Welcome to Tableau Server

Tableau Server, by Tableau Software®, is an online solution for sharing, distributing, and collaborating on content created in Tableau.

- **Shareable.** Tableau Server users can create workbooks and views, dashboards, and data sources in Tableau Desktop, and then publish this content to the server.
- **Secure.** Tableau Server site and server administrators control who has access to server content to help protect sensitive data. Administrators can set user permissions on projects, workbooks, views, and data sources.
- **Mobile.** Users can see and interact with the most up-to-date server content from anywhere, whether they use a browser or mobile device.

New to Tableau Server? Go to Get Started with Tableau Server on page 19 to learn more.

A Tableau Server content page, where users can explore and interact with up-to-date data.
What's New in Tableau Server

What's New in Version 10.0

The following features and enhancements are new in Tableau Server 10.0:

User

- Revision history for data sources and workbooks
- Enhancements for data source discovery
- Web authoring enhancements

Site administrator

- Site-specific SAML

Server administrator

- Administrative views for Tableau Desktop license reporting
- Administrative views for Tableau Server performance
- Subscribe other users to views and workbooks
- Customize server logos independently
- Cluster deployment
- Performance improvements
- Kerberos enhancements
- Updated server menus and navigation

Updated server menus and navigation

Top-level menus in Tableau Server have changed. For example, users who are signed in as server administrators now access individual sites by using the Manage All Sites menu. For information, see Navigate Server Admin Pages on page 662, Navigate Site Admin Pages on page 318, and Navigate Tableau Server on page 27.

Revision history for data sources and workbooks

As with revision history for workbooks (added in version 9.3), Tableau Server now supports revision history for data sources. Every time you publish a data source, Tableau can save a version of that data source to its revision history. In addition, workbook revisions can now be restored online in Tableau Server. For more information, see Keep Content Revisions and Maintain Content Revisions.
Enhancements for data source discovery

Published data sources can be marked as favorites, which means you can now filter and view favorite data sources. For more information, see Mark and Search Favorites on page 70.

On the Data Sources page, you can now sort data sources based on popularity—number of connected workbooks, page views based on the number of page views for workbooks, and views that are using a data source. For more information, see Quick Start: Discover Popular Content.

Web authoring enhancements

Web authoring now supports creating dashboards online, connecting to published data sources, cross-data filtering, data blending, and more.

- Create and edit dashboards
- Connect to published data sources
- Filter across data sources
- Blend data sources
- Rename fields
- Workbook formatting
- Data highlighting
- Show and hide cards for captions, filters, and highlighters
- Clear or duplicate sheets (for views or dashboards)
- Full support for creating and editing table calculations

Subscribe other users to views and workbooks

Content owners, project leaders, and administrators can now subscribe other users to workbooks and views in Tableau Server. For more information, see Set Up Subscriptions to Views and Subscribe to Views.

Performance improvements

Tableau Server includes the following performance improvements to reduce load times and provide faster feedback after some workbook edits:

- To display views more quickly, the initial page load time for views has been improved.
- To display story points more quickly and provide a better experience when you navigate between them, Tableau Server now caches and pre-fetches story points.
To provide faster feedback when you edit workbooks, the changes that you make to colors, shapes, and global transparency display more quickly.

**Administrative views for Tableau Server performance**

You can use two new administrative views to help optimize background tasks and view load times. The **Performance of Views** administrative view displays how long it takes for views to load and how many sessions are running at a time on the server. The **Background Task Delay** view displays how long extract refresh tasks and subscription tasks are delayed—that is, the amount of time between when they are scheduled to run and when they actually run. For more information, see **Background Task Delay** on page 1108 and **Performance of Views** on page 1109.

**Customize server logos independently**

You can now specify custom images for the main header, sign-in, and web authoring logos. For more information, see **Change the Name or Logo** on page 655 and **customize** on page 1266.

**Administrative views for Tableau Desktop license reporting**

Two new administrative views provide visibility into how Tableau Desktop licenses are being used in your organization, including which licenses are due for maintenance renewal or upgrade from trial versions. For more information, see **Quick Start: Tableau Desktop License Reporting** on page 1078.

**Note:** Desktop License Reporting is disabled on Tableau Server by default. See **Enable Desktop License Reporting** for details on how to enable it.

**Cluster deployment**

Cluster installations no longer require Active Directory domain deployment. You can now install a cluster in a Windows Workgroup. See **Distributed Requirements** on page 696 for more information.

**Site-specific SAML**

Tableau Server now includes more options for authenticating users with SAML. For a Tableau Server installation with multiple sites, you can configure each site to use a separate SAML identity provider (IdP). For more information, see **SAML** on page 1010.
Kerberos enhancements

Tableau Server now includes the following enhancements to Kerberos support:

- Kerberos authentication for Oracle data sources. See Enable Kerberos for Oracle on page 1000.
- Multi-domain support for Kerberos delegation. See Kerberos delegation multi-domain configuration on page 1009.
- The ability to use JDBC to connect to SAP HANA on a Mac. You can now connect to SAP HANA on a Mac using JDBC drivers and Kerberos authentication. For more information, see SAP HANA.

What's New in Version 9.3

The following features and enhancements are new in Tableau Server 9.3:

- Revision history for workbooks
- Support for manual-only repository failover
- PostgreSQL database verification
- Monitoring and alerts for low disk space
- Content metrics and relevance in server pages
- Zooming improvements for maps
- Kerberos support for PostgreSQL and Teradata connections
- On-demand connections
- Progressive loading of dashboards
- Hardware-based defaults for single-server installations
- Project Leader enhancements

Revision history for workbooks

Every time you publish a workbook, Tableau can now save a version of that workbook to its revision history. Then, if you want to revert to a previous version of the workbook, you can go to the workbook in Tableau Server, view its revision history, download the version you need, and then republish it to the server. For more information, see Quick Start: Revision History.

Support for manual-only repository failover

You can now disable automatic failover of the active PostgreSQL repository. You can disable automatic repository failover to control failover manually by using the failover/repository on page 1274 command. For more information, see the
clustercontroller.pgsq1.failover option in tabadmin set options on page 1294.

PostgreSQL database verification

You can now use the tabadmin verify_database command to verify that the PostgreSQL database does not contain any errors that would cause a restore to fail. For more information, see Verify the Tableau Postgres Database on page 1148.

Monitoring and alerts for low disk space

Because low disk space can reduce server performance or even cause the server to stop responding, you can specify disk space thresholds and get alerts when space falls below them. Alerts continue until disk space rises above the warning threshold. You can also use a new Server Disk Space Administrative View to track recent changes in disk space. For more information, see Quick Start: Disk Space Alerts on page 1091.

Content metrics and relevance in server pages

Find the right content quickly on Tableau Server and Tableau Online through metrics-based search results and sort options. List view and thumbnail view display metrics as well. For more information, see Quick Start: Discover Popular Content.

Zooming improvements for maps

Zooming in map views just got easier. You can now scroll or pinch and stretch to zoom in and out of a point on a map.

Kerberos support for PostgreSQL and Teradata connections

Tableau Server now supports Kerberos for connections to PostgreSQL and Teradata databases. For more information, see Kerberos Requirements on page 992.

On-demand connections

When you load a workbook published to Tableau Server, Tableau connects only to the data sources that are required to display your data. This means that if there are multiple tabs in a workbook, Tableau only connects to the data sources for the current tab so that you can view your data as soon as possible. For data sources that require authentication, now you only have to enter your credentials for the current tab or worksheet. As a result, when you view a shared workbook, you can view the workbook tabs for which you have data credentials. Previously, you needed to have credentials to all the data sources in the workbook before you could view any of the workbook tabs.
Progressive loading of dashboards
Dashboards load progressively so that you can view and interact with your data faster. Rather than wait for the entire dashboard to load, you can start analyzing your data as soon as the first section has finished loading.

Hardware-based defaults for single-server installations
The Tableau Server installer detects your computer’s hardware and then creates an optimal default configuration for single-server installations. The default configuration determines the number of Tableau Server processes that will run, based on the hardware resources Tableau finds on the computer. If you run a distributed installation, only the primary computer in the cluster is configured with the new defaults. The optimal defaults are intended to serve as a good starting point for you to further optimize performance. For more information, see Primary Server Installation Defaults on page 614.

Project Leader enhancements
Users with the Project Leader permission can now change ownership of content, move workbooks between projects, and run extract refresh schedules. For more information, see Projects on page 756.

Added in Version 9.2
The following features and enhancements were added in Tableau Server 9.2.

- Default permissions in projects
- Lock content permissions to the project
- Support for OpenID Connect
- Updated toolbars for views and web editing
- Web editing enhancements
- Find your current location in a map view
- Enhanced logging by Server Resource Manager

Default permissions on projects
Administrators and project leaders can set the permissions for a project, and set the default permissions for the workbooks and data sources in the project. As a result:

- Permission capabilities are now relevant for each content type in the project. For example, only the View, Save, and Project Leader capabilities are available for
projects.
- The default permissions apply to all content published to the project. Default permissions can only be changed at the project level.
- Administrators and project leaders can give each project a unique set of default permissions.

Note: New projects in the site still start with a copy of the permissions defined for the Default project.

For more information, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Lock content permissions to the project

Administrators and project leaders can lock content permissions in a project, and prevent users from changing the permissions of any content in the project. When project permissions are locked:

- The default permissions are used for all workbooks (including views) and data sources in a project.
- Users (including content owners) cannot modify permissions for individual workbooks, views, and data sources in the project.

Note: Because you can lock content permissions to the project, the Assign Permissions to Contents button has been removed for projects and workbooks.

For more information, see Quick Start: Lock Project Permissions and Lock Content Permissions to the Project on page 869.

Support for OpenID Connect

Tableau Server now supports OpenID Connect for authentication and single sign-on (SSO). You can configure the server to redirect users to an OpenID Connect identity provider (IdP) (for example, Google), where a user signs in with the user name and password that he or she normally uses with that provider. The user is then automatically signed in to Tableau Server. For more information, see OpenID Connect on page 1050.

View and authoring toolbar enhancements

The view and authoring toolbars and interfaces have been updated and a few commands have changed.
Manage and create custom views by clicking **Original View** or the name of the current custom view. Custom views no longer are referred to as **Remember my changes**.

The **Export** menu has been replaced with **Download**. To export a view as an Image, Crosstab, Data, or PDF, and to download a workbook, use the **Download** menu.
Web editing enhancements

The following functionality is now available as you edit views on the web:

- Drag all or part of a formula to the Data pane to create a new field.
- Right-click on a dimension in the Data pane and convert it to a measure, or right-click on a measure and convert it to a dimension:

You can also right-click to change a field’s data type, to set a default aggregation or geographic role, or to covert a measure or a date field from continuous to discrete—or from discrete to continuous.

- If you upload a workbook that uses blended data, you can see a link icon next to the field or fields in the primary data source that are being used to link the two data sources:
You can click on the link icon to activate or deactivate specific fields. When fields that can be used as linking fields are not being used, the link icon changes appearance:

- The colors that you set for the sheet tabs in Tableau Desktop also display when you edit the views in Tableau Server.

**Find your current location in a map view**

You can now quickly zoom to your current location in a map view.
Enhanced logging by Server Resource Manager

Administrators can now see incremental increases and decreases in memory and CPU usage for each of the processes that Server Resource Manager monitors. Entries are written into the logs for each process. Logs are located in:

<install directory>\ProgramData\Tableau\Tableau Server-\data\tabsvc\vizqlserver\Logs.

Added in Version 9.1

The following features and enhancements were added in Tableau Server 9.1.

- Scheduled synchronization of Active Directory groups
- Authentication for connected devices
- Mutual (two-way) SSL authentication
- SAML Logout
- Single Sign-on for SAP HANA
- Web data connectors

Scheduled synchronization of Active Directory groups

Server Administrators can synchronize all Active Directory groups in Tableau Server on-demand or on a scheduled basis. For more information, see Quick Start: Synchronize All Active Directory Groups on a Schedule.

Authentication for connected devices

When mobile users connect to Tableau Server, administrators can control whether mobile users must sign in and provide their credentials every time they connect to Tableau Server, or if users can connect with their devices to Tableau Server without signing in after their device is successfully authenticated. For more information, see Authentication for Connected Devices.

Mutual (two-way) SSL authentication

Tableau Server now supports mutual SSL authentication between Tableau Server and clients (Tableau Desktop, web browsers, and tabcmd.exe). For more information, see Quick Start: Mutual (Two-Way) SSL Authentication on page 970.
Note: When Tableau Server is configured for mutual SSL, you cannot sign into Tableau Server using a version of Tableau Desktop earlier than 9.1. If you sign in with an earlier version of Tableau Desktop and fallback authentication is not configured, an error displays: "Invalid username or password". If fallback authentication is configured, you are prompted for your user name and password. See Quick Start: Mutual (Two-Way) SSL Authentication on page 970 for details about fallback authentication.

SAML Logout

Users who sign in to Tableau Server using SAML SSO functionality can now terminate their session by signing out of server. This standards-compliant solution helps ensure SOX compliance through providing support for both Service Provider (SP) initiated logout and Identity Provider (IdP) initiated logout, with Tableau Server being the Service Provider and a third party solution like Ping Federate being the SAML Identity Provider. For more information about SAML and Tableau Server, see SAML Requirements.

Note: If you are using SAML with a version of Tableau Server prior to 9.1, see What's Changed - Things to Know Before You Upgrade on page 665.

Single Sign-On for SAP HANA

Tableau Server now supports single sign-on (SSO) for SAP HANA when SAP HANA is configured to support single sign-on (SSO). Users can sign in to their SAP HANA server and then access and publish data to Tableau Server, without having to re-enter their user name and password. For more information, see SAP HANA SSO. Note: Tableau Server requires SAP HANA driver version 1.00.9 or later to support SSO for SAP HANA.

Web data connectors

You can import web data connectors to Tableau Server. This gives you an opportunity to vet web data connectors before you make them available for users. Extracts that were created by imported web data connectors can also be refreshed on the server. You can manage web data connectors on the server using new tabadmin commands. For more information, see Web Data Connectors in Tableau Server on page 899.

Added in Version 9.0

The following features and enhancements were added in Tableau Server 9.0.

- Streamlined user interface
- Improved server management
• Improved user, group, and permission management
• Scheduled synchronization of Active Directory groups
• New server processes
• Faster performance
• High availability
• New tabadmin and tabcmd commands

Streamlined user interface
Tableau Server 9.0 features improved workflow, quick access to information, and simplified search and content management.

• A server-based service supports the rewritten web client, which is faster, more scalable, and more extensible.
• New content pages show all content by type (projects, workbooks, views, data sources) in a single place. Each page includes key information about the content, including related items and actions. For more information, see Access and Manage Your Content on page 72 and Navigate Tableau Server on page 27.
• Site navigation is streamlined. Administrators and users can switch more easily between sites and content. For more information, see Get Around: Explore Menus and Content on page 27 and Multi-Site Navigation on page 36.
• Search all content types at once using the new quick search box at the top of the page. For more information, see Quick Start: Quick Site Search on page 53 and Search Content on page 64.
• View content faster by scrolling instead of paging through content.

Improved server management
Improved views, alerts, and secure communication options simplify server management

• Built-in administrative views are redesigned for improved loading and faster performance, more in-depth analytics, and an interface that's easier to understand. For information, see Administrative Views on page 1097.
• Email alerts for server processes are consolidated so that it's easier to check server health. For information, see Configure Alerts and Subscriptions on page 619.
• The server supports enabling SSL for communication between server components and the PostgreSQL repository. For information, see Configure Internal SSL on page 976.
Improved user, group, and permission management

Manage users, groups, and permissions with a streamlined workflow. For more information, see Users on page 781 and Groups on page 821.

- Site roles for users replace license levels and user rights. For information, see Site Roles for Users on page 788. Note that license levels are still used when you use the tabcmd addusers command to import users. For more information, see CSV Import File Guidelines on page 810.
- If a user is disabled in or is deleted from Active Directory, the user is removed from Tableau Server groups and becomes unlicensed.
- A redesigned permissions interface provides responsive visual feedback on effective and resulting permissions. You can now view and modify permissions for content in a single view. For information, see Quick Start: Permissions on page 842 and Manage Permissions on page 834.
- Monitor the status of server licenses on the Licenses page, and by using the tabadmin licenses command. For information, see View Server Licenses on page 1169.

New server processes

The following new processes help manage API requests, clusters, and high availability.

- **API Server.** This process is used when you interact with the server via REST API.
- **Cache Server.** This distributed process manages a shared query cache across the server cluster and is used by the VizQL Server, Backgrounder, and Data Server processes.
- **Cluster Controller.** This process runs on every computer in a cluster and works with the Coordination Service to report process status and coordinate failover for high availability.
- **Coordination Service (zookeeper).** This process runs on every computer in a cluster, manages leader election when needed, and ensures that there is a quorum for making decisions during failover.
- **File Store.** This process ensures that extracts are available on all nodes of a cluster that is configured with a File Store process.

For information about the new processes, see Tableau Server Processes on page 1240 and Improved High Availability and Data Engine Scale.

You can see process status on the server Status page or by using the tabadmin status --verbose command.

The following processes from earlier versions of Tableau Server are changed or removed:
In the Data Engine process, you are no longer limited to running only two data engine nodes per cluster. This new flexibility can improve server clusters that are used for extract-heavy scenarios.

- **Faster performance**
  Performance enhancements make workbooks load faster, improve dashboard performance, and make interactivity more seamless with faster results and response.

  - Parallel queries take advantage of the capabilities of source databases to execute more queries at the same time. Each query is faster, and independent queries start at the same time.
  - Data engine vectorization speeds up the performance of extract-based workbooks.
  - Parallel aggregation uses multiple cores in Tableau extracts and other file-based data sources.
  - Temp table support in the Data Server makes querying more efficient with databases that use temp tables, such as SQL Server.
  - External query caching re-opens workbooks with Tableau extracts more quickly because queries are not re-run every time you open a workbook.
  - Query Fusion recognizes related queries in a dashboard and combines them so there are fewer queries to execute.
  - Shadow extracts create faster data source access for large text files and Excel files.

  For information about optimizing Tableau Server performance, see Performance Tuning Examples on page 1135.

- **High availability**
  Tableau Server provides a more secure, more robust high-availability solution with straightforward configuration and management features for administrators.

  - More robust failover support:
    - An updated user experience improves cluster configuration.
    - The Cluster Controller process centralizes failure detection and response.
    - You can manually trigger Repository failover and failback using the `tabadmin` utility.
    - You can set a preferred active Repository (via a new user interface) for better support of asymmetrical hardware configurations.
    - New workflows and feedback during configuration make it easier to configure a server.
deployment for high availability and ensure that data and server information are safe in the process.

- A new data extract storage process (File Store) provides significant speed improvements and makes data replication more reliable.
- Tableau Server no longer restricts the number of Data Engine nodes per cluster.
- All Data Engine nodes are active and data is replicated between them, which eliminates the need for failover of the Data Engine processes.

For information, see Improved High Availability and Data Engine Scale and High Availability on page 709.

New tabadmin and tabcmd commands

The tabadmin utility has new commands, including:

- decommission
- failoverrepository
- manage_global_credentials
- recommission
- regenerate_internal_tokens

In addition, the cleanup command includes a new --reset-coordination option that extends the cleanup procedure to data maintained by the new Coordination Service.

The tabcmd utility features these improvements:

- Faster startup and execution.
- Support for Unicode.
- Support for the new --role option for the createusers, createsiteusers, and syncgroup commands.
- The following new commands:
  - deletesiteusers
  - editdomain
  - listdomains
Get Started with Tableau Server

Different types of users interact with Tableau Server in different ways. We’ve organized the documentation for our readers with an eye to what types of tasks different readers are probably interested in.

- **User Guide.** In this documentation, the User Guide is for business users—people who go to Tableau Server to see and interact with views. This guide is also useful for analysts/authors, or as we also refer to them, *content owners*—people who create workbooks and views and publish them.

- **Site Administrator Guide.** The Site Administrator Guide is for people who manage one or more individual sites on Tableau Server—they customize existing sites, add and manage users, manage permissions for users and content, and monitor site usage. Some of the information in this guide is also useful for analysts/authors, who create workbooks and views and publish them (*content owners*), and for *data stewards*, who create, manage, and publish data sources.

- **Server Administrator Guide.** The Server Administrator Guide is for people who install, configure, and manage Tableau Server itself. Tasks for the server administrator include installing and upgrading the server, managing licensing, backing up the server, adding and managing nodes in a cluster, managing security and authentication (including SSL).

**Suggested starting points**

The following lists provide suggested starting points and topics if interest in the documentation for each type of user.

**Users**

- Sign In
- Get Around
- Create or Interact with Published Views on page 82

**Content owners, data stewards, and publishers**

- Get Around
- Navigate Tableau Server
- Create Views and Dashboards
- Control Access to Server Content
- Keep Content Revisions
- Refresh Data on a Schedule

**Site administrators**
Server administrators

If you’re new to being a server administrator, if you’re installing a trial version of Tableau Server, or if you’re an IT pro who wants a high-level overview of the how to install and configure Tableau Server, start here:

- Everybody's Install Guide

This guide walks you through the basic planning and installation tasks for a single-server installation of Tableau Server.

If you are comfortable as a server administrator and if you want more in-depth information about installing and configuring Tableau Server, we recommend the Server Administrator Guide, which provides comprehensive information about how to install, manage, and monitor Tableau Server.

- Before You Install
- Navigate Server Admin Pages on page 662
- Distributed Environments on page 694
- Security on page 953

Get information on using Tableau Server

For information on using Tableau Server, click the help menu at the top of the page.

Product version information

To find your Tableau Server version and build number, click About Tableau Server.

Your version and build number appear in the About Tableau Server dialog box:
User Guide

The topics in this guide describe how to sign in and find your way around Tableau Server to discover, view, and interact with workbooks and data visualizations (known as views or "vizzes" in Tableau). If you're familiar with Tableau Desktop, you'll be pleased to discover that many of the ways that you can interact with views in Tableau Desktop can be also be done online, through your web browser.

Video introduction to Tableau Server

You're probably reading this because Tableau Server is installed in your organization. If you're not sure what Tableau Server is or what you should or can do with it, watch this five-minute video.

To view more training and introductory videos, go to Free Training Videos on the Tableau website.
Sign In

You use a *Tableau Server account* to browse content on the server. Your Tableau Server administrator assigns you a site role, which determines how you can work with Tableau Server—for example, whether you can publish, interact with, or view content.

Get sign-in credentials

When you sign in, you provide your sign-in credentials, that is, your user name and password. Your sign-in credentials are based either on your Active Directory user account, or on the Tableau Server user management system. Contact your administrator if you do not know your user name or password.

Change the sign-in account

Depending on how your server is configured, you might be signed in automatically. If you are signed in automatically to the wrong account, you can switch to the correct account.

1. Click **Sign Out** on the user menu in the upper-right corner of the page.
2. Type your user name and password, and then click **Sign In**

![Sign In Screen](image)

**Note**: Tableau Server always stores a session cookie when you sign in. For this reason, your web browser must be configured to allow first-party cookies for you to sign in.

**Sign in to a site**

You might have access to one or multiple sites on Tableau Server. When you sign in to a single-site server, you will not need to select the site.

If you have access to multiple sites on Tableau Server, you must select the site you want to work with during the sign-in process.

![Select a Site](image)

The name of the current site is displayed on the site menu.
Sign in with SAML

If Tableau Server is configured to use SAML authentication, you will see a sign-in prompt from an external identity provider (IdP) instead of the Tableau Server Sign In prompt. For example:

![Sign in with SAML](image)

When you sign in with SAML, your ability to sign out of Tableau Server depends on whether or not Tableau Server is configured for SAML logout. If you do not see a **Sign Out** command on the Tableau Server user menu, you must use the sign-out interface provided by the IdP. If Tableau Server is configured for SAML logout, you will see the **Sign Out** command. For more information on SAML logout, see [SAML Requirements](#).

Sign in with mutual SSL

If Tableau Server is configured to use mutual SSL authentication, you might see a certificate selection dialog box. (Whether you see the dialog box, and whether you see it each time you log in, depends on the browser you are using and if there are multiple certificates on your
computer. If Tableau Server is configured for fallback authentication, you can cancel out of the certificate selector and use your user name and password to sign in. For more information about mutual SSL, see the Mutual SSL Authentication Quick Start.
Get Around: Explore Menus and Content

When you sign in to Tableau Server as a user, you will see the content page. Explore the Projects, Workbooks, Views, or Data Sources menus to see the content that is available to you.

The menus and options that you see in Tableau Server can vary depending on your site role, whether you own content in the site, and the permissions that are set on the site content. If you belong to only one site, these are the menus you will see:

If you have access to multiple sites, you will see the site menu with the name of currently selected site:

Note: For information on the menus and pages available to users, see Navigate Tableau Server below.

For information on the menus and pages available to administrators, see Navigate Server Admin Pages on page 662 and Navigate Site Admin Pages on page 318.

If this is the first time you've ever seen Tableau Server and you're not sure what Tableau Server is or what you can do with it, watch this five-minute video.

To view more training and introductory videos, go to Free Training Videos on the Tableau website.

The following topics provide a visual tour of the content pages in Tableau Server.

Navigate Tableau Server

This topic describes how to navigate the content pages in Tableau Server.
When you sign in to Tableau Server, the pages you can see and the options available to you are based on your site role and content permissions.

Every Tableau workbook or data source published to, or created in Tableau Server, lives in a site. That site serves as a central repository for views, workbooks, dashboards, stories, and even data sources that you and your coworkers can then access and share with each other.

As a site user, you can explore the Projects, Workbooks, Views, or Data Sources pages to see the content that is available to you. You can also click your name and then click My Content to see the content you have created and published in Tableau Server, and your account information. For more information, see Access and Manage Your Content on page 72

Note: For information on the menus and pages available to administrators, see Navigate Server Admin Pages on page 662 and Navigate Site Admin Pages on page 318.

Content pages

When you sign in to Tableau Server, the first page you see will look something like the following example. From this page, you can explore the content that is available to you by searching, filtering, and sorting the view.

The Content page includes the Projects, Workbooks, Views, or Data Sources menus. As you click these menus, each of these pages shows the content you can access.

- A - Site menu available (if you are a user in more than one site)
- B - Quick search
- C - Favorites search
- D - Your user settings and content
- E - Content menus
- F - Filtered search

Navigate sites
If you have access to multiple sites, you must select a site when you sign in. The content for that site will be displayed, along with the site menu and the name of the current site. You can then navigate to other sites by selecting a different site from the site menu.

Navigate to views

To get to a view, do one of the following:

- Click Projects, click a workbook, and then click a view thumbnail or a view name in a list.
- Click Workbooks, and then click a view thumbnail or a view name in a list.
- Click Views, and the click a view thumbnail or a view name in a list.

View, sort, and select content

On a site’s Content page, select the Projects, Workbooks, Views, or Data Sources menu to see the content available to you. Within these pages, you can search, filter, sort, view, and
select content.

For details on searching and filtering content, see Search Content on page 64.

View content in lists or as thumbnails

As you browse Tableau Server, you will notice that items are displayed either as thumbnails or in lists. Icons at the top of Projects, Workbooks, and Views pages let you specify whether contents should be displayed as lists or as thumbnails. Click the thumbnail or list icon to toggle the view.

![Thumbnail view](image1)

**Thumbnail view**

![List view](image2)

**List view**

Sort content

Depending on the type of content displayed on the page, you can sort by different characteristics, such as by number of views, name, order of sheets, owner, project, creation or modification date, and so on.
To sort the items in a view, click the **Sort By** drop-down arrow, and then select the sort option. This example shows the sorting options for views.

You can also sort a list by clicking the column heading. An arrow indicates the order: ascending (up arrow) or descending (down arrow). To sort using multiple options, click the sort type in inverse order. For example, to sort on project first, workbook second, and name last, click those sort types in inverse order (name, workbook, project).
Select content and use actions

On the Projects, Workbooks, Views, and Data Sources pages, click a selection box to select content and perform various actions, such as to tag content or to assign permissions. Click the Actions menu ( . . . ) to access commands available for the selected content.
Notes When you select multiple items, the Actions menu might display commands that don't apply to one or more items in the set of selected items. When you select multiple items and apply an action command to those items, Tableau Server will attempt to perform the action on all of the items and notify you of the results (whether the action succeeds or not) for the individual items.

To select all content on the page, click the drop-down n selected arrow, and then click Select All. To clear a content selection, click the drop-down arrow and then click Clear.

If many items are available on a page, click the Actions menu ( . . .) to select the item and see the available commands.

In thumbnail view, the Actions menu is displayed in the right corner of the thumbnail.
In list view, the **Actions** menu is displayed to the right of the workbook name.

**Interact with views**

With the view open, different actions will be available in the view depending on the settings for the server and site, and your permissions. Possible actions include: share, download (data, crosstab, image, PDF, or workbook), subscribe to the view, create custom views, and edit the
An view open for interaction

Example of content actions available for a view

Workbooks often contain multiple sheets, including individual views, dashboards, and stories. Authors can optionally publish these workbooks where the sheets are shown as tabs along the top of the page (using the Tableau Desktop option **Show Sheets as Tabs**). You can navigate a tabbed workbook by using the sheet tabs.

Sheets shown as tabs in the view

Whenever you are looking at a view, you can navigate back to a **Projects**, **Workbooks**, or **Views** page by clicking a breadcrumb link.
Breadcrumb links above the view

Edit a view

To make changes to a view, click **Edit**. This action is available to users who have web editing permissions.

The view opens in the web editing workspace.

The web editing workspace

Multi-Site Navigation

These tips will help you navigate from site to site, and identify which site you're using.
Site Sign-In

If you are a member of multiple sites, when you sign in to the server, you are prompted to select a site. Click the site name to select it. You can also enter a name to select from a list of sites with similar names.

If you only belong to one site, you are not prompted to choose your site at server sign-in. After you sign in, you will not see a site menu at the top of the page.

Navigate to other sites

If you belong to multiple sites, the site menu is displayed at the top of the page:

To sign in to a different site, click the site menu and select the site:
Identify your site

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

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

Work in the Web Authoring Workspace

The web authoring workspace in Tableau Server is a lot like Tableau Desktop. You can create and edit views and dashboards in your workbook, but not stories.

Note: Only the workbook owner can save changes to an existing workbook on Tableau Server or Tableau Online.

Worksheets

Like in Tableau Desktop, Data and Analytics panes appear on the left side of the workspace.
The Data pane includes the names of the data sources included in the workbook, and the fields, parameters, and sets included in the active data source.
The Analytics pane provides analytic features in Tableau. You can drag reference lines, forecasts, trend lines, and other objects into your view from the Analytics pane.

A toolbar with options for editing the view appears at the top of the workspace. You can also use some of the toolbar options to connect to published data sources or create new worksheets and dashboards. For more information about the toolbar options, see Web Authoring Toolbar on page 185.
The Marks card and the Pages and Filters shelves appear on the left of the view. Columns and Rows for measures and dimensions shelves appear above the view. Tabs for worksheets and dashboards in the workbook appear at the bottom of the workspace.

To create a new view, click the New Worksheet button [ ] at the bottom of the workspace, or on the toolbar, click the drop-down arrow on the New Worksheet [ ] button and select New Worksheet from the list.

Dashboards

Select a dashboard tab [ ] in your workbook, and then click Edit in the toolbar to begin authoring. You can interact with dashboards as you would in Tableau Desktop. For more information, see Create Views and Dashboards on page 148.
For example, sort data by selecting the sort button \( \text{ sorter } \) next to a view name on your dashboard. Or, select field names to **Keep Only**, **Exclude**, **View Data**, or **Group Members**. These actions affect associated underlying worksheets.

<table>
<thead>
<tr>
<th>Country</th>
<th>Ease of Business</th>
<th>Days to Start Bus.</th>
<th>Hours to do Tax</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singapore</td>
<td>1</td>
<td>5</td>
<td>82</td>
<td></td>
</tr>
<tr>
<td>Hong Kong SAR, C.</td>
<td>2</td>
<td>8</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>New Zealand</td>
<td>3</td>
<td>0</td>
<td>170</td>
<td></td>
</tr>
<tr>
<td>United States</td>
<td>4</td>
<td>8</td>
<td>237</td>
<td></td>
</tr>
<tr>
<td>Denmark</td>
<td>( \checkmark )</td>
<td>Keep Only</td>
<td>Exclude</td>
<td></td>
</tr>
<tr>
<td>Korea, Rep.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norway</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malaysia</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Georgia</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To easily update or review views in your workbook, select a view on your dashboard then click the **Go to Sheet** icon.
Ease of Business

To create a new dashboard, on the toolbar, click the drop-down arrow on the New Worksheet button and select New Dashboard from the list or click the dashboard button at the bottom of the workspace.

To rename an existing dashboard, double-click the tab for the dashboard and type a new name.

Use the Download menu to save your dashboard for offline viewing or to download underlying data.

Select Download > Image to save your dashboard as a PNG file. Select Download > PDF to save your dashboard as a PDF.

- To download a view’s underlying data, first select the view on your dashboard. The context-sensitive Download menu updates to add the options Data and Crosstab.
- Select **Data** to open a new window where you can preview and download both summary and underlying data for the view you've selected.

- Select **Crosstab** to download summary data for the view you've selected as a text table (CSV file).

**Web Authoring Toolbar**

When you are creating or editing a view, you can use the toolbar at the top of the view to perform common actions. The table below explains the function of each toolbar button.

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>←</td>
<td><strong>Undo</strong>: Reverses the most recent action in the workbook. You can undo an unlimited number of times, back to the last time you opened the workbook, even after you have saved.</td>
</tr>
<tr>
<td>→</td>
<td><strong>Redo</strong>: Repeats the last action you reversed with the <strong>Undo</strong> button. You can redo an unlimited number of times.</td>
</tr>
</tbody>
</table>
| 📦             | **New Data Source**: Opens the **Connect to a Data Source** page, where you can connect to a published data source. For more information, see [Connect to Published Data Sources](#) on page 894.
<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Pause Button" /></td>
<td>Pause Auto Updates: Controls whether Tableau updates the view in real time when changes are made. For more information, see Pause Automatic Updates on page 116.</td>
</tr>
<tr>
<td><img src="image" alt="New Worksheet Button" /></td>
<td>New Worksheet: Creates a new blank worksheet or dashboard. For more information, see Create Views and Dashboards on page 148.</td>
</tr>
<tr>
<td><img src="image" alt="Duplicate Sheet Button" /></td>
<td>Duplicate Sheet: Creates a new worksheet or dashboard containing the same view as the current sheet.</td>
</tr>
<tr>
<td><img src="image" alt="Clear Button" /></td>
<td>Clear: Clears the current worksheet or dashboard. Use the drop-down menu to clear the entire sheet or just the sheet formatting.</td>
</tr>
<tr>
<td><img src="image" alt="Swap Button" /></td>
<td>Swap: Moves the fields on the Rows shelf to the Columns shelf</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>and vice versa. Most used with view types that are based on x- and y-axes.</td>
</tr>
</tbody>
</table>

**Totals**: You can compute grand totals and subtotals for the data in a view. Select from the following options:

- **Show Column Grand Totals**: Adds a row showing totals for all columns in the view.
- **Show Row Grand Totals**: Adds a column showing totals for all rows in the view.
- **Row Totals to Left**: Moves rows showing totals to the left of a crosstab or view.
<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Column Totals to Top:</td>
<td>Moves columns showing totals to the top of a crosstab or view.</td>
</tr>
<tr>
<td>• Add All Subtotals:</td>
<td>Inserts subtotal rows and columns in the view, if you have multiple dimensions in a column or row.</td>
</tr>
<tr>
<td>• Remove All Subtotals:</td>
<td>Removes subtotal rows or columns.</td>
</tr>
<tr>
<td>Highlight:</td>
<td>Turn on highlighting for the selected sheet. Use the options on the drop-down menu to define how values are highlighted. For more information, see Highlight Marks, Legends &amp; Actions.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image1.png" alt="Toolbar Button" /></td>
<td><strong>Show Mark Labels</strong>: Switches between showing and hiding mark labels for the current sheet.</td>
</tr>
<tr>
<td><img src="image2.png" alt="Toolbar Button" /></td>
<td><strong>Format Workbook</strong>: Open the Format Workbook pane to change how fonts and titles look in every view in a workbook by specifying format settings at the workbook level instead of at the worksheet level.</td>
</tr>
<tr>
<td><img src="image3.png" alt="Toolbar Button" /></td>
<td><strong>Fit</strong>: Change the proportions of your view within the browser window, and go back and forth between seeing details and seeing the whole picture. The <strong>Cell Size</strong> commands have different effects depending on the type of visualization.</td>
</tr>
<tr>
<td><img src="image4.png" alt="Toolbar Button" /></td>
<td><strong>Show/Hide Cards</strong>: Shows and hides <strong>Titles, Captions, Filter</strong>, and <strong>Highlighter</strong> cards that are included in</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image.png" alt="Image Button" /></td>
<td><strong>Download</strong>: Use the options under <strong>Download</strong> to capture parts of your view for use in other applications.</td>
</tr>
</tbody>
</table>
| ![Data Button](data.png) | **Image**: Displays the view, dashboard, or story as an image in a new browser tab.  
**Data**: Displays the data from the view in a new browser window with two tabs: **Summary**, showing aggregated data for the fields shown in the view, and **Underlying**, showing underlying data for the selected marks in the visualization. If the new window does not open, |
<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>you may need to disable your browser's popup blocker.</td>
</tr>
<tr>
<td><strong>Crosstab:</strong></td>
<td>Saves the underlying data for the selected marks in the visualization to a CSV (comma-separated values) file which can then be opened in Microsoft Excel.</td>
</tr>
<tr>
<td><strong>PDF:</strong></td>
<td>Opens the current view as a PDF in a new browser window. From there you can save it to a file. If the new window does not open, you may need to disable your browser's popup blocker.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>![Show Me](Show Me)</td>
<td><strong>Show Me</strong>: Opens a control that shows a range of visualization types that you can use in Tableau. When you display the Show Me list, Tableau uses the data in the current view to determine which visualization types to make available for you to select. Among the available types, it draws an orange outline around the recommended chart type that is the best match for your data. You can also hover over a visualization type to see what field types are required to make that visualization type available.</td>
</tr>
</tbody>
</table>

**Tiled and Floating Layouts**

Each object in a dashboard can use one of two types of layouts: tiled or floating. Tiled objects are arranged in a single layer grid that adjust in size based on the total dashboard size and the objects around it. Floating objects can be layered on top of other objects and can have a fixed size and position.

**Tiled Layout**

All objects are tiled on a single layer. The top three views are in a horizontal layout container.

**Floating Layout**

While most objects are tiled on this dashboard, the map view is floating. It's layered on top of the bar chart, which uses a
Switch Between Layouts

By default, objects in a dashboard use the Tiled layout. Tiled is selected by default when you drag views and objects onto your dashboard sheet:

Objects
- Horizontal
- Vertical
- Blank

Tiled  Floating

To switch an object that's already in your dashboard to floating:
Select the Floating option from the object's short-cut menu:
Discover and Find Content

You can instantly find content anywhere on a site by searching for it using keywords, a title, the date that the content was modified, or tags. You can narrow down your searches to help find specific content—for example, you can focus on content that's popular and see which workbooks and views have been visited recently. To help you find the content you work with frequently, you can mark workbooks and views as favorites.

Quick Start: Quick Site Search

You can instantly find content anywhere on a site by entering keywords. For more information about search, see Search Content on page 64.
1 Go to the site you want to search

Open your browser and navigate to the site you want to search. The search box at the top of the page lets you search across all resources that you have access to on the current site, including projects, workbooks, views, data sources, and users.

2 Enter text

Enter a search keyword. As you type, the top three results of each resource type are displayed. The server searches resource names, descriptions, owner, tags, captions, comments, and other information.
Search results are listed by relevance and categorized by the type of resource. Click See All to see all search results for each resource type.

---

3 Learn about the search results

Search results show the number of views and favorites for the resource. Hover over an item in the search results list to see a tooltip that tells you more about it.
4 Go directly to the item

Click an item in the list to go directly to it. You can click a view to open it, without having to go to the workbook first.
Quick Start: Discover Popular Content

Find the right content quickly on Tableau Server and Tableau Online through metrics-based search results and sort options. List view and thumbnail view display metrics as well.

Relevant Search

Open your browser and navigate to the site on the server that you want to search. Enter a search keyword. Tableau Server displays the top three resources that match what you’ve entered, based on relevance. To see all results, click See All.
Search results are automatically sorted by relevance. See All results are also sorted by relevance.

For workbooks and views, relevancy is based on number of views, recent activity, favorites.

For published data sources, relevancy is based on number of workbooks connected to the data source, number of views of workbooks connected to the data source, recent activity, and favorites.

Note: There can be a delay between when users view content and when the view count is refreshed.

You can also use Favorites search to find favorite workbooks, views, and published data sources. Results are listed in alphabetical order (not by relevance).
Metrics-based Sort and View

Sort workbooks and views by the number of views (all-time, last 1 month, last 3 months, or last 12 months). You can sort available columns from most to least relevant or vice versa.

In thumbnail view, sparklines show historical data on the popularity of workbooks and views for the past 12 weeks.
Sort data sources by the number of workbooks or views connected to a data source, or by Live / Last Extract, modified date, and alerts. You can sort available columns from most to least relevant or vice versa. Data source connections can be sorted by connection type or data source.
View, Sort, and Select Content

Tableau Server gives you different options for viewing and sorting your content. As you find content of interest, you can select it to perform actions on it, such as setting permissions or changing the owner.

View Content in Lists or Thumbnails

As you browse Tableau Server you will notice that items are either displayed as thumbnails or in lists. Icons at the top of Projects, Workbooks, and Views pages let you specify whether contents should be displayed as lists or as thumbnails. Click the thumbnail or list icon to toggle the view.

This image shows thumbnail view selected.

![Thumbnail View](image1.png)

This image shows list view selected.

![List View](image2.png)

Sort Content

Depending on the type of content displayed on the page, you can sort by different characteristics, such as by number of views, name, order of sheets, owner, project, creation or
modification date, and so on.

To sort in thumbnail view, click the **Sort By** drop-down arrow, and then select the sort option. This example shows the sorting options for views.

You can also sort a list by clicking the column heading. An arrow indicates the order: ascending (up arrow) or descending (down arrow). To sort using multiple options, click the sort type in inverse order. For example, to sort on project first, workbook second, and name last, click those sort types in inverse order (name, workbook, project).
Select Content and Use Actions

On the **Projects, Workbooks, Views, and Data Sources** pages, click a selection box on a thumbnail or list item to select content and perform various actions, such as to tag content or to assign permissions. Click the **Actions** menu (….) to access commands available for the selected content.
Notes on selecting multiple items
When you select multiple items, the Actions menu may display commands that don’t apply to one or more items in the set of selected items.

When you select multiple items and apply an Action command to those items, Tableau Server will attempt to perform the action on all of the items and notify you of the results (whether the action succeeds or not) for the various items.

To select all content on the page, click the drop-down n selected arrow, and then click Select All. To deselect content, click the drop-down arrow, and then click Clear.

Search Content
You can search for content on Tableau Server in a variety of ways:

- **Quick search.** Use quick search to search the entire site for resources such as projects, workbooks, views, published data sources, and users. The results match what you’ve entered, based on relevance. For workbooks and views, relevancy is based on number of views, recent activity, favorites. For published data sources, relevancy is based on number of workbooks connected to the data source, number of views of workbooks connected to the data source, recent activity, and favorites.

- **Filtered search.** Use filtered search to find site resources by using a combination of
search criteria that are specific to the resource, such as name and tag for a project, and owner and date last modified for a workbook. The search results show relevant items for each content page. Click the Projects, Workbooks, Views, and Data Sources menus to see the results for each resource type.

- **Favorites.** You can quickly find any workbook or view that you’ve marked as a favorite. For information, see Mark and Search Favorites on page 70.

Both quick search and filtered search support attributes and operators that help you set the scope of the search. For example, limit a search to just titles of views, or use operators like and and not.

**Note:** Search in Tableau Server is not case sensitive.

**Quick search**

You can use the quick search field at the top of the page to find resources anywhere in the site.

When you use quick search, you are searching for resource names, descriptions, owner, tags, captions, comments, and other information across all resources on the site. As you type, a list appears, showing the resources that match your search text, based on relevance. To see all results, click See All.
When you select a resource from the list, Tableau directs you to that resource. For example, if you select a view, you are taken to the view, rather than the workbook that contains the view.

**Note:** Quick search restricts searches to published data sources only. To search on embedded data sources (workbooks that use a data connection only), use a filtered search.

### Filtered search

With filtered search, you can use a combination of search text and filters to find your content.

Each content page has different filters available that are relevant to the resource type. When you enter a search term and select General Filters, those search options apply to every content type. For example, you could type "sales" and you could get relevant results for that term in the **Projects**, **Workbooks**, **Views**, and **Data Sources** pages.

1. Go the page for the resource type you want to search and filter.
   
   For example, if you want to search for data connection information, click the **Data Sources** menu, and then use the Data Source Filters section of the Filters pane.
The list of filters changes to match the resource type.

2. Enter text in the main search field, and then press the **Enter** key.

3. Click a filter, and enter a search term for filter value. Select a filter value, or select from other filter options that are available.

To remove a filter, click **Clear** at the top of the Filters pane.

To remove all filters, click **Clear All Filters** at the top of the Filters pane.
Click the Filters button to collapse or open the Filters pane and reset the filters to their defaults.

The toggle is a dark icon ( darken ) when the Filters pane is open and a light icon ( lighten ) when the Filters pane is closed.

**Search attributes**

In addition to a general search, you can limit search on Tableau Server to a specific attribute such as name, workbook, data source, and so on.

To include a search attribute, use the following syntax in the search box:

\[
<\text{attribute-name}>: \text{search-text} \quad \text{(with no spaces on either side of the colon)}
\]

For example, to return only items whose names contain the words *sales* or *projections*:

name: sales projections.

You can include multiple attributes to further limit a search. For example, to find all dashboards that are owned by Smith, you can type the following into the search field:

sheettype: dashboard owner: smith

A complete list of attributes is shown below.

<table>
<thead>
<tr>
<th>This attribute...</th>
<th>Followed by...</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>name:</td>
<td>search term</td>
<td>Items with names that match the search term</td>
</tr>
<tr>
<td>title:</td>
<td>search term</td>
<td>Views with titles that match the search term</td>
</tr>
<tr>
<td>caption:</td>
<td>search term</td>
<td>Applies to views with captions.</td>
</tr>
<tr>
<td>owner:</td>
<td>user name</td>
<td>Items that are owned (published) by the specified users.</td>
</tr>
</tbody>
</table>

**Note:** Prior to 8.2, owners were listed as publishers in Tableau Server. The publisher...
<table>
<thead>
<tr>
<th>This attribute...</th>
<th>Followed by...</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>search attribute is still supported and returns the same results as the <code>owner</code> attribute.</td>
</tr>
<tr>
<td>publisher:</td>
<td>user name</td>
<td>(See owner above)</td>
</tr>
<tr>
<td>project:</td>
<td>search term</td>
<td>Items that are part of a project whose name matches the search term</td>
</tr>
<tr>
<td>comment:</td>
<td>search term</td>
<td>Views whose comments match the search term</td>
</tr>
<tr>
<td>tag:</td>
<td>search term</td>
<td>Items whose tags match the search term</td>
</tr>
<tr>
<td>field:</td>
<td>search term</td>
<td>Views with matching fields on the rows, columns, level of detail, pages, or encoding shelves</td>
</tr>
<tr>
<td>sheettype:</td>
<td>view, dashboard, or story</td>
<td>Views that are of the matching sheet type</td>
</tr>
<tr>
<td>class:</td>
<td>type of data source (e.g., mysql)</td>
<td>Views and data sources that are associated with the matching type of data source</td>
</tr>
<tr>
<td>dbname:</td>
<td>name of database</td>
<td>Published data sources that are associated with the matching data source</td>
</tr>
<tr>
<td>nviews:</td>
<td>number</td>
<td>Workbooks that contain the specified number of views</td>
</tr>
</tbody>
</table>

**Note:** Quick search restricts the searches of `dbname` to published data sources only. To search on embedded data sources (workbooks that use a data connection only), use filtered search.

**Search operators**

You can use `and`, `or`, `not`, and `*` with search phrases, including combining them with attributes, to build search expressions. For example, if you want to search for all items that do not match a specific phrase, or to match one phrase or another but not necessarily both.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Definition</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>and</code></td>
<td>Returns items that match both search terms.</td>
<td>sales and marketing; pens and paper</td>
</tr>
<tr>
<td><code>or</code></td>
<td>Returns items that match either search</td>
<td>west or east; soccer and foot-</td>
</tr>
<tr>
<td>Operator</td>
<td>Definition</td>
<td>Examples</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>term.</td>
<td>term.</td>
<td>ball</td>
</tr>
<tr>
<td>not</td>
<td>Excludes items that match the search term following this operator</td>
<td>not sheettype:dashboard</td>
</tr>
<tr>
<td>*</td>
<td>Acts as a substitute for any character or word following or as part of the search term. This operator can be used by itself or at the beginning or end of the search term. This operator is useful when you don’t know the exact term you are searching for.</td>
<td>dev* sales*</td>
</tr>
</tbody>
</table>

**Spaces and search**

If your search phrase includes spaces, punctuation, or reserved terms such as and, or, or not, enclose your search phrase in double quotation marks.

**Mark and Search Favorites**

To make it easy to find your most-used workbooks, views, or published data sources, you can mark them as favorites.

**Note:** You cannot mark an embedded data source as a favorite.

**Mark a workbook, views, or published data source as a favorite**

- Select the favorites star next to the content.
Alternatively, from the list view, create a favorite by clicking the star next to the workbook, view, or published data source.

<table>
<thead>
<tr>
<th>Name</th>
<th>Views: All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tale of 100 Start-ups</td>
<td>5</td>
</tr>
<tr>
<td>Economic Indicators</td>
<td>4</td>
</tr>
<tr>
<td>Investing in the Dow</td>
<td>0</td>
</tr>
<tr>
<td>Volatile Year for Technology</td>
<td>0</td>
</tr>
</tbody>
</table>

The content is added to your Favorites menu.

Remove an item from Favorites

- Click the star that you used to mark the favorite.

Search Favorites

- Click the Favorites menu in the upper-right corner of the page. Type a search term to filter the list.
On the Favorites menu, indicates a view, indicates a workbook, and indicates a published data source. Results are listed in alphabetical order. If you have a large number of favorites, type a search term to filter the list or scroll to see all of them.

**Access and Manage Your Content**

You can see available projects, workbooks, views, and data sources in Content pages in Tableau Server. In each page, you can browse thumbnails or lists for content, use the Filters pane to search and filter available content, or use quick search. For related information, see [Search Content on page 64](#) and [Navigate Tableau Server on page 27](#).

You can also view the content you own and your subscriptions on your private content page. The content and options you see on these pages are based on your site role and the permissions for the item you are viewing.

**To navigate to content**

- **Users:** In a site, click the **Projects, Workbooks, Views, and Data Sources** menus to browse content in these pages.
Administrators: In a site, click Content, and then click the Projects, Workbooks, Views, and Data Sources menus to browse content in these pages.

To see the content you own, click your name at the top of the page, and then click My Content.

**Note**: For information on the menus and pages available to administrators, see Navigate Server Admin Pages on page 662 and Navigate Site Admin Pages on page 318.

**Use content pages**

Projects, workbooks, and data sources each have their own content pages with options that are specific to that content type. For example, a workbook content page includes menus for the views in the workbooks, data sources connected to the workbook, subscriptions, permissions settings, and details.
Each content page provides information such as the owner, and number of views and favorites.

You can see actions that you’re allowed to take on the content.
A breadcrumb trail at the top each content page shows your location in the site.

Each content item has a Details page. For example, with the workbook selected, you can click Details to open the details page. If you own the content, you might be able to update the description (use the About command), move the workbook to another project, change the owner, edit tags, manage its revision history, and select whether users can navigate between views in the workbook.
Projects

A project page includes the workbooks, views, and data sources in that project, as well as the default project permissions and project details. Open a project to see its content page.
Workbooks

A workbook page includes the views and data sources used by the workbook, and the workbook permissions and details. Open a workbook to see its content page.

Views

The Views page shows the view and the actions available to you for interacting with and sharing that view. In a workbook, click a view open it.
Data sources

The Data Sources page includes the connections for the data source, refresh schedules, connected workbooks, permissions, and details. In a site or a project, click Data Sources, and then open a data source to go to its content page.

Use your content page

Your content page includes your user information, the workbooks and data sources that you own, your subscriptions, and your account settings.

Click Settings to go to your account settings from your content page.
Select content and perform actions on it

On the Projects, Workbooks, Views, and Data Sources pages, click a selection box on a thumbnail or list item to select content and perform various actions, such as tagging content or assigning permissions. Click the Actions menu (… ) to access commands available for the selected content.
**Note:** When you select multiple items, the **Actions** menu might display commands that don't apply to one or more items in the set of selected items. When you select multiple items and apply an action to those items, Tableau Server will try to perform the action on all of the items and notify you of the results (whether the action succeeds or not).

To select all content on the page, click the drop-down $n$ selected arrow, and then click **Select All**. To clear a content selection, click the drop-down arrow, and then click **Clear**.

If many items are available on a page, click the **Actions** menu ($\ldots$) to select the item and see the available commands.
In thumbnail view, the **Actions** menu is displayed in the right corner of the thumbnail.

In list view, the **Actions** menu is displayed to the right of the workbook name.
Create or Interact with Published Views

This section contains topics for people who can sign in to a Tableau Server site to work with published views in the Tableau browser environment. As someone who interacts with views, you might be a Tableau Desktop user as well, or your job might depend on the data analysis published by Tableau Desktop users.

On Tableau Server some of the ways you can work with views include the following:

- Use the interaction tools to learn about the data presented in the view. For example, select a mark to view information specific to that selected segment of the view; email a view to others; subscribe to receive email updates; get code that you can use to embed a view into a webpage or blog post.

  If filters are provided, you can use them to present the data in a different way, and you can even save custom variations of the view that you find particularly compelling.

- Edit existing workbooks that are published to your Tableau Server site, or connect to published data sources to create new workbooks.

  When you publish views to Tableau Server, you can continue analyzing your data even when you are working remotely and don't have access to your Tableau Desktop installation.

The specific capabilities you have access to depend on the permissions your Tableau Server site administrator assigned to you when setting up your account.
Interact with Views

When you find content that interests you, you can view and interact with the data in many different ways, depending on the content you are allowed access.

With the view open, you can choose from options such as sharing, creating custom views, exporting, downloading, subscribing, and editing the view. You can also interact with the view to explore its data. If you have web editing permissions, you can edit existing workbooks or create new workbooks from data sources.

Quick Start: Share Views

Easily share your published views with others. Click Share in a workbook or view to create links that you can embed in a blog or webpage or email to a friend or co-worker.

1 Open the share options

Click the Share button in the upper right corner of the view.

The Share button may be below the view if it is embedded in a webpage or you are logged in as a Guest user.

2 Email the view

Copy the URL in the Link field and paste it into an email or click the envelope icon in the lower left corner of the dialog box to automatically open a blank email that includes the link to the view in the email body.
Make sure that the people you are sharing with have access to the server and the right permissions to see the view.

3 Embed the view

Copy the HTML code in the Embed Code field and paste it into your blog or webpage. The interactive view will display inline on the page.

Share Views

Every published view and workbook can be shared via email or embedded into another webpage, wiki, or web application. Anyone viewing a shared view must have an account on Tableau Server and permission to access the view.

Email a view

1. Click Share in the upper-right corner of the view.
2. Copy and paste the provided link into your email message or click the envelope icon in the lower left corner of the dialog box to open a blank email that includes the link to a view in the email body.

Embed a view

You can share a view by embedding it into another webpage such as your wiki, blog, or web application.

1. Click **Share** in the upper-right corner of the view.
2. Copy the provided embed code, and then paste it into the source code of the page in which you want to embed the view.

**Note:** The embed code generated by Tableau will automatically refer to the current view. For information about how embedded custom views are displayed in Tableau, see **Embed Code for Custom Views.**
**Download Views**

You can download a view in the following formats:

- Image
- PDF
- Crosstab. This option opens a file in Microsoft Excel.
- Data (a data file). This option opens a new tab in the browser window and displays the data in summary and in detail. You can also download the data from this view as a comma-separated value (.csv) file.
- Tableau Workbook. For information about how to download a workbook, see Download Workbooks.

Some formats may not be available to you if you do not have permission to download that format.

**Note:** Downloading large amounts of data can affect server performance or not complete successfully. If you encounter these issues, try exporting the data directly from the underlying data source.

**To download a view**

- On the toolbar at the top of the view, select an option on the Download menu.

If you are downloading a dashboard to a PDF and the dashboard includes a web page object, the web page object is not included. Also, when you select a download option, the image, PDF, or crosstab must be generated. A dialog box opens when it is done generating so you can continue downloading the file.
To download a view as a PDF

1. Open a view, click Download on the toolbar, and select PDF from the context menu.

2. Select either a Portrait or Landscape orientation and a Paper Size. Under Content, select the part of the workbook to download (current dashboard, selected sheets in the dashboard, or selected sheets in the workbook). Then, under Sheets to Download, select the specific sheets to download. Click a sheet thumbnail to select or deselect the sheet or press Ctrl and click to select multiple sheets.
Selected sheets are indicated with a blue highlight.

3. Click **Download**, then, in the **Download PDF** dialog box, click **Download** again.

4. Open the downloaded file.

Quick Start: Subscribe to Views

Subscribe to a view to receive a snapshot of that view by email periodically—without having to sign in to Tableau Server. When you open a view in Tableau Server and you see a subscription icon (✉️) in the upper-right corner, you can subscribe to that view or to all of the views in the
workbook. You can subscribe yourself to views. In addition, workbook owners, project leaders, and administrators can subscribe other users. For more information, see **Subscribe to Views** on page 930.

**Note:** Server administrators determine whether subscriptions are enabled for a site, and they create the subscription schedules that are available to users.

1 **Open a view**

In a site, click **Views** or **Workbooks**, open a view, and then click **Subscribe**.

Enter the subject for the email message. Select a schedule. Click **This View** to send a snapshot of the current view, or **Entire Workbook** to send an email with snapshots of every view in the workbook.
2 Subscribe yourself

Click **Subscribe me** to subscribe yourself, and then click the **Subscribe** button. Enter your email address if needed.

If you don't own the workbook, you will not see the **Subscribe me** and **Subscribe Others** options. You will only need to click the **Subscribe** button.

3 Subscribe others

If you own the workbook (or you are the project leader or an administrator), click **Add** to subscribe other people. For each user you want to subscribe, type the initial letters of each name, and then select the name from the results list. When you are done adding subscribers, click **OK**, and then click the **Subscribe** button.
If a user doesn't have permission to view the content, their subscription will not be saved.

Click **Manage** under **Current Subscribers** to pick a different schedule, to change the email subject line, or to unsubscribe a user.

**4 Open the view from email**

When you receive your email, click the snapshot of the view in your email to open it in Tableau Server.
An email with a snapshot of the subscribed view is sent at the scheduled time.

To unsubscribe, click **Manage my subscriptions** in the email. For more information, see **Unsubscribe yourself from a view** on page 935.

**Subscribe to Views**

When you open a view in Tableau Server and a subscription icon is available in the upper-right corner, you can subscribe to that view or to all of the views in the workbook.

**Note:** Server administrators determine whether subscriptions are enabled for a site, and they create the subscription schedules that are available to users.

When you subscribe to a view or to all of the views in a workbook, a snapshot of that content is automatically delivered to you periodically via email. You don't need sign in to Tableau Server to see it.

If you are the owner of the workbook, you can also **subscribe other users** to your workbook and its views. If you’re a project leader, you can subscribe other users to workbooks and views in your projects. If you’re an administrator, you can subscribe users to any workbooks or view in a site.
You can also choose to unsubscribe from views you no longer want to receive.

Subscribe yourself to a view

1. In a site, click Views or Workbooks.

2. Open a view, or open a workbook and then open one of its views.

3. Click Subscribe in the toolbar.
If you don't own the workbook, you will not see the **Subscribe me** and **Subscribe Others** options.

4. If your Tableau Server account doesn't already have an email address, enter your email address.

5. To subscribe to the current view, click **This View**. To subscribe to all views in the workbook, click **Entire Workbook**.

6. Pick a schedule, enter a subject line for your email, select **Subscribe me**, and then click the **Subscribe** button.

   **Note:** If you don't own the workbook, you will not see the **Subscribe me** or **Subscribe Others** options. You will only need to click the **Subscribe** button.

When you receive the subscription by email, click the snapshot of the view to open it in Tableau Server.

   **Note:** If a dashboard size is set to **Automatic**, the image included in the subscription email is fixed at 800 pixels by 600 pixels.

---

**Subscribe others to a view**

If you are the owner of the workbook, you can subscribe other users to your workbook and its views. If you're a project leader, you can subscribe other users to workbooks and views in your
If you're an administrator, you can subscribe users to any workbooks or view in a site. If a user doesn't have permission to view the content, their subscription will not be saved.

1. In a site, click **Views** or **Workbooks**.
2. Open a view, or open a workbook and then open one of the views.
3. Click **Subscribe**.
4. Select the current view (**This View**), or to include all views in the workbook, select **Entire Workbook**. Pick a schedule and enter a subject line for the email that users will receive.
5. Under **Subscribers Others**, click **Add**.

---

**Add Subscribers**

**Jane Johnson**

**agar**

Ashley Garcia

[Add button] [Cancel button] [OK button]
For each user you want to subscribe, type the initial letters of each name, and then select the name from the results list. When you are done adding subscribers, click **OK**, and then click the **Subscribe** button.

**Note**: Subscriptions can't be added for subscribers who do not have email addresses in their Tableau Server account. Also, if a user doesn't have permission to view the content, their subscription will not be saved.

6. To change the subscription schedule or email subject line for a subscriber, or to unsubscribe a user, click **Manage** under **Current Subscribers**.
Unsubscribe yourself from a view

1. Open your account settings on Tableau Server in one of the following ways:
   - Click the Manage my subscriptions link at the bottom of a subscription email.
   - Sign in to Tableau Server, select your name, and then from the drop-down list, select My Content.

2. Click Subscriptions.

3. Select the check box next to the view you want to unsubscribe from, click Actions, and
then click **Unsubscribe**.

![Unsubscribe Action](image)

You can also change your subscriptions in your users settings page, such as selecting a different schedule or changing the email subject line. For more information, search for "Manage Your Subscription Settings" in the Tableau Server Help.

**Quick Start: Keep Content Revisions**

Every time you publish a workbook or data source to Tableau Server, Tableau can save a version of that content in its revision history. If you want to revert to a previous version, you can go to the workbook or data source in Tableau Server, view its revision history, and restore that version.

**Note:** A server administrator must enable **Revision History** in **Site Settings** to make this feature available.

1 **Publish your content**

In Tableau Desktop, click **Server > Publish Workbook** or **Server > Publish Data Source**. Make changes to the workbook or data source, and then publish it again to the same project, with the same name. Workbook revisions are also saved when a content owner edits and save a workbook in a project on Tableau Server.
Because your content has the same name, you need to confirm that you want to overwrite the workbook or data source when you publish the workbook or data source.

2 View revision history

Sign in to Tableau Server. Select the workbook or data source, and then in the actions menu (…), click Revision History.
Workbooks include the option to preview past versions.

3 Restore a previous version of a workbook

In the revision history for the workbook, select the revision, and then click Restore.
The restored version becomes the current version.

4 Restore a previous version of a data source

Select and then download the data source. Open the downloaded file in Tableau Desktop, and then republish it with the same name, to the same location in Tableau Server. This makes it the current revision of that data source.
More about revision history

- To access revision history, a user must have a site role of Publisher, plus the following permissions:
  - In the project: View and Save
  - Workbooks in the project: View, Save, and Download Workbook/Save As
  - Data sources in the project: View, Save, and Download Data Source

- Workbooks and data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you might need to make changes in the downloaded workbook or data source.

- You can delete specific revisions from revision history. Server administrators can clear all revisions for every workbook or data source in a site. The most recent revision of each published workbook and data source is always retained.

- When a workbook or data source is deleted from a site, all previous revisions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the data source are saved.

- If revision history has been turned on and then turned off, saved revisions are still retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

- Published workbooks can be previewed and restored online, in Tableau Server, or can optionally be downloaded, opened in Tableau Desktop, and then republished to the same location, using the same name. For more information, see Manage Workbook Revisions on page 945.

  Published data sources must be downloaded, opened in Tableau Desktop, and then republished to be restored. For more information, see Manage Data Source Revisions on page 950.

Manage Workbook Revisions

When revision history is enabled on your site and you publish a workbook from Tableau Desktop or save a workbook on Tableau Server, Tableau Server saves a version of that content in its revision history. Each time you publish or save that workbook again in the same location, with the same name, another revision is saved.

You can restore previous revisions of workbooks online, in Tableau Server. Optionally, you can download a revision, open it in Tableau Desktop, and then republish the workbook to the same location in Tableau Server, using the same name.

Permissions for revision history

To access revision history, you must have a site role of Publisher, plus the following permissions:
- In the project: **View** and **Save**
- Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**

**View revision history**

- In Tableau Server, select a workbook, and then click **Revision History** in the actions menu ( . . . ).

**Preview a revision of a workbook**

1. Select a workbook, and then click **Revision History** in the actions menu ( . . . ).
2. In the revision history, click **Preview** in the actions menu ( . . . ).
If a preview is available, it is displayed from Tableau Server in a new tab in the browser.

3. Click **Close Preview** in the preview page to return to the **Revision History** dialog box.

**Note:** If a workbook cannot be previewed online, you can download the workbook, and open it in Tableau Desktop to preview it.

### Restore a revision

1. Select a workbook, and then click **Revision History** in the actions menu ( . . ).
2. In the revision history, select a revisions, and then click **Restore**.

The restored version becomes the current version.
Note: If the revision can't be restored online, you can download the workbook, open it in Tableau Desktop, and then republish it to make it the current version.

About restoring workbooks that require credentials
When you restore a workbook that uses a live connection and prompts for a user name and password, you have the option to embed the credentials for the connection. If the workbook uses a data source with multiple connections, you might need to provide a user name and password for each connection that prompts for credentials.

Extracts with embedded credentials and scheduled refreshes
When you restore an extract that uses embedded credentials and scheduled extract refreshes, you will need to edit the data connection and provide your credentials as part of the restore process. This ensures that the workbook has the credentials it needs to complete the refresh successfully.

1. Restore the workbook that uses embedded credentials and has scheduled extract refreshes.
2. Go to the Data Source page for data source used by the workbook.
3. Click the Connections tab.
4. For each connection in the data source, select the connection and click Actions > Edit Connection.
5. Enter (or keep) the connection information, and then click Save.

Workbooks with OAuth connections
When you restore a workbook that uses an OAuth connection, you cannot preview the workbook. If you are not able to restore the workbook in Tableau Server, download the workbook and then republish it to make it the current revision.

Download a revision
1. Select a workbook, and then click Revision History in the actions menu (. . .).
2. In the revision history, click Download in the actions menu (. . .).
3. Open the workbook file in Tableau Desktop.

You can also republish the with the same name, in the same location in Tableau Server, to make it the most current version.

**Note:** When you publish the workbook from Tableau Desktop, because your content has the same name, you will have to confirm that you want to overwrite the workbook. Proceed by clicking **Yes**.

**Delete a revision from history**

You can delete any previous version of a workbook from its revision history. You cannot delete the most current revision except by deleting the entire workbook.

1. Select a workbook, and then click **Revision History** in the actions menu (. . .).
2. In the revision history, click **Delete** in the actions menu (. . .).
The revision history list updates to indicate a revision has been deleted.

More about revision history

- If a different author publishes over a workbook with the same name, the most recent author becomes the owner and can see its entire revision history.

- Workbooks are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a workbook is deleted from a site, all previous versions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the workbook are saved.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

Manage Data Source Revisions

When revision history is enabled on your site and you publish a data source, Tableau Server saves a version of the data source in its revision history.

To restore a published data source, you download it, open it in Tableau Desktop, and then republish it to the same location in Tableau Server, with the same name.
Permissions for revision history

To access revision history, you must have a site role of Publisher, plus the following permissions:

- In the project: View and Save
- Data sources in the project: View, Save, and Download Data Source

View revision history

- Select a data source, and then click Revision History in the actions menu ( . . ).

Download and restore a revision

1. Select a workbook or data source, and then click Revision History in the actions menu ( . . ).
2. In the revision history, click Download in the actions menu ( . . ).
3. Open the data source file in Tableau Desktop, and then republish it with the same name, to the same location in Tableau Server. The uploaded version becomes the most current version.

**Note:** When you publish from Tableau Desktop, because your content has the same name, you will have to confirm that you want to overwrite the data source. Proceed by clicking **Yes**.

**Delete a revision from history**

You can delete any previous version of a data source from its revision history. You cannot delete most current revision except by deleting the entire data source.

1. Select a data source, and then click **Revision History** in the actions menu (…).
2. In the revision history, click **Delete** in the actions menu (…).
The revision history list updates to indicate a revision has been deleted.

More about revision history

- If a different author publishes over a data source with the same name, the most recent author becomes the owner and can see its entire revision history.

- Data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook or data source.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a data source is deleted from a site, all previous versions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the data source are saved.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

Tag Content

Tags are keywords you can create for workbooks, views, and data sources to help you find, filter, and categorize content in Tableau Server. Authors can add tags to content when they publish the content. After content is published to a Tableau server, you can add tags to any workbook, view, or data source that you are allowed to access, and you can delete any tags you have added.
View tags

In a published workbook or view, its tags appear in the tooltip when you hover over the thumbnail. You can also view tags in the Tags area located below a view.

Click a tag link to filter the view to any content that uses that tag.

Find tagged content

Use tags to find content with quick search and filtered search.
Add tags

You can add tags to any view or data source on the server that you have permission to access.

To add tags to a single view

In the Tags area below the view, click **Edit Tags**, and then enter one or more words in the text box. Click **Save** when you are done.

![Tags interface](image)

To add tags to multiple views

If you have an Interactor-level access, you can also add tags to multiple views at once.

1. Navigate to a list of views or workbooks.
2. Select the views you want to tag, and then select **Actions > Tag**.

![View interface with tag options](image)
Enter one or more words in the text box, and then click **Save Tags** when you are done.

**Change Tags**

Change the tags for view “Tale of 100 Start-ups”.

DOW  ×  market  ×  returns  ×  stocks  ×  
Add a tag

**To add tags to a data source**

In the Data Source page, select a data source, and then select **Actions > Tag**. Enter one or more words in the text box. Click **Save Tags** when you are done.

**Delete tags**

You can delete any tags that you have created.
1. Open the tagged view.
2. Scroll to the Tags area below the view.
3. Click the 'x' next to the tag you want to remove.

**Download Workbooks**

Workbooks can be downloaded using the **Download** button (ダウンロード) in a view. The downloaded workbook can be opened with a version of Tableau Desktop. Downloading the workbook from the server is the same as selecting **Server > Open Workbook** in the desktop application.

This option is only available if you've been given the **Download/Save** permission by the author of the workbook or an administrator.
Pause Automatic Updates

As you interact with the view on the server, it will sometimes have to send a query to the data source to update the data in the view. If you are working with a dense view with a lot of data or a very large data source, the automatic update may take a long time. To avoid waiting for each update while you make several changes you can click Pause on the toolbar to pause automatic updates.

When you Resume automatic updates using the same toolbar button you only have to wait for a single query to the data source.

Refresh Data

If the data source is changed, such as new fields have been added or data values and field names have been modified, the view will reflect those changes the next time you load the page. However, you may need to manually update the view using the Refresh Data button on the toolbar.

When you refresh the data, you clear any cache that may exist and retrieve the latest data from the data source. This option is different than the Pause Automatic Updates above option, which still may load the view based on cached data. Depending on the size of your data source and the view, refreshing the data may take longer than other queries that operate on cached data.

Sort Data

You can quickly change the sort order for items in a view using the Sort buttons on an axis, field label, or header. For example, the view below shows oil output by well. If you mouse over the vertical axis, a sort button appears that displays the sort it will perform when you click it—in this case, an ascending sort:
Once you sort, the sort icon stays on the vertical axis:

Click the Sort icon again, and it sorts in descending order, from lowest output to highest output:

A third click returns the bar chart to its original state:

You can always revert to the entire view’s original state using the Revert button on the toolbar.
Filter Data

With an Interactor license level, you can filter data in a view so you can narrow the visualization to the data of interest. For example, a regional sales report may show sales for several different regions. You can focus on how your region is doing using filters. Then you can revert the filters to return to the original view and compare your region to others. There are two ways to filter the view: filters in the view and the Keep Only and Exclude tooltip commands.

Types of Filters

There are many different types of filters. The author of the view chooses the type of filter that best shows the values you are filtering. Each type of filter and how you can interact with it is described below.

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Multiple Values List</strong></td>
<td>Shows a list of check boxes. Select the values to include.</td>
</tr>
<tr>
<td><strong>Single Value List</strong></td>
<td>Shows a list of radio buttons. Select one value at a time to include.</td>
</tr>
<tr>
<td><strong>Compact List</strong></td>
<td>Shows a drop-down list of values. Select one value at a time to include.</td>
</tr>
<tr>
<td><strong>Slider</strong></td>
<td>Shows values along a range. Drag the slider or use the arrows to select</td>
</tr>
<tr>
<td><strong>Filter Type</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>a single value to include. For quantitative values use the two sliders to specify a range of values.</td>
<td></td>
</tr>
<tr>
<td>Wildcard Match</td>
<td>Shows a text field. Include all values that contain a specific set of characters. You can use the asterisk symbol (*) as a wildcard character.</td>
</tr>
<tr>
<td>Multiple Values</td>
<td>Shows a text field. Type a value and click the plus symbol to include it in the filter. Alternatively, copy and paste a list of values into the text field.</td>
</tr>
<tr>
<td>Date Filters</td>
<td>Date filters can be shown in most of the filter types described above. However, there are a few</td>
</tr>
</tbody>
</table>
**Filter Type** | **Description**
--- | ---
| | more types that are specific to dates. For example, relative date filters allow you to include common date ranges such as “last 3 weeks” or “year to date.” Or you can select from a collection of predefined date periods such as 1 day, 1 week, 3 months, etc.

**Multiple Value List (Hierarchical)**

If the view uses a multidimensional data source, the multiple value list shows the hierarchical levels. Use the level selectors at the top to select all values at the given level.

**Filtering Options**

Many filters have additional options that display when you hover the pointer over the filter area on the page. For example, sometimes it is easier to select what you don't want to see than all the things that you do want. Options on the drop-down menu for a multiple value list let you switch between an **Inclusive** and **Exclusive** filter. Each type of filter has its own set of options available in the drop-down menu in the upper right corner of the filter area. The availability of these options is controlled by the author of the view.
In addition, if a filter contains a lot of values, you can use the search to quickly find and select what you’re looking for. Click the **Search** button that shows when you hover the pointer over the filter in the view. Then start typing the value you are looking for. The results show directly below the search box making it easy to select.

Finally, you can limit the values shown in a filter using the **Show More Values/Show Fewer Values** button. When you show more values, all values from the field are displayed. When you show fewer values, the other active filters are considered. Only values that pass all of the filters are shown. For example, a view may be filtered to show sales greater than $5000 in the Western and Central regions. When you show more values, the Region filter will show all regions. When you show fewer values, the Region filter will only show regions where sales are greater than $5000.

This option shows only if the author of the view turns it on for the filter.

**Reverting Filters**

If at any time you want to restore the filters to how they were when the view was published, use the **Revert** button. This action reverts the entire workbook to how it was when it was last published, so any work you did in the workbook may also be reverted.
Explore Data in Maps

Tableau maps can help you quickly find locations and analyze data worldwide. There are many ways you can explore and interact with map views. You can zoom in and out, pan, and select marks with the view toolbar, and even search for locations worldwide with map search.

![Map View](image)

If you don't see the view toolbar or the map search icon, or if you can't pan or zoom, it's probably because the workbook author has chosen to customize how others can interact with the view.

The workbook author can choose to hide the view toolbar and the map search icon so they do not appear in the view. Similarly, the workbook author can choose to turn off pan and zoom to control how you interact with the view.

For more information, see Customize How People Interact with your Map.

Select marks, pan, and zoom

You can use the view toolbar in the upper left corner of the view to select marks, pan, and zoom in and out of a map view. For more information, see Select Marks and Pan and Zoom.

Search for locations

You can use the map search icon to search for locations in your map view. For more information, see Set Map Search Options.
Find your current location

You can use the locate me button to quickly pan and zoom to your current location. To do so, click the map search icon in the top-left corner of the view, and then, next to the search box that appears, click the locate me button.

Note: When you click the locate me button, you might be prompted by your web browser to allow Tableau access to find your current location. If you block access, Tableau will be unable to zoom to your current location.

Return to the initial view

After exploring, you can return to the initial view of your map. To do so, click the zoom home button on the view toolbar.

Explore Data via Tooltips

Tooltips are data details that appear when you hover over one or more marks in a view.
Click a mark to access options for filtering or removing marks, and viewing data:

- **Keep Only**: Displays only what you’ve selected in the view.
- **Exclude**: Removes what you’ve selected from the view.
- **View Data**: Opens a window displaying the data. You can view the summarized data or the underlying data.

To filter or exclude a group of marks, select them and click the action you want to perform.
To look at the summarized or underlying data, make a selection and click the View Data icon. Then do one of following:

- Click the **Summary** tab to view the aggregated data in the view or in your selection within the view.
• Click the **Full data** tab to view the values for each row in the data source—as restricted by the mark or marks you’ve selected.

![Table showing data](image)

• Select **Show all columns** to view all columns in the underlying data source, whether or not the data is in the view.

![Table showing data](image)

• Click **Download all rows as a text file** using the links at the top and bottom of the table. A CSV file is generated that you can open and view the data in an Excel spreadsheet.
Follow Hyperlinks

Many views contain hyperlinks to either outside URLs or other sheets. These hyperlinks are added by the author and can be useful for getting more information about a set of data points. Based on how the view was authored, hyperlinks can be launched in one of the following three ways:

- **Select** - Click a mark in the view.
- **Hover** - Rest the pointer over the mark in the view.
- **Screen tip** - The hyperlink is available in the screen tip that displays when you rest the pointer over a selection of marks.

Links to Web Pages

A view may link to an outside webpage to show more information about the data. For example, a map view might link from a specific location to show a satellite image. Or, a sales report may link from a selection of products to show the current inventory status on an internal intranet. When you click a link it takes you to the external webpage, which is outside of Tableau Server. Webpages can also be embedded into a dashboard. In this case, clicking a link loads the webpage in the same window.
Links to Other Views

In addition to linking to external webpages, some views will have hyperlinks to other views on Tableau Server. These links filter data in the target view based on your selection in the source view. For example, in the dashboard shown below, selecting a product in the list of Top Selling Products filters the map view to show where the product is sold.

Highlight Marks, Legends & Actions

Highlighting is a way to call attention to a subset of data in a view. There are several ways to highlight data on Tableau Server. You can use marks, legends, or actions, or you can explore
your data interactively using the Highlighter to search for data points in context.

Use Marks to Highlight

When you select one or more marks in a view, all other marks are dimmed so that your selection stands out. Select individual marks or click and drag the pointer to select a bunch of marks. Hold the Ctrl key on your keyboard as you select marks, or use an advanced selection tool to select multiple marks in the view. For more information on advanced selection tools, see View Toolbar in the Tableau Desktop Help.

![Bar Chart](image.png)

Use Legends to Highlight

You can use the legends (color, shape, and size) to highlight the marks that correspond to each item in the legend. For example, if a view is colored by product category, you can quickly highlight a certain type of product using legend highlighting. Legend highlighting is enabled by default. To highlight items in the view using the legend, do the following:

1. Select an item in the legend:
2. You can hold the Ctrl key on your keyboard to select multiple items in the legend.

To turn off legend highlighting, do the following steps:

1. In the top menu, click **Edit** to open the view in web editing mode.
2. In the tool menu, click the highlight icon and select either **Disable Workbook Highlighting** to turn off legend highlighting for the entire workbook, or **Disable Sheet Highlighting** to turn off legend highlighting for the specific worksheet.

---

**Highlight data in context**

For particularly large or data dense views, you might want to highlight marks or groups of marks while still maintaining the context of the data in your view. You can turn on the highlighter for discrete dimensions that are shown in your view and then use keywords to search for and highlight marks.
To turn on the highlighter for a field, do the following:

1. In the top menu, click **Edit** to open the view in web editing mode.
2. Right-click on a discrete field (dimension) that is included in the view and impacts the level of detail in the view. Then select **Show Highlighter** from the context menu.

3. **Click Done.**

   In the Highlighter field, enter key words or partial keywords to search for marks. You can
also hover over the items in the drop-down list to do ad hoc analysis.

Highlight Actions

Some views may have Highlight Actions, which highlight related data in one or more views based on your selection in a source view. Highlight Actions are primarily used in dashboard views where multiple views are shown at once. For example, in the dashboard view below there is a highlight action set up to highlight on Region. When West is selected in the scatter plot all the other views highlight the West category.
Depending on the how the view was authored, Highlight Actions can be launched either by selecting a mark, resting the pointer over a mark, or in the screen tip that displays when you rest the pointer over a mark for a period of time.

**Comment on Views**

You can add comments to any view you have access to on Tableau Server. You can also see any comments associated with a particular view.

Type your text in the *Comment* text box located below the view and click *Post Comment*.

![Comments (0)](image)

You can add formatting to your comment by inserting hyperlinks, bolding, italics, and underlining. Examples of how to add each of these types of formatting are shown in the table below.

<table>
<thead>
<tr>
<th>Format</th>
<th>What to Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperlink</td>
<td>“My Link”:<a href="http://www.tableau.com">http://www.tableau.com</a></td>
<td>My Link</td>
</tr>
<tr>
<td>Bold</td>
<td><em>Bold Text</em></td>
<td>Bold Text</td>
</tr>
<tr>
<td>Italics</td>
<td><em>Italic Text</em></td>
<td><em>Italic Text</em></td>
</tr>
<tr>
<td>Underline</td>
<td>+Underlined Text+</td>
<td>Underlined Text</td>
</tr>
</tbody>
</table>

**Republishing workbooks with comments**

If you want a workbook to keep its comments, use the same workbook name when you republish the workbook. If you republish a workbook that has comments with a new name, the comments are not retained with the new workbook.

**Quick Start: Custom Views**

If you regularly look at certain views and find you need to make the same changes each time you open the views, you can create custom views that "remember" your changes. Each time you open a custom view, it displays the information you want to see. You can then share custom views with other Tableau Server users who have permission to see the original view.
1 Create a Custom View

Open a view and make the changes you want to save. For example, modify filters, sorting, or zoom level. Click *Original View* on the toolbar and enter a name for your custom view. Select **Make it my default** to make the new view your default view. Click **Save** when you are done.

Your custom view is always displayed with the saved changes. Click a different view in the list to display it. The currently selected view is indicated by a gray background.

Your custom view does not change the original, but is related to it. If the original view is updated or republished, your view is also updated. If the original view is deleted from the server, your custom view is also deleted.

2 Share Custom Views

To make a custom view available for others to view it, select the **Make it public** option when you create the view.
The public view icon ( ⊘ ) indicates that the view is shared. The private view icon ( ⊘ ) in the Manage dialog box indicates that the view can only be seen by you.

As you browse views on the server, you can see and use custom views that other people have shared.

3 Manage Custom Views

To change the default view, edit a custom view name, delete a custom view, or make it private, click the view name and then click **Manage**.
The default view is indicated by ⬤. To change the default view, click ⬤ next to the custom view name in the Manage dialog box.
4 Delete Custom Views

Click the delete icon (🗑️) in the Manage dialog to delete the view.

A confirmation message appears asking if you are sure that you want to delete the custom view. Click Delete to remove the view from your custom views list.

Use Custom Views

If you notice you are making the same changes to a view every time you open it, you might want to consider saving the changes as a custom view.

For example, in a particular view, you might apply a filter to include only data relevant to you, or you might sort a view differently than how it was published. You might also want to keep different versions of the same view—for example, one with two filters selected, and another with only one selected. Each of these different configurations can be saved as a custom view.

A custom view is always displayed with the latest saved changes. A custom view does not change the original, but is related to it. If the original view is updated or republished, the custom view is also updated. If the original view is deleted from the server, the custom view is also deleted.

You can also choose whether your custom views are visible to other users (public), or only to you (private).

Save custom views
When you interact with a view, you have the option to save any changes you’ve made to the view (sorting, filtering) as a custom view.

If you make a change to the original view, the **Original View** menu in the toolbar indicates the view has changed with an asterisk. You can access custom views by clicking the **Original View** menu, or by clicking the name of the custom view in the same menu location.

Custom views are always associated with the original view. As the original view is updated or republished, customized versions of the view are also updated.

- If the original view is deleted from the server, its associated custom views are also deleted.
- If filters are removed from the original view and it’s republished, the filters will be unavailable in customized versions of the view.
• If filters are restored and the view is republished, customized versions of the view include the restored filters.

To save a custom view

1. Open the individual view that you want to customize.
2. Filter the data, change sort orders, highlight, zoom in or out, or make any other modifications.
3. Click **Original View** or the name of the currently selected custom view as it is shown in the toolbar. Enter a name for the custom view. Select whether you want it to be the default view, or if it should be public, and then click **Save**.

Access and manage custom views

To see the custom views available for a view, click the **Original View** menu, or click the name of the custom view in the same menu location.

This example shows the original view.
This example shows a custom view name in the same location as *Original View*.

The custom views that you have saved are displayed under *My Views*. Views created by other people, including the *Original View*, are listed under *Other Views*.

**To select a view**

- Click a view name in the list of views to display it. The currently selected view is indicated by a gray background.

**To change the default view**

The default view is indicated by ![default View](default.png). To change the default view, click ![default View](default.png) next to the custom view name in the Manage dialog box.

1. In a view, click *Original View* or click the name of the custom currently selected view as it is shown in the toolbar.

2. Click *Manage*.
3. Click next to the custom view name to make the view the default view. The icon will change to to indicate the view is now the default view.
4. Click **Done**.

**Advertise a custom view**

By default, your custom view is private so only you can see it on your list. If your site role is Interactor or Publisher, you can advertise the view to other users. Anyone who has access to the original published view will be able to see your advertised custom view.

To advertise a custom view, click Make it public when you first create the view. Even if you don't advertise your custom view, you can still share it by copying the URL or clicking **Share**.

**To change the public or private status of a custom view**

1. Click **Original View** or the name of the current view in the toolbar. Select the view you want to edit, and then click **Manage**.
2. In the Manage dialog box, click the public or private view icon to change the view status.

The public view icon (☺) indicates that the view is shared.
The private view icon (🔒) indicates that the view can only be seen by you.

3. Click **Done**.

**Make a custom view private**

You can always make a public view private. When a custom view is private, it no longer shows in the drop-down list for others and only you can see it on your list of custom views.

1. Click **Original View** or the name of the current view in the toolbar. Select the view you want to edit, and then click **Manage**.
2. In the Manage dialog box, click the public view icon (🔒) next to the view to make it private (🔐).

The public view icon (🔒) indicates that the view is shared.
The private view icon (🔒) indicates that the view can only be seen by you.

3. Click **Done**.

**Delete a custom view**

You can delete a custom view you created at any time. Removing your custom view does not affect the original view.

**Note:** If the original view is deleted from the server, your custom view is also deleted.

1. When you're looking at the custom view you want to delete, click the view name in the toolbar, and then click **Manage**.
2. Click the delete icon (🗑️) next to the custom view name.

3. Click Done.
Create Views and Dashboards

Users with the appropriate permissions for the web authoring environment can edit views and dashboards or create new ones.

A view is a single data visualization, such as map of sales territories or a line chart showing profit over time.

A dashboard is a collection of related views, shown in a single place for easy access. For example, you may have a set of views on sales and customers that you look at every day. Instead of flipping through each one individually, you can create a dashboard that displays everything at once.

When you sign in to Tableau Server, the Content page appears by default. Content (workbooks and data sources) that you have access to appear here as a result of either of the following processes:

- A Tableau Desktop user publishes a workbook or data source to Tableau Server.
- A Tableau Server user creates and saves a workbook in the web editing environment.

Quick Start: Create a Dashboard on the Web

As a Tableau Server user, you can not only connect to data and create views, you can pull those views into a completely new dashboard and use it to ask questions of data.

1 Connect to Data

On the Content page of a site, select Data Sources and select a data source by checking the check box at the left, such as Sample - Superstore. From the Actions menu, select New Workbook.
2 Create Views

Create two or more views to pull into your dashboard. For example, if you’re using the Superstore data source, you can create a view of customer sales by dragging Sales to Columns and Customer Name to Rows. Then click the New Sheet icon at the bottom of the workbook to create another view. For details on how to create more views, see Edit and Create Views.
3 Create a Dashboard

Click the **New Dashboard** icon at the bottom of your workbook, then click and drag views from the **Dashboard** pane to the dashboard on the right. As you drag a view, a gray shaded area shows you where you can drop it.

To replay, click the image.

4 Fine-Tune Your Dashboard

Add interactivity by selecting the **Use as Filter** command from the upper-right corner of one of the views. When the person viewing the workbook clicks on a mark in that view, marks in the rest of the dashboard filter accordingly.
You can also use this menu to control whether items such as filters and the toolbar are available to those who view your dashboard.

Click **Save As** to save your workbook, then **Done** to exit authoring mode. To share your work with others, click **Share** in the upper right.

**Connect to Published Data Sources**

You can publish data sources to Tableau Server or Tableau Online from Tableau Desktop. Publishing data sources to Tableau Server or Tableau Online enables sharing data among colleagues; including those who don't use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

You can connect to these published data sources on Tableau Server or Tableau Online if you have permissions to create and edit views.

**Note:** Many of the topics in this section use the Sample-Superstore data source that comes with Tableau Desktop. To use this data source, you can publish it to Tableau Server or Tableau Online from Tableau Desktop. Follow the procedure in the **Publish a Data Source** topic in the Tableau Desktop Help to learn more.

If you do not have access to the Sample-Superstore data source, you can connect to
your own published data and follow the procedures using similar measures and
dimensions from it.

Connect to a published data source on the web

At any time while you’re creating or editing a view on Tableau Server or Tableau Online, you
can connect to one or more published data sources.

1. Sign in to Tableau Server or Tableau Online and select a view to edit.

2. In editing mode, click the New Data Source icon.

3. In the Connect to Data dialog box, select a published data source from the list, and then
   click Add.

   Note: By default, only data sources that have been published to the server are
displayed in the list.

You can also connect to a published data source when you’re creating a new workbook on
Tableau Server or Tableau Online.

1. Sign in to Tableau Server or Tableau Online.

2. Navigate to the Content page and select Data Sources.

3. In the list of data sources, select the check box next to the one you want to use, and then
   click Actions and select New Workbook.

Connect to a published data source from Tableau Desktop

1. On the Connect to Data page in Tableau Desktop, click Tableau Server, and then
   provide the server name and your credentials.
2. Select a data source you want to use.

   The data source opens in the Data pane in the workbook. Tableau Server data sources show a Tableau icon instead of a database icon.
Try it: Build Views

You can create a new workbook and connect to a data source on Tableau Server. You can then use that data to build one or more views and pull those views into a dashboard.

**Note:** Only the workbook owner can save changes to an existing workbook on Tableau Server or Tableau Online.

The following procedures describe how to build three different views using the **Sample - Superstore** data source. Every view will be about sales, but each will give a slightly different perspective.

Create a bar chart

This procedure shows how to build a bar chart view that displays a list of technology products and how much profit each has generated.

To build a bar chart:

1. On the **Dimensions** pane, open **Product** and drag **Category**, then **Sub-Category** to the **Columns** shelf. From the **Measures** pane, drag **Profit** to the **Rows** shelf.
Click the image to replay it.

Tableau now has enough to convert the data into a visualization (view), in this case, a horizontal bar chart.

2. To display only the items under Technology, click the Technology header and select **Keep Only**.

3. To display each bar as a separate color, drag **Sub-Category** to the **Color** card. As a final touch, click the **Color** card to edit the color palette and choose **Blue**.
4. Right-click the sheet tab, click Rename sheet, type Bar Chart, and then click OK.

Create a highlight table

This procedure shows how to create a table that lists product profits by category and uses color to highlight the range of sales sizes.

1. Create a new sheet by clicking the New worksheet icon.

2. From Dimensions, drag Segment to the Columns shelf and Category and Sub-Category to the Rows shelf, then drag Profit to the Text card.
Click the image to replay it.

Your view now displays profit data in a table.

3. Add color to your table by dragging Profit to the Color card and changing the Marks type to square.
Your view now displays data about profits in a table that uses color to highlight the different sales amounts.

4. Give the view a name by right-clicking the sheet tab, clicking **Rename sheet**, and typing **Highlight Table**. Click **OK**.

## Create a map view

This procedure shows how to create a map view from the **Sample - Superstore** data source. This view will show profits by U.S.state.

1. Create a new sheet by clicking the New worksheet icon.
2. In the **Data** pane, double-click the **State** dimension, then under Measures, double-click **Profit**.
3. On the **Show Me** menu, click **Filled Maps**.
Tableau looks for geographic fields in any data source. For more information on how Tableau interprets geographic data in data sources, see **Assign Geographic Roles** in the Tableau Desktop help.

4. Give the view a name by right-clicking the sheet tab, clicking **Rename sheet**, typing **Map**. Click **OK**.

5. Click **Save As** to save the workbook. In the **Save Workbook** dialog box, complete the following steps:
   - Specify the workbook name, and leave **Project** set to **Default**.
   - Select **Show sheets as tabs** if you created multiple sheets and want their tabs to appear at the bottom of the view.
   - Select **Embed password for data source** if you want users who do not have an account on the database to be able to see the view.
• When you are finished, click **Save**.

Now that you’ve created some views, you can use them in a dashboard. See [Best Practices for Effective Dashboards](#) on the next page for some tips to consider for dashboards, then go to [Try it: Create a Dashboard](#) on page 166 for steps on how to build a dashboard.

**Edit a view**

On the Views page, you can open a view for editing in the following ways:

• In list view, select the check box for the view, then select **Actions > Edit Sheet**.

• Click a view to open it, and then click **Edit**.

If the workbook publisher did not embed database credentials, you are prompted to provide them.

When you save your work, even though you may have entered the authoring environment from a single view, the complete workbook is saved, including any other views and dashboards you may or may not have edited.
The options available to you to save workbooks depend on the permissions granted by your administrator. For more information, see Grant Web Edit, Save, and Download Permissions on page 878.

**Best Practices for Effective Dashboards**

A well-designed dashboard can align your organization’s efforts, help uncover key insights, and speed up decision-making. Use this topic for tips on best practices for creating effective dashboards in Tableau.

What's your goal?

**Know your purpose and audience**

The best visualizations have a clear purpose and work for their intended audience. What will you be trying to say with this dashboard? Are you presenting a conclusion or a key question?

In addition to knowing what you’re trying to say, it’s important to know who you’re saying it to. Does your audience know this subject matter extremely well or will it be new to them? What kind of cues will they need? Thinking about these questions before you head into the design phase can help you create a successful dashboard.

**Leverage the most-viewed spot**

Most viewers scan web content starting at the top left of a web page. Once you know your dashboard's main purpose, be sure to place your most important view so that it occupies or spans the upper-left corner of your dashboard. In the dashboard below, the author decided that the map view holds the key message.
Design for the real world

**Author at your final display size**

By default, Tableau dashboards are set to use a fixed size and if you keep this setting, be sure to construct your visualization at the size it will be viewed at. You can also set **Size** to **Automatic**, which makes Tableau automatically adapt the overall dimensions of a visualization based on screen size. This means that if you design a dashboard at 1300 x 700 pixels, Tableau will resize it for smaller displays—and sometimes this results in scrunched views or scrollbars. The **Range** sizing feature is helpful for avoiding this.
Limit the number of views

In general, it's a good idea to limit the number of views you include in your dashboard to two or three. If you add too many views, visual clarity and the big picture can get lost in the details. If you find that the scope of your story needs to grow beyond two or three views, you can always create more dashboards.

Too many views can also interfere with the performance of your dashboard. See Make Visualizations Faster in the Tableau Desktop help for more details.
Save time

Use the Go to Sheet command

The views in a dashboard are connected to the worksheets they represent. Tableau provides short-cut menus to help you quickly accomplish basic tasks, like jumping from the dashboard you're working on to the original view, where you can perform other actions.

To jump to a sheet:
1. In your dashboard, select the view you want to go to.

2. Select **Go to Sheet**.

**Hide sheets as you go**

When you’re working with a large number of worksheets to build dashboards, as you finish off a dashboard, you can hide the sheets that went into the dashboard. This can make your workbook easier to navigate.

To hide a dashboard’s sheets, right-click (control-click on Mac) the dashboard’s tab at the bottom of the workbook and select **Hide All Sheets**.
Try it: Create a Dashboard

After you've created one or more views, you can pull them into a dashboard, add interactivity, and much more. This topic walks you through the steps.

Open a dashboard sheet

You create a dashboard in much the same way you create a new worksheet. After you create a dashboard you can add views and objects.

To open a new dashboard sheet and start creating a dashboard, click the New Dashboard icon at the bottom of the workbook:

The Dashboard pane appears on the left and lists the sheets in your workbook.

Add views

After you have a dashboard sheet, click the views you built (listed under Sheets on the left) and drag them to your dashboard sheet on the right. A gray, shaded area indicates where you can drop views:
Add objects

In addition to adding views to your dashboard, you can add blank space and layout containers. Layout containers are helpful for fine-tuning how your dashboard resizes itself when users interact with it. See Refine Your Dashboard on page 169 for tips on when and how to use layout containers.

To add an object:

Select an item under Objects on the left and drag it to the dashboard sheet on the right:

Add interactivity

Part of the power of dashboards is that you can set up filters and interactivity to associate the different views on your sheet and enhance your users' analysis. The people looking at your dashboard can stay in a single spot—they don't need to click outside the world you've created.
for them.

In a dashboard you can use the **Use as Filter** option to make one of the views act as a filter on all the other views in the dashboard. To do this, select the filter button.

You can also select the drop-down arrow and choose **Use as Filter**.

**Edit a dashboard**

On the Views page, you can open a dashboard for editing in the following ways:

- In list view, select the check box for the dashboard, then select **Actions > Edit Sheet**.
- Click a dashboard to open it, and then click **Edit**.

If the workbook publisher didn’t embed database credentials, you are prompted to provide them.
Notes

- Although it's possible to edit a workbook that has device layouts, it's not advised. This is because device layouts inherit their content, formatting, and action from the parent (that is, from the default dashboard). Because you can't currently edit the device layouts, it's possible to inadvertently break a device layout and not know it. To edit a dashboard that has device layouts, download the dashboard to Tableau Desktop and make your edits there.
- When you save your work, even though you may have entered the authoring environment from a single view, the complete workbook is saved, including any other views and dashboards you may or may not have edited.
- The options available to you to save workbooks depend on the permissions granted by your administrator. For more information, see Grant Web Edit, Save, and Download Permissions on page 878.

Refine Your Dashboard

After you’ve created your dashboard, take a moment to step back and evaluate it. This topic lists some areas to check and refine.

Is everything in the right spot?

Confirm placement

Make sure that the view with your key finding or main question spans or occupies the upper-left corner of your dashboard. In your views, the most important data should be on the X or Y axis and your less important data should be on color, size, or shape.
Check label orientation and readability

Important labels should be horizontal so that your viewers can easily read them.
Review sizing and layout

Test your work and adjust sizing

Once you finish your dashboard, it's a best practice to test it and experience it as your users will. For example, will they be using Tableau Reader on a laptop to view your dashboard, will the dashboard be published on Tableau Server, or will it be embedded in a web application and viewed from a tablet?

After you test your dashboard, you'll probably find some things to change—sizing and how and where items are positioned is usually one of them.

Sizing often needs adjustment. Tableau dashboards are set to a fixed default size that's intended to work well on a typical desktop. However, when you publish (to the web, in a blog, for a presentation, etc.) you may find yourself more limited.

You can specify the overall size of the dashboard using the settings under Size in the Dashboard area on the left. Select a new size using the drop-down menu.

- **Automatic** - The dashboard automatically resizes to fill the window it's displayed in.
- **Fixed size** - The dashboard always remains a specific size. If the dashboard is larger than the window the dashboard becomes scrollable.
- **Range** - The dashboard scales between minimum and maximum sizes that you specify, after which scroll bars or white space display.

Use a layout container

A layout container can create a smoother experience for your users by helping dashboard objects reposition and resize when one view is filtering another.

The image below shows how a dashboard behaves when two views are placed in a vertical layout container versus how they behave when they are not placed in a layout container. Notice how, in the dashboard that uses a layout container, the views adjust vertically as different filters are applied.
With a Layout Container

Sub-Category

<table>
<thead>
<tr>
<th>Sub-Category</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bookcases</td>
<td>-4,436</td>
</tr>
<tr>
<td>Chairs</td>
<td>13,235</td>
</tr>
<tr>
<td>Furnishings</td>
<td>7,919</td>
</tr>
<tr>
<td>Tables</td>
<td>-9,728</td>
</tr>
<tr>
<td>Appliances</td>
<td>6,982</td>
</tr>
<tr>
<td>Art</td>
<td>3,454</td>
</tr>
<tr>
<td>Binders</td>
<td>17,996</td>
</tr>
<tr>
<td>Envelopes</td>
<td>3,264</td>
</tr>
<tr>
<td>Fasteners</td>
<td>577</td>
</tr>
<tr>
<td>Labels</td>
<td>3,076</td>
</tr>
<tr>
<td>Paper</td>
<td>15,536</td>
</tr>
<tr>
<td>Storage</td>
<td>7,104</td>
</tr>
<tr>
<td>Supplies</td>
<td>-1,658</td>
</tr>
<tr>
<td>Accessories</td>
<td>20,736</td>
</tr>
<tr>
<td>Copiers</td>
<td>24,084</td>
</tr>
<tr>
<td>Machines</td>
<td>2,141</td>
</tr>
<tr>
<td>Phones</td>
<td>23,837</td>
</tr>
<tr>
<td>Bookcases</td>
<td>635</td>
</tr>
<tr>
<td>Chairs</td>
<td>8,345</td>
</tr>
</tbody>
</table>

Click the image to replay it.

To add a layout container
1. Drag a horizontal or vertical layout container to the dashboard.

<table>
<thead>
<tr>
<th>Dashboard</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size</strong></td>
<td></td>
</tr>
<tr>
<td>Automatic</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Sheets</strong></th>
<th>Map - Profit by State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Heat Map - Profit by Seg.</td>
</tr>
<tr>
<td></td>
<td>Line Chart - Profit Over ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Objects</strong></th>
<th>Horizontal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Tiled - Floating

- Show dashboard title

2. Add sheets and objects to the layout container. As you hover over the layout container, a blue box indicates that the object is being added to the flow of the layout container.
Buy space with a floating layout

Tableau uses a tiled layout by default in dashboards, which means that each view, legend, and object is arranged in a single-layer grid, similar to a tiled floor.

If you want an item to overlap another, you can use a Floating layout for one or more items instead:
Legends can sometimes be floated over other items instead of remaining tiled.

See Tiled and Floating Layouts on page 178 for more details.

Highlight the essentials

**Show just what your users need**

Legends, titles, captions, and filters can be helpful for your viewers. Make sure you give your users what they need and no more.

To change what's displayed for your users:
1. Select a view in the dashboard.

2. Click the drop-down menu in the upper-right corner of the selected view and select the items you want to show. For example, you can show the title, caption, legends, and a variety of filters.

Note: You can display or hide a dashboard title on the web, but you can't edit it.

Note: Filters are only available for the fields used in the original view.

Eliminate clutter

In general, you should try to use no more than two color palettes in a single dashboard, and ideally, just one palette if your data is quantitative. Remove unnecessary text, lines, or shading that don't provide actionable information for your viewers. Make sure every legend you provide is really needed, remembering to take into account the final, published size of your dashboard.
Cluttered

Month of Date
January 1850   March 2016
Difference from Median (°C)

Difference from median global temperature (°C)

Highlight Year
Highlight Year of Date

Scatter Plot

The trend of Median for Date. Color shows Median. Details are shown for Year of Date and Month of Date. The data is filtered on Month of Date, which ranges from January 1850 to March 2016 and keeps Null values. The view is filtered on Year of Date, which excludes 2016. The trend of Median for Date.

Click the image to replay it.

Make a sheet sorter

A sheet sorter can be a time-saver for the people viewing your dashboard. It allows them to quickly display individual worksheets on the dashboard. For information, see Creating a Sheet Selector for a Dashboard.
Tiled and Floating Layouts

Each object in a dashboard can use one of two types of layouts: tiled or floating. Tiled objects are arranged in a single layer grid that adjust in size based on the total dashboard size and the objects around it. Floating objects can be layered on top of other objects and can have a fixed size and position.

**Tiled Layout**

All objects are tiled on a single layer. The top three views are in a horizontal layout container.

**Floating Layout**

While most objects are tiled on this dashboard, the map view is floating. It’s layered on top of the bar chart, which uses a tiled layout.

---

**Switch Between Layouts**

By default, objects in a dashboard use the Tiled layout. Tiled is selected by default when you drag views and objects onto your dashboard sheet:
To switch an object that’s already in your dashboard to floating:

Select the Floating option from the object’s short-cut menu:
**Work in the Web Authoring Workspace**

The web authoring workspace in Tableau Server is a lot like Tableau Desktop. You can create and edit views and dashboards in your workbook, but not stories.

**Note:** Only the workbook owner can save changes to an existing workbook on Tableau Server or Tableau Online.

**Worksheets**

Like in Tableau Desktop, **Data** and **Analytics** panes appear on the left side of the workspace.

The **Data** pane includes the names of the data sources included in the workbook, and the fields, parameters, and sets included in the active data source.
The **Analytics** pane provides analytic features in Tableau. You can drag reference lines, forecasts, trend lines, and other objects into your view from the **Analytics** pane.

A **toolbar** with options for editing the view appears at the top of the workspace. You can also use some of the toolbar options to connect to published data sources or create new worksheets and dashboards. For more information about the toolbar options, see [Web Authoring Toolbar](#) on page 185.
The **Marks** card and the **Pages** and **Filters** shelves appear on the left of the view. **Columns** and **Rows** for measures and dimensions shelves appear above the view. **Tabs** for worksheets and dashboards in the workbook appear at the bottom of the workspace.

To create a new view, click the **New Worksheet** button at the bottom of the workspace, or on the toolbar, click the drop-down arrow on the **New Worksheet** button and select **New Worksheet** from the list.

**Dashboards**

Select a dashboard tab in your workbook, and then click **Edit** in the toolbar to begin authoring. You can interact with dashboards as you would in Tableau Desktop. For more information, see *Create Views and Dashboards* on page 148.
For example, sort data by selecting the sort button next to a view name on your dashboard. Or, select field names to **Keep Only, Exclude, View Data, or Group Members**. These actions affect associated underlying worksheets.

<table>
<thead>
<tr>
<th>Country</th>
<th>Ease of Business</th>
<th>Days to Start Bus.</th>
<th>Hours to Do Tax</th>
<th>Business</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singapore</td>
<td>1</td>
<td>2</td>
<td>53</td>
<td>62</td>
</tr>
<tr>
<td>Hong Kong SAR, C.</td>
<td>2</td>
<td>6</td>
<td>3</td>
<td>60</td>
</tr>
<tr>
<td>New Zealand</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>170</td>
</tr>
<tr>
<td>United States</td>
<td>4</td>
<td>0</td>
<td>237</td>
<td></td>
</tr>
<tr>
<td>Denmark</td>
<td></td>
<td></td>
<td>Keep Only</td>
<td></td>
</tr>
<tr>
<td>Korea, Rep.</td>
<td></td>
<td></td>
<td>Exclude</td>
<td></td>
</tr>
<tr>
<td>Norway</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malaysia</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Georgia</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To easily update or review views in your workbook, select a view on your dashboard then click the **Go to Sheet** icon.
Ease of Business

To create a new dashboard, on the toolbar, click the drop-down arrow on the New Worksheet button and select New Dashboard from the list or click the dashboard button at the bottom of the workspace.

To rename an existing dashboard, double-click the tab for the dashboard and type a new name.

Use the Download menu to save your dashboard for offline viewing or to download underlying data.

Select Download > Image to save your dashboard as a PNG file. Select Download > PDF to save your dashboard as a PDF.

To download a view's underlying data, first select the view on your dashboard. The context-sensitive Download menu updates to add the options Data and Crosstab.
- Select **Data** to open a new window where you can preview and download both summary and underlying data for the view you've selected.

- Select **Crosstab** to download summary data for the view you've selected as a text table (CSV file).

**Web Authoring Toolbar**

When you are creating or editing a view, you can use the toolbar at the top of the view to perform common actions. The table below explains the function of each toolbar button.

<table>
<thead>
<tr>
<th>Toolbar Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>←</td>
<td><strong>Undo</strong>: Reverses the most recent action in the workbook. You can undo an unlimited number of times, back to the last time you opened the workbook, even after you have saved.</td>
</tr>
<tr>
<td>→</td>
<td><strong>Redo</strong>: Repeats the last action you reversed with the <strong>Undo</strong> button. You can redo an unlimited number of times.</td>
</tr>
<tr>
<td>📊</td>
<td><strong>New Data Source</strong>: Opens the <strong>Connect to a Data Source</strong> page, where you can connect to a published data source. For more information, see <strong>Connect to Published Data Sources</strong> on page 894.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="Pause Auto Updates" /></td>
<td><strong>Pause Auto Updates</strong>: Controls whether Tableau updates the view in real time when changes are made. For more information, see <a href="#">Pause Automatic Updates</a> on page 116.</td>
</tr>
<tr>
<td><img src="image" alt="New Worksheet" /></td>
<td><strong>New Worksheet</strong>: Creates a new blank worksheet or dashboard. For more information, see <a href="#">Create Views and Dashboards</a> on page 148.</td>
</tr>
<tr>
<td><img src="image" alt="Duplicate Sheet" /></td>
<td><strong>Duplicate Sheet</strong>: Creates a new worksheet or dashboard containing the same view as the current sheet.</td>
</tr>
<tr>
<td><img src="image" alt="Clear" /></td>
<td><strong>Clear</strong>: Clears the current worksheet or dashboard. Use the drop-down menu to clear the entire sheet or just the sheet formatting.</td>
</tr>
<tr>
<td><img src="image" alt="Swap" /></td>
<td><strong>Swap</strong>: Moves the fields on the <strong>Rows</strong> shelf to the <strong>Columns</strong> shelf</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>and vice versa. Most used with view types that are based on x- and y-axes.</td>
</tr>
<tr>
<td><strong>Σ</strong></td>
<td><strong>Totals</strong>: You can compute grand totals and subtotals for the data in a view. Select from the following options:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Show Column Grand Totals</strong>: Adds a row showing totals for all columns in the view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Show Row Grand Totals</strong>: Adds a column showing totals for all rows in the view.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Row Totals to Left</strong>: Moves rows showing totals to the left of a crosstab or view.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>Column Totals to Top</td>
<td>Moves columns showing totals to the top of a crosstab or view.</td>
</tr>
<tr>
<td>Add All Subtotals</td>
<td>Inserts subtotal rows and columns in the view, if you have multiple dimensions in a column or row.</td>
</tr>
<tr>
<td>Remove All Subtotals</td>
<td>Removes subtotal rows or columns.</td>
</tr>
<tr>
<td>Highlight</td>
<td>Turn on highlighting for the selected sheet. Use the options on the drop-down menu to define how values are highlighted. For more information, see Highlight Marks, Legends &amp; Actions.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td><img src="image" alt="Show Mark Labels" /></td>
<td><strong>Show Mark Labels:</strong> Switches between showing and hiding mark labels for the current sheet.</td>
</tr>
<tr>
<td><img src="image" alt="Format Workbook" /></td>
<td><strong>Format Workbook:</strong> Open the <strong>Format Workbook</strong> pane to change how fonts and titles look in every view in a workbook by specifying format settings at the workbook level instead of at the worksheet level.</td>
</tr>
<tr>
<td><img src="image" alt="Fit" /></td>
<td><strong>Fit:</strong> Change the proportions of your view within the browser window, and go back and forth between seeing details and seeing the whole picture. The <strong>Cell Size</strong> commands have different effects depending on the type of visualization.</td>
</tr>
<tr>
<td><img src="image" alt="Show/Hide Cards" /></td>
<td><strong>Show/Hide Cards:</strong> Shows and hides <strong>Titles, Captions, Filter</strong>, and <strong>Highlighter</strong> cards that are included in</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>the view.</td>
</tr>
<tr>
<td></td>
<td><strong>Download</strong>: Use the options under <strong>Download</strong> to capture parts of your view for use in other applications.</td>
</tr>
<tr>
<td>![Image]</td>
<td>- <strong>Image</strong>: Displays the view, dashboard, or story as an image in a new browser tab.</td>
</tr>
<tr>
<td>![Data]</td>
<td>- <strong>Data</strong>: Displays the data from the view in a new browser window with two tabs: <strong>Summary</strong>, showing aggregated data for the fields shown in the view, and <strong>Underlying</strong>, showing underlying data for the selected marks in the visualization. If the new window does not open,</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>you may need to disable your browser's popup blocker.</td>
</tr>
<tr>
<td><strong>Crosstab:</strong></td>
<td>Saves the underlying data for the selected marks in the visualization to a CSV (comma-separated values) file which can then be opened in Microsoft Excel.</td>
</tr>
<tr>
<td><strong>PDF:</strong></td>
<td>Opens the current view as a PDF in a new browser window. From there you can save it to a file. If the new window does not open, you may need to disable your browser's popup blocker.</td>
</tr>
<tr>
<td>Toolbar Button</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| ![Show Me](image) | **Show Me**: Opens a control that shows a range of visualization types that you can use in Tableau. When you display the Show Me list, Tableau uses the data in the current view to determine which visualization types to make available for you to select. Among the available types, it draws an orange outline around the recommended chart type that is the best match for your data.
You can also hover over a visualization type to see what field types are required to make that visualization type available. |

The **Side Bar**

The Side Bar provides two panes: the Data pane and the Analytics pane.

You can hide and show the Side Bar by clicking the minimize button ![minimize](image) in the upper right corner. The Data pane collapses to the bottom of the workbook. Click the minimize button again to show the Data pane.

To open the Side Bar again click the same button, which is now at the extreme left of the workspace:
Data Pane

At the top of the **Data** pane is a list of available data sources for the workbook. If you are editing an existing workbook, there may be multiple data sources. If you are creating a new workbook, you see only the data source from which you created the workbook.

Select a data source to see the dimensions and measures for that data source.

All data sources contain fields. These fields appear below the list of data sources in the **Data** pane. Dimensions and measures always appear, other field types appear if they exist in the data source:

- **Dimensions** are fields that contain discrete qualitative data. Examples of dimensions include dates, customer names, and customer segments.

- **Measures** are fields that contain numerical data that can be aggregated. Examples of measures include sales, profit, number of employees, temperature, frequency, and pressure.

- **Sets** are custom fields that define a subset of data based on some conditions. A set may be based on a computed condition, which updates as the data changes, or a constant list of values. Sets may exist in workbooks that you edit, but you cannot create sets.

- **Parameters** are dynamic values that can replace constant values in calculations, filters, and reference lines. Parameters may exist in workbooks that you edit, but you cannot create parameters.

By default, Tableau treats all relational fields containing numbers as measures. However, you might decide that some of these fields should be treated as dimensions. For example, a field containing ages may be categorized as a measure by default in Tableau because it contains numeric data. However, if you want to look at each individual age rather than an axis, you can convert the **Age** field to a dimension. To do this, drag the **Age** measure and drop it into the **Dimensions** area in the **Data** pane. Now if you drag the **Age** field to the **Rows** or **Columns** shelf it will create column headers (1, 2, 3, etc.) instead of a continuous axis.
To build visualizations, you drag fields from the **Data** pane to the **Rows** and **Columns** shelves, the **Marks** card, or one of the other available shelves. For a demonstration, see Try it: Build Views on page 154.

**Manage fields in the Data pane**

Right-click a field in the **Data** pane to see a set of options for modifying that field. You can do the following:

- Convert a dimension to a measure or vice versa.
- Convert a discrete field to continuous or vice versa. This option is available for measures and date dimensions.
- Change the data type for a field.
- Change the geographic role for a field.
- Change the default aggregation for a measure.
- Create a calculated field.

**Data blending**

If you upload a workbook that uses blended data sources, you can see a link next to the field or fields in the primary data source that are being used to link the two data sources:

You can click on the link icon to activate or deactivate specific fields. When fields that can be used as linking fields are *not* being used, the link icon changes to show a broken link:
Data sources are linked using a left join, where the view is using all data rows from the primary data source but only those data rows from the secondary data source that have values for fields that are in the view or for fields that are designated as linking fields. So changing the linking field, or designating multiple linking fields, can actually pull in different or additional data rows from the secondary data source, thereby changing the values returned by aggregations.

Analytics Pane

The Analytics pane provides quick and easy access to common analytic features in Tableau. You can drag reference lines, box plots, trend lines forecasts, and other items into your view from the Analytics pane, which appears on the left side of the workspace. Toggle between the Data pane and the Analytics pane by clicking one of the tabs at the top of the Side Bar:

Add Analytics Pane Item

To add an item from the Analytics pane, drag it into the view. When you drag an item from the Analytics pane, Tableau shows the possible destinations for that item in a drop target area in the upper left section of the view—drop the item somewhere in this area. The range of choices varies depending on the type of item and the current view.
In a simple case, the drop target area would offer these three options:

The terms **Table**, **Pane** and **Cell** define the scope for the item:

For a more complicated view—for example, if the view contained a line chart with multiple or dual axes—Tableau would show you a drop target area that looked like this:
If you dropped the item in one of the three larger boxes in the header at the top of the drop target area—for example, the Table box—a separate median with quartiles would be added for each axis:

But if you drop the item in any of the six lower boxes aligned with a specific measure, the median with quartiles would only be added on the corresponding axis, with the specified scope.

**Delete Analytics Pane Item**

You can delete an item you just added from the Analytics pane by clicking the Undo button on the web editing toolbar. See [Web Authoring Toolbar](#) on page 185.

You can also click on an item and choose **Remove** from the tooltip.

**Note:** Some Analytics pane items *(Median with Quartiles and Average with 95% CI)* actually add both a reference line and a reference distribution. Unless you are using Undo, you would need to delete these items separately.

**Edit Analytics Pane Item**

For information on how to edit an Analytics pane item, see the section for a particular item type under Analytics Pane Item Definitions, below.

**Analytics Pane Items**

The following items can be dragged from the Analytics pane and dropped in the view.

**Constant Line**

Adds one or more constant lines to the view. You can add a constant line for a specific
measure, for all measures, or for date dimensions. When you add a constant line, Tableau displays a Value prompt where you specify the value for the constant:

Click on a constant line to change its value or to remove it. You can also click the Undo button on the web editing toolbar to remove the line. See Web Authoring Toolbar on page 185.

**Average Line**

Adds one or more average lines to the view. You can add an average line for a specific measure or for all measures.

You can click on a resulting average line and choose a different aggregation, such as Total or Sum, or choose to remove the line entirely. You can also click the Undo button on the web editing toolbar. See Web Authoring Toolbar on page 185.

**Median with Quartiles**

Adds one or more sets of median lines and distribution bands to the view. You can add a median with quartiles for a specific measure or for all measures.

The distribution bands are computed as quartiles; the middle two quartiles are shaded.

To remove the median and the quartiles, click the Undo button on the web editing toolbar. See Web Authoring Toolbar on page 185. You can also click on either the median line or the quartiles to see its value or to remove it. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect.

**Box Plot**

Adds one or more box plots to the view. You can add box plots for a specific measure or for all measures. The scope for a box plot is always Cell (and never Table or Pane).

Click or hover over any of the horizontal lines in the box plot to see statistical information about the whiskers, quartiles, and median.

Clicking a horizontal line also shows a Remove option. You can also click the Undo button on the web editing toolbar. See Web Authoring Toolbar on page 185.

**Totals**

Adds totals to the view. When you add totals, the drop options are Subtotals, Column Grand Totals, and Row Grand Totals.

To remove totals, click the relevant column or row header and choose Remove. You can also click the Undo button on the web editing toolbar. See Web Authoring Toolbar on page 185.

**Average with 95% CI**

Adds one or more sets of average lines with distribution bands; the distribution bands are
configured at a 95% confidence interval. You can add these items for a specific measure or for all measures.

The confidence interval distribution bands shade the region in which the population average will fall 95% of the time.

To remove the average line and confidence interval, click the **Undo** button on the web editing toolbar. See **Web Authoring Toolbar on page 185**. You can click on a resulting average line and choose a different aggregation, such as Total or Sum, or choose to remove the line entirely. You can also click on a distribution to see its value or to remove it. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect.

**Median with 95% CI**

Adds one or more sets of median lines with distribution bands; the distribution bands are configured at a 95% confidence interval. You can add these items for a specific measure or for all measures.

The confidence interval distribution bands shade the region in which the population median will fall 95% of the time.

To remove the median line and confidence interval, click the **Undo** button on the web editing toolbar. See **Web Authoring Toolbar on page 185**. You can click on a resulting median line and choose a different aggregation, such as Total or Sum, or choose to remove the line entirely. You can also click on a distribution to see its value or to remove it. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect.

**Trend Line**

Adds one or more trend lines to the view. When you add trend lines, the drop options identify the trend line model types available in Tableau: **Linear**, **Logarithmic**, **Exponential**, and **Polynomial**. For some views, only a subset of these options is available.

Click on a trend line to remove it or to see a statistical definition. You can also click the **Undo** button on the web editing toolbar. See **Web Authoring Toolbar on page 185**.

**Columns and Rows Shelves**

Drag fields to the **Columns** shelf to create the columns of a table, or to the **Rows** shelf to create the rows of a table. You can drag multiple fields to either shelf.

Discrete values (typically, dimensions) appear in blue on the **Columns** and **Rows** shelves; continuous values (typically, measures) appear in green.

At the right end of any field you place on the **Columns** or **Rows** shelf is a drop-down menu that you can use to configure the dimension or measure:
The options available depend on the type of field. The complete list of options includes:

- **Show Filter**
  Choose this option to add a filter for this field to the view so that users can specify which data to include and exclude for this dimension or measure.

- **Include in Tooltip**
  By default, all fields on the **Columns** and **Rows** shelves are included in the tooltips that appear when you move your mouse over one or more marks in the view. Clear this option to remove a field from tooltips.

- **Discrete/Continuous**
  Use these options to convert a continuous range of values into a set of discrete values, or a discrete set into a continuous range.

- **Dimension/Attribute/Measure**
  Use this range of options to convert a dimension to a measure or a measure to a dimension.
  You can also define the field as an Attribute, which returns the value of the given expression if it only has a single value for all rows in the group, and otherwise displays an asterisk (*) character. Null values are ignored.

- **Edit in Shelf**
  Use this option to edit the calculated field directly on the shelf. For more information about editing calculated fields, see [Create or Edit a Calculated Field](#) on page 207.

- **Add Table Calculation**
Opens the Table Calculation dialog box, where you can apply a calculation that transforms the measure’s values.

- **Quick Table Calculation**
  Provides a set of options for redefining the meaning of the marks for the value.

- **Remove**
  Removes the value from the **Columns** or **Rows** shelf.

**Options for Date Dimension**

An additional set of options is available for date dimensions:

Choose one of the options from the upper group to define the granularity of the data as discrete values. For example, if you choose **Month** your view will combine the data for each named month in your data across the full range of years:
There are exactly twelve marks in the data—one for each month. The November mark combines the data from November 2008, November 2009, etc.

Choose one of the options from the lower group to define the granularity of the data as continuous values. For example, if you choose Month your view will show your data sequentially, over the range of available months.

In this case there are 48 marks in the data—one for each month since November 2008.
Marks

When you drag fields to the view, the data are displayed using marks. Each mark represents the intersection of all of the dimensions in the view. For example, in a view with Region and Year dimensions, there is a mark for every combination of those two dimensions (East 2011, East 2012, West 2011, West 2012, etc.).

Marks can be displayed in many different ways including lines, shapes, bars, maps, and so on. You can show additional information about the data using mark properties such as color, size, shape, labels, etc. The type of mark you use and the mark properties are controlled by the Marks card. Drag fields to the Marks card to show more data. For example, the same view above is shown again below but this time with Profit on Color. With this additional information, it is clear that the West region had the highest profit in 2014.

Control the marks in the view using the Marks card. Use the drop-down menu to specify the type of mark to show. Drag fields to the Marks card and use the drop-down controls to add more information to the view and control the color, shape, size, labels, and number of marks in the view.
Filters Shelf

Use the Filters shelf to specify which data to include and exclude for a dimension or measure. For example, you might want to analyze the profit for each customer segment, but only for certain shipping containers and delivery times. By placing the Container dimension on the Filters shelf you can specify which containers to include. Similarly, you can put the Delivery Date field on the Filters shelf to define which delivery times to include.

When you drag a dimension or measure to the Filters shelf, Tableau inserts a filter control into the view so users can select the values to display. For example:

For dimensions, the filter control shows discrete values, as above. For measures, the control shows a continuous range:

Hover your mouse to the right of the title for the filter control to specify how to display values in the control:
Pages Shelf

Drag a dimension or measure to the Pages Shelf to break a view into a series of pages so you can better analyze how a specific field affects the rest of the data in a view. Dragging a dimension to the Pages shelf is like adding a new row for each member in the dimension. Dragging a measure to the Pages shelf automatically converts the measure into a discrete measure that can be broken into individual pages.

When you drag a dimension or measure to the Pages shelf, Tableau automatically inserts a control into the view to let you navigate the pages in your view. For example:

You can manually advance through the sequence of pages in any of the following ways:
• Use the drop-down menu to select a value.
• Use the forward and back buttons on either side of the drop-down list to navigate through the pages one at a time.
• Use the Page slider to quickly scroll forward and backward in the sequence of pages.

Select **Show History** to show marks from previous pages in addition to marks for the current page.

**Tooltips**

Place your cursor over a mark in the view to see the tooltip for that mark.

Tooltips provide information on the values of dimensions and measures for the selected mark:

![Example Table]

Tooltips also provide these options:

• **Keep Only**
  Exclude all marks from the view except this one.

• **Exclude**
  Exclude this mark only.

• **Group Members**
  Choose the paperclip icon to create a new group, which is a dimension, from the selected mark. Typically, you would select multiple marks and then create a group. For example, if you have a dimension Region with values North, South, East and West, you could select South and West and then create a group from them.

• **View Data**
  Choose the table icon to open a new browser window to display two tabs: **Summary**, which shows only data for the current mark, and **Underlying**, which shows data for the entire view.
Calculated Fields

If your underlying data doesn't include all of the fields you need to answer your questions, you can create new fields in Tableau and then save them as part of the data source. For example, you could create a new calculated field called **Profit** that calculates the difference between the **Sales** and the **Cost** fields, or you could create a formula that sorts values for the **Sales Budget** field depending on whether they are under budget or over budget.

You created calculated fields in Tableau by defining a formula that is based on existing fields and other calculated fields, using standard functions and operators. You can create calculated fields using the calculation editor, or by double-clicking a field on a shelf and building an ad-hoc calculation.

You can create calculated field with Tableau Desktop or in Tableau web editing environments in products like Tableau Server and Tableau Online.

**Note:** You can use all types of calculations if you are using a relational data source, however multidimensional data sources do not support aggregations and binned data.

Create or Edit a Calculated Field

Use the calculation editor to create or modify a calculated field.

To display all collapsed content, click the (Expand all) button at the top of the page.

**Create a Calculated Field**

To open the calculation editor, click the drop down to the right of Dimensions on the **Data** pane and choose **Create Calculated Field**.

See **Create Formulas in Tableau** on page 211 for information on how to bring the different element that make up a formula into the calculation editor.

When the workbook is connected to multiple data sources, the current data source is listed next to the calculation name box in the upper left--in the image above, the current data source is **Sample - Superstore**. As you work in the editor, any fields that you that are not from the
current data source are shown with the data source prepended to the field name. For example: [DS1],[Sales].

Click **Apply** in the calculation editor to save the field as currently defined and add it to the **Data** pane without closing the editor; click **OK** to save the calculation and close the editor.

Calculations that return a string or date are saved as dimensions, and calculations that return a number are saved as measures.

Tableau will allow you to save an invalid calculation; however, a red exclamation point appears next to it in the **Data** pane:

Until you correct an invalid calculated field, you will not be able to drag it into the view.

**Edit a Calculated Field**

To edit a calculated field, click on the drop-down to the right of the field in the **Data** pane and select **Edit**:

You can only edit calculated fields--that is, named fields created in Tableau (as opposed to named fields that were part of the original data source).

**Drag and Drop Options with the Calculation Editor**

You can work on a calculation in the editor and do other things in Tableau at the same time. Here is an example of the kind of workflow that is possible with the calculation editor:
1. Start by creating or editing a view.
2. Open the calculation editor and begin working on a calculated field.
3. Drag all or part of your formula to a shelf, dropping it on an existing field, to see how it changes the view.
4. Double-click the field you just dropped on the shelf to open it as an ad-hoc calculation (see Ad-Hoc Calculations on page 215 for details). Then tweak the calculation.
5. Drag the ad-hoc calculation back to the calculation editor and drop it on the original formula in the calculation editor, thereby replacing the original formula.

You can also drag all or part of a formula to the Data pane to create a new field.

It is not possible to drag numeric bins, generated latitude and longitude fields, Measure Names, or Measure Values into the calculation editor.

Copy and Paste Calculated Fields

Calculated fields are available to all sheets that use the same data source in a single workbook. To copy and paste calculated fields between workbooks, right-click the field in the Data window of the source workbook and choose Copy. Then right-click in the Data window in the destination workbook and choose Paste. You can copy and paste all custom defined fields, including calculated fields, ad hoc groups, user filters, and sets.

Working in the Calculation Editor

When you use the calculation editor, the result must always be a calculated field—that is, a user-created named field (as opposed to a named field that was part of the original data source). If you are editing an existing calculated field, then that field is updated when you click OK or Apply. If you are creating a new calculated field, or wish to save your modifications as a new field, type a new field name in the box at the top of the calculation editor before clicking OK or Apply. Ad-hoc calculations, by contrast, do not need to be named. See Ad-Hoc Calculations on page 215.

Customizing the Calculation Editor

You can customize the calculation editor in the following ways:

- **Expand or collapse the function list and help area**
  
  You can expand (open) or collapse (close) the function list and the help area on the right side of the calculation editor by clicking the angle control that is between the working area and the function list:
Click the same control (which now faces left) to reopen the function list and help area.

- **Resize or move the calculation editor**
  You can resize the calculation editor by dragging from the lower right corner.
  You can move the editor by clicking in the upper-left section (anywhere in the box shown in red, below) until you see the four-way arrow cursor (outlined in blue), and then dragging:

**Visual Feedback in the Calculation Editor**

The calculation editor has built-in coloring and validation to help you avoid syntax errors. As you create a formula, syntax errors are underlined with a red line. Hover over the error to see a suggestion for resolving it:

Feedback on formula validity is also displayed at the bottom of the calculation editor. A green check mark indicates that the formula is valid; a red X indicates that it is not. You can click the drop down box to see details:
When you are writing formulas, any part that displays in bold indicates that it will be computed locally within Tableau on the aggregated results. Any normal weight text will be computed at the database level.

**Sheets Affected**

As you edit a calculated field, you can click **Sheets Affected** in the editor’s status bar to see which other sheets are using the field, which will be updated when you commit your changes:

The **Sheets Affected** drop-down is only shown if the field you are editing is also being used in other sheets.

**Create Formulas in Tableau**

You create formulas from the following elements:

- **Functions**
  
  Click the small arrow on the right side of the calculation editor to display a list of functions and their definitions:
You can also type in the **Enter Text to Search** area to find functions that match the characters that you type. By default all functions are displayed. Certain functions are only available with specific data sources. See **Additional Functions** in the Tableau Desktop help for details. For complete help on Tableau functions and other keywords, see **Functions, Operators, & Data Types** in the Tableau Desktop help.

Click a function in the function list to view a brief description and an example on the right. Double-click a function in the list to include it at the cursor location in the current formula.

Functions are colored light blue in formulas.

**Note**: Autocomplete is available in the calculation editor. For details, see **Auto-Completion for Formulas** on page 214.

- **Fields**

You can incorporate a field in a formula by dragging it from the **Data** pane and dropping it in the calculation editor.

You can also drag a field from a shelf in the view to the calculation editor. The result will not necessarily be the same as when you drag a field from the **Data** pane. For example, if you drag the **Sales** field from the **Data** pane to the calculation editor, this is what you will see:

```
[Sales]
```

But if you drag **Sales** to the **Rows** shelf and then drag it from there to the calculation editor, the field will have changed as a result of being aggregated in the view:

```
SUM([Sales])
```
You may also choose to modify a field on a shelf before dragging it to the calculation editor. For example, if you have dragged Sales to the Rows shelf and then added a quick table calculation to the field, you can drag the modified field to the calculation editor to see how the field is expressed in the Tableau calculation language, and to make further modifications. This is what you might see:

```
RUNNING_SUM(SUM([Sales]))
```

You can also drag part or all of an ad-hoc calculation from a shelf to the calculation editor. In addition to the various drag-and-drop options, you can also just type a field name in the calculation editor. Field names that include special characters or spaces must be delimited with square brackets—for example, `SUM([Store Profit])`. For field names that actually include bracket characters, just type two additional brackets. For example, to specify a field named “Store Profit[]” you would type `[Store Profit][]`; two of the brackets say “include the bracket character in the field name,” and the third bracket delimits the field name. For information on ad-hoc calculations, see Ad-Hoc Calculations on page 215.

Fields are colored orange in formulas.

- **Operators**
  
  Type any necessary operators into your formula. All standard operators such as addition (+), subtraction (−), multiplication (·), and division (/) are supported.

  Operators are colored black in the formula.

- **Parameters**
  
  Parameters are placeholder variables that can be inserted into calculations to replace constant values. When you use a parameter in a calculation, you can then expose a parameter control in a view or dashboard to allow users to dynamically change the value.

  Parameters are colored purple in formulas.

  **Note:** You cannot create parameters in web editing environments.
• **Comments**

To add a comment to a calculation, type two forward slash characters into the formula pane.

For example:

\[ \text{Sales} \times \text{Profit} //\text{John’s calculation} \]

In this example //John’s calculation is a comment.

A comment starts at the two forward slashes (/\/) and goes to the end of the line. A multiline comment can be written by starting each line with two forward slashes (//).

Comments are colored green in the formula.

**Auto-Completion for Formulas**

As you type a formula, either in the calculation editor or in an ad-hoc calculation, Tableau displays a list of options for completing the formula.

As you scroll the list, using mouse or keyboard, Tableau shows a short description when the current item is a function:

![Auto-Completion example](image)

When the current item is a field, set, or bin, and that keyword has a comment attached, that comment will appear as the description.

Click a keyword in the list or press Enter to select it. If the keyword is a function, Tableau displays syntax information when you select it:

![Function syntax](image)

As you type, the list of suggested keywords is organized into the following categories, in the following order:
• Functions, dimensions, measures, parameters, sets, and bins that begin with the string you have typed so far, from the current data source, sorted alphabetically.

• Functions, dimensions, measures, parameters, sets, and bins that contain the string you have typed so far, from the current data source, sorted alphabetically.

• If the workbook connects to multiple data sources, dimensions, measures, sets, and bins that begin with or contain the string you have typed so far are displayed, sorted by data source and sorted alphabetically within each data source.

Typing certain characters will either turn off auto-completion or cause it to behave differently:

• Auto-completion will not offer suggestions as you type a quoted string.

• Auto-completion will not offer suggestions if you begin by typing a number.

• If you begin by typing an open square bracket, [, auto-completion shows fields, parameters, sets, and bins, but not functions.

Auto-Completion with Multiple Data Sources

If the workbook is using multiple data sources, auto-completion behaves as follows:

• If the selected field is from a secondary source, auto-completion adds the field with its aggregation and fully qualified name. For example:

  ATTR([secondaryDataSource].[sate])

• Matches from secondary sources are only shown if there is an explicit blend relationship set with the currently active sheet.

• Fields that are being used to blend the two data sources are only shown once in the search results (the field shown is from the primary data source).

Ad-Hoc Calculations

Ad-hoc calculations are calculations that you can create and update as you work with a field on a shelf in the view. Ad-hoc calculations are also known as type-in or in-line calculations.

Double-click on an existing field to start editing.
You can also double-click on an empty shelf or on an empty part of a shelf to create a new calculation.

Type to update the expression, or drag new fields into the expression from the Data pane or elsewhere in the view.

Press Enter or click outside the expression to commit the expression, close the calculation, and update the view.

Press Ctrl+Enter to commit the expression and update the view without closing the calculation.

Ad-hoc calculations are supported on the Rows, Columns, Marks, and Measure Values shelves; they are not supported on the Filters or Pages shelves.

**Note:** Ad-hoc calculations are not named, but are saved when you close the workbook. If you want to save an ad-hoc calculation for use in other workbook sheets, copy it to the Data pane. You will be prompted to name the calculation. (On server it just deposits it w/o rename prompt.) Once you name an ad-hoc calculation, it is the same as a calculation you created with the calculation editor, and is available on other sheets in the workbook. See [Create or Edit a Calculated Field](#) on page 207

**Multi-Line Ad-Hoc Calculations**

As you’re typing an ad-hoc calculation, you can press Shift+Enter to start a new line. However, only the current line is ever visible in an ad-hoc calculation, so this can be confusing for anyone viewing or editing the calculation who has no way of knowing that it contains multiple lines.

The first line of a multi-line ad-hoc calculation can be a comment that serves as a title for the calculation. This is the only line that is visible on the shelf after the calculation is committed:

```
//City and State
```
[City] + ', ' + [State]

**Ad-Hoc Calculations and Aggregation**

If Tableau determines that the expression you enter is a measure (that is, returns a number), it automatically adds an aggregation to the expression when you commit the expression. For example, if you type \texttt{DATEDIFF('day', [Ship Date], [Order Date])} in an ad-hoc calculation and then press Enter, what you will see is the following:

\texttt{SUM(DATEDIFF('day', [Ship Date], [Order Date]))}

If you use a field that is already an aggregated field (for example, \texttt{SUM([Profit])}) in an ad-hoc calculation, the result is an aggregate calculation. For example, when you commit an ad-hoc calculation \texttt{SUM([Profit])/SUM([Sales])}, the result is:

\texttt{AGG(SUM([Profit])/SUM([Sales]))}

**Ad-Hoc Calculations for Insight and Experimentation**

Typically you create ad-hoc calculations on-the-fly to do things like

- Test a hunch
- Try a what-if scenario
- Debug a complex calculation

**Managing Ad-Hoc Calculations**

Use the following keystrokes to manage ad-hoc calculations.

- Double-click in an existing field on the \textit{Rows, Columns, Marks, or Measure Values} shelf to open it as an ad-hoc calculation.
- Double-click elsewhere on any of these shelves to create a new ad-hoc calculation from scratch.
- Press Esc to cancel an ad-hoc calculation.
- Press Enter to commit an ad-hoc calculation, which updates the view and closes the ad-hoc calculation. Press Ctrl+Enter to commit the change and update the view without closing the ad-hoc calculation.
- When you double-click a named field on a shelf to edit it, you are not changing the original named field.

In addition, the following rules govern the use of ad-hoc calculations.

- Only one ad-hoc calculation can be open at a time.
- If a field that has a geographic role or a fiscal year setting associated with it is added to an ad-hoc calculation, the ad-hoc calculation inherits that role or setting.
• The right-click context menu for an ad-hoc calculation supports the same options that would be available for any other field in the view of the same type, including the ability to change aggregation, show a filter, or format.

• Ad-hoc calculations are not available when you create groups, sets, hierarchies, or parameters.

• Ad-hoc calculations are valid for creating trend lines, forecasts, and reference lines, bands, and distributions.

Transform Values with Table Calculations

A table calculation is a transformation you apply to the values for a measure in the view.

You can use table calculations for a variety of purposes, including:

• Transforming values to rankings
• Transforming values to show running totals
• Transforming values to show percent of total

Table calculations are available as you work with views in Tableau Desktop and also as you edit views in Tableau Server or Tableau Online.

Table calculations are a special type of calculated field that computes on the local data in Tableau. They are calculated based on what is currently in the view and do not consider any measures or dimensions that are filtered out of the view.

Note: The “table” is determined by the dimensions in the view—not the tables in your data source. Also, a “table calculation” is an entirely different thing than a calculated field. For information on calculated fields, see Calculated Fields.

Table Calculation Enhancements in Tableau 10

Tableau 10 provides some key enhancements to the experience of creating table calculations:

• As you work in the Table Calculation dialog box to configure a calculation, the view updates in real time to show the results of your choices.

• Highlighting shows you the scope of your calculation by setting off part of the view with a colored background as you change the Compute Using option in the Table Calculation dialog box:
If you select a different mark in the view while the Table Calculations dialog box is still open, highlighting will switch to the partition containing that mark.

**What is a Table Calculation?**

A table calculation is a transformation you apply to the values of a single measure in your view, based on the dimensions in the level of detail.

For any Tableau view, there is a virtual table that is determined by the dimensions in the view. This table is not to be confused with the tables in your data source. Specifically, the virtual table is determined by the dimensions within the “level of detail,” which means the dimensions on any of the following shelves or cards in a Tableau worksheet:

Consider this simple view:
In this view, the dimensions are **Order Date** (with values aggregated up to **YEAR**) on **Columns**, and **Segment** on **Rows**. The individual cells in the table show the value of the **Profit** measure for each combination of **Order Date** and **Segment**. There are four years’ worth of **Order Date** data, and three **Segments**; multiply these numbers to get 12 individual cells, each showing a **Profit** value.

Suppose you want to see not absolute dollar values, but the percentage of the total profit that each of these 12 individual profit values contributes to the total profit, so that when all the cell values are added, they total 100%. To do this, you can add a table calculation. Here’s how you could add a table calculation to show this.

1. You always add a table calculation to a measure in the view. In this case there is only one measure in the view—**SUM(Profit)**—so when you right-click that measure on the Marks card (or Control-click on a Mac) you see two options that mention table calculations:
   - **Add Table Calculation**
   - **Quick Table Calculation**

If you choose **Quick Table Calculation**, you will see a range of options:
2. **Percent of Total** looks right. Choose that, and the view updates to show percentages, instead of absolute dollar values:
Notice the triangle icon that now appears next to $\text{SUM(Profit)}$ on the Marks card:

This indicates that a table calculation is currently being applied to this measure.

3. But what about when the view is not a text table? Are table calculations still an option? They are.

   Back up one step (click the Undo button on the toolbar) to remove the quick table calculation.

4. Now use Show Me to change the chart type to horizontal bars:
You might not be as likely to refer to this view as a “table,” but the dimensions in the level of detail are the same, so the view’s virtual table is the same.

5. You can now apply the exact same table calculation as before—right-click $\text{SUM(Profit)}$, choose **Quick Table Calculation**, and then **Percent of Total**. The view changes to show percentages along the horizontal axis, instead of dollar values:

![Sheet 1](image)

**Note:** When you are learning about table calculations, or experimenting with different options, a text table is usually going to provide more intuitive insight than other chart types.

**Quick Table Calculations**

A quick table calculation is a one-step process where you choose a common table calculation type from a list. Tableau automatically applies the most typical settings for that calculation type.

To apply a quick table calculation, do the following:

1. Click a measure in the view and choose **Quick Table Calculation** from the context menu.

2. Choose a Calculation type from the sub-menu.

If the results are satisfactory, you’re done. If not, you can continue working with the calculation by clicking the measure again and choosing Edit table calculation.

The list of available quick calculation types does not exactly match the list of default calculation types you see in the Table Calculation dialog box. For example, a **Compound growth rate**
quick table calculation is a **Percent Difference From** table calculation, but with the **Compute compounded rate** option selected. If you want to know how any given quick table calculation is defined, add the calculation and then go back and edit it to see how it is configured in the Table Calculation dialog box.

**Choose Your Approach to Table Calculations**

The easiest way to add a table calculation is to use a quick table calculation. See **Quick Table Calculations** on the previous page. But if you need more flexibility than a quick table calculation affords, you can base your calculation on the visual structure of your view, or you can reference the specific dimensions in the view.
Table Calculations: Use the Visual Structure of Your View

When you configure a table calculation using the **Compute Using** options in the Table Calculation dialog box, you define a table calculation using the visual structure of your view.

As you add or edit a table calculation in the Table Calculation dialog box, Tableau highlights the effects of the **Compute Using** options you choose in the view.

For example, a **Table (Across)** calculation moves from left to right across the entire width of the view:
A Pane (Down then Across) calculation moves down the first column in a pane, then down the second column in a pane, and so on:
The **Compute Using** approach should be convenient and intuitive for most users because the calculation aligns with what you see in your view.

The list of options under **Compute Using** changes according to the content of your view. For example, if you have only two dimensions in your view (one on Rows and one on Columns), there will be no Pane options because your view is not complex enough for there to be multiple panes—in this case, your entire view is single pane.
Table Calculations: Reference Specific Dimensions in Your View

With this approach, you define a table calculation by referencing the dimensions in the view. Choose **Specific Dimensions** in the Table Calculation dialog box and then select or clear fields in the box below.

![Table Calculation Dialog Box]

To configure a table calculation using specific dimensions, click Specific Dimensions and then select dimensions in the box below. You can also drag dimensions up or down in the list to set the sequence for the calculation:
You can still see the effect of your choices in the view, but because the visual structure and the table calculation are not necessarily aligned (as they always are with the **Compute Using** options), the results can sometimes look a bit exotic. For example:

If you are configuring table calculations using specific dimensions, then you should become familiar with the concepts of addressing and partitioning. See **Table Calculations: Addressing and Partitioning** on the next page.
Table Calculations: Addressing and Partitioning

If you are configuring table calculations using specific dimensions, then you should become familiar with the concepts of addressing and partitioning.

Background: How Level of Detail Affects the Direction and Scope of Your Calculations

Your view contains marks; these marks are typically aggregated. This means that an individual mark collects data not from a single row in your data source, but from all rows that have the same values for the dimensions in your view.

The number of marks in your view is determined by the view’s level of detail. Dimensions located on any of the shelves or cards highlighted in the following image contribute to the level of detail:

So if your view has State and Segment on Columns, and Region on Rows, then a single mark represents all rows in your data source that share the same value for these three dimensions. For example, one mark would aggregate individual values for rows where the value of State is Indiana, the value of Segment is Consumer, and the value of Region is Central. The measure that Tableau is aggregating—it might be Profit, Sales, or any other measure in the data source—is for you to decide, as is the aggregation itself—it might be Sum, Average or another aggregation.
When you add a table calculation, you must use all dimensions in the level of detail either for partitioning (scoping) or for addressing (direction):

- The dimensions that define how to group the calculation, that is, define the scope of data it is performed on, are called partitioning fields. The table calculation is performed separately within each partition.
- The remaining dimensions, upon which the table calculation is performed, are called addressing fields, and determine the direction of the calculation.

Partitioning fields break the view up into multiple sub-views (or sub-tables), and then the table calculation is applied to the marks within each such partition. The direction in which the calculation moves (for example, in calculating a running sum, or computing the difference between values) is determined by the addressing fields. So when you order the fields in the Specific Dimensions section of the Table Calculation dialog box from top to bottom, you are specifying the direction in which the calculation moves through the various marks in the partition.

When you add a table calculation using the **Compute Using** options, Tableau identifies some dimensions as addressing and others as partitioning automatically, as a result of your selections. But when you use Specific Dimensions, then it’s up to you to determine which dimensions are for addressing and which for partitioning.

The following example demonstrates how experimenting with the options in the Table Calculations dialog box can help you understand how your calculation is working.

**Example - Addressing and Partitioning**

In this example, using the Sample – Superstore data source provided with Tableau Desktop, create the following simple view:

1. Drag **Order Date** and **Segment** to Columns.
2. Drag **State** to Rows.
3. Drag **Sales** to Text.

   This gives you a basic view:
4. Click **SUM(Sales)** on the Marks card and then select **Add table calculation** from the context menu.

This opens the Table Calculation dialog box.

Note the yellow highlighting running across the top row of the view. Highlighting shows you exactly how the current **Compute Using** option partitions the view.

5. Set **Calculation Type** to **Running Total**.

6. Click **Specific Dimensions**, the last item in the **Compute Using** list.

The list box below the **Compute Using** list, which lists all the dimensions in the view’s level of detail, is now available for edit:
Any field with a check mark is an addressing field. **Year of Order Date** and **Segment** are checked, so this leaves **State** as the only partitioning field. This configuration matches **Table (Across)**, which was the **Compute Using** value that was selected when you clicked **Specific Dimensions**.

**Note**: You can select any of the **Compute Using** options and then click **Specific Dimensions** to see how your selected option translates in terms of specific dimensions.

**Year of Order Date** and **Segment** are both on Columns in the view, so these dimensions define the columns in the view. Notice that the values running from left to
right in the view are ascending—the running total increases across the entire width of the view.

But when you get to the second row (Arizona) you enter a new partition, so the calculation restarts.

To gain insight, try changing things up in the list of specific dimensions.

7. Clear the check mark from **Year of Order Date**.

**Year of Order Date** is now a partition field, leaving only **Segment** for addressing. **Year of Order Date** and **State** are both now partitioning fields, which means the partitions are more plentiful—and smaller, because you are using two fields to partition (think “divide up”) the view. There are only three marks per partition now, so as you continue moving left-to-right across the view, you enter a new partition with the fourth value, and the calculation restarts.

8. Reselect **Year of Order Date** and then drag **Segment** to the top of the list—above **Year of Order Date**. Highlighting shows that the partition is the same as in the default view, stretching left to right across the entire width of the view. This makes sense because the same two fields—**Segment** and **Year of Order Date**—are used for addressing.

But notice that the values are not the same as in the previous view:

Because you reversed the order in which the fields are listed for **Specific Dimensions**, Tableau computes the running total using values for **Segment** first and then **Year of Order Date**. The arrows in the image show the order in which Tableau is computing the running sum. The logic of the calculation is now somewhat at odds with the visual
alignment of the view, but that is what can happen when you configure a table calculation using Specific Dimensions.
Customize How Tableau Sorts the Results of Table Calculations

When you add a table calculation using specific dimensions from your view, you can also customize the way Tableau sorts the values. This option is available for all table calculation types except Rank and Percentile.

With custom sorting, you can sort the results of a table calculation using values from a field in the data source. With **Specific Dimensions** selected in the Table Calculations dialog box, do the following:

1. Click **Custom Sort** to display drop-down options.

2. Select **Custom Sort**.
3. From the drop-down list on the left, choose the field with the values you want to sort on.
4. From the drop-down list on the right, choose the aggregation you want to use with the field.
5. Specify whether to sort by **Descending** or **Ascending** values.

Create a Table Calculation

You can add a table calculation to a measure in the view. To add a table calculation:

1. Click a measure in the view and choose **Add table calculation** from the field menu.

   Tableau opens the Table Calculation dialog box and also adds highlighting to the view. As you work in the Table Calculation dialog box, highlighting shows the scope and direction of the calculation.

2. Choose a Calculation Type.

   For details on available calculation types, see **Table Calculation Types** on page 240.

   For each calculation type there is a specific set of options for you to consider. For example, for a **Difference From** calculation, there are two associated fields:
For information on these fields, go to Table Calculation Types on page 240, and then click the link for Difference From Calculation.

3. To define the table calculation using the visual structure of your view, choose one of the Compute Using options.
Alternatively, to define the table calculation by referencing specific dimensions in the view, choose **Specific Dimensions** and then select and order dimensions in the box further down in the Table Calculation dialog box.
For information on these two alternative approaches, see Choose Your Approach to Table Calculations on page 224. **Compute Using** is the recommended option for most users.

4. When you are satisfied with your table calculation, click the X in the upper-right corner of the Table Calculations dialog box to close it. Highlighting disappears. To indicate that the field now includes a table calculation, Tableau displays a small triangle next to the field name. Hover the cursor over this triangle to see a description of the table calculation:
Table Calculation Types

As you add or modify a table calculation, you must choose a calculation type. Each type has a specific set of options.
Table Calculation Type: Difference From

For each mark in the view, a **Difference From** table calculation computes the difference between the current value and another value in the table.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of O.</th>
<th>Month of Or.</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
</tr>
<tr>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
</tr>
<tr>
<td>March</td>
<td>$55,691</td>
<td>$38,467</td>
</tr>
<tr>
<td>Q2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
</tr>
<tr>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
</tr>
<tr>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
</tr>
<tr>
<td>Q3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
</tr>
<tr>
<td>August</td>
<td>$27,909</td>
<td>$36,898</td>
</tr>
<tr>
<td>September</td>
<td>$81,777</td>
<td>$64,596</td>
</tr>
<tr>
<td>Q4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
</tr>
<tr>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
</tr>
<tr>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see by how much values go up or down from month-to-month during the course of each year.

**Click for Information on How to Create the Basic View**

1. Connect to the **Sample - Superstore** data source.
2. Click and drag the **Order Date** field in the Data pane and drag it to the Columns shelf.
   
   The default date level is YEAR(Order Date).
3. Click and drag **Order Date** again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose **Quarter**.
You will see two options named **Quarter**. Be sure to choose the first one.

The field should now read **QUARTER(Order Date)**.

**Note:** If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Modify the basic view to show how by how much values go up or down from month-to-month during the course of each year.
1. Click the **SUM(Sales)** field on the Marks card and then select **Add table calculation**.

2. In the Table Calculation dialog box, verify that **Difference From** is the **Calculation Type**.
   
   This is the default Calculation Type, so you really shouldn't have to change this field value.

3. Choose **Table (Down)** from the **Compute Using** list.

   The highlighting shows how this **Compute Using** value sets the scope of the calculation in the view:

<table>
<thead>
<tr>
<th>Quarter of 0..</th>
<th>Month of Or..</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2011</td>
</tr>
<tr>
<td>Q1</td>
<td>January</td>
<td>-$9,136</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>-$5,963</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$50,880</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>-$27,396</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>-$4,647</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$10,947</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>-$649</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>-$6,037</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$53,868</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>-$50,324</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$47,175</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>-$9,083</td>
</tr>
<tr>
<td></td>
<td>2012</td>
<td>$3,968</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-$990</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$169</td>
</tr>
<tr>
<td></td>
<td>2013</td>
<td>$8,133</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-$5,175</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$13,088</td>
</tr>
<tr>
<td></td>
<td>2014</td>
<td>$27,698</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$39,564</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$28,973</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$33,191</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-$16,445</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$12,895</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$44,568</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$25,729</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$34,533</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$1,053</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$15,045</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$21,852</td>
</tr>
</tbody>
</table>

   Comparing the values in the original text view with the values in this view, you can verify that this is the result that you want.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

**Difference from what?**

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different:

<p>| Previous | Calculates the difference between the current value and the previous value in the partition. This is the default value. |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Calculates the difference between the current value and the next value in the partition.</td>
</tr>
<tr>
<td>First</td>
<td>Calculates the difference between the current value and the first value in the partition.</td>
</tr>
<tr>
<td>Last</td>
<td>Calculates the difference between the current value and the last value in the partition.</td>
</tr>
</tbody>
</table>

**At the level**

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension. How is this different from just using that dimension for partitioning? In fact, it is partitioning, but it’s partitioning by position rather than by value, which is how partitioning is defined with the Compute Using options.

The At the level option is only available when you select Specific Dimensions in the Table Calculations dialog box, and when more than one dimension is selected in the field immediately below the Compute Using options — that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with Compute Using, because those values establish partitions by position. But with Specific Dimensions, because the visual structure and the table calculation are not necessarily aligned, the At the level option is available to let you fine-tune your calculation.

For example, if you take the result of the Difference From calculation you added above, you can see the effect of At the level by doing the following:

1. Click the SUM(Sales) field on the Marks card and then select Edit table calculation.
2. In the Table Calculation dialog box, choose Specific Dimensions.

Notice that two dimensions are now checked in the list box of dimensions: Quarter of Order Date and Month of Order Date. These are the addressing fields, and because more than one field is being used for addressing, At the level is now available.

The choices available from the At the level drop-down list are:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Deepest</td>
<td>Specifies that the calculation should be performed at the level of finest granularity. This is the default option.</td>
</tr>
<tr>
<td>Quarter of Order Date</td>
<td>Specifies that the calculation should be performed at the quarter level.</td>
</tr>
<tr>
<td>Month of Order Date</td>
<td>Specifies that the calculation should be performed at the month level.</td>
</tr>
</tbody>
</table>
3. With this particular view, **Deepest** and **Month of Order Date** are equivalent options, and they are both equivalent to just choosing the **Table (Down)** option from **Compute Using**. The only option that yields a different result is **Quarter of Order Date**:

<table>
<thead>
<tr>
<th>Quarter of O.</th>
<th>Month of Or.</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>February</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>March</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$14,849</td>
<td>$16,021</td>
<td>$20,706</td>
<td>-4,591</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$18,838</td>
<td>$17,921</td>
<td>$33,823</td>
<td>$25,368</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>-21,096</td>
<td>-15,670</td>
<td>-11,756</td>
<td>-5,649</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$5,651</td>
<td>-5,430</td>
<td>-808</td>
<td>$8,316</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$4,261</td>
<td>$6,767</td>
<td>-23,426</td>
<td>$15,865</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$47,182</td>
<td>$39,799</td>
<td>$33,478</td>
<td>$42,229</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>-$2,493</td>
<td>$2,640</td>
<td>$18,022</td>
<td>$29,365</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$50,719</td>
<td>$39,074</td>
<td>$48,927</td>
<td>$50,810</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>-$12,232</td>
<td>$10,324</td>
<td>$24,329</td>
<td>-$14</td>
</tr>
</tbody>
</table>

The difference is now calculated between the first month in a given quarter and the first month in the previous quarter.
Table Calculation Type: Percent Difference From

For each mark in the view, a **Percent Difference From** table calculation computes the difference between the current value and another value in the table, as a percentage.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of O.</th>
<th>Month of Or.</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
<td>$22,868</td>
<td>$20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
<td>$38,467</td>
<td>$51,186</td>
<td>$53,909</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
<td>$56,691</td>
<td>$45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
<td>$39,430</td>
<td>$48,260</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
<td>$38,441</td>
<td>$48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
<td>$36,896</td>
<td>$33,266</td>
<td>$61,516</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$61,777</td>
<td>$64,596</td>
<td>$72,908</td>
<td>$90,489</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
<td>$82,192</td>
<td>$112,326</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
<td>$97,237</td>
<td>$90,475</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see how values change, in terms of percentage, from month to month during the course of each year.

**Click for Information on How to Create the Basic View**

1. Connect to the **Sample - Superstore** data source.
2. Click and drag the **Order Date** field in the Data pane and drag it to the Columns shelf. The default date level is YEAR(Order Date).
3. Click and drag **Order Date** again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose **Quarter**.
You will see two options named Quarter. Be sure to choose the first one.

The field should now read QUARTER(Order Date).

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag Order Date a third time and drop it on the Rows shelf to the right of QUARTER(Order Date).

6. Click on the right side of the field to open the context menu and this time choose Month (again, choose the first of two options named Month). The field should now read MONTH(Order Date).

7. Drag Sales from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Follow these steps to add a Percent Difference From table calculation to the basic view.

1. Click the SUM(Sales) field on the Marks card and then select Add table calculation.

2. In the Table Calculation dialog box, choose Percent Difference From as the
Calculation Type.

3. Choose **Table (Down)** from the **Compute Using** list.

The highlighting in the view shows how this Compute Using value sets the scope of the calculation in the view:

```
<table>
<thead>
<tr>
<th>Quarter of O..</th>
<th>Month of Or..</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>-66%</td>
<td>-33%</td>
<td>23%</td>
<td>-55%</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>1,058%</td>
<td>215%</td>
<td>124%</td>
<td>166%</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>-49%</td>
<td>-11%</td>
<td>-23%</td>
<td>-26%</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>-16%</td>
<td>-12%</td>
<td>44%</td>
<td>14%</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>46%</td>
<td>-18%</td>
<td>-30%</td>
<td>6%</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>-2%</td>
<td>16%</td>
<td>-3%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>-18%</td>
<td>28%</td>
<td>-13%</td>
<td>27%</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>193%</td>
<td>75%</td>
<td>119%</td>
<td>47%</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>-62%</td>
<td>-51%</td>
<td>-23%</td>
<td>-14%</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>150%</td>
<td>142%</td>
<td>46%</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>-12%</td>
<td>-1%</td>
<td>18%</td>
<td>-19%</td>
</tr>
</tbody>
</table>
```

Comparing the values in the original text view with the values in this view, you can see that the result is correct. For example, in the original view we see that the January 2010 value is 530,288 and the February 2010 value is 318,309. In the updated view, the value for February 2010 is -39.9%, which is 39.9% less than 530,288.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

You also have the option to click Compute compounded rate if you want the calculation to be compounded.

**Difference from what?**

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different:

<p>| Previous | Calculates the difference between the current value and the previous value in the partition. This is the default value. |</p>
<table>
<thead>
<tr>
<th></th>
<th>Calculates the difference between the current value and the next value in the partition.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Calculates the difference between the current value and the first value in the partition.</td>
</tr>
<tr>
<td></td>
<td>Calculates the difference between the current value and the last value in the partition.</td>
</tr>
</tbody>
</table>

**At the level**

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension. How is this different from just using that dimension for partitioning? In fact, it is partitioning, but it’s partitioning by position rather than by value, which is how partitioning is defined with the **Compute Using** options.

The **At the level** option is only available when you select **Specific Dimensions** in the Table Calculations dialog box, and when more than one dimension is selected in the field immediately below the **Compute Using** options — that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with **Compute Using**, because those values establish partitions by position. But with **Specific Dimensions**, because the visual structure and the table calculation are not necessarily aligned, the **At the level** option is available to let you fine-tune your calculation.

For an example of how the **At the level** option can affect a table calculation, see **Table Calculation Type: Difference From** on page 241.
Table Calculation Type: Percent From

For each mark in the view, a Percent From table calculation computes a value as a percentage of some other value—typically, as a percentage of the previous value in the table.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of O..</th>
<th>Month of Or..</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1 January</td>
<td></td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
<td>$22,868</td>
<td>$20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
<td>$38,467</td>
<td>$51,186</td>
<td>$53,909</td>
</tr>
<tr>
<td>Q2 April</td>
<td></td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
<td>$56,691</td>
<td>$45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
<td>$39,430</td>
<td>$48,260</td>
</tr>
<tr>
<td>Q3 July</td>
<td></td>
<td>$33,946</td>
<td>$28,765</td>
<td>$38,441</td>
<td>$48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
<td>$36,896</td>
<td>$33,266</td>
<td>$61,516</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$81,777</td>
<td>$64,596</td>
<td>$72,908</td>
<td>$90,489</td>
</tr>
<tr>
<td>Q4 October</td>
<td></td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
<td>$82,192</td>
<td>$112,326</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
<td>$97,237</td>
<td>$90,475</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see how each Year/Quarter/Month value differs from the previous month’s value, as a percentage.

Click for Information on How to Create the Basic View

1. Connect to the Sample - Superstore data source.
2. Click and drag the Order Date field in the Data pane and drag it to the Columns shelf.
   The default date level is YEAR(Order Date).
3. Click and drag Order Date again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose Quarter.
You will see two options named Quarter. Be sure to choose the first one.

The field should now read QUARTER(Order Date).

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag Order Date a third time and drop it on the Rows shelf to the right of QUARTER(Order Date).

6. Click on the right side of the field to open the context menu and this time choose Month (again, choose the first of two options named Month). The field should now read MONTH(Order Date).

7. Drag Sales from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Follow these steps to add a Percent From table calculation to the basic view.

1. Click the SUM(Sales) field on the Marks card and then select Add table calculation.
2. In the Table Calculation dialog box, choose Percent From as the Calculation Type.
3. Choose **Table (Down)** from the **Compute Using** list.

The highlighting in the view shows how this **Compute Using** value sets the scope of the calculation in the view:

<table>
<thead>
<tr>
<th>Quarter of O...</th>
<th>Month of Or...</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>34%</td>
<td>67%</td>
<td>123%</td>
<td>45%</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td></td>
<td>315%</td>
<td>224%</td>
<td>266%</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>1,156%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>51%</td>
<td>83%</td>
<td>111%</td>
<td>17%</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>84%</td>
<td>88%</td>
<td>144%</td>
<td>114%</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>146%</td>
<td>82%</td>
<td>70%</td>
<td>106%</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>98%</td>
<td>116%</td>
<td>97%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>82%</td>
<td>128%</td>
<td>87%</td>
<td>127%</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>293%</td>
<td>175%</td>
<td>219%</td>
<td>147%</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>38%</td>
<td>49%</td>
<td>77%</td>
<td>86%</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>250%</td>
<td>242%</td>
<td>146%</td>
<td>144%</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>88%</td>
<td>99%</td>
<td>118%</td>
<td>81%</td>
</tr>
</tbody>
</table>

Comparing the values in the original text view with the values in this view, you can verify that the result is correct. For example, in the original view we see that the January 2011 value is 13,946 and the February 2011 value is 4,811. In the updated view, the value for February 2010 is 34%; you can verify that 4,811 is about 34% of 13,946.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

**Difference from what?**

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different:

<table>
<thead>
<tr>
<th>Previous</th>
<th>Calculates the difference between the current value and the previous value in the partition. This is the default value.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>Calculates the difference between the current value and the next value in the partition.</td>
</tr>
<tr>
<td>First</td>
<td>Calculates the difference between the current value and the first value in the partition.</td>
</tr>
<tr>
<td>Last</td>
<td>Calculates the difference between the current value and the last value in the partition.</td>
</tr>
</tbody>
</table>

**At the level**

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension. How is this different from just using that dimension for partitioning? In fact, it is partitioning, but it’s partitioning by position rather than by value, which is how partitioning is defined with the **Compute Using** options.

The **At the level** option is only available when you select **Specific Dimensions** in the Table Calculations dialog box, and when more than one dimension is selected in the field immediately below the **Compute Using** options — that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with **Compute Using**, because those values establish partitions by position. But with **Specific Dimensions**, because the visual structure and the table calculation are not necessarily aligned, the **At the level** option is available to let you fine-tune your calculation.

For an example of how the **At the level** option can affect a table calculation, see **Table Calculation Type: Difference From** on page 241.
Table Calculation Type: Percent of Total

For each mark in the view, a Percent of Total table calculation computes a value as a percentage of all values in the current partition.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of O.</th>
<th>Month of Or.</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2011 2012 2013 2014</td>
</tr>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946 $18,174 $18,542 $44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811 $12,211 $22,868 $20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691 $38,467 $51,186 $53,909</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295 $34,195 $39,249 $40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648 $30,132 $56,691 $45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595 $24,797 $39,430 $48,260</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946 $28,765 $38,441 $48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909 $36,898 $33,266 $61,516</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$81,777 $64,596 $72,908 $90,489</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453 $31,405 $56,463 $77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629 $75,973 $82,192 $112,326</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546 $74,920 $97,237 $90,475</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see the percentage that each month contributes to each year’s total sales. So, for example, you want to see what percent of 2010 sales were realized in January of 2010, what percent in February, and so on.

Click for Information on How to Create the Basic View

1. Connect to the Sample - Superstore data source.
2. Click and drag the Order Date field in the Data pane and drag it to the Columns shelf.
   - The default date level is YEAR(Order Date).
3. Click and drag Order Date again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose Quarter.
You will see two options named **Quarter**. Be sure to choose the first one.

The field should now read **QUARTER(Order Date)**.

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Follow these steps to add a **Percent of Total** table calculation to the basic view.

1. Click the **SUM(Sales)** field on the Marks card and then select **Add table calculation**.
2. In the Table Calculation dialog box, choose **Percent of Total** as the **Calculation Type**.
3. Choose **Table (Down)** from the **Compute Using** list. The highlighting in the view shows how this **Compute Using** value sets the scope of the calculation in the view:

<table>
<thead>
<tr>
<th>Quarter of D..</th>
<th>Month of Or..</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>2.88%</td>
<td>3.86%</td>
<td>3.05%</td>
<td>6.09%</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>0.99%</td>
<td>2.60%</td>
<td>3.76%</td>
<td>2.76%</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>11.50%</td>
<td>8.18%</td>
<td>8.41%</td>
<td>7.95%</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>5.84%</td>
<td>7.27%</td>
<td>6.45%</td>
<td>5.47%</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>4.88%</td>
<td>6.40%</td>
<td>9.32%</td>
<td>6.22%</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>7.14%</td>
<td>5.27%</td>
<td>6.48%</td>
<td>6.58%</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>7.01%</td>
<td>6.11%</td>
<td>6.32%</td>
<td>6.60%</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>5.76%</td>
<td>7.84%</td>
<td>5.47%</td>
<td>8.38%</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>16.89%</td>
<td>13.73%</td>
<td>11.98%</td>
<td>12.33%</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>6.50%</td>
<td>6.67%</td>
<td>9.28%</td>
<td>10.60%</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>16.24%</td>
<td>16.15%</td>
<td>13.51%</td>
<td>15.30%</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>14.36%</td>
<td>15.92%</td>
<td>15.98%</td>
<td>12.33%</td>
</tr>
</tbody>
</table>

Comparing the values in the original text view with the values in this view verifies that the result is correct. For example, the 12 values in the view for 2011 add up to 100%.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

Choose **Compute across all pages** if you want the calculation to take account of any dimension on the Pages shelf.

**At the level**

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension. How is this different from just using that dimension for partitioning? In fact, it is partitioning, but it’s partitioning by position rather than by value, which is how partitioning is defined with the **Compute Using** options.

The **At the level** option is only available when you select **Specific Dimensions** in the Table Calculations dialog box, and when more than one dimension is selected in the field immediately below the **Compute Using** options —that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with **Compute Using**, because those values establish partitions by position. But with **Specific Dimensions**, because the visual structure and the table calculation are not necessarily aligned, the **At the level** option is available to let you fine-tune your calculation.
Table Calculation Type: Rank

For each mark in the view, a Rank table calculation computes a ranking for each value in a partition.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of O.</th>
<th>Month of O.</th>
<th>Order Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Column</td>
<td>2011</td>
</tr>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$81,777</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see how each month within a given year ranks in total sales, from most sales (ranked 1) to least sales (ranked 12).

**Click for Information on How to Create the Basic View**

1. Connect to the **Sample - Superstore** data source.
2. Click and drag the **Order Date** field in the Data pane and drag it to the Columns shelf.
   The default date level is YEAR(Order Date).
3. Click and drag **Order Date** again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose **Quarter**.
You will see two options named **Quarter**. Be sure to choose the first one.

The field should now read **QUARTER(Order Date)**.

**Note:** If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Follow these steps to add a **Rank** table calculation to the basic view.

1. Click the **SUM(Sales)** field on the Marks card and then select **Add table calculation**.
2. In the Table Calculation dialog box, choose **Rank** as the **Calculation Type**.
3. Choose **Table (Down)** from the **Compute Using** list.

   The highlighting in the view shows how this **Compute Using** value sets the scope of the calculation in the view:

   Comparing the values in the original text view with the values in this view shows that the result is correct. The values run from 1 to 12, and the highest sales total in 2011, for September (81,777), is ranked 1 in the updated view, while the second highest value, for November (78,629) is ranked 2 in the updated view.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

**Descending vs. Ascending**

**Ascending** order ranks values from least to most. **Descending** order ranks values from most to least. For Rank table calculation, the default value is **Descending**.

**Rank Type**

One issue with **Rank** calculations is that there may be more than one mark with the same value. What would happen, for example, if Tables in the Central region and Appliances in the South region both had sales of exactly $36,729? Tableau lets you specify how to handle such cases by including an additional field in the Table Calculation dialog box when you set **Calculation Type** to **Rank**.

The choices are listed below. The number sequence at the beginning of each option show how each option would rank a hypothetical set of four values where two of the values are identical:
<table>
<thead>
<tr>
<th>Option</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Competition (1, 2, 2, 4)</td>
<td>Identical values are assigned an identical rank. The highest value is ranked 1 and then the next two, identical values, are both ranked 2. The next value is then ranked 4.</td>
</tr>
<tr>
<td>Modified Competition (1, 3, 3, 4)</td>
<td>Identical values are assigned an identical rank. The highest value is ranked 1 and then the next two, identical values, are both ranked 3. The next value is then ranked 4.</td>
</tr>
<tr>
<td>Dense (1, 2, 2, 3)</td>
<td>Duplicate values are all given the same rank, which is the next number in the ranking sequence. The next value after the duplicate values is computed as though the duplicate values were a single value.</td>
</tr>
<tr>
<td>Unique (1, 2, 3, 4)</td>
<td>Duplicate values are given unique rankings, according to the direction in which the ranking is being computed.</td>
</tr>
</tbody>
</table>
Table Calculation Type: Percentile

For each mark in the view, a Percentile table calculation computes a percentile rank for each value in a partition.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of O.</th>
<th>Month of Or.</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
<td>$22,868</td>
<td>$20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
<td>$38,467</td>
<td>$51,186</td>
<td>$53,909</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
<td>$56,691</td>
<td>$45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
<td>$39,430</td>
<td>$48,260</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
<td>$38,441</td>
<td>$48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
<td>$36,898</td>
<td>$33,266</td>
<td>$61,516</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$81,777</td>
<td>$64,596</td>
<td>$72,908</td>
<td>$90,489</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
<td>$82,192</td>
<td>$112,326</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
<td>$97,237</td>
<td>$90,475</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see the values for each month in a given year expressed as a percentile. With this kind of table calculation, the lowest ranked value is assigned the 0 percentile and the highest ranked value is assigned the 100 percentile (assuming the calculation is working in ascending order, which is the default). All other values are assigned a percentile based on their ranking. So for a set of four values, 1, 3, 22, 67, the percentiles would be 0%, 33%, 67%, and 100%, respectively.

**Click for Information on How to Create the Basic View**

1. Connect to the Sample - Superstore data source.
2. Click and drag the Order Date field in the Data pane and drag it to the Columns shelf. The default date level is YEAR(Order Date).
3. Click and drag Order Date again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose Quarter.
You will see two options named **Quarter**. Be sure to choose the first one.

The field should now read **QUARTER(Order Date)**.

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Follow these steps to add a **Percentile** table calculation to the basic view.

1. Click the **SUM(Sales)** field on the Marks card and then select **Add table calculation**.
2. In the Table Calculation dialog box, choose **Percentile** as the **Calculation Type**.
3. Choose **Table (Down)** from the **Compute Using** list.

The highlighting in the view shows how this Compute Using value sets the scope of the calculation in the view:

<table>
<thead>
<tr>
<th>Quarter of Q..</th>
<th>Month of Or..</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>9.1%</td>
<td>9.1%</td>
<td>0.0%</td>
<td>18.2%</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>0.0%</td>
<td>0.0%</td>
<td>9.1%</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>72.7%</td>
<td>72.7%</td>
<td>54.5%</td>
<td>54.5%</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>36.4%</td>
<td>54.5%</td>
<td>35.4%</td>
<td>9.1%</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>18.2%</td>
<td>36.4%</td>
<td>72.7%</td>
<td>27.3%</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>63.6%</td>
<td>18.2%</td>
<td>45.5%</td>
<td>36.4%</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>54.5%</td>
<td>27.3%</td>
<td>27.3%</td>
<td>45.5%</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>27.3%</td>
<td>63.6%</td>
<td>13.2%</td>
<td>63.6%</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>100.0%</td>
<td>81.8%</td>
<td>81.8%</td>
<td>90.9%</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>45.5%</td>
<td>45.5%</td>
<td>63.6%</td>
<td>72.7%</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>90.9%</td>
<td>100.0%</td>
<td>90.9%</td>
<td>100.0%</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>81.8%</td>
<td>90.9%</td>
<td>100.0%</td>
<td>81.8%</td>
</tr>
</tbody>
</table>

Comparing the values in the original text view with the values in this view verifies that the result is correct. The lowest value in the original text table was February (4,711); in the new text view, February is 0.0%; the next lowest value in the original table was January (13,946); in the new view, January is 9.1%.

**Descending vs. Ascending**

Ascending order ranks values from least to most. Descending order ranks values from most to least. For a Percentile table calculation, the default value is **Ascending**.
Table Calculation Type: Running Total

For each mark in the view, a **Running Total** table calculation aggregates values cumulatively in a partition. It can do this by summing values, averaging values, or replacing all values with either the lowest or highest actual value.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of Year</th>
<th>Month of Order</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
<td>$22,868</td>
<td>$20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
<td>$38,467</td>
<td>$51,186</td>
<td>$53,909</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
<td>$56,691</td>
<td>$45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
<td>$39,430</td>
<td>$48,260</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
<td>$30,441</td>
<td>$48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
<td>$36,898</td>
<td>$33,266</td>
<td>$61,516</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$81,777</td>
<td>$64,596</td>
<td>$72,908</td>
<td>$90,489</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
<td>$82,192</td>
<td>$112,326</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
<td>$97,237</td>
<td>$90,475</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to see a running total of sales for each year, such that each month’s sales are added to all previous months’ sales.

**Click for Information on How to Create the Basic View**

1. Connect to the **Sample - Superstore** data source.
2. Click and drag the **Order Date** field in the Data pane and drag it to the Columns shelf.
   - The default date level is YEAR(Order Date).
3. Click and drag **Order Date** again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose **Quarter**.
You will see two options named **Quarter**. Be sure to choose the first one.
The field should now read **QUARTER(Order Date)**.

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Follow these steps to add a **Running Total** table calculation to the basic view.

1. Click the **SUM(Sales)** field on the Marks card and choose **Add table calculation**.
2. In the Table Calculation dialog box, choose **Running Total** as the **Calculation Type**.
3. Choose **Table (Down)** from the **Compute Using** list.

The highlighting in the view shows how this **Compute Using** value sets the scope of the calculation in the view:

Comparing the values in the original text view with the values in this view shows that the result is correct. The monthly values ascend steadily and the December value (484,247) is the same value you see if you show column grand totals (from the Analysis menu, select **Totals > Show column grand totals**).

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

**The Running Total Doesn’t Have to Be a Sum**

For a **Running Total** table calculation, Tableau can update values cumulatively in other ways than summing. Choose one of the options from the drop-down list just below the **Calculation Type** field:

<table>
<thead>
<tr>
<th>Option</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>Each value is added to the previous value.</td>
</tr>
<tr>
<td>Average</td>
<td>The running total averages the current and all previous values.</td>
</tr>
<tr>
<td>Minimum</td>
<td>All values are replaced with the lowest value in the original partition.</td>
</tr>
<tr>
<td>Maximum</td>
<td>All values are replaced with the highest value in the original partition.</td>
</tr>
</tbody>
</table>
Restarting Every

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension.

**Restarting every** can be useful in the following situations:

- With dates or other hierarchies, if you restart every month, as you bring in Year or Quarter, Tableau knows to partition automatically.
- With non-hierarchies, Restarting every affects the sorting. If you want to address on **Products** and partition by **State**, but you want the products sorted by **SUM(Sales)** within each state, you need to include **States** as an addressing field under Specific Dimensions, but then restart every state. Otherwise, the sort by **SUM(Sales)** would be based on each product's sum of sales across all states.

The Restarting every option is only available when you select **Specific Dimensions** in the Table Calculations dialog box and when more than one dimension is selected in the field immediately below the **Compute Using** options—that is, when more than one dimension is defined as an addressing field.

This option is not available when you’re defining a table calculation with **Compute Using**.

For example, if you take the result of the Running Total calculation you added above, you can see the effect of Restarting every by doing the following:

1. Click the **SUM(Sales)** field on the Marks card and choose **Edit table calculation**.
2. In the Table Calculation dialog box, choose **Specific Dimensions**.
   
   Notice that two dimensions are now checked in the list box of dimensions: Quarter of Order Date and Month of Order Date. These are the addressing fields, and because more than one field is being used for addressing, Restarting every is now available.

   The choices available from the At the level drop-down list are:

<table>
<thead>
<tr>
<th>None</th>
<th>Specifies that the calculation should be performed at the level of greatest granularity. This is the default option. This option does not change the view.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter of Order Date</td>
<td>Specifies that the calculation should be performed at the quarter level.</td>
</tr>
</tbody>
</table>

3. If you choose Quarter of Order Date, the view updates to show the effect of this change:
The calculation now restarts after every quarter. If you click out of the Table Calculations dialog box (to dismiss the highlighting) you can see this more clearly.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

**Add Secondary Calculation**

With *Running Total* and *Moving Calculation* table calculations, you have the option to transform values twice to obtain the result you want—that is, to add a secondary table calculation on top of the primary table calculation. For example, you could add an initial table calculation to calculate the running total for sales per month within each individual year, and then a secondary calculation to calculate the year-over-year percent difference for each month from one year to the next.

To do this, first add the primary table calculation, as shown above. Then continue as follows:

1. Click the **SUM(Sales)** field on the Marks card and select **Edit table calculation**.
2. In the Table Calculation dialog box, click **Add Secondary Calculation**.

   The Table Calculation dialog box expands to show a second panel:
3. In the second panel, choose **Percent Difference From** as the **Secondary Calculation Type**.

4. You do not need to change the **Compute Using** selection: **Table (Across)** is the right option.

5. Click the X in the upper-right corner to dismiss the Table Calculation dialog box.

   Now your view shows what you needed: a year-over-year percent difference of a running total:
<table>
<thead>
<tr>
<th>Quarter of Order</th>
<th>Month of Order</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$4,228</td>
<td>$368</td>
<td>$26,161</td>
<td></td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$11,628</td>
<td>$11,025</td>
<td>$23,576</td>
<td></td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>-$5,596</td>
<td>$23,745</td>
<td>$26,299</td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$304</td>
<td>$28,798</td>
<td>$27,163</td>
<td></td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$6,787</td>
<td>$55,357</td>
<td>$16,123</td>
<td></td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>-$3,011</td>
<td>$69,991</td>
<td>$24,952</td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>-$8,192</td>
<td>$79,666</td>
<td>$34,940</td>
<td></td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$797</td>
<td>$76,033</td>
<td>$63,191</td>
<td></td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>-$16,384</td>
<td>$84,345</td>
<td>$80,771</td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>-$16,433</td>
<td>$109,404</td>
<td>$102,102</td>
<td></td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>-$19,089</td>
<td>$115,623</td>
<td>$132,236</td>
<td></td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>-$13,715</td>
<td>$137,941</td>
<td>$125,473</td>
<td></td>
</tr>
</tbody>
</table>
Table Calculation Type: Moving Calculation

For each mark in the view, a Moving Calculation table calculation determines the value for a mark in the view by performing an aggregation (sum, average, minimum, or maximum) across a specified number of values before and/or after the current value.

A moving calculation is typically used to smooth short-term fluctuations in your data so that you can see long-term trends. For example, with securities data there are so many fluctuations every day that it is hard to see the big picture through all the ups and downs. You can use a moving calculation to define a range of values to summarize using an aggregation of your choice.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

<table>
<thead>
<tr>
<th>Quarter of Or.</th>
<th>Month of Or.</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
<td>$18,174</td>
<td>$18,542</td>
<td>$44,703</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$4,811</td>
<td>$12,211</td>
<td>$22,868</td>
<td>$20,284</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$55,691</td>
<td>$84,467</td>
<td>$51,186</td>
<td>$53,909</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$28,295</td>
<td>$34,195</td>
<td>$39,249</td>
<td>$40,112</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$23,648</td>
<td>$30,132</td>
<td>$56,691</td>
<td>$45,651</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$34,595</td>
<td>$24,797</td>
<td>$39,430</td>
<td>$48,260</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$33,946</td>
<td>$28,765</td>
<td>$38,441</td>
<td>$48,428</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$27,909</td>
<td>$36,898</td>
<td>$33,266</td>
<td>$61,516</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$41,777</td>
<td>$64,596</td>
<td>$72,908</td>
<td>$90,489</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$31,453</td>
<td>$31,405</td>
<td>$56,463</td>
<td>$77,794</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$78,629</td>
<td>$75,973</td>
<td>$82,192</td>
<td>$112,326</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$69,546</td>
<td>$74,920</td>
<td>$97,237</td>
<td>$90,475</td>
</tr>
</tbody>
</table>

Instead of absolute sales values, you want to find out how sales totals are trending over time. To do this, you transform each monthly total so that it averages the monthly total for the two previous months and the current month. So instead of seeing just raw sales totals for October, 2010 (for example), you want to replace that number with a number that averages sales for August, September, and October.

**Click for Information on How to Create the Basic View**

1. Connect to the Sample - Superstore data source.
2. Click and drag the Order Date field in the Data pane and drag it to the Columns shelf.
   - The default date level is YEAR(Order Date).
3. Click and drag **Order Date** again and drop it this time on the Rows shelf.

4. Click on the right side of the field to open the context menu. Then choose **Quarter**.

You will see two options named **Quarter**. Be sure to choose the first one.

The field should now read **QUARTER(Order Date)**.

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.

6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.

7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.
Follow these steps to add a Moving Calculation table calculation to the basic view.

1. Click the SUM(Sales) field on the Marks card and then click Add table calculation.
2. In the Table Calculation dialog box, choose Moving Calculation as the Calculation Type.
3. Click in the drop-down list below the Calculation Type option to display the options that you use to configure the calculation:

![Calculation Type](image)

4. Choose from the drop-down list of available aggregations: Sum, Average, Minimum, or Maximum. In this case, you want Average.

5. Using the Previous values option, specify how many marks preceding the current mark should be included in calculating the average. In this case, you want 2.

6. Using the Next values option, specify how many marks following the current mark should be included in calculating the average. In this case, you want 0.

7. Click Current value to indicate that you want the current value to be included in the calculation.

8. Do not click Null if there are not enough values. You can select this option if you want the current value to be null if there are not enough previous or next values to perform the calculation as specified. When you do not click this option, Tableau will use as many of the values within the specified range as are available.

When you close the drop-down list, Tableau shows a summary of the options you have chosen:
Choose **Table (Down then Across)** from the **Compute Using** list.

This will cause the calculation to run through the entire time range—from January 2010 to December 2013.

Your view is now “smoothed” to show how sales numbers are trending over time:

<table>
<thead>
<tr>
<th>Quarter of Year</th>
<th>Month of Order</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>January</td>
<td>$13,946</td>
<td>$166,348</td>
<td>$169,435</td>
<td>$224,133</td>
</tr>
<tr>
<td></td>
<td>February</td>
<td>$18,757</td>
<td>$99,931</td>
<td>$116,330</td>
<td>$162,224</td>
</tr>
<tr>
<td></td>
<td>March</td>
<td>$74,448</td>
<td>$68,852</td>
<td>$92,596</td>
<td>$118,896</td>
</tr>
<tr>
<td>Q2</td>
<td>April</td>
<td>$88,797</td>
<td>$84,873</td>
<td>$113,303</td>
<td>$114,305</td>
</tr>
<tr>
<td></td>
<td>May</td>
<td>$107,635</td>
<td>$102,794</td>
<td>$147,126</td>
<td>$139,673</td>
</tr>
<tr>
<td></td>
<td>June</td>
<td>$86,539</td>
<td>$89,124</td>
<td>$135,870</td>
<td>$134,023</td>
</tr>
<tr>
<td>Q3</td>
<td>July</td>
<td>$92,190</td>
<td>$33,694</td>
<td>$134,562</td>
<td>$142,339</td>
</tr>
<tr>
<td></td>
<td>August</td>
<td>$96,451</td>
<td>$90,461</td>
<td>$111,137</td>
<td>$158,204</td>
</tr>
<tr>
<td></td>
<td>September</td>
<td>$143,633</td>
<td>$130,260</td>
<td>$144,614</td>
<td>$200,433</td>
</tr>
<tr>
<td>Q4</td>
<td>October</td>
<td>$141,140</td>
<td>$132,899</td>
<td>$162,637</td>
<td>$229,799</td>
</tr>
<tr>
<td></td>
<td>November</td>
<td>$191,859</td>
<td>$171,973</td>
<td>$211,564</td>
<td>$280,609</td>
</tr>
<tr>
<td></td>
<td>December</td>
<td>$179,628</td>
<td>$132,297</td>
<td>$235,893</td>
<td>$280,595</td>
</tr>
</tbody>
</table>

10. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

**Add Secondary Calculation**

With **Running Total** and **Moving Calculation** table calculations, you have the option to transform values twice to obtain the result you want—that is, to add a secondary table calculation on top of the primary table calculation. For example, you could add an initial table calculation to calculate the running total for sales per month within each individual year, and then a secondary calculation to calculate the year-over-year percent difference for each month from one year to the next.

For an example showing how to create a secondary calculation, see **Table Calculation Type: Running Total** on page 264.
Customize Table Calculations

You can always customize a table calculation by editing it in the Table Calculations dialog box, but there are other, more specialized ways to customize a table calculation.

Customizing a Table Calculation Using Its Context Menu

Click any field in the view to see a context menu listing ways to customize the field. For a field in the view that has a table calculation, you can change the Compute Using option—that is, the option that determines the direction and scope of the calculation relative to the visual structure of the view. Do this by clicking the field and then choosing an option from the Compute Using list.

For Difference From, Percent Difference From, and Percent From table calculations, you can also specify a different field from which the difference should be computed. Do this by clicking the field and then choosing an option from the Relative to list. The options are Previous, Next, First, and Last.

Customizing a Table Calculation Using the Calculation Editor

You can customize a table calculation by dragging it into the calculation editor:

When you edit a table calculation in the calculation editor, you can click Default Table Calculation in the lower-right corner of the editor to open the calculation in the Table
Calculation dialog box. This will allow you to create a new named calculated field that uses the same table calculation as the one you are basing the calculation on.

**Nested Table Calculations**

A nested table calculation can be one of two types of calculated fields:

- A calculated field that includes more than one calculated field with a table calculation (as in the example below), or
- A calculated field that itself has a table calculation and includes at least one calculated field with a table calculation.

With nested table calculations, you can set **Compute Using** configurations for individual calculations independently.

Here is a scenario you can try, using the Sample - Superstore data source that is included with Tableau Desktop, which results in a nested table calculation.

1. Drag **Sub-Category** to Columns and **Region** to Rows.
2. Create a calculated field, **1-nest**, with the definition `TOTAL(SUM([Sales]))`.
   `TOTAL` is a table calculation function, so this calculated field automatically has a table calculation—when you use it in the view, the field will have the tell-tale table triangle, indicating a table calculation:

   ![1-nest](image)

3. Create a second calculated field, **2-nest**, with the definition `TOTAL(SUM([Profit]))`.
4. Create a third calculated field, **3-nest**, with the definition `[1-nest] + [2-nest]`.
5. Drag **3-nest** and drop it to the right of **Sub-Category** on Columns.
6. Click **3-nest** on Columns and choose **Edit Table Calculation**.
   In the Table Calculations dialog box, you can now separately configure the underlying
Manage Your User Settings and Profile

This section provides information about how to set and change a user password, email address, display name, and other information.

Manage Your Account Settings

Use your Account Settings page to change your display name and password (local users only), add or change your email address, manage your subscription settings, change your start page, change the language and locale you see in Tableau Server, or clear cookies for data connection passwords.

Go to your Account Settings page

- At the top of the browser window, select your name, and then select My Account Settings.
Quickly Access Your Content

Use your Content page to quickly browse items that you’ve published and your subscriptions.

At the top of the browser window, select your name, and then select My Content.

Manage Your Credentials and Passwords

When you access a workbook or data source that has a live connection to data and requires you to sign in, Tableau offers to save your password for you. If you accept, it stores your credentials in a cookie or an access token, depending on the data type. You can remove these credentials if you no longer use the data, or you have exceeded the maximum number of saved credentials and want to make room for a new one.

Under Saved Credentials, do either of the following:

- Select the Delete link next to an individual access token.
- Select Clear All Saved Credentials.

When you clear all credentials, the following items are removed from your user account:
Passwords you used to access published data sources or workbooks that connect to them.

Access tokens for OAuth data connections, such as to Google or Salesforce.com data.

**Caution:** Removing an access token is effectively like “changing the locks.” If the token is stored with workbooks or data sources you published, deleting the token also removes access to the data from those workbooks and data sources. If the token is embedded in an extract connection, and the extract is refreshed on a schedule, the refreshes will not be able to complete until you embed the new credentials or access token into the connection.

**Remove connected clients**

The first time you sign in to Tableau Server from a Tableau client, a secure token is created and stored in your account. This token allows you to access the site directly, without having to sign in. A client can be the Tableau Mobile app, Tableau Desktop, the Tableau Online sync client, and others.

You can delete a connected client (token) if you no longer use it, or if you want to add a new client but get an error that says you’re already using the maximum number of connected clients. After you remove a connected client from your account, you will need to provide your credentials the next time you access Tableau Server from that client.

- In the **Connected clients** section, next to the client you want to remove, select **Delete**.

**Opt in to or out of extract refresh failure email notification**

If you own published data sources that contain extracts on a refresh schedule, you can receive email notification when the scheduled refresh does not complete successfully.

- Under **Email Notification**, select the **Send email when extract refreshes fail** check box to opt in, or clear it to opt out.

The email contains the following information:

- Extract or workbook name.
- The last successful refresh time.
- The number of consecutive times the refresh has failed.
  
  After five consecutive failures, refreshes are suspended until you take an action to resolve the cause of the failure.
- A suggested action to take to address the cause of the failure—such as updating embedded credentials or a path to the original data file—and a link to the server to take the action.

**Manage Your Subscription Settings**
Go to **Subscriptions** in your content to change the schedule for any subscriptions you're receiving.

1. Click **Subscriptions**, and then select the workbook or view.
2. Select **Actions > Change Schedule**.

3. Select the new schedule from the list of available schedules, and then click **Change Schedule**.

To change the subject name of the subscription, click **Change Subject**. To unsubscribe from a view or workbook, by click **Delete**.

**Change Your Start Page**

To change the start page that appears when you sign in to Tableau Server, navigate to the page you want, click your name in the upper right area of the page, and then click **Make This My Start Page**.
Server administrators can specify a different default start page for all users.

To return to using the default start page, click your user name, and then click **My Account Settings**. For start page, click **Reset to Default**.

The URL for your current start page is displayed here. You can click the link to go to the page.

**Language and Locale**

The **Language** setting controls the language you see for the Tableau Server user interface and **Locale** affects views, such as how numbers are formatted, or which currency is used. Your administrator can configure these settings for all server users, but you can change them here, just for yourself. If you do change the settings, note that they will only take effect if they are a supported language. See **Language and Locale** on page 661 to learn more.

Change the **Language** and **Locale**, and then click **Save Changes**. The language and locale update immediately and will be continue to be used for your server sessions the next time you sign in.

**Change Your Display Name**

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your display name. Select the display name text and enter the new display name, and then click **Save Changes**.

**Change Your Password**

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your Tableau Server password by clicking **Change Password**. When you click this link you are asked to enter your **Current Password** and the **New Password** (twice). After you've typed in the required information, click **Save Password** to save the changes.
Change Your Email Address

If you have a subscription for a Tableau Server view or workbook, the email account that receives the subscription is listed on the Account Settings page.

To enter or change the email address that Tableau Server sends subscriptions to, enter the new email address in the Email text box, and then click Save Changes.

Save Passwords for Database Access

Sometimes a view requires you to enter a database user name and password. If you have access to the database you should enter your user name and password into the appropriate text boxes. If you select the Remember my password option you will be automatically signed in each time you look at the view. Your sign in information is stored encrypted on the server so you will be automatically signed in even between browser sessions and when accessing the view from multiple computers. This is convenient when you have a select number of views that you access all the time.

Administrators can restrict whether to allow users to remember database passwords. If you are an administrator, see Server Settings (General) on page 1177 to learn more.
Clearing and Resetting Saved Passwords

If your passwords are being saved (Allow users to save data source passwords is enabled on the General tab of the Settings page for the server), you can clear your saved passwords. When you do this, the next time you visit the server, you are prompted to enter your user name and password. You may want to do this if your user name and password change so you can begin using and saving your new credentials.

1. Click your name at the top of the page, and then select My Account Settings.

2. In Manage Credentials, click Clear All Saved Credentials.

Note: Administrators can also clear all saved passwords on the server using the Clear All Saved Credentials for all users link on the Server Settings - General page.
Manage Credentials for OAuth Data Connections

If your server administrator has allowed you to save access tokens for OAuth data connections, you can find and manage them in the Saved Credentials section on your Account Settings page. Access tokens represent OAuth connections to Salesforce.com, Google BigQuery, or Google Analytics data sources.

**Note:** If you do not see the Saved Credentials section, consult with your Tableau Server administrator about allowing saving access tokens.

Remove saved credentials

To revoke Tableau access to data, delete the associated saved access token from your account.

After you delete the credentials, you will need to sign in to the data the next time you access it. This will create a new saved token.

Your administrator might choose for all users to use the same shared credentials for connecting to a data source. If this is the case, the access token is associated with the data connection for all users, and it does not appear under Saved Credentials on your Account Settings page.

**Note:** If you are not able to delete saved credentials, your administrator might have cleared the Allow users to save data source access tokens option in the server settings. The administrator will need to enable the option again for you to successfully delete the saved credentials.

Test Connections for Saved Credentials

1. While you’re signed in to Tableau Server, display your Account Settings page.
2. In the Manage Credentials section, click Test link next to the stored connection that you want to test.

This test confirms whether your credentials are approved for OAuth connections and can use access tokens. If a test succeeds, but you cannot access your data through this managed connection, confirm that the credentials you provided for this connection can access your data.

For example, if you accidentally created the connection using your personal Gmail account, but you use a different account to access a Google Analytics database, you will need to delete the saved credentials and sign in to the data using the appropriate Gmail account.

**Clear all Saved Credentials**

When you select Clear All Saved Credentials, the following items are removed from your user account:
All access tokens for OAuth connections that are stored in your account.

**Caution:** If any of these tokens are stored with published workbooks or data sources, deleting them also removes access to the data source from those locations. Effectively, this is like “changing the locks” anywhere the affected tokens are used.

Passwords you have used to access published data extracts or workbooks that connect to them.

**Embed Views into Webpages**

You can embed interactive Tableau views and dashboards into web pages, blogs, wiki pages, web applications, and intranet portals. Embedded views update as the underlying data changes, or as their workbooks are updated on Tableau Server. Embedded views follow the same licensing and permission restrictions used on Tableau Server. That is, to see a Tableau view that's embedded in a web page, the person accessing the view must also have an account on Tableau Server. As an alternative, if you have a core-based license you can select Enable Guest account, which allows users to load the view without signing in.

You can embed views the following ways:

- **Use the Share embed code:** The Share button at the top of each view provides embed code that you can copy and paste into your webpage. (If you don't see the Share button, ask your administrator if they've used the tabadmin command to change the vizqlserver.showshare option.)

- **Write your own embed code:** You can enhance the embed code that Tableau provides, or you can build your own code. Either way you can use parameters that control the toolbar, tabs, and more.

**Note:** For users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

**Writing Embed Code**

If you’re writing your own embed code, you can take one of two approaches:

- **Use Tableau JavaScript:** This is the preferred approach. Use the embed code that Tableau generates as the starting point for your own code, adding or editing object parameters that control the toolbar, tabs, and more. The default embed code, which relies on a Tableau JavaScript file, is also the only way to control the load order of multiple embedded views.

- **Specify the View URL:** Embed a view using an Iframe or Image tag, where the source
is the URL from the Link box of the Share dialog box. You may want to do this if you can’t use JavaScript on your website. There may also be situations when all you can specify is an URL—such as if you’re embedding a view using the SharePoint Page Viewer Web Part.

When you embed a view, you should define a width and height that the view will be displayed in. If you do not do this, the client browser will arbitrarily pick a width and height.

Tableau JavaScript

The following code shows an example of embed code that is generated when you click Share on a published view. Special characters in the host_url parameter are URL encoded, and those in the site_root and name parameters are notated as HTML numeric character references.

```
<script type='text/javascript' src='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='host_url' value='http%3A%2F%2Fmyserver%2F' />
  <param name='site_root' value='/t/Sales' />
  <param name='name' value='MyCoSales/SalesScoreCard/' />
  <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.

View URL as the Source

Here’s an example of embedding the same view using an IFrame, where the source is the URL from the Link box of the Share dialog box:

```
<iframe src='http://myserver/t/Sales/views/MyCoSales/SalesScoreCard?embed=yes&amp;:tabs=yes&amp;toolbar=yes' width='800' height='600'></iframe>
```

The `embed` URL parameter is required, and you can optionally include parameters that control the toolbar and revert options, among others. You can also add filters to the URL that control the specific data that shows when a view is loaded.
See also

For examples, see the Parameters You Can Use to Embed a View on page 504 and search for "Script Tag Examples" in the Tableau Server Help.

Parameters You Can Use to Embed a View

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.

<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>customViews</td>
<td>:customViews</td>
<td>no</td>
<td>Hides the Remem-&lt;br&gt;ber my changes option.</td>
<td>&lt;param name='customViews' value='no'/&gt;&lt;br&gt;<a href="http://tabserver/views/Date-Time(DateCalc-s?:embed=yes:&amp;amp;customViews=no">http://tabserver/views/Date-Time(DateCalc-s?:embed=yes:&amp;amp;customViews=no</a></td>
</tr>
<tr>
<td>device</td>
<td>:device</td>
<td>desktop&lt;br&gt;tablet&lt;br&gt;phone</td>
<td>If a dash-&lt;br&gt;board has layouts for mobile devices, it displays a specific layout, regardless of the device type. If this parameter isn't set, Tableau Server detects screen size and loads the appropriate layout. See Embed Dash-&lt;br&gt;-board?</td>
<td>&lt;param name='device' value='phone'/&gt;&lt;br&gt;<a href="http://tabserver/views/sales/sales_dashboards?device=tablet">http://tabserver/views/sales/sales_dashboards?device=tablet</a></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>boards</td>
<td></td>
<td></td>
<td>para values</td>
<td>boards on page 519 for examples.</td>
</tr>
<tr>
<td>:embed</td>
<td>yes</td>
<td></td>
<td>Required for URL parameter. Hides the top navigation area, making the view blend into your web page better.</td>
<td><a href="http://tabserver/views/Date-Time/DateCalcs?:embed=yes">http://tabserver/views/Date-Time/DateCalcs?:embed=yes</a></td>
</tr>
<tr>
<td>filter</td>
<td>:string</td>
<td></td>
<td>Customizes what data is displayed when the view opens. You can also filter by URL parameters.</td>
<td>&lt;param name='filter' value='Team=Blue'/&gt;</td>
</tr>
<tr>
<td>:format</td>
<td>:pdf; png</td>
<td></td>
<td>Displays a view as a PDF or .png file.</td>
<td><a href="http://t-%D0%B0%D0%B1%D0%BB%D0%B5au-server/views/Sales/Q2?:format=pdf">http://t-аблеau-server/views/Sales/Q2?:format=pdf</a></td>
</tr>
<tr>
<td>:high-dpi</td>
<td>false</td>
<td></td>
<td>Renders a view using standard DPI (dots</td>
<td><a href="http://t-%D0%B0%D0%B1%D0%BB%D0%B5au-server/views/Sales/Q2?:highdpi=false">http://t-аблеau-server/views/Sales/Q2?:highdpi=false</a></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>-</td>
<td>:original_view</td>
<td>yes</td>
<td>If the name parameter refers to a workbook or sheet URL (and does not explicitly refer to a custom view) including this parameter displays the view as the original view, ignoring other custom views that might be available.</td>
<td>&lt;param name='filter' value=':original_view=yes'/&gt;</td>
</tr>
</tbody>
</table>
| host_url        | -            | string | The server name as it appears in the URL. | <param name='host_url' value='http://myserver.exampleco.com'/>  
<param name="host_url" value="http://localhost/"> |
| link-target     | :link-target | string | The target window name for external | <param name="linktarget" value="_blank"/>  
http://tabserver/views/Date- |
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| load-order | - | number | When multiple views are embedded, the default load order is the order in which the views are listed. Use this setting to override that order. Negative numbers are allowed. | `<param name="load-order" value="2"/>
` |
<p>| name | - | string | Required for object parameter. Workbook and sheet name and optionally, a custom view (username@domain/[custom view name]). If you refer to the Tableau | <code>&lt;param name='name' value='ExampleCoSales/Sales'/&gt;</code>&lt;br&gt;<code>&lt;param name='name' value='ExampleCoSales/Sales/jsmith@example.com/EastCoastSales'/&gt;</code> |</p>
<table>
<thead>
<tr>
<th><strong>Object Parameter</strong></th>
<th><strong>URL Parameter</strong></th>
<th><strong>Values</strong></th>
<th><strong>Description</strong></th>
<th><strong>Examples</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Server URL to confirm the value of name, exclude the session ID ( :iid= &lt;n&gt; ) at the end of the URL.</td>
<td>path</td>
<td>string</td>
<td>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 Trusted Authentication examples.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>:record_performance</td>
<td>yes</td>
<td>Starts a performance recording for a view. Add this at the end of</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the URL, immediately before the session ID (:iid=\textless n\textgreater). See Create a Performance Recording on page 1139 for more information.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>true; false; number</td>
<td>If client-side rendering is enabled (which is the default), setting this to false forces server-side rendering</td>
<td><a href="http://tabserver/views/DateTime/DateCalculs?:render=false">http://tabserver/views/DateTime/DateCalculs?:render=false</a></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>-</td>
<td>:revert</td>
<td>all;</td>
<td>Returns the item to its original state.</td>
<td><a href="http://tabserver/views/DateTime/DateCalc-s?:embed=yes&amp;:revert=all">http://tabserver/views/DateTime/DateCalc-s?:embed=yes&amp;:revert=all</a></td>
</tr>
<tr>
<td>showShareOptions</td>
<td>-</td>
<td>true;</td>
<td>Controls whether the Share options are displayed</td>
<td>&lt;param name='showShareOptions' value='true' /&gt;</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>in an embedded view. The default is true.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If Share options still appear when you change this parameter to false, ask your Tableau Server administrator to change...
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| site_root        | -             | string| Required. The site name. The Default site value is null (value=''). If your server is multi-site and you want to use trusted authentication, see the Trusted Authentication examples. | `<param name='site_root' value='#/Sales'/>`  
`<param name='site_root' value=''/>` |
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>tabs</td>
<td>:tabs</td>
<td>yes; no</td>
<td>Displays or hides tabs.</td>
<td><code>&lt;param name='tabs' value='yes'/&gt;</code></td>
</tr>
<tr>
<td>ticket</td>
<td>-</td>
<td>number</td>
<td>For trusted authentication only, cannot be used with the path object parameter. Must be used with name object to construct the trusted ticket redemption URL. See the Trusted Authentication examples.</td>
<td><code>&lt;param name='ticket' value='Etdpsm_Ew6rJY-9kRrALjauU'/&gt;</code>&lt;br&gt;<a href="http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes">http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes</a></td>
</tr>
<tr>
<td>toolbar</td>
<td>:toolbar</td>
<td>yes; no; top</td>
<td>The toolbar is displayed by default on the bottom when this parameter is not set. When no the toolbar is excluded from the</td>
<td><code>&lt;param name='toolbar' value='top'/&gt;</code>&lt;br&gt;<a href="http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:toolbar=no">http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:toolbar=no</a></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>embedded view. When top, the toolbar is placed above the view.</td>
<td></td>
</tr>
<tr>
<td>tooltip</td>
<td>:tooltip</td>
<td>yes; no</td>
<td>Tooltips are displayed by default in a view when this parameter is not set. If set to no, tooltips are excluded from the embedded view.</td>
<td></td>
</tr>
</tbody>
</table>

**Examples**

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

**Add Filters**

You can pass filter values so the view opens showing just the data you want. For example, you may want to include a hyperlink from another part of your web application to an embedded sales performance view that only shows a specific region.

**Script Tag Example**

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style='display:none;'>
```
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/Superstore/Product?:embed=y&Region=East" width="800" height="600"></iframe>

<iframe src="http://myserver/views/Superstore/Product?:embed=yes&Region=East,West" width="800px" height="600px"></iframe>

For more information, see Filter on Multiple Fields on page 516.

Filter on Multiple Fields

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

Script Tag Example

<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>

<object class='tableauViz' width='800' height='600' style='display:none;'
  <param name='host_url' value='http://myserver/' />
  <param name='site_root' value='' />
  <param name='name' value='Superstore/Product' />
  <param name='filter' value='Region=Central,South&Customer_Segment=Consumer,Home Office' />
</object>

Iframe Tag Example

<iframe src="
The first box below shows an example of the URL you might get when you click **Share** on a view and copy the link in the **Link** field.

The second box shows how you might modify the URL and add it to an Iframe by deleting the `showShareOptions` and `display_count` parameters, adding filter parameters for **Region** and **Segment**, and adding width and height parameters, to create an embed link that displays only **Consumer** and **Home Office** products from the Central and South regions.

**Note:** If a filter value contains a special character, such as a comma, replace the character with the URL encoding sequence for \ (backslash, `%5C`) followed by the URL encoding sequence for the special character. The backslash is needed to escape the special character. For example, the URL encoding sequence for \, (backslash, comma)
Filter Dates and Times

If you want to filter on a Date/Time field, include the value using the default Tableau format shown below:

`yyyy-mm-dd hh:mm:ss`

The time part uses a 24-hour clock. Many databases store all date values as Datetime fields, so you may need to pass a time value along with your date.

Script Tag Example

```html
<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='host_url' value='http://myserver/'/>
  <param name='site_root' value=''/>
  <param name='name' value='Sales/Sales-Performance'/>
  <param name='filter' value='Date=2012-12-01'/>
</object>
```

This example filters on both a date field and a datetime field:

```html
<param name='filter' value='2012-12-01%2022:18:00'/>
```

Iframe Tag Example

```html
<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.

Filter Measures

You can filter measures by including one or more values. There is no support for greater than, less than, or ranges. The example below filters to show only $100 and $200 sales.

Script Tag Example

```html
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600'
```
Iframe Tag Example

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

Control the Load Order of Multiple Views

You can control the order in which multiple views load for the people working with your views. This feature can only be accessed using embed code that relies on the Tableau JavaScript file.

In the following example, two views are embedded. The second view loads first, followed by the top view. If you embed multiple views and give them all the same load order value, or if you don't specify load order parameters, they are loaded in the order in which they appear on the page.

Script Tag Example

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

You can embed a Tableau dashboard in your own web site or wiki page and you can have it display in different layouts based on screen size.

- To automatically make the right layout appear for your users, regardless of what device they are using, create device-specific layouts for your dashboard in Tableau Desktop. Tableau Server automatically displays the correct layout based on screen size.
- To always make a certain layout appear, regardless of screen size, use the `device` parameter in your embed code.

In the following example, the embed code displays a dashboard. If the dashboard uses device-specific layouts, the correct layout displays, that is, no additional parameters are needed:

```html
<param name='name' value='MyCoSales/SalesScoreCard' />
<param name='tabs' value='yes' />
<param name='toolbar' value='yes' />
<param name='load-order' value='1' />
</object>

The following example assumes that the embedded dashboard has device-specific layouts. The `device` parameter is set to `phone`. This means that, no matter which device displays the dashboard, the layout created for phones will be the one that's displayed.

```html
<param name='host_url' value='http://mysite.myserver.com' />
<param name='site_root' value='' />
<param name='name' value='ProfitAnalysis/Sales_Dashboard' />
<param name='tabs' value='yes' />
<param name='toolbar' value='yes' />
<param name='filter' value=':original_view=yes' />
</object>
```
Screen Dimensions and Device Layouts

If a dashboard used device-specific layouts, Tableau uses the largest dimension of the device screen, whether it's the height or width, to determine which one to load.

If the largest dimension is... This dashboard device layout is displayed...

500 px or less Phone

501–800 px Tablet

801+ px Desktop/Laptop

For details on how to create a dashboard that uses device-specific layouts, see Create Dashboards in the Tableau Desktop help.

Embed Code for Custom Views

When you embed a view of a workbook or sheet that has custom views available:

- If the embed code URL for the view explicitly refers to a custom view, that custom view will be displayed by default.
- If the embed code URL does not explicitly refer to a custom view, and a Default custom view has been defined, the Default custom view will be displayed in the embedded view by default.
- If no Default custom view has been defined, the original view will be displayed in the embedded view by default.

**Note:** To ensure the original view will be displayed by default in an embedded view, make sure the embed code URL for the name parameter does not explicitly refer to a custom view, and include the following `filter` parameter in the embed code: `<param name='filter' value=':original_view=yes'/>`.

In the following example, the embed code will always display the original view of the Profit Analysis sheet in the Profit Analysis workbook, because the `filter` parameter is set to `:original=yes`, and the `name` parameter does not refer to a specific custom view in the URL for the sheet.

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
```
In this example, the setting for the name parameter in this example specifically refers to the URL for a custom view named Furniture (in the Profit Analysis sheet in the Profit Analysis workbook).

In this example, the name parameter does not refer to a specific custom view in the URL for the sheet, and the original_view parameter has not been specified. The embed code here will display the custom view that has been set to Default in the Profit Analysis sheet in the Profit Analysis workbook. However, if the original view is still the Default (no other custom view has been set to Default), then the original view will be displayed as the default view.
Embed Views into Wikis

You can easily embed a view into a wiki or other web page simply by putting the view inside an `<iframe>` tag.

1. Navigate to the wiki page you want to embed a view into.
2. Edit the page and add an `<iframe>` where the source is the URL from the Email box of the Share View dialog box. For example:

   `<iframe src="http://myserver/views/Date-Time/DateCalc-s?;embed=yes&;toolbar=no" width="800" height="600"></iframe>`

3. Save your changes.

   The view is embedded into the wiki page.

   If both Tableau Server and the wiki are configured to use Microsoft SSPI, users accessing an embedded view on the wiki will be automatically signed in so they can see the view.

   If the server and the wiki are not using the same method for authentication, users will first be asked to sign in to the server before they can see the view.

Embed Images

In addition to embedding a view into a `<script>` or `<iframe>` tag you can also embed the view as an image. When you embed an image the view is not interactive, however, it is updated
every time the page fully reloads. That way the image shows the latest data even if the underlying data changes.

1. Navigate to the page where you want to embed the image.
2. Edit the page and add an `<img>` tag where the source is the URL from the Email box of the Share View dialog box for the view, plus the .png file extension. For example:

   `<img src="http://tableauserver/views/Date-Time/DateCalcs.png" width="900" height="700">`

   **Note:** This approach will only work if the user accessing the embedded image also has an active web browser session with Tableau Server and is signed in to Tableau Server using Microsoft SSPI.

**Embed Views into SharePoint (Microsoft SSPI)**

You can embed a Tableau Server view in a SharePoint page. To automatically authenticate Tableau Server users who access the embedded view you have two choices, both of which depend on which user authentication method was selected during Tableau Server Setup. You can use either Active Directory with Enable automatic logon to authenticate Tableau Server users (also known as using Microsoft SSPI), or you can use Local Authentication—and then also configure Tableau Server for trusted authentication.

This topic applies to the first option, where both Tableau Server and SharePoint are using Microsoft SSPI. If your Tableau Server is using Local Authentication, see **Embed Views into SharePoint (Local Authentication)** on page 527 for steps.

**Requirements**

**Licensed users:** Anyone who accesses an embedded view must be a licensed user on Tableau Server.

**SharePoint version:** Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server’s security requirements.

**TableauEmbeddedView web part:** You must have a TableauEmbeddedView web part deployed to your SharePoint server before you can embed Tableau views in a SharePoint page. For sample SharePoint code and instructions for how to create a web part and deploy it to your SharePoint server, see C:\Program Files\Tableau\Tableau Server\<version>\extras\embedding\sharepoint. **Note:** The sample SharePoint code is provided as an example, and may require modification to work in your SharePoint deployment.

**Embedding a View into SharePoint**

You can embed the Tableau web part in a new or existing SharePoint page.

- 306 -
1. Open the page where you want to embed a view and switch to edit mode.

2. In the section of the page where you want to embed the view, on the Insert tab, click Web Part.

3. Under Categories, in the Custom (or Miscellaneous) folder, select TableauEmbeddedView, and then click Add in the lower-right corner.

4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select Edit Web Part.
5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.

- In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
- In **View Path**, enter the path to the view you want to embed.
- Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.
- In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed
height (for example, 700 Pixels) and adjust the width to fit the zone.

6. Click OK to apply the changes and exit edit mode.

The view will be embedded into the web part that you just created. Your users will not need to log in to Tableau Server to see the embedded view, rather they will be automatically authenticated using Microsoft SSPI.

**Embed Views into SharePoint (Local Authentication)**

You can embed a Tableau Server view in a SharePoint page. If Tableau Server is using Local Authentication for user authentication, there are some extra steps you need to take before you start embedding views.
This topic describes how to complete the following steps:

- Edit the security permissions for the TableauEmbeddedView.dll file.
- Install and deploy the TableauEmbeddedView.wsp file.
- Verify the web part's deployment.
- Embed a view in SharePoint using the Tableau web part.

**Note:** If your Tableau Server installation is using Active Directory for user authentication, you can start embedding views right away. For more information, see Embed Views into SharePoint (Microsoft SSPI) on page 524.

**Requirements**

**Users:** To access an embedded view, users must be licensed Tableau Server users and their user name on SharePoint must be the same as their user name on Tableau Server.

**SharePoint version:** Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server’s security requirements.

**Edit Security Permissions for TableauEmbeddedView.dll**

Edit the security permissions for TableauEmbeddedView.dll so that all users of the operating system can use it.

1. Locate the TableauEmbeddedView.dll and TableauEmbeddedView.wsp files that install with Tableau Server. If Tableau Server is installed on drive C, the files will be in the following directory:
   
   C:\Program Files\Tableau\Tableau Server\10.0\extras\embedding\sharepoint\ 

2. Copy the files to the root directory of your SharePoint server. The root directory is usually located at C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin, for example:
   
   C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin

3. To edit the security permissions on TableauEmbeddedView.dll, right-click TableauEmbedded.dll and then select Properties > Security.

4. Under Group or user names, select Everyone, and then click Edit.
5. Under **Permissions for Everyone**, for the **Full control** permission, select **Allow**.

6. Click **OK**.
Install and Deploy TableauEmbeddedView.wsp

The TableauEmbeddedView.wsp file gives SharePoint more information about what to do with the .dll file. You copied the TableauEmbeddedView.wsp file to the SharePoint root directory in the previous procedure. To install and deploy the .wsp file, follow these steps:

1. Open SharePoint 2013 Management Shell and enter the following command:
   ```
   Add-SPSolution -LiteralPath "C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin\TableauEmbeddedView.wsp"
   ```
2. On the SharePoint Central Administration home page, click System Settings.
3. In the Farm Management section, click Manage farm solutions.
4. On the Solution Management page, click the solution that you want to deploy.
5. On the Solution Properties page, click Deploy Solution.
6. On the Deploy Solution page, in the Deploy When section, select one of the following options:
   - Now
   - At a specified time. Specify a time by using the date and time boxes.
7. In the Deploy To? section, in the A specific web application list, click All web applications or select a specific Web application, and then click OK.
8. Open your SharePoint site. Click the settings icon, and then select Site settings.
9. Under Site Collection Administration, click Site collection features.
10. Scroll to the TableauEmbeddedView feature and then click Activate to activate the feature.

Verify the Web Part's Deployment

In the following procedure, you will verify that the Tableau web part is installed.
1. Open your SharePoint site in a web browser.
   It may take a few moments for the site to appear.

2. Click the settings icon, and then select Site settings.


4. Confirm that TableauEmbeddedView.webpart is listed.

**Embed a View Using the Tableau Web Part**

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.

2. In the section of the page where you want to embed the view, on the Insert tab, click Web Part.

3. Under Categories, in the Custom (or Miscellaneous) folder, select TableauEmbeddedView, and then click Add in the lower-right corner.
4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select Edit Web Part.

5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
   - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
   - In **View Path**, enter the path to the view you want to embed.
• Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.

• In the Appearance section you can specify a Title for the web part, the Height, Width, Chrome State, and Chrome Type. In general you should specify a fixed height (for example, 700 Pixels) and adjust the width to fit the zone.

6. Click OK to apply the changes and exit edit mode.

Now the view is embedded in the page and users who access it will be automatically signed in based on their user name and password for SharePoint.
This is an example of embedding views into SharePoint using the provided .dll file. You can also embed views into other types of web application. For more information, see JavaScript API on the Tableau Developer Portal.
What is a Site?

You might be used to using the term site to mean "a collection of connected computers," or perhaps as the short form of "website." But in Tableau-speak, we use site to mean a collection of content (workbooks, data sources, users, etc) that's walled off from any other content on that instance of Tableau Server. (Another way to say this is that Tableau Server supports multitenancy by allowing server administrators to create multiple sites on the server for different sets of users and content.)

Each site has its own URL and its own set of users, and each site has completely segregated content, projects, and data sources. You can set permissions per user or group on a project, workbook, view, or data source. All server content is published, accessed, managed, and controlled on a per-site basis.

What is a site administrator?

A site administrator is in charge of creating and maintaining the framework on Tableau Server that enables Tableau Desktop users in the organization to publish, share, manage, and connect to data sources and workbooks. Their duties can include creating and managing users and groups, creating projects to organize content on the site, assigning permissions to allow users to access the content they need, scheduling extract refreshes, and a few other tasks.

Site administrators and server administrators

In addition to a site administrator, there's also a server administrator. The server administrator sets up Tableau Server—they install and upgrade it, configure the services that run on Tableau Server, back it up, and perform other tasks that pertain to running Tableau Server as a whole. Server administrators also create sites as needed. (Site administrators don't have permissions to create sites.)

In some organizations, the same person might be both a server administrator and the site administrator for one or more sites. Even so, the tasks performed by a site administrator and a server administrator are distinct.

About this guide

This guide tells you, a site administrator, how to plan, create, and manage sites on an instance of Tableau Server. Note the following:

- We don't cover the duties of a server administrator. We have a separate guide that covers those tasks.
- We don't discuss how to publish content to the server. Users do this from Tableau
Desktop. However, we do discuss how to set up users on the site and give them permissions to publish and view the content that they need. For information about how to publish to Tableau Server, see *Publish Data Sources and Workbooks* in the Tableau Desktop documentation.

**Navigate Site Admin Pages**

When you sign in to Tableau Server as a site administrator, you see additional menus for site and content management that are not available to other users. If your server is configured for multiple sites, a site menu is available for navigation.

A site administrator can:

- Manage content and assign permissions.
- View and manually run schedules set up by the server administrator for extract refreshes and subscriptions.
- Manage extract refreshes and subscriptions.
- Add and manage site users (if allowed by the server administrator; see *Add or Edit Sites* on page 740).
- Add and manage site groups.
- Monitor site activity.

For information on navigating content pages in Tableau Server, see *Navigate Tableau Server* on page 27 and *Access and Manage Your Content* on page 72.

**Site Administrator Pages**

Site administrators manage all site content and access on a per-site basis. If you have access to multiple sites on the server, all of the sites that you manage will be available from the site menu. If you manage a single site on the server, no site selection is necessary, all site-related menus will be available.

On a single-site server, these are the menus a site administrator sees:

On a multi-site server, with a site selected, these are the menus a site administrator sees:
The site menu displays the name of currently selected site. To go to another site, click the site menu, and then select the site name.

Planning a Site

Before you add users and content to a site, we recommend that you plan the following aspects of a site. Details about each of these aspects of site administration are provided in this guide.

- Projects
- Users and groups
- Site roles and permissions
- Extract refresh schedules
- Steps for setting up your site

Projects

You can create projects on a site, which lets you organize related content. For example, you might set up a project to contain all the data sources and workbooks for a project that a group of your colleagues are working on together. Or you might set up different projects for different departments.

Projects are also useful because you can set up different permissions for each project. If you know what projects you'll need and who needs access to the content in those projects, it's usually easier to set up permissions before users publish content.

Every site has a default project named Default. As we explain later, we recommend that you do not use the Default project for content. Instead, use it to set up default permissions; when you create projects, the new projects get their initial set of permissions from the default project. In effect, the default project is a template for new projects.
Users and groups

Obviously, it's important to know who needs to access content on your site. Any user who will publish to the site must be able to sign in. If the user already has an account on the server, you'll need to add that user to the site. If the user doesn't already exist, you'll need to create a user account. Either way, make a list of the users who will need to be able to sign in to your site. (Users can belong to more than one site.)

Note: The server license might restrict how many users you can have. Tableau Server licenses are based on either cores or users. If the server has a user-based license, there's an upper limit to how many users can have active accounts on the server. Check with the server administrator to make sure that you'll be able to have an account for all your users.

In general, we recommend that you create groups on the server and then assign users to the groups. This makes it much easier to manage permissions, since you can assign permissions to a group, and all the users in that group automatically get those permissions. (See the next section.) It's typical to create groups for users who use content in similar ways. For example, you might create a group named SalesWBPublishers for all the users in the Sales department who publish workbooks, and a separate group named SalesDSPublishers for people in the Sales department who publish data sources. (These groups need different permissions, so it makes sense to have different groups for these functions.)

Site roles and permissions

Each user has a site role that determines the maximum permissions that they can have on the site. For example, if you have the role of Site Administrator, you have full rights to work on the site. A user whose site role is Publisher can publish to the site, whereas a user whose site role is Interactor can interact with content (for example, change filter settings in a view), but can't publish. A user whose site role is Viewer can view content, but can't change settings in the content and can't publish.

As part of your site planning, decide what site role each user will have. (You can change a user's site role later if you need to.) A user with a site role that's too restrictive might not be able to do the work they need on your site. But by the same token, it's a security best practice to restrict users' permissions to only what they need in order to do their work (that is, to follow the principle of least privilege).

You must also determine what permissions a user needs in order to able to work with content. Each piece of content on the site (each workbook, data source, and project) supports certain capabilities. For example, a workbook has capabilities like View, Save, Filter, Web Edit, Add Comments, and Download, among others. Before a user can use a workbook—view it, save it, download it, add comments to it, and so on—that user must have permission for the specific capability. Therefore, you should map out what permissions users will need in order to be able work with content.
As we just noted, site roles act as an upper limit on permissions. It's actually the *combination* of site role and permissions that determines what a user can do. A user whose site role is **Interactor** can never publish to the site, no matter what permissions you grant that users. But a user whose site role is **Publisher** can publish a workbook to the site only if that user has permission to save and view workbooks.

To make it easier to manage permissions, create groups and assign the permissions to those groups. You can then add users to the groups that have the permissions that those users need. (Site administrators automatically have permissions for all the capabilities of all content, so they don't need to be explicitly assigned any permissions.)

If you are new to using permissions in Tableau Server, see Projects and Content Permissions in the *Everybody's Install Guide* for a walkthrough that uses a best practice approach to setting up permissions.

**Extract refresh schedules**

If users publish data sources or workbooks that include extracts, you usually want to make sure that the extracts are refreshed so that they contain the latest data. Users can manually refresh an extract, but this isn't always a good idea if the extract is large and the refresh takes a long time. Instead, you can set up schedules for when an extract should be refreshed. Another planning task for a site administrator is therefore to think about when extracts should be refreshed and to work out schedules.

**Steps for setting up your site**

The table below shows a loose sequence of steps for setting up a site. You can complete the steps in any order that makes sense for you. At the bottom of this topic you'll find a list of links to more resources for each of the steps.

Before you configure the site, we recommend spending some time learning about site authentication, site roles, projects, and permissions. Create and document a plan for your projects, groups, and overall permissions strategy. Setting up a test project to experiment with different settings is a good way to iron out these issues. You can change many site settings after your users are working with the site, but try to go in with the intention of minimizing post-production changes.

<table>
<thead>
<tr>
<th>Configure site access</th>
</tr>
</thead>
<tbody>
<tr>
<td>If your organization uses single sign-on, you can configure your site to use SAML authentication. Otherwise, you can use the default Tableau ID authentication, where each user signs in using a user name and password.</td>
</tr>
<tr>
<td>Talk to your server administrator about whether single sign-on is something you want for your site.</td>
</tr>
</tbody>
</table>
Projects

As an administrator, you can create projects to collect and organize related content. Content in Tableau Server refers to workbooks, views, and data sources, and the projects that contain them.

You access projects from the Content page in Tableau Server.

As an administrator, you can do the following for projects:
• Create projects.
• Rename projects.
• Change project owners.
• Set permissions for projects and default permissions for their content.
• Lock content permissions.

**Note**: Only administrators can create and own projects.

**Project Leader**

Users who have the **Project Leader** permission in a project can:

• Control who has access to project content by setting default permissions for project content at any time, even when content permissions are locked to the project.
• Lock content permissions to the project.
• Move workbooks between projects.
• Change the name of a project.
• Change owners of workbooks or data sources in the project.
• Run, add, or remove extract refreshes for workbooks and data sources in the project.

**Note**: Users with a site role of Interactor or Publisher get full project leader permissions. A project leader does not have to be the project owner.

**Default project**

Tableau creates every site with a **Default** project. The Default project serves as a template for new projects in that site. It defines the default settings and permissions that are applied to new projects and to the workbooks and data sources within those projects.

When you create a new project, the new project uses a copy of the Default project permissions.
Default permissions

As an administrator or project leader, you can set permissions for every project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project.

Each project can have its own set of default permissions.

For more information, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Only administrators and Project Leaders can edit the default permissions for a project and its workbooks and data sources.

For information on using projects to control permissions for content, see Create Project-Based Permissions on page 881.
Project content permissions

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.

When permissions are locked to the project, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are managed by the owner ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

**Note:** If a workbook or data source with editable permissions is moved to a locked project, the default permissions in the locked project are applied to the moved content and its permissions will then be locked.

For more information, see Lock Content Permissions to the Project on page 869.
Add Projects

1. Click Content > Projects, and then click New Project.

   ![New Project Dialog]

2. Enter a name and description for the project, and then click New Project.

   You also can include formatting and hyperlinks in the project description. Click Show formatting hints for syntax. For information on adding a image for the project, see Add a Project Image on page 761.

   To edit a project, click the Project to open it, click Details, and then click Edit Description.

Move Workbooks into Projects

All workbooks must be in a project. By default, workbooks are added to the Default project. After you create your own projects, you can move workbooks from one project to another. You can move workbooks into projects if you are an administrator, or if you have the site role of Publisher or Interactor and at least one of the following is true:
You have been given the Move permission for the project.

You have been given Project Leader permission for the project.

To move a workbook into a project

1. Click Content > Workbooks. In the Workbooks page, select one or more workbooks, and then select Actions > Move.
2. Select a different project for the workbook, and then click **Move**.

Because all workbooks must be part of a project, you can remove a workbook from a project by moving it to the Default project. Each workbook can only be contained in a single project.

**Add a Project Image**

Projects can have images that are displayed in thumbnail view in Tableau Server.
To set a project image:

1. In a site, click Projects, and then open a project.
2. Click Details, and then click Edit Description.
3. Add the URL for your image in the About field. Click **Show formatting hints** for syntax examples that show how you can format the description text.

Type the URL using this syntax:

```
!http://www.example.com/image.png!
```
Set Default Permissions for a Project, and its Workbooks and Data Sources

As an administrator or project leader, you can set a project's permissions and the default permissions for its workbooks and data sources.

4. Click Save.
Each project can have its own set of default permissions. The permissions that you set are the default permissions for all content in the project, including content that is being published to the project from Tableau Desktop.

**Note:** New projects are always created with the default permissions set for the Default project.

For additional information on working with permissions, see Manage Permissions on page 834 and Projects on page 756. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

**Notes on default permissions in locked projects**

You can choose to have the default permissions apply to all workbooks and data sources in a project, and ensure that no one can change those settings, by locking content permissions to the project. For more information, see Lock Content Permissions to the Project on page 869.

- Workbooks and data sources in a locked project always use the default permissions set for content in that project. Views in a locked project always use the workbook permissions. This applies to workbooks and data sources when they are being published from desktop.
- Administrators and users with the Project Leader permission can always edit default permissions, even when a project is locked.
- Users, including content owners, cannot edit individual workbook, view, and data source permissions when content is locked to the project.

**To set default permissions in a project**

1. In the Content page of a site, click a project, and then click Permissions in the project place page.
2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

![Add a user or group rule](image)

For an existing user or group, click the actions menu (…), and then click **Edit**.

![Edit user or group](image)

3. Select a permission role template for **Project**, **Workbooks**, or **Data Sources**, and then click **Save**.
Or, to create a custom set of capabilities, click the **Project**, **Workbooks**, or **Data Sources** labels to expand the permissions view. Click capabilities to set them to **Allowed**, **Denied**, or **Unspecified**. Click **Save**.

This example shows how to set project permissions. The same general steps apply for workbooks and data sources.

**Note:** To change the settings after saving, click the actions menu (…), and then click **Edit**.

4. View the user permissions, which are the effective permissions.

Click a group name or user name in the permission rules to see the resulting user permissions.
Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.

Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
5. Follow the same steps to configure additional permission rules for more users or groups.

**Set Permissions for a Project**

Every project includes permissions that can be set for the project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project, and each project can have its own set of default permissions. For more information, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Administrators and users with the Project Leader permission can lock content permissions to a project. For more information, see Quick Start: Lock Project Permissions, Lock Content Permissions to the Project on page 869.

For more information on working with permissions, see Manage Permissions on page 834 and Projects on page 756.

**Note:** When you create a new project, it initially will have the same permissions as the Default project in the site, which are the default permissions for the project, and its workbooks and data sources.
The three capabilities you can set specifically for a project are: **View**, **Save**, and **Project Leader**.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View</strong></td>
<td>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</td>
</tr>
<tr>
<td><strong>Save</strong></td>
<td>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project.</td>
</tr>
<tr>
<td></td>
<td>When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining all permissions.</td>
</tr>
<tr>
<td></td>
<td>Subsequently, the original owner’s access to the workbook is determined by that user's group permissions and any further permissions the new owner might set.</td>
</tr>
<tr>
<td></td>
<td>This permission also determines the user’s or group’s ability to overwrite a workbook after editing it on the server. For related information, see <a href="#">Grant Web Edit, Save, and Download Permissions</a> on page 878.</td>
</tr>
<tr>
<td><strong>Project Leader</strong></td>
<td>Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions.</td>
</tr>
</tbody>
</table>
To set permissions for the project

1. On the Projects page, select a project, and then select **Actions > Permissions**.

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.
The available permission role templates for projects are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>Allows the user or group to view the workbooks and views in the project.</td>
</tr>
<tr>
<td>Publisher</td>
<td>Allows the user or group to publish workbooks and data sources to the server.</td>
</tr>
<tr>
<td>Project Leader</td>
<td>Allows the user or group to set permissions for all items in a project.</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to <strong>Denied</strong>.</td>
</tr>
<tr>
<td>Data Source Con-</td>
<td>Allows the user or group to connect to data sources in the project.</td>
</tr>
<tr>
<td>nector</td>
<td></td>
</tr>
<tr>
<td>Data Source Editor</td>
<td>Allows the user or group to connect to, edit, download, delete, and set permissions for a data source in the projects. They can also publish data sources, and as long as they are the owner of a data source they publish, can update connection information and extract refresh schedules. This permission is relevant for views when the view they access connects to a data source.</td>
</tr>
</tbody>
</table>

4. To further customize the rule, click the actions menu (…) next to the permission rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
5. **View the resulting permissions.**

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
6. Follow the same steps to configure additional permission rules on the content for more users or groups.

Quick Start: Lock Content Permissions to a Project

As an administrator or project leader, you can lock content permissions in a project to prevent users from changing the permissions of any content in the project. When permissions are locked to the project, the default permissions are applied to all workbooks and data sources in a project and cannot be modified by users (including the content owners).

**Note:** Content owners always get full access to the content they've published, but cannot change permissions for their workbooks and data sources when the parent project permissions are locked.

For related information on setting permissions, see Manage permissions. For more information on setting default permissions and locking content permissions to the project, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861 and Lock Content Permissions to the Project on page 869. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.
1 Set Default Permissions for the Project

Because the content inside locked projects always uses the default permissions, first verify that your default permissions are set appropriately. In a site, click Content > Projects. Open a project, and then click Permissions. Add a user or group and select a permission role template for that content type, or click Edit, and then set capabilities to Allowed, Denied, or Unspecified.

Administrators and Project Leaders can edit default permissions at any time.

2 Lock Content Permissions to the Project

In a project’s permissions, click the Managed by the owner button. The button label indicates whether content permissions are currently locked to the project or managed by the content owner. Select Locked to the project, and then click Save.
When permissions are locked to the project, all content in the project uses the default permissions. No users can change permissions for individual workbooks (including views) or data sources in the project.

3 View Locked Permissions

Open a project, select a workbook or data source in the project, and then click Actions > Permissions. When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

In this example, the workbook owner has full permissions for the workbook, but cannot change the workbook permissions while they are locked to the project.
Unlock Content Permissions for the Project

In a site, click Content > Projects. Select a project, and then click Actions > Permissions. Click the Locked to the project button. Select Managed by the owner, and then click Save.

When a project's content permissions are Managed by the owner, individual workbooks, views, and data sources in the project start with the default permissions and can be modified by users.

Notes on project permissions:

- Only administrators and project leaders can lock content permissions, and set and edit default permissions in a project.
- Administrators and project leaders can edit default permissions for the project, its workbooks, and its data sources at any time, at the project level.
- Individual workbook, view, and data source permissions cannot be edited by users (including content owners) when a project is locked.
- Workbooks and data sources in a locked project always use the default permissions. Views in a locked project always use the workbook permissions.

Lock Content Permissions to the Project

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.
When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

**Note:** Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For information on default permissions, see *Set Default Permissions for a Project, and its Workbooks and Data Sources* on page 861. For a best-practice walkthrough on how to implement permissions, see *Projects and Content Permissions*.

**Note:** Administrators and project leaders can set and edit default permissions for the project, and its workbooks and data sources when it is locked.

1. In the Content page of a site, open a project, and then click **Permissions** in the project place page.

2. Click the **Managed by the owner** button.

   The padlock icon on the button label indicates whether content permissions are currently locked to the project or managed by the content owner.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.
When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

4. To unlock content permissions for the projects, open the project permissions again. Click the **Locked to the project** button. In the **Content Permissions in Project** dialog box, select **Managed by the owner**, and then click **Save**.
The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

Create Project-Based Permissions

As an administrator, you can organize a collection of related workbooks and data sources in a project. You can then control access to that content by creating permission rules for groups of users who need similar access levels to publish or interact with that content.

Note: For this scenario, you set the permission rule for the All Users group for the project to None, which means that permissions are Unspecified for the All Users group.

Preparation

Before you begin creating projects and project-based permissions, document the projects and permission levels that you want users to have in each project.

This roadmap exercise helps you organize permissions to be most efficient to manage over time, and can help you identify any user or permission gaps in your solution. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Also read the following topics in the Tableau Server Help:

- Manage Permissions on page 834 and permissions-related topics
- Projects on page 756 and projects-related topics
- Grant Web Edit, Save, and Download Permissions on page 878

Step 1: Create projects and user groups

1. Sign in to Tableau Server with your administrator user name and password.

2. On the Projects page, click New Project.

3. Click Groups, and then click New Group.

   Create groups that correspond to each project and access level. For example, for a project that allows users only to access the views, you might use a name similar to Project1Viewer. For a project that allows interaction with the views, Project1Interactor.

4. Click Users, and then click Add Users. Select one or more users in the list, select Actions > Group Membership, and then select a group for the users. Click Save to confirm the group membership.

   Repeat this step to add users to other groups.

Step 2: Assign permissions at the project level

After you set up your projects and user groups, you can start assigning permissions. Repeat these steps for each project. Also see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.
1. On the Projects page, select a project, and then select **Actions > Permissions**.

2. For the **All Users** group permission rule, set the permission role template to **None**. Click the actions menu (…) next to **All Users**, and then click **Edit**. Select **None** for **Project, Workbooks**, and **Data Sources**, and then click **Delete**. This means that all capabilities will be set to **Unspecified**.

3. Click **Add a user or group rule**, select **Group**, and then select the group name in the list.

   To edit an existing rule, click the actions menu (…) next to the permission rule name, and then click **Edit**.

4. Select a permission role template for **Project, Workbooks**, and **Data Sources** to specify a predefined set of capabilities for the group or user.

5. To further change capabilities included in the rule, click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**.

   Click **Save** when you are done.

   Repeat steps 3-5 for each group or user requiring project permissions.

   **Note**: You can optionally lock content permissions to the project to enforce the default permissions for all content in the project. This overwrites any previous permissions assigned to workbooks and views in the project. For more information, see **Lock Content Permissions to the Project** on page 869.

**Step 3: Check project permissions**

- View the resulting user permissions.

  Click a group name or user name in the permission rules list to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

When you publish workbooks to the project, the permissions are updated accordingly.

For information on granting Save permissions to users, see **Grant Web Edit, Save, and Download Permissions** on page 878.

**Delete Projects**

Only administrators can delete projects. When you delete a project, all of the workbooks and views that are part of the project are also deleted from the server.

1. Click **Content > Projects**. In the Projects page, select a project, and then select **Actions > Delete**.
2. Click **Delete** in the confirmation dialog box.

The **Default** project cannot be deleted.

**Control Access to Published Content**

Administrators can control access to Tableau Server content by assigning permissions to projects, workbooks, views, and data sources. They also can specify and change owners for projects, workbooks, and data sources.

Content owners have control over the permissions for the content that they publish to the server.
Manage Permissions

In Tableau Server, you set content permissions in order to specify who is allowed to work with what content in a site.

About content permissions

Content permissions ensure that only the right people can see and interact with your content. For example, you can tightly restrict who has access to your company’s financial information, but widely share organizational development content.

You assign content permissions to the following items:

- Projects
- Workbooks
- Views
- Data sources

About permission rules, site roles, and user permissions

You assign content permissions by setting permission rules. Permissions rules are the explicit capabilities you assign to a user or group for a given content item. A capability is a task that you want a user to be able to perform, such as editing a view. Every project, workbook, view, or data source can have a unique set of permission rules.

In addition to content permissions, a user’s site role and whether the user is a content owner also affects what tasks a user can perform and what actions are available to the user for each content type.

User permissions are the effective permissions that determine what a user can actually do with the content. They are the result of how Tableau evaluates each user or group permission rule that applies to a user for a given content item.

For more information, see Site Roles for Users on page 788 and How Permissions are Evaluated on page 839. Also see Projects and Content Permissions for a walkthrough that uses a best practice approach to setting up permissions.
Example: A permission rule set for the "Finance" group.

Who can set permissions

User who have the **Set Permissions** capability can change permissions for content items in projects that aren't locked. Administrators, content owners, and users with the **Project Leader** capability automatically receive the **Set Permissions** capability.

**Note** Project Leader is a permissions capability that you can set for a user or group at the project level.

Default permissions and projects

The permissions assigned to content when it is published or created on the server are the item's **default permissions**. Default permissions are set only at the project level, and only by administrators and users with the Project Leader capability.

- New projects get a copy of content permissions from the **Default** project in the site. These permissions include the permissions for the project, and the default permissions for its workbooks and data sources.
- New workbooks and data sources use the default permissions from their project. When content permissions are not locked, the individual workbook and data source permissions can be edited to differ from the defaults.
- New views use the default permissions from their workbook. When content permissions
are not locked and the views aren’t shown as tabs in the workbook, the individual view permissions can be edited to differ from the defaults. Note that tabbed views always use their workbook permissions.

When the content permissions are locked to the project, workbooks and data sources in the project will always use the default permissions. Views in the workbooks will always use their workbook permissions. The default permissions can only be changed at the project level.

If you are new to using permissions in Tableau Server, see Projects and Content Permissions for a walkthrough that uses a best practice approach to setting up permissions.

For more information on the Default project, see Projects on page 756.

For more information on default permissions, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

For more information on locking content permissions, see Lock Content Permissions to the Project on page 869.

Permission Rules and User Permissions

When you specify permissions for a project, workbook, view, or data source, you specify who is allowed to work with that resource through a permission rule. Permission rules are the explicit capabilities that can be set for an individual user, or for a group—for each resource.

The Permissions window has two sections: Permission Rules (upper section) and User Permissions (lower section). You set permissions in Permission Rules, and you view the effective or resulting permissions in User Permissions.
Permission Rules

The permission rules you set up include the user or group and the set of capabilities you want users to have for on that content item (such as the ability to edit a view). Available capabilities vary depending on the type of content selected, and can be set to Allowed, Denied, or Unspecified.

For information about setting and viewing permissions, see Quick Start: Permissions on page 842, Edit Permission Rules on page 872, and View Permission Rules and User Permissions on page 873.

Click the ellipses next to the permission rule name.

Select a permission role template and edit capabilities (the actions allowed on the content).

- **User / Group**: Lists users or groups of users the rule applies to.

- **Permissions**: Lists available permission role templates for a specific project, workbook, view, or data source. Each permission role template (such as Editor, Interactor, Viewer) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to Custom. For more information about permission role templates and capabilities, see Set Permissions for Workbooks and Views on page 846, Set Permissions for a
Project on page 856, and Set Permissions for a Data Source on page 851.

- View / Interact / Edit: Categories for the sets of capabilities that can be set to Allowed, Denied, or Unspecified. (Unspecified results in Denied if no other permissions are specified for a user or group on the content.)

User Permissions

The User Permissions area of the Permissions window shows the effective permissions for each user. These are the actual permissions for each user, after the user’s site role and permission rules have been evaluated.

To view the user permissions for a group or user, click a user or group name in the permission rules list. The effective permissions for users in the group are displayed in the lower half of the Permissions window.

Effective user permissions for a resource are determined by:

- The maximum capabilities allowed for a user’s site role. The site role acts as the "ceiling" for what permissions are allowed. For more information, see Site Roles for Users on page 788.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

For example, if a user is granted Editor-level permissions for a workbook (which allows all available capabilities), but has the site role of Viewer and does not own the workbook, the user will only be allowed the capabilities of View, Export Image, Summary Data, View Comments, Add Comments, and Save.

In the following example, a permission rule has been created for the Finance group. The permission role template of Editor was initially applied to the group, which granted all capabilities. The administrator then set the Save capability to Denied, so the name for the set of permissions applied to the group became Custom. The User Permissions section for the Finance group shows that most of the users in the group have all capabilities, except for the Save capability. One user has even fewer capabilities because that user has a site role of Viewer.
Note that the **All Users** group permission rule in this example has been set to **None**, which leaves all of the permissions as **Unspecified** for the **All Users** group. This approach requires the administrator to specifically assign permissions for only the groups or users that should see the content.

**How Permissions are Evaluated**

Permissions in Tableau Server are assigned to resources, also known as content—projects, workbooks, views, and data sources. You specify who can work with a resource using permission rules.

The views, workbooks, projects, and data sources on Tableau Server that users can access, and the actions available for these different content types, are affected by:

- **Site role.** A user’s site role determines whether a user can publish, interact with, or only view resources and the different levels of permission capabilities allowed for a user. The site role acts as the “ceiling” for what permissions are allowed. For more information, see [Site Roles for Users](#) on page 788.

- **Content permissions.** Every resource, that is, every project, workbook, view, or data source, can have a unique set of permission rules.

  A permission rule includes the user or group, and the set of capabilities you want to grant users for a resource (such as the ability to edit a view). Each permission role template
(such as Editor, Interactor, Viewer) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to Custom.

Available capabilities vary depending on the resource. Capabilities can be set to Allowed, Denied, or Unspecified. Denied always takes precedence over Allowed, and Unspecified results in Denied if no other permission rules allow a capability for a user.

- **Ownership.** Content owners always get full access to the content they’ve published. In projects with locked permissions, content owners cannot edit permissions for their workbooks and data sources.

Users with the Set Permissions capability can change permissions for content items in projects that aren’t locked. Administrators, content owners, and users with the Project Leader capability automatically have the Set Permissions capability.

You can set permission rules for an individual user or group for each resource. This diagram illustrates how permission rules are evaluated in Tableau Server.

Effective user permissions are determined by:

- Maximum permissions allowed for a user’s site role. For more information, see Site Roles for Users on page 788.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

**Notes on permissions**

- Server and site administrators can access all the resources in a site with full permissions.
- You cannot set permissions at the site level; permissions are assigned to resources only.
- Publishers (content owners) always get full access to their content. Content owners can change permissions on their workbooks and data sources, unless the parent project permissions are locked. For more information, see Lock Content Permissions to the Project on page 869.
- Individual user permissions on resources take precedence over group permissions on
resources. In other words, user permissions trump group permissions.

- Workbook permissions serve as templates for view permissions. When content permissions are locked to the project, and when a workbook uses tabbed views, views inherit their workbook permissions. When permissions are not locked, and when a workbook is saved without tabs, the workbook and view permissions can be edited independently.

- Project default permissions serve as templates for content in a project. When content permissions are locked to the project, the workbooks and data sources always use the default permissions. When permissions are not locked, workbook and data source permissions can be edited independently.

- For each content item, every site user is automatically included in the All Users group. As a result, the All Users permission rule affects how permissions are evaluated for users when you create additional group permission rules for that content item.

If you use Tableau Server in an environment where openly sharing knowledge and information across the organization is important, set the permission rule for the All Users group in the Default project to the Publisher permission template. Users can publish to and consume content from new projects.

If you use Tableau Server in an environment where restricting access is important, set the permission rule for the All Users group in the Default project to the role of None. Then, add explicit permissions for groups and users to allow them to publish and work with content in new projects.

**Tableau Server evaluates permissions in the following order of precedence:**

1. **Server and Site Administrator**: Administrators can access all site content with full permissions.

2. **User - Unlicensed, Viewer license, or Guest**: If a user is Unlicensed, has a Viewer license (different than Viewer site role), or is a Guest, there are certain capabilities they are never allowed to perform. If the capability is explicitly denied for the user because of licensing, they are denied.

3. **Project Owner**: If the user owns the project that contains the content, the capability is allowed. Otherwise,

4. **Project Leader**: If the user has the Project Leader capability, or is in a group that has the Project Leader capability, they are allowed. If the user is explicitly denied the Project Leader capability, they are denied. Otherwise,

5. **User - Authorizable Owner**: If the user is the owner of the content, they are allowed. Otherwise,

6. **User - Capability Denied**: If the user has been explicitly denied the capability for the content, they are denied. Otherwise,
7. **User - Capability Allowed**: If the user has been explicitly allowed the capability for the content, they are allowed. Otherwise,

8. **Group - Capability Denied**: If the user belongs to a group that has been explicitly denied the capability for the content, they are denied. Otherwise,

9. **Group - Capability Allowed**: If the user belongs to a group that has been explicitly allowed the capability for the content, they are allowed. Otherwise,

10. The user is denied access to the content.

**Quick Start: Permissions**

You can use permission rules to control access to specific content on a site. Every user has a set of allowed capabilities based on their site role. Each content type—projects, workbooks, views, and data sources—can have permission rules assigned to groups or to specific users. The easiest and most efficient way to manage permissions is to create permission rules for groups.

**1 Add Users to Groups**

Within a site, click **Groups**. Create groups for users who should have the same permissions, and then add the users to these groups. Click a group name, and then click **Add Users** to select the users to be included in the group.
2 Select the Content

On the Content page for a site, click Workbooks, Views, Projects, or Data Sources. Select an item in the page. Select Actions > Permissions to view the permission rules for that content.
A permission rule is a set of capabilities (such as the ability to edit a view) that are allowed or denied to a user or group of users. Available capabilities vary depending on the type of content selected.

3 Create a Permission Rule

Click Add a user or group rule, select Group, enter search text, and then select a name from the list. Select a permission role template to apply an initial set of capabilities for the group. Click a capability to set it to Allowed or Denied, or leave it Unspecified. Click Save when you are done.
Whether a user can set permissions is based on their site role and how their Set Permissions capability is set.

4 View User Permissions

After you save the permission rule for the group, you can view the effective permissions for that content.

Click a group name to see the group’s users and their permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
Custom indicates a user's capabilities have been changed from the initial settings for their site role or content role.

Site roles
A user's site role determines the maximum permissions allowed for that user.

- Server and site administrators can access all site content with full permissions.
- Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For more information, see Site Roles for Users on page 788.

Permissions evaluation

- Denied takes precedence over Allowed.
- Unspecified results in Denied if no other permissions are specified.
- Specific user permissions on content take precedence over group permissions on content. In other words, user permissions trump group permissions.

For more information on working with permissions, see Manage Permissions on page 834, How Permissions are Evaluated on page 839, Permission Rules and User Permissions on page 836, and Projects on page 756. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Set Permissions for Workbooks and Views
As an administrator or user with the Set Permissions capability, you can set permission rules for a workbook or a view.
For more details on working with permissions, see Manage Permissions on page 834 and Projects on page 756.

**Note:** When project content permissions are locked, permissions cannot be changed for individual workbooks and views in the locked project. For more information, see Lock Content Permissions to the Project on page 869 and Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Use permission rules to set these capabilities for workbooks:

Use permission rules to set these capabilities for views:

**To set permissions on a workbook or view**

1. In the Content page of a site, click Workbooks or Views. Select a workbook or view, and then click Permissions to view the current permission rules.
Note: If you select multiple items and some of the items are read-only, you cannot view the permissions. Instead, select one view at a time.

2. Click Add a user or group rule, select Group or User, and then select the group or user name from the list.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click Save.
The list of capabilities and the available permission role templates vary depending on whether you are setting permissions for a workbook or a view. For more information on capability definitions, see Permissions Reference on page 874.

**Note:** For workbooks and views that contain confidential data, it is good practice to set the All Users group permissions to **None** (all permissions **Unspecified**). You can then add other group permission rules to allow access.

The available permission role templates for workbooks and views are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Applies to...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>workbooks</td>
<td>Allows the user or group to view the workbook or view on the server.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Interactor</td>
<td>workbooks</td>
<td>Allows the user or group to view the workbook or view on the server, edit workbook views, apply filters, view underlying data, export images, and export data. All other permissions are inherited from the user's or group's project permissions.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Editor</td>
<td>workbooks</td>
<td>Sets all capabilities for the rule to <strong>Allowed</strong>.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>workbooks</td>
<td>Sets all capabilities for the rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Denied</td>
<td>workbooks</td>
<td>Sets all capabilities for the rule to <strong>Denied</strong>.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
</tbody>
</table>

4. To further customize the rule, click the actions menu (….) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

6. Follow the same steps to configure additional permission rules on the content for more users or groups.

**Note:** Tabbed views are views in a workbook that is published to the server with **Show Sheets as Tabs** enabled. Tabbed views use the workbook permissions instead of the view permissions. When you view the permissions for a tabbed view in a workbook, you see the workbook’s permission rules in the Permissions window, not the view’s permission rules.

To edit tabbed view permissions, you must open the tabbed view’s workbook permissions. The changes that you make to the workbook permissions affect all tabbed views in that workbook. When the workbook is saved again without tabs (or tabs are hidden), the default permissions are again applied to the workbook and views, and view
permissions can then be edited.

Views in a workbook in a project with locked permissions will also use the workbook permissions. For more information, see Lock Content Permissions to the Project on page 869.

Set Permissions for a Data Source

As an administrator or user with the Set Permissions capability, you can change permissions for a data source.

For information on how data source authentication interacts with data source permissions, see How "Embedded password" and "Prompt user" settings affect permissions for published data source connections on page 371.

For more information on permissions in general, see Manage Permissions on page 834 and Projects on page 756.

Note: When project content permissions are locked, permissions cannot be changed for data sources in the locked project. For more information, see Lock Content Permissions to the Project on page 869 and Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Use permission rules to set the following capabilities for a data source:
<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>View the data source on the server.</td>
</tr>
<tr>
<td>Connect</td>
<td>Connect to the data source. The <strong>Connect</strong> permission allows a user to connect to a data source from an editor (in Tableau Desktop or Tableau Server web editing).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If a workbook author embeds credentials in a workbook or view, users who also have the <strong>Web Edit</strong> permission will be able to access to the workbook's data source regardless of their <strong>Connect</strong> permissions.</td>
</tr>
<tr>
<td>Save</td>
<td>Publish data sources to the server and overwrite data sources on the server.</td>
</tr>
<tr>
<td>Download Data Source</td>
<td>Download the data source from the server.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the data source.</td>
</tr>
<tr>
<td>Set Permissions</td>
<td>Specify permissions for the data source.</td>
</tr>
</tbody>
</table>

**To set permissions for a data source**

1. In the Data Sources page, select one or more data sources, and then select **Actions > Permissions**.
2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.
The permission role templates for data sources are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector</td>
<td>Allows the user or group to connect to the data source on the server.</td>
</tr>
<tr>
<td>Editor</td>
<td>Allows the user or group to connect to, download, delete, and set permissions on data sources on the server. They can also publish data sources, and as long as they are the owner of a data source they publish, they can update connection information and extract refresh schedules. (The latter two capabilities are no longer available if an administrator or project leader changes data source ownership.)</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to <strong>Denied</strong>.</td>
</tr>
</tbody>
</table>

**Note:** Cube data sources, like those for Microsoft Analysis Services or Oracle Essbase connections, must be used locally. To download the published data source to Tableau Desktop, you need the **Download** permissions. You must explicitly grant the **Download** permissions because the Data Source Connector role does not provide these. For more information, see *Cube Data Sources* on page 893.

4. To further customize the rule, click the actions menu (…) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
5. Follow the same steps to configure additional permission rules on the content for more users or groups.

6. View the resulting permissions.

   Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

**How "Embedded password" and "Prompt user" settings affect permissions for published data source connections**

When a Tableau Desktop user publishes a workbook or data source to Tableau Server, the authentication mode (if used) affects how the Connect permission is evaluated.
If a workbook author uses the **Embedded password** option when publishing a workbook, users will always be able to access the workbook, regardless of those users' unique **Connect** permissions on the published data source connection that is used by the workbook. In other words, the author is effectively giving users permission to access the workbook's published data source connection and those users' **Connect** permissions are irrelevant.

In cases where a workbook author uses the **Prompt users** option when publishing a workbook, access to the workbook's published data source connection will follow their **Connect** permissions. Users will have access to the workbook to the data source connection when the published data source uses an embedded password and **Connect** is **Allowed**. Users will be prompted for the data source password when **Connect** is **Allowed**.

The following table summarizes how the **Connect** permission interacts with different modes of authentication for a workbook's published data source connection.

<table>
<thead>
<tr>
<th>Workbook</th>
<th>Published Data Source Connection</th>
<th>Connect capability</th>
<th>Access to data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded password</td>
<td>Embedded password</td>
<td>Allowed</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td></td>
<td>Prompt user</td>
<td>Allowed</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td>Prompt user</td>
<td>Embedded password</td>
<td>Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Denied</td>
</tr>
<tr>
<td></td>
<td>Prompt user</td>
<td>Allowed</td>
<td>Prompt user for credentials</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Denied</td>
</tr>
</tbody>
</table>

**Set Permissions for a Project**

Every project includes permissions that can be set for the project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the
For more information on working with permissions, see Manage Permissions on page 834 and Projects on page 756.

Note: When you create a new project, it initially will have the same permissions as the Default project in the site, which are the default permissions for the project, and its workbooks and data sources.

The three capabilities you can set specifically for a project are: View, Save, and Project Leader.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</td>
</tr>
<tr>
<td>Save</td>
<td>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project.</td>
</tr>
</tbody>
</table>
When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining all permissions.

Subsequently, the original owner's access to the workbook is determined by that user's group permissions and any further permissions the new owner might set.

This permission also determines the user’s or group’s ability to overwrite a workbook after editing it on the server. For related information, see Grant Web Edit, Save, and Download Permissions on page 878.

| Project Leader | Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions. |

To set permissions for the project

1. On the Projects page, select a project, and then select Actions > Permissions.

2. Click Add a user or group rule, select Group or User, and then select the group or user name from the list.
3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click Save.

The available permission role templates for projects are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>Allows the user or group to view the workbooks and views in the project.</td>
</tr>
<tr>
<td>Publisher</td>
<td>Allows the user or group to publish workbooks and data sources to the server.</td>
</tr>
<tr>
<td>Project Leader</td>
<td>Allows the user or group to set permissions for all items in a project.</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>-------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to <strong>Denied</strong>.</td>
</tr>
<tr>
<td>Data Source Connector</td>
<td>Allows the user or group to connect to data sources in the project.</td>
</tr>
<tr>
<td>Data Source Editor</td>
<td>Allows the user or group to connect to, edit, download, delete, and set permissions for a data source in the projects. They can also publish data sources, and as long as they are the owner of a data source they publish, can update connection information and extract refresh schedules. This permission is relevant for views when the view they access connects to a data source.</td>
</tr>
</tbody>
</table>

4. To further customize the rule, click the actions menu (... ) next to the permission rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

- 376 -
6. Follow the same steps to configure additional permission rules on the content for more users or groups.

**Set Default Permissions for a Project, and its Workbooks and Data Sources**

As an administrator or project leader, you can set a project's permissions and the default permissions for its workbooks and data sources.

Each project can have its own set of default permissions. The permissions that you set are the default permissions for all content in the project, including content that is being published to the project from Tableau Desktop.

**Note:** New projects are always created with the default permissions set for the Default project.

For additional information on working with permissions, see Manage Permissions on page 834 and Projects on page 756. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.
Notes on default permissions in locked projects

You can choose to have the default permissions apply to all workbooks and data sources in a project, and ensure that no one can change those settings, by locking content permissions to the project. For more information, see Lock Content Permissions to the Project on page 869.

- Workbooks and data sources in a locked project always use the default permissions set for content in that project. Views in a locked project always use the workbook permissions. This applies to workbooks and data sources when they are being published from desktop.
- Administrators and users with the Project Leader permission can always edit default permissions, even when a project is locked.
- Users, including content owners, cannot edit individual workbook, view, and data source permissions when content is locked to the project.

To set default permissions in a project

1. In the Content page of a site, click a project, and then click Permissions in the project place page.

2. Click Add a user or group rule, select Group or User, and then select the group or user name from the list.
For an existing user or group, click the actions menu (..), and then click **Edit**.

3. Select a permission role template for **Project**, **Workbooks**, or **Data Sources**, and then click **Save**.
Or, to create a custom set of capabilities, click the Project, Workbooks, or Data Sources labels to expand the permissions view. Click capabilities to set them to Allowed, Denied, or Unspecified. Click Save.

This example shows how to set project permissions. The same general steps apply for workbooks and data sources.

Note: To change the settings after saving, click the actions menu (…), and then click Edit.

4. View the user permissions, which are the effective permissions.
   Click a group name or user name in the permission rules to see the resulting user permissions.
Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.

Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
5. Follow the same steps to configure additional permission rules for more users or groups.

Quick Start: Lock Content Permissions to a Project

As an administrator or project leader, you can lock content permissions in a project to prevent users from changing the permissions of any content in the project. When permissions are locked to the project, the default permissions are applied to all workbooks and data sources in a project and cannot be modified by users (including the content owners).

Note: Content owners always get full access to the content they've published, but cannot change permissions for their workbooks and data sources when the parent project permissions are locked.

For related information on setting permissions, see Manage permissions. For more information on setting default permissions and locking content permissions to the project, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861 and Lock Content Permissions to the Project on page 869. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

1 Set Default Permissions for the Project

Because the content inside locked projects always uses the default permissions, first verify that your default permissions are set appropriately. In a site, click Content > Projects. Open a project, and then click Permissions. Add a user or group and select a permission role template for that content type, or click Edit, and then set capabilities to Allowed, Denied, or Unspecified.
Administrators and Project Leaders can edit default permissions at any time.

2 Lock Content Permissions to the Project

In a project's permissions, click the Managed by the owner button. The button label indicates whether content permissions are currently locked to the project or managed by the content owner. Select Locked to the project, and then click Save.
When permissions are locked to the project, all content in the project uses the default permissions. No users can change permissions for individual workbooks (including views) or data sources in the project.

3 View Locked Permissions

Open a project, select a workbook or data source in the project, and then click **Actions > Permissions**. When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

In this example, the workbook owner has full permissions for the workbook, but cannot change the workbook permissions while they are locked to the project.
Unlock Content Permissions for the Project

In a site, click **Content > Projects**. Select a project, and then click **Actions > Permissions**. Click the **Locked to the project** button. Select **Managed by the owner**, and then click **Save**.

![Content Permissions in Project](image)

When a project's content permissions are **Managed by the owner**, individual workbooks, views, and data sources in the project start with the default permissions and can be modified by users.

**Notes on project permissions:**

- Only administrators and project leaders can lock content permissions, and set and edit default permissions in a project.
- Administrators and project leaders can edit default permissions for the project, its workbooks, and its data sources at any time, at the project level.
- Individual workbook, view, and data source permissions cannot be edited by users (including content owners) when a project is locked.
- Workbooks and data sources in a locked project always use the default permissions. Views in a locked project always use the workbook permissions.

Lock Content Permissions to the Project

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that
When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

**Note:** Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For information on default permissions, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

**Note:** Administrators and project leaders can set and edit default permissions for the project, and its workbooks and data sources when it is locked.

1. In the Content page of a site, open a project, and then click **Permissions** in the project place page.

2. Click the **Managed by the owner** button.

   The padlock icon on the button label indicates whether content permissions are currently locked to the project or managed by the content owner.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.
When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

4. To unlock content permissions for the projects, open the project permissions again. Click the **Locked to the project** button. In the **Content Permissions in Project** dialog box, select **Managed by the owner**, and then click **Save**.
The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

**Edit Permission Rules**

1. In the Content page of a site, select a project, workbook, view, or data source, and then select **Actions > Permissions** to view the current permission rules.

To select an item in the page, select the checkbox for the item.

Example of permission rules for a workbook.

2. For the permission rule that you want to change, click the actions menu (…) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

3. View the resulting permissions.
Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

### View Permission Rules and User Permissions

At any time, you can view the permissions for a user or group, for a view, workbook, project, or data source. The permissions shown are specific to the view, workbook, data source, or project you have selected.

1. **On the Content page for a site, click Workbooks, Views, Projects, or Data Sources.**
   To select an item in the page, select the checkbox for the item.

2. **Select Actions > Permissions** to view the current permission rules.
3. Click a group or user name in the permission rules area to see the resulting permissions. Hover over a capability box in User Permissions to see a tooltip with details on whether a capability is allowed or denied.

Custom indicates a user's capabilities have been changed from the initial settings for their site role or content role.

Permissions Reference

Administrators and other authorized users can allow or deny permissions on resources in Tableau Server. Permissions can also be set in Tableau Desktop when publishing a workbook or data source to Tableau Server.

Administrators always have full control of all resources on Tableau Server, and site administrators have full control of all resources on a site. If you publish a workbook or data source to Tableau Server, you are the owner of that resource, and you retain full control over
that resource, with the exception of setting permissions on resources in locked projects. For more information, see Lock Content Permissions to the Project on page 869.

The following table shows which permissions apply to which resources in Tableau Server, and describes the capabilities (that is, the actions users can perform) with each permission.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Applies to...</th>
<th>When allowed, users can...</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>workbooks, data</td>
<td>View the item on Tableau Server.</td>
</tr>
<tr>
<td></td>
<td>sources, views,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>projects</td>
<td></td>
</tr>
<tr>
<td>Web Edit</td>
<td>workbooks, views</td>
<td>Edit views in workbooks. See Grant Web Edit, Save, and Download Permissions on page 878.</td>
</tr>
<tr>
<td>Save</td>
<td>workbooks, data</td>
<td>Overwrite the resource on the server. When allowed, the user can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining access to all permissions. Subsequently, the original owner’s access to the workbook is determined by that user's group permissions and by any further permissions the new owner might set.</td>
</tr>
<tr>
<td></td>
<td>sources, views,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>projects</td>
<td></td>
</tr>
<tr>
<td>Download Workbook/Save As</td>
<td>workbooks</td>
<td>Download a workbook from the server, and also save an edited workbook as a new workbook on the server. For more information, search for &quot;Download Workbooks&quot; in the Tableau</td>
</tr>
</tbody>
</table>

- 391 -
<table>
<thead>
<tr>
<th>Permission</th>
<th>Applies to...</th>
<th>When allowed, users can...</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="User" /></td>
<td>Server Help and see Grant Web Edit, Save, and Download Permissions on page 878.</td>
<td></td>
</tr>
<tr>
<td>Download Data Source</td>
<td>data sources</td>
<td>Download the data source from the server.</td>
</tr>
<tr>
<td><img src="image" alt="Delete" /></td>
<td>workbooks data sources views</td>
<td>Delete the resource.</td>
</tr>
<tr>
<td>Filter</td>
<td>workbooks views</td>
<td>Modify filters in the view, keep only filters, and exclude data.</td>
</tr>
<tr>
<td>Add Comments</td>
<td>workbooks views</td>
<td>Add comments to views in a workbook. Search for &quot;Comment on Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td>View Comments</td>
<td>workbooks views</td>
<td>View the comments associated with the views in a workbook. Search for &quot;Comment on Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td>Download Summary Data</td>
<td>workbooks views</td>
<td>View the aggregated data in a view, or in the user’s selection within the view, and download that data as a text file.</td>
</tr>
<tr>
<td>Download Full Data</td>
<td>workbooks views</td>
<td>View the raw data behind each row in a view, as restricted by any marks the user has selected, and download the data as a text file.</td>
</tr>
<tr>
<td>Permission</td>
<td>Applies to...</td>
<td>When allowed, users can...</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><img src="image" alt="Download Image/PDF" /></td>
<td>workbooks, views</td>
<td>Download each view as an image. For more information, search for &quot;Download Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td><img src="image" alt="Share Customized" /></td>
<td>workbooks, views</td>
<td>Make saved customizations to a view available for others to see. Users can create custom views using <strong>Custom Views</strong> in Tableau Server. For more information, Search for &quot;Custom Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td><img src="image" alt="Move" /></td>
<td>workbooks</td>
<td>Move workbooks between projects. <strong>Note:</strong> Only administrators can move data sources between projects.</td>
</tr>
<tr>
<td><img src="image" alt="Set Permissions" /></td>
<td>workbooks, data sources, views</td>
<td>Specify permissions for the resource. For workbooks, this permission extends to the views in a workbook.</td>
</tr>
<tr>
<td><img src="image" alt="Connect" /></td>
<td>data sources</td>
<td>Connect to the data source on the server. The <strong>Connect</strong> permission allows a user to connect to a published data source from an editor (in Tableau Desktop or Tableau Server web editing). If a workbook author embeds credentials in a workbook or view, users who also have the <strong>Web Edit</strong> permission will be able to access to the workbook's data source regardless of their <strong>Connect</strong> permissions. <strong>Note:</strong> If a workbook is configured to show sheets as tabs, all views use the workbook permissions, even if different permissions are specified on an individual view.</td>
</tr>
<tr>
<td><img src="image" alt="Project Leader" /></td>
<td>projects</td>
<td>Set permissions for all resources in a project and for the project itself. Can lock project permissions and edit default permissions.</td>
</tr>
<tr>
<td>Permission</td>
<td>Applies to...</td>
<td>When allowed, users can...</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>----------------------------</td>
</tr>
<tr>
<td>✗</td>
<td></td>
<td>Also can change the owner, move content, and run refresh schedules.</td>
</tr>
</tbody>
</table>

Note: Tabbed views are views in a workbook that is published to the server with Show Sheets as Tabs enabled. Tabbed views use the workbook permissions instead of the view permissions. When you view the permissions for a tabbed view in a workbook, you see the workbook’s permission rules in the Permissions window, not the view’s permission rules. To edit tabbed view permissions, you must open the tabbed view’s workbook permissions. The changes that you make to the workbook permissions affect all tabbed views in that workbook. When the workbook is saved again without tabs (or tabs are hidden), the default permissions are again applied to the workbook and views, but view permissions can then be edited.

Views in a workbook in a project with locked permissions will also use the workbook permissions. For more information, see Lock Content Permissions to the Project on page 869.

Grant Web Edit, Save, and Download Permissions

For a user to be able to edit, save, and download workbooks, they must have a site role that allows those actions, and specific capabilities in a user or group permission rule.

The following capabilities control whether a user can edit, save, and download views:

- **Web Edit**—determines whether the user can edit workbook views in Tableau Server.

  To edit an existing workbook, a user must have a site role of Interactor or Publisher. The Web Edit capability must be set to Allowed in the workbook permissions.

  Note: Users with a site role of Interactor are not allowed to save or download workbooks.

- **Download/Save As**—determines whether users see the Save As command while they are editing a view, and whether they can save their changes to a new workbook. It also determines whether users can open a workbook on the server using Tableau Desktop.

  To save changes to a workbook or save a workbook as a new workbook on Tableau Server, a user must have a site role of Publisher. The Save and Download/Save As capabilities must be set to Allowed in the workbook permissions.

- **Save**—determines whether users can save changes to an existing workbook on the
server (overwrite a workbook).

**Note:** The **Save** permission determines whether a user can overwrite the content on the server. This permission does not determine whether a **Save** button is displayed for users who do not own the content. Only the workbook owner can save changes to an existing workbook on Tableau Server.

To save changes to a workbook, a user must have a site role of **Publisher**. The **Save** capability must be set to **Allowed** in the workbook permissions.

**Note:** Setting the **Save** capability to **Denied** for a project disables saving to the entire project, as well as disabling overwriting the existing workbook.

**To grant Web Edit permissions**

1. Set the site role of the user to **Interactor** or **Publisher**. For more information, see [Change Site Roles on page 820](#).

2. In the permission rules for a group or user at the workbook level, set the **Web Edit** capability to **Allowed**.

3. Save the rule.

**To grant Save and Download/Save As permissions**

1. Set the site role of the user to **Publisher**. For more information, see [Change a Site Role](#).

   **Note:** **Interactors** are not allowed to save or download workbooks.

2. Create a permission rule for a group or user at the project and workbook level. Set the following capabilities:

To allow users (Publisher site role) to edit and save changes to existing and new workbooks

<table>
<thead>
<tr>
<th>Permission</th>
<th>For the project</th>
<th>For specified workbooks in the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Save</td>
<td>Allowed</td>
<td>Allowed</td>
</tr>
</tbody>
</table>
Note: To apply the default permissions to all workbooks within the project, lock content permissions to the project. For more information, see Lock Content Permissions to the Project on page 869.

To allow users (Publisher site role) to edit and save changes to new workbooks, but not overwrite existing workbooks

<table>
<thead>
<tr>
<th>Permission</th>
<th>For the project</th>
<th>For specified workbooks in the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Save</td>
<td>Allowed</td>
<td>Denied</td>
</tr>
</tbody>
</table>

Important: In this scenario, permissions must be set manually on each workbook and the project permissions are not locked. If project permissions are locked, the permissions apply to all workbooks in the project.

3. Save the rule.

Note: When you deny Save permissions for a workbook, users can still click Save As when editing the workbook in Tableau Server, but a message appears that tells users they do not have permission to overwrite the workbook and the changes will not be saved.

About permissions for views in workbooks

Permissions for views in workbooks are inherited from the workbook permissions.

If a user selects Show sheets as tabs when publishing a workbook from Tableau Desktop or saving it on Tableau Server, the workbook permissions override the permissions on individual views. When the workbook is saved again without tabs, the default permissions are applied to the workbook and views, but view permissions can then be edited.

See also

Permissions Reference on page 874

Quick Start: Permissions

Quick Start: Lock Content Permissions to a Project on page 866
Permission Rules and User Permissions

Set Permissions for Workbooks and Views on page 846
Set Permissions for a Project on page 856
Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861
Create Project-Based Permissions on page 881
Site Roles for Users on page 788

Create Project-Based Permissions

As an administrator, you can organize a collection of related workbooks and data sources in a project. You can then control access to that content by creating permission rules for groups of users who need similar access levels to publish or interact with that content.

Note: For this scenario, you set the permission rule for the All Users group for the project to None, which means that permissions are Unspecified for the All Users group.

Preparation

Before you begin creating projects and project-based permissions, document the projects and permission levels that you want users to have in each project.

This roadmap exercise helps you organize permissions to be most efficient to manage over time, and can help you identify any user or permission gaps in your solution. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Also read the following topics in the Tableau Server Help:

- Manage Permissions on page 834 and permissions-related topics
- Projects on page 756 and projects-related topics
- Grant Web Edit, Save, and Download Permissions on page 878

Step 1: Create projects and user groups

1. Sign in to Tableau Server with your administrator user name and password.
2. On the Projects page, click New Project.
3. Click Groups, and then click New Group.

Create groups that correspond to each project and access level. For example, for a project that allows users only to access the views, you might use a name similar to Project1_Viewer. For a project that allows interaction with the views, Project1_Interactor.

4. Click Users, and then click Add Users. Select one or more users in the list, select
**Actions > Group Membership**, and then select a group for the users. Click **Save** to confirm the group membership.

Repeat this step to add users to other groups.

**Step 2: Assign permissions at the project level**

After you set up your projects and user groups, you can start assigning permissions. Repeat these steps for each project. Also see **Set Default Permissions for a Project, and its Workbooks and Data Sources** on page 861.

1. On the Projects page, select a project, and then select **Actions > Permissions**.

2. For the **All Users** group permission rule, set the permission role template to **None**.
   
   Click the actions menu ( . . . ) next to **All Users**, and then click **Edit**. Select **None** for **Project, Workbooks, and Data Sources**, and then click **Delete**. This means that all capabilities will be set to **Unspecified**.

3. Click **Add a user or group rule**, select **Group**, and then select the group name in the list.
   
   To edit an existing rule, click the actions menu ( . . . ) next to the permission rule name, and then click **Edit**.

4. Select a permission role template for **Project, Workbooks, and Data Sources** to specify a predefined set of capabilities for the group or user.

5. To further change capabilities included in the rule, click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**.
   
   Click **Save** when you are done.

Repeat steps 3-5 for each group or user requiring project permissions.

**Note**: You can optionally lock content permissions to the project to enforce the default permissions for all content in the project. This overwrites any previous permissions assigned to workbooks and views in the project. For more information, see **Lock Content Permissions to the Project** on page 869.

**Step 3: Check project permissions**

- View the resulting user permissions.

  Click a group name or user name in the permission rules list to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

When you publish workbooks to the project, the permissions are updated accordingly.
For information on granting Save permissions to users, see **Grant Web Edit, Save, and Download Permissions** on page 878.

**Enable Web Authoring**

The ability for users to edit views in Tableau Server is a setting that administrators control. In addition to this setting being enabled, user must also have the **Web Edit** permission for a given content item.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.

2. In a site’s Settings page, make sure **Allow users to use web authoring** is selected.

3. In the permissions for a workbook or a view, make sure the permission rule for a user or group allows the **Web Edit** capability.

4. If your site is already in production, and you want the change to take effect immediately, restart the server.

To confirm which sites allow web authoring, on the site menu, click **Manage All Sites**, and then click the **Sites** menu.
For more information on web authoring and web editing in Tableau Server, also see these topics:

- **Disable web authoring**
- **The Web Authoring workspace**
- **Grant edit and save permissions**

## Disable Web Authoring

If you want users to be able to view published workbooks on Tableau Server but not access the web editing environment, you can use a site-level setting to disable authoring.

For example, you might have a select group of data analysts who use Tableau Desktop to create and publish workbooks, and a group of sales managers working in the field, who do not use Tableau Desktop but need to access the published dashboards from a web browser.

1. In a web browser, sign in to the server as an administrator and go to the site for which you want to disable authoring.

2. With Site selected, display the **Settings** page.

3. In the Site Settings page, clear the check box for **Allow users to use web authoring**, and then click **Save**.

```
Web Authoring

Users with the appropriate permissions can edit workbooks in their browser.

Allow users to use web authoring
```

If you disable web authoring while creating a new site, no cached sessions exist, and the setting takes effect immediately.

Otherwise, the change takes effect after server session caching expires or the next time a user signs in after signing out.

Until the change takes effect, users might have authoring access if they see an Edit link on a view, or if they enter the URL for the view’s edit mode. For example, they bookmarked the URL while they had the view open for editing.

4. If your site is already in production, and you want the change to take effect immediately, restart the server.
To confirm which sites allow web authoring, on the site menu, click **Manage All Sites**, and then click the **Sites** menu.

![Tableau Server Site Management](image)

**Users**

Everyone who needs access to Tableau Server must be added as a user.

**Guest user**

A Guest user is available in Tableau Server (core-based licenses only) in each site to allow users who don’t have an account on the server to see and interact with embedded views. When enabled, the user can load a webpage that contains an embedded visualization without signing in. For more information, see **Guest User** on page 792.

**Server users and site users**

Server administrators can add users to the server, and server administrators and site administrators (if enabled under Guest Access in **Settings** for the server) can add users to individual sites. For details on allowing site administrators to add users to sites, see step 4 in **Add or Edit Sites** on page 740.

In a single-site environment, server and site administrators can add users on the **Users** page.
In a multi-site environment, server and site administrators can add users in the **Site Users** page.

Server administrators can add users in the **Server Users** page. When you add a user to the server, you can assign the site membership and site roles per site for the user.
If you add a user without assigning site membership and role, the user is assigned the Unlicensed role and won't use a server license (user-based licensing only). The user will exist in Server Users, but will not be a member of any site until you add that user to the site.

**Note:** Every user who is added to a site is also automatically added to the server. Site administrators can remove users from their sites, but they cannot delete users from the server. Server administrators can delete users from the server.

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

When you publish a data source or workbook on Tableau Server or when you create a project, you become its owner. Ownership can be changed. For example, if an employee who is the original owner leaves, the administrator can reassign ownership to another user. After you change ownership, the original owner has no special connection to the item, and their ability to access it is determined by their Tableau Server permissions.

**Note:** You cannot delete a Tableau Server user if the user owns any items. When you attempt to delete the user, their site role is set to Unlicensed. You must first change the ownership of the items and then delete the user. For more information, see Deleting a User from Tableau Server.

If you change the ownership of a workbook or data source that has embedded credentials, the embedded credentials will be deleted. You will need to download the workbook or data source, update the embedded credentials for the new owner, and then re-upload the workbook or data source.

Your ability to change or be given ownership depends on your permissions and your relationship to the item, as described in the following table.

<table>
<thead>
<tr>
<th>Item type</th>
<th>Who can change ownership</th>
<th>Who can be given ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Projects</strong></td>
<td>Server administrator</td>
<td>Server administrator</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td>Site administrator</td>
</tr>
<tr>
<td><strong>Workbooks and Data Sources</strong></td>
<td>Server administrator</td>
<td>Server administrator</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td>Site administrator</td>
</tr>
<tr>
<td></td>
<td>Owner of the item</td>
<td>Member of the site that contains the item (Guest user excluded).</td>
</tr>
<tr>
<td></td>
<td>Project leader for the project that contains the item</td>
<td></td>
</tr>
</tbody>
</table>

Change a Workbook Owner

By default, the publisher of a workbook is its owner. Administrators, project leaders, and the current owner of the workbook can change ownership. The new owner must be a server administrator or a site administrator, or be any user other than Guest on the same site as the workbook.

**To change the owner for a workbook**
1. On the Content page for a site, select **Workbooks**.
2. Select one or more workbooks, and then select **Actions > Change Owner**.

3. Type the name of a user or select a user from the list.

4. Click **Change Owner**.

**Change a Data Source Owner**

By default, the publisher of a data source is its owner. Administrators, project leaders, and the current data source owner can change ownership. The new owner must be a server or site administrator, or be any user other than Guest on the same site as the data source.

**To change the owner for a data source**
1. On the Content page for a site, select **Data Sources**.

2. Select one or more data sources, and then select **Actions > Change Owner**.

3. Type the name of a user or select a user from the list.

4. Click **Change Owner**.

**Change a Project Owner**

By default, the creator of a project is its owner. Administrators can change project ownership. The new owner must be a server administrator or an administrator for the project's site.

**To change the owner for a project**

1. On the Content page for a site, select **Projects**.

2. Select one or more projects, and then select **Actions > Change Owner**.
3. Type the name of a user or select a user from the list.
4. Click **Change Owner**.

**Site Roles for Users**

Every user added to Tableau Server must have an associated site role. The site role is assigned by the administrator. The site role determines the levels of permissions allowed for a user, including whether a user can publish, interact with, or only view content published to the server. Administrators are also defined based on the site role.

**Note:** Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.

Users are accounts on the server that can be associated with one or more sites, and with groups in those sites. Any user that is added to Tableau Server or to a site becomes member of the All Users group. The All Users group is present in every site and cannot be deleted.

**Who can publish content**

Users with the following site roles can publish to Tableau Server:

- Server Administrator
- Site Administrator
- Publisher
- Viewer (can publish)
- Unlicensed (can publish)

**Note:** A system change is required on computers that Unlicensed (can publish) users will use to publish. For more information, see the corresponding quick fix article.

Users with a site role of Interactor, Viewer, and Unlicensed cannot publish content to the server.

**Site roles and permissions**

Effective user permissions for a resource are determined by:

- The maximum capabilities allowed for a user's site role. The site role acts as the "ceiling" for what permissions are allowed.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

When you select a site role for a user, help is available to remind you of the general level of permissions for that site role.

![Site role permissions table](image)

When you set permissions for a content item, the User Permissions section in the Permissions window indicates when a permission capability is not allowed for that site role.

The User Permissions area of the Permissions window shows the effective permissions for each user. These are the actual permissions for each user, after the user’s site role and permission rule has been evaluated.

For details, see Permission Rules and User Permissions on page 836.
General capabilities

- **Server Administrator**: The server administrator can access all server features and settings on the server and all sites. Server administrators can create sites, add users of any site role type, control whether site administrators can add users, create additional server administrators, and they can administer the server itself. This includes handling maintenance, settings, schedules, and the search index.

  Server administrators can perform operations on all content anywhere on the server, regardless of what permissions have been assigned to the content. Server administrators can also manage other users on the server.

- **Site Administrator**: Site administrators can manage groups, projects, workbooks, and data connections. By default, site administrators can also add users and assign site roles and site membership. This setting can be enabled or disabled by the server administrator (see step 4 in Add or Edit Sites on page 740).

  Site administrators have unrestricted access to content on a specific site. A user can be specified as a site administrator on multiple sites.

- **Publisher**: Publishers can sign in, browse the server, and interact with the published views. They also can connect to Tableau Server from Tableau Desktop in order to publish and download workbooks and data sources.

  Publishers can publish (upload) workbooks and data sources to the server. Publishers aren't allowed to manage other users.

- **Interactor**: Interactors can sign in, browse the server, and interact with the published views. It's important to note that specific views, workbooks, and projects may have been published with permissions that restrict a user's capabilities. Permission settings can be edited by the workbook author or an administrator.

  Interactors can view workbooks and can interact with views. They are not allowed to publish to the server.

- **Viewer**: Viewers can sign in and see published views on the server but cannot interact with the views. Users with this site role can only be given permission to view, add comments, and view comments. They cannot interact with filters in the view or sort data in a view.

- **Unlicensed**: Unlicensed users cannot sign in to the server. When you import server users from a CSV file, all are assigned a site role of Unlicensed.

  If an insufficient number of licenses are available when an administrator creates a user (though CSV import of a site user, or import from Active Directory, or when a local user is created) the user will be assigned the Unlicensed site role.

  Attempting to remove a user who owns content from a site will demote the user to Unlicensed. The user will still own the content.

- **Viewer (can publish)**: The user can connect to Tableau Server from Tableau Desktop
to publish and download workbooks and data sources, but cannot interact with content on the server.

- **Unlicensed (can publish)**. The user can connect to Tableau Server Tableau Desktop to publish workbooks to the server, but cannot sign in to Tableau Server directly.

**Site roles and Active Directory import and synchronization**

When you import Active Directory users to a site, either as a single user or as member of a group, you can specify a site role for the user. If a user is not yet a member of any site on the server, the user is added to the site with the assigned role. When you synchronize Active Directory groups, the site role is applied through the **Minimum Site Role** setting on the **Groups - Details** page.

If a user already exists in a Tableau Server site, the site role assigned during the import or sync process will be applied if it gives the user more access in a site. Importing or synchronizing users and groups will promote a user’s site role, but not demote a user’s site role.

If a user already has the ability to publish, that ability will always be maintained. For example, if a user with the current site role of **Unlicensed (can publish)** is imported with the new site role of **Interactor**, that user’s site role will be promoted to **Publisher** on import.

To guarantee a user maintains a site role with equal or greater capabilities in server after an import, the following matrix shows the rules applied for site roles on import. Bold indicates that a site role was promoted to preserve the user’s ability to publish.

**Note:** The **Import Site Role** row headers indicate the site role specified for import. The **Current Site Role** column headers represent the current user site role. The table values represent the resulting site role. A bold site role in the table indicates a site role promotion that preserves the ability to publish.

<table>
<thead>
<tr>
<th>Import Site Role</th>
<th>Site Administrator</th>
<th>Publisher</th>
<th>Interactor</th>
<th>Viewer</th>
<th>Viewer (can publish)</th>
<th>Unlicensed</th>
<th>Unlicensed (can publish)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td></td>
</tr>
<tr>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td></td>
</tr>
<tr>
<td>Inter-</td>
<td>Publisher</td>
<td>Inter-</td>
<td>Inter-</td>
<td>Pub-</td>
<td>Inter-</td>
<td>Pub-</td>
<td></td>
</tr>
</tbody>
</table>
## Change Site Roles

Server administrators and site administrators with the ability to add site users can change the site role of a user at any time. For details on site roles, see [Site Roles for Users on page 788](#).

Only server administrators can change the site membership of users. For details, see [Assign Site Membership on page 801](#).

1. In a site, click **Users**.
2. Select one or more users, and then select **Actions > Site Role**.

### Current Site Role

<table>
<thead>
<tr>
<th>Import Site Role</th>
<th>Site Administrator</th>
<th>Publisher</th>
<th>Interactor</th>
<th>Viewer</th>
<th>Viewer (can publish)</th>
<th>Unlicensed</th>
<th>Unlicensed (can publish)</th>
</tr>
</thead>
<tbody>
<tr>
<td>actor</td>
<td>actor</td>
<td>actor</td>
<td>actor</td>
<td>lisher</td>
<td>lisher</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewer (can publish)</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>Viewer</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Interactor</td>
<td>Viewer</td>
<td>Viewer (can publish)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unlicensed (can publish)</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
<td>Unlicensed (can publish)</td>
<td>Unlicensed (can publish)</td>
</tr>
<tr>
<td>Unlicensed</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Interactor</td>
<td>Viewer</td>
<td>Viewer (can publish)</td>
<td>Unlicensed</td>
<td>Unlicensed (can publish)</td>
</tr>
</tbody>
</table>
3. Select a site role, and then click **Change Site Role**.

---

**Guest User**

A Guest user is available in Tableau Server to allow users who don’t have an account on the server to see and interact with an embedded view. When enabled, the user can load a webpage that contains an embedded visualization without signing in.

**Note:** The Guest user option is available only with a core-based license.

When you embed a Tableau Server view into an internal website page, every person who views that page will need a Tableau Server account (they'll be asked for a user name and password)
unless you have purchased a core-based (hardware) license. In that case you can have as many accounts as you want, as well as the ability to enable Guest user access, which does not require log in or authentication.

Guest is a special account that is used only to allow users to see views. The Guest user cannot browse the Tableau Server interface and won't see server interface elements in the view, such as user name, account settings, comments, and so on.

Note: Enabling the Guest user for a site can increase the number of potential simultaneous viewers beyond the user list you might be expecting. The administrative view Status > Traffic to Views can help you gauge the activity.

A Guest user can have the following permissions

Projects, Workbooks, and Views: View, Export Image, Summary Data, View Comments, Filter, Full Data, Web Edit, Download (to save a local copy)

Data Sources: View and Download

When a Guest user is included in a group that has a permission rule set on a content item, Guest user permissions do not affect the permission-levels of other users in that group.

To enable Guest access

1. Single-site: Click Settings > General.
   Multisite: In the site menu, click Manage All Sites and then click Settings > General.

2. For Guest Access, select Enable Guest account to allow people who are not signed into a Tableau Server account to see views with Guest access permissions.

3. Click Save.

The Guest user is unique in the following ways:

- The Guest user represents all unauthenticated users accessing content on the server.
- Tableau Server must use a core license for Guest to be available.
- Server administrators can enabled/disable Guest across the server; it is not controllable per site.
- The Guest user cannot be edited and can never own content.
- The Guest user can be made a member of one or more groups in a site.
- Only the server administrator can enable or disable Guest access (in All Sites > Settings > General).
- The Guest user, when enabled, is a member of the All Users group.
The Guest user cannot be deleted; it must be disabled by the server administrator in All Sites > Settings > General.

If the Guest user needs to be able to access a workbook that uses an extract data source, make sure Guest has the View permission for the data source. The Guest user is not allowed to connect to published data sources, unless the publisher embedded their credentials when publishing the content.

The Guest user is not allowed to save customized views.

The "Guest" user is not a user name. It cannot be used to log in, to request trusted tickets, or in a user filter.

Add Users to a Site

Administrators can add users to sites in the following ways:

- By adding a local user account or a user account from Active Directory, as described in this topic. You can also add users by importing an Active Directory group. For details, see Create a Group via Active Directory on page 822.
- By importing a CSV file that contains user information. For details, see Import Users on page 804 and CSV Import File Guidelines on page 810.

In a single-site environment, administrators can add users to a site on the Users page. In a multi-site environment, you will use the Site Users page. Server administrators must give site administrators the ability to add users to sites. This setting can be enabled or disabled by the server administrator (see step 4 in Add or Edit Sites on page 740).

Note: Users can be added to sites, or to the server. To add users to the server, see Add Users to the Server on page 794. The options available for adding users depends on the authentication method that you select when you first configure Tableau Server. If you are using local authentication, you cannot add Active Directory users. If you are using Active Directory, you cannot add local users.

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

Note: When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn’t own any content.
Note: This screenshot is from a single-site environment. In a multi-site environment, this would be the Site Users page.

To add local users to a site

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

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

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

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

3. Select a site role. For details on site roles, see **Site Roles for Users** on page 788.

4. Click **Add User**.

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

To add Active Directory users to a site

Before adding users to a site, be sure to review **User Management in Active Directory Deployments** on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.
1. In a site, click **Users**, and then click **Add Users**, and then click **Active Directory User**.

![Add Users to this Site](image)

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

![Import Users from Active Directory](image)

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

3. Select a site role. For details on site roles, see **Site Roles for Users** on page 788.
4. Click **Import Users**.

**Assign Site Membership**

Server administrators and site administrators with the ability to add site users can change a user’s site role. For details on site roles, see **Site Roles for Users** on page 788 and **Change Site Roles** on page 820.

Only server administrators can change the site membership of users.

1. In the site menu, click **Manage All Sites**, and then click **Users**.
2. Select one or more users, and then select **Actions > Site Membership**.

![Server Users screenshot]

**Server Users 77**

- **Add Users** - 2 selected
- **Actions**
  - **Site Membership**
  - **Delete**

**General Filters**

- **Max Site Role**
- **Any site role**

<table>
<thead>
<tr>
<th>Username</th>
<th>Max site role</th>
<th>Sites</th>
<th>Last Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>Admin</td>
<td>Server Administrator</td>
<td>Aug 9</td>
</tr>
<tr>
<td>alejandrogrove</td>
<td>Interactor</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>andrewallen</td>
<td>Publisher</td>
<td>1</td>
<td>Jul 25</td>
</tr>
<tr>
<td>andrewsmith</td>
<td>Publisher</td>
<td>1</td>
<td>Jun 17</td>
</tr>
<tr>
<td>ashleygarcia</td>
<td>Site Administrator</td>
<td>2</td>
<td>Jun 16</td>
</tr>
<tr>
<td>brendangweed</td>
<td>Publisher</td>
<td>1</td>
<td>Sep 1</td>
</tr>
</tbody>
</table>
3. Select one or more sites, and a role for each site, and then click **Save**.

![Site Membership](image)

**Import Users**

To automate the process of adding users to a site, you can create a CSV file that contains user information, and then import the file. You can import users to a site, or, to the server (if you are a server administrator).

*Note:* This topic contains the steps for importing, assuming that you have already created the CSV file. If you have not created the file yet, see [CSV Import File Guidelines](#) on page 810 for a list of file format requirements and import options.
Add users from a CSV file to a site

1. In a site, click Users, click Add Users.

![Add Users to this Site](image)

2. Click Import From File, click Browse and navigate to the file, and then click Import Users.

![Import users from file](image)

The results of the import are displayed.
For a single-site server, the site roles assigned to the users during the import process will be imported with the users. If a user already exists in the Tableau Server site, the site role assigned during the import process will be applied only if it gives the user more access to the server. Importing users and groups will promote a user’s site role, but not demote a user’s site role.

3. Click **Done**.

### Add users from a CSV file to a server

1. **Single-site**: Click **Users**, and then click **Add Users**.
   
   **Multisite**: In the site menu, click **Manage All Sites**, click **Users**, and then click **Add Users**.
2. Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

The results of the import are displayed.

For a multi-site server, when you import users in the Server Users page, you are creating server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a server user can have is either Unlicensed or Server Administrator. When you assign site membership to a server user, you can specify the site role for that user per site. For details, see Assign Site Membership on page 801. If you import the users in the Site Users page, the users will be assigned the site roles you specify in the CSV file, for that site.

3. Click **Done**.
**Multi-site environments**

If the server is running multiple sites and you are a server administrator, you can import a CSV file from two different locations. Where existing user accounts are concerned, each location has different capabilities.

- The **Server Users** page appears in a multi-site environment. Only server administrators can access this page.

![Server Users page](image)

You can import the CSV file from here if you want to update existing user accounts in addition to adding new ones. For example, if you import a file that has a new password for each existing user, their passwords will be reset.

- The **Site Users** page.

![Site Users page](image)

Server administrators can add new user accounts with CSV imports and, if existing users are part of the import, the **Password** and **Display Name** fields must either match or be left blank. If new passwords or full names are used, the import will fail.

**Single-site environments**

Server and site administrators on a single-site server perform CSV user imports from the **Users** page in a site.
**Multi-site versus single-site import**

Users can belong to more than one site on the same server, but they must use the same credentials for each site. This becomes important when you’re adding users to a site and those users might already be members of a different site. If you try to import a user who already exists, and if the user’s credentials in the CSV file don't match the existing credentials, the import fails for that user.

**Note:** The issue of credentials mismatch during import doesn't apply if the server is configured to use Active Directory for authentication. In that case, the CSV file should never contain a password, because user passwords are managed by Active Directory.

If you're importing users into a site and you think that the users might already exist on the server, you can try leaving the Password column in the CSV file blank. When you import the users, if a user who is defined in the CSV already exists in another site, the user is added to the site where you're importing. However, if the user doesn't already exist on the server, the user is created, and the CSV import window alerts you that the new user doesn't have a password. You can then use the server environment to assign a password to any user who doesn't have one.

**Multi-site**

For a multi-site server, when you import users in the Server Users page, you are creating server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a server user can have is either Unlicensed or Server Administrator.

**Single site**

For a single-site server, the site roles assigned to the users during the import process will be imported with the user. If a user you are importing already exists in Tableau Server, the site role assigned during the import process will be applied only if it gives the user more access to the server. Importing users and groups will promote a user's site role, but not demote a user's site role.
CSV Import File Guidelines

You can automate adding users by creating a comma-separated values (CSV) file with user information and then importing the file. You can include attributes in the CSV file, such as site role and the ability to publish, to apply to the users at the same time you import them.

To import users, you can use the server administration pages or the `tabcmd` utility. For details, see Import Users on page 804 or createsiteusers filename.csv on page 1322.

**Note:** If you use the `tabcmd` utility to import users, you can pass options on the command line that can specify default values for the users' site roles. For more information, see the createsiteusers filename.csv on page 1322 documentation.

You can import users into a site or into the server. If you import users into a site, site roles are applied to the user. If you specify site roles, but importing users would exceed your license limits, users are imported as Unlicensed. If you import users into the server (not into a specific site), the user isn't assigned to a site, and site roles in the CSV file like Publisher and Interactor are treated as Unlicensed.

CSV File Format Requirements

When you create the CSV file for importing users, make sure that the file meets the following formatting requirements:

- The file does not include column headings. Tableau Server assumes that every line in the file represents a user.
- The file is in UTF-8 format, and includes the byte-order mark (BOM).
- Character encodings such as BIG-5 have been converted to UTF-8. You can do this by opening the file in a text editor and using the `Save As` command.
- If a name includes the "@" character other than as a domain separator, you need to refer to the symbol using the hex format: \0x40

  For example, `user@fremont@myco.com` should be `user\0x40fremont@myco.com`

Required Columns in the CSV File

The following values are required for each user:

- User name
- Password: If Tableau Server is configured to use Active Directory authentication, there must be a `Password` column, but the column itself should be empty. If the server is using local authentication, you must provide passwords for new users.
Additional Import File Options

The CSV file can contain the following fields, in the order shown here:

- **User name.** The user name. If the server is configured to use Active Directory, this value must match a user defined in Active Directory. If the user name is not unique across domains, you must include the domain as part of the user name (for example, example\Adam or adam@example). This is the only required field.

- **Password.** A password for the user. If the server is configured to use Active Directory, this value is not used.

- **Display name.** The display name is part of the information that's used to identify a user on the server. If the user's display name is already in use, Tableau Server updates the existing user information with the settings in the CSV file. If the server is configured using Active Directory, this value is not used.

- **License level (Interactor, Viewer, or Unlicensed).** This setting determines the role for a non-administrator user. If you are using the server administration pages to import users, the license level is set only if you are importing into an individual site. If you are using the server administration pages to import users while managing the server (not a specific site), and if the user is not set to be an administrator, the site role is set to Unlicensed. *(You can change the site role later.)*

  **Note:** In Tableau Server 9.0, license levels have been replaced with site roles. If you create a user using the server UI, you select a site role like Site Administrator, Publisher, Interactor, and View (can publish). For information about site roles, see **Site Roles for Users** on page 788. For more information about how the license levels and other values in the CSV file are converted to site roles, see **Settings and Site Roles** later in this topic.

- **Administrator level (System, Site, or None).** This setting determines whether the user is imported as an administrator. If you are using the site administration pages, you can set the administrator role to **System** only if you are importing while managing the server. If you are using the server administration pages to import users while you are managing a site, and if the administrator role for a user in the CSV file is set to **System**, Tableau Server imports the user as a site administrator.

- **Publisher permissions (yes/true/1 or no/false/0).** This setting determines whether the user has publisher permissions. If you are using the site administration pages, the publisher setting is used only if you are importing into an individual site. If you are importing users while managing a server, this value isn't used.

- **Email address.** The email address is part of the information that's used to identify a user
on the server. If the email address is already in use, Tableau Server updates the existing user information with the settings in the CSV file.

The order of the columns is significant. The first column is treated as the user name, the second as the password, the third as display name, etc., regardless of the content in the columns.

**Settings and Site Roles**

The license level, administrator, and publisher settings for a user are used during the import process to set a user’s site role. The following table shows how the settings are converted to site roles.

<table>
<thead>
<tr>
<th>CSV settings</th>
<th>Site role</th>
</tr>
</thead>
<tbody>
<tr>
<td>License level=(any)</td>
<td>System (server) administrator. This setting is valid only if you are</td>
</tr>
<tr>
<td>Administrator=System</td>
<td>importing users while managing the server. If you set a user to be a</td>
</tr>
<tr>
<td>Publisher=(any)</td>
<td>system administrator, the other values are ignored.</td>
</tr>
<tr>
<td>License level=(any)</td>
<td>Site administrator. This setting is valid only if you are importing users</td>
</tr>
<tr>
<td>Administrator=Site</td>
<td>while managing a specific site. If you set a user to be a site admin-</td>
</tr>
<tr>
<td>Publisher=(any)</td>
<td>istrator, the other values are ignored.</td>
</tr>
<tr>
<td>License level=Interactor</td>
<td>Publisher</td>
</tr>
<tr>
<td>Administrator=None</td>
<td>Interactor</td>
</tr>
<tr>
<td>Publisher=true</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>License level=Viewer</td>
<td>Viewer</td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=false</td>
<td></td>
</tr>
<tr>
<td>CSV settings</td>
<td>Site role</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>License level=Unlicensed</td>
<td>Unlicensed (can publish)</td>
</tr>
<tr>
<td>Administrator=None Publisher=true</td>
<td></td>
</tr>
<tr>
<td>License level=Unlicensed</td>
<td>Unlicensed</td>
</tr>
<tr>
<td>Administrator=None Publisher=false</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

- If you are importing users while managing the server, you can create users with only two site roles: system (server) administrator and Unlicensed. All other settings are site specific. In that case, if the administrator level for a user in the CSV file is not System, the user’s site role is set to Unlicensed.

- If you have a user-based server installation, and if adding users would exceed the number of users allowed by your license, the users are added as unlicensed users.

**Example**

The following example shows a CSV file that contains information for several users.

```plaintext
henryw,passw0rd,Henry Wilson,Interactor,None,yes,henryw@example.com
freds,pa$$word,Fred Suzuki,Viewer,None,no,freds@example.com
alanw,p@ssword,Alan Wang,Interactor,Site,yes,alanw@example.com
michellek,mypassword,Michelle Kim,Interactor,System,yes,michellek@example.com
```

If you import this file while managing a site, four users are added to that site. The Administrator mode for user Michelle is set to System. However, because you are importing the users into a site, Tableau Server sets user Michelle to be a site administrator, not a system administrator. Three of the users are allowed to publish.

If you import this file while managing the server, four users are added to the server, but they are not added to any site. The site roles in the CSV file (Interactor and Viewer) must be associated with site users, so the site role for the users who are not administrators is set to Unlicensed.
View, Edit, and Delete Users

View and edit site users

Sign in to a site as an administrator, and then click **Users**. In this page you can set group membership, set site role, or remove the user from the site.

Click a user name to see the content they own.

Click **Settings** in a user page to view their account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.
If Tableau Server is running multiple sites, **Server Users** lists all users on the server system, and **Site Users** displays all users for the current site.

If the server is configured to use the internal user management system (Local Authentication), you can edit the **Display Name, Email, and Password** for users after they have been added. If you are making many changes, you may find it easier to import the changes from a CSV file. For details, see **Import Users** on page 804 and **CSV Import File Guidelines** on page 810.

**For multi-site servers**: Site administrators can edit an existing user’s account as long as the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings. For example, if User Joe is a member of Site A and Site B and the site administrator is only an administrator of Site B, the site administrator cannot edit Joe’s Full Name or reset his password.

**View and edit server users**

Sign into Tableau Server as a server administrator. On the site menu, click **Manage All Sites**, and then click **Users**. In this page you can set site membership or delete the user from the server.
Click a user name to view account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.

![User Settings](image)

**Search for users**

To search for a specific user, in the **Search** box on the left, type all or part of the user’s name, and then press **Enter**.

The search operation checks the display name and user name attributes.

You can use the asterisk (*) character as a search wildcard. For example, searching for John* will return all user names that start with John.
Remove users from a site

You can remove a user from a site only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but not removed.

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

1. In a site, click **Users**. Select one or more users to delete, and then select **Actions > Remove**.

2. Click **Remove** in the confirmation dialog.
Delete users from the server

You can delete a user from Tableau Server only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to delete a user who owns content, the user site role will be set to Unlicensed, but the user will not be deleted.

If a user is a member of multiple sites, and owns content in one or more of those sites, the user will be removed from the sites in which they don't own content. The user will remain a member in sites where they do own content, but demoted to the Unlicensed site role.

1. In the site menu, click Manage All Sites, and then click Users. In a single-site environment, click Users.

   Select one or more users to delete, and then click Actions > Delete.

   ![Servers Users with Delete option]

2. Click Delete in the confirmation dialog box.

Change passwords for users of a single site

To change the password for a user with membership to a single site, sign in to Tableau Server as a site administrator or a server administrator.

1. Ensure that the correct site is selected in the menu.
2. Click Users.
3. Click the display name of a user.
4. Click Settings.
5. Click the **Change Password** link, edit the password, and then click **Save Password**.

**Change passwords for users of multiple sites**

To change the password of a user with membership to multiple sites, sign in to Tableau Server as a server administrator.

1. In the site menu, click **Manage All Sites**.
2. Click **Users**.
3. Click the display name of a user.
4. Click the **Change Password** link, edit the password, and then click **Save Password**.
Groups

You can organize Tableau Server users into groups to make it easier to manage multiple users. You can either create groups locally on the server or import groups from Active Directory.

To keep Active Directory group membership up-to-date:

- Site administrators can synchronize selected groups on demand in a site. For more information, see Synchronize Active Directory Groups on a Site.
- Server administrators can synchronize all Active Directory groups on the server based on a schedule or on-demand. For more information, see Synchronize All Active Directory Groups on the Server.

You can also assign permissions to a group for a project, workbook, view, or data source. For details, see Manage Permissions on page 834.

The All Users group exists in every site by default. Every user added to the server becomes a member of the All Users group automatically. You cannot delete this group, but you can set permissions for it.

Create a Local Group

Local groups are created using the Tableau Server internal user management system. After you create a group you can add and remove users.
1. In a site, click **Groups**, and then click **New Group**.

![Create a new group](image)

2. Type a name for the group and click **Create**.

### Add Users to a Group

One way to simplify user management is to assign users to groups. For example, you can assign permissions to a group to apply them to all users in the group.

To add a user to a group, the group must already exist. For information, see **Groups** on page 821.

**Add users to a group (Users page)**

1. In a site, click **Users**.
2. Select the users you want to add to a group, and then click **Actions** > **Group Membership**.
3. Select the groups and then click Save.

Add users to a group (Groups page)

1. In a site, click Groups, and then click the name of the group.
2. In the group's page, click Add Users.
3. Select the users to be added, and then click **Add Users**.

**Create a Group via Active Directory**

When you import Active Directory groups, a matching group is created on the server and a user is created on the server for each member of the group that is not already on the server.

Each user is assigned a site role as part of the import process. If a user already exists on the site without a group affiliation, the user is added to the group with the assigned site role, and the same permissions in the site.

Before importing groups, be sure to review **User Management in Active Directory Deployments** on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

**Note:** Importing users and groups will promote a user's site role, but never demote a user's site role. If any of the users to be imported already exist in Tableau Server, the site role assigned during the import process will be applied only if it gives the user more access to the server. For more information, see **Site Roles for Users** on page 788.
1. In a site, click **Groups**, and then click **Add Groups**

2. Type the name of the Active Directory group you want to import, and then select the group name in the resulting list.

3. Select the site role for the users.
4. Click **Import**.

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

**Quick Start: Synchronize All Active Directory Groups on a Schedule**

After you import Active Directory groups in Tableau Server, you can make sure they stay synchronized in Tableau Server by setting up a schedule. You can also synchronize all Active Directory groups on the server on-demand, at any time. The minimum site role setting for the group is applied when users are synchronized.

**Note:** To use this feature, your Tableau Server installation must be set up for Active Directory.

1. **Set a minimum site role for synchronization**

In a site, click **Groups**. Select a group, and then click **Actions > Minimum Site Role**. Select the minimum site role, and then click **Change Site Role**. Server and site administrators can set the minimum site role for group users to be applied during Active
Directory synchronization. If you don't set a minimum site role, new users are added as *Unlicensed*.

Synchronizing can promote a user's site role, but will never demote a user's site role.

2 Set the schedule

Server administrators can enable synchronization for all Active Directory groups on the **General** tab of the **Settings** page for the server. Enable synchronization, select the frequency settings, and then click **Save**.
All Active Directory groups on the server are synchronized according to the same schedule.

3 Run synchronization on-demand (optional)

On the General tab of the Settings page, click Synchronize All Groups to synchronize all Active Directory groups on Tableau Server immediately. Click this button at any time to ensure new users and changes are reflected in all Active Directory groups on the server.

Click Synchronize All Groups to synchronize all Active Directory groups on the server outside of a schedule.
4 View the status of synchronization tasks

Server and site administrators can view the results of Active Directory synchronization jobs in the **Background Tasks for Non Extracts** administrative view. On the server or in a site, click **Status**. Under **Analysis**, click **Background Tasks for Non Extracts** and filter on the **Queue Active Directory Groups Sync** and **Sync Active Directory Group** tasks.

Queue Active Directory Groups Sync queues the Sync Active Directory Group tasks to be run.

**Synchronize Active Directory Groups in a Site**

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

1. In a site, click **Groups**.
   
   On the Groups page, select one or more groups.
2. Click Actions > Synchronize.

Set the minimum site role for users in an Active Directory group

In the Groups - Details page, administrators can set the minimum site role for group users to be applied during synchronization.

This setting does not run synchronization; it sets the minimum site role to applied to the group every time synchronization runs. When you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role will be applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as Unlicensed by default.

**Note:** A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization on page 791.

1. In a site, click Groups.
2. On the Groups page, select a group.

   Click Actions > Minimum Site Role.
3. Select the minimum site role, and then click **Change Site Role**.

What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user’s content before removing the user’s account completely. For more information, see **Sync behavior when removing users from Active Directory** on page 1253.

What happens when an Active Directory group is removed from Tableau Server?

Many Tableau administrators use Active Directory groups to import and create users. After the users are imported into Tableau Server, administrators will then delete the group in Tableau Server. Deleting a group does not delete the users in it.
Synchronize All Active Directory Groups on the Server

As a server administrator, you can synchronize all Active Directory groups on a regular schedule or on-demand on the General tab of the Settings page for the server.

The Last synchronized time indicates the time that synchronization most recently began.

Synchronize Active Directory groups on a schedule

1. **Single-site**: Click Settings > General.

   **Multisite**: In the site menu, click Manage All Sites and then click Settings > General.

2. Scroll down the page to Active Directory Synchronization, and then select Synchronize Active Directory groups on a regular schedule.
3. Select the frequency and time of synchronization.

4. Click Save.

Synchronize all Active Directory groups on demand

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. **Single-site**: Click Settings > General.

   **Multisite**: In the site menu, click Manage All Sites, and then click Settings > General.

2. Under **Active Directory Synchronization**, click Synchronize All Groups.
View synchronization activity

You can view the results of synchronization jobs in the Background Tasks for Non Extracts administrative view. Queue Active Directory Groups Sync is the task that queues and indicates the number of Sync Active Directory Group tasks to be run.

1. **Single-site:** Click Status.
   - **Multisite:** In the site menu, click Manage All Sites and then click Status.
2. Click the Background Tasks for Non Extracts link.
3. Set the Task filter to include Queue Active Directory Groups Sync and Sync Active Directory Group.

You can quickly navigate to this administrative view by clicking the View synchronization activity link in the Settings page for the server.

Set the minimum site role for users in an Active Directory group

In the Groups - Details page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as Unlicensed by default.

**Note:** A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization on page 791.

1. In a site, click Groups.
2. On the Groups page, select a group.
   - Click Actions > Minimum Site Role.
3. Select the minimum site role, and then click **Change Site Role**.

What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see **Sync behavior when removing users from Active Directory** on page 1253.

**Delete Groups**

You can delete any group from the server (with the exception of the All Users group). When you delete a group, the users are removed from the group but they are not deleted from the server.

1. In a site, click **Groups**.
2. On the Groups page, select one or more groups to delete.
3. Select **Actions > Delete**.

![Tableau interface showing actions](image)

**Data Sources**

A Tableau data source consists of metadata that describes the following:

- **The data connection information** that describes what data you want to bring in to Tableau for analysis.

- **Customization and cleanup** that helps you and others use the data source efficiently. For example, calculations, sets, groups, bins, and parameters, custom field formatting, and so on.

- **Information about how to access or refresh the data**, such as a path to an Excel file, credentials for accessing data on-premises or in the cloud, and so on.

**Sharing data sources**

After you create and customize a data source that you want others to use for their Tableau analysis, you publish it from Tableau Desktop. After it’s published, your team can connect to it when they create or edit workbooks.

If your data source contains an extract connection, set up a refresh schedule, so that when it is refreshed, workbooks that connect to it show the updates as well.

You can also publish a workbook that contains the data source if you want users to connect to the data source only from that workbook. This is also referred to as an *embedded* data source. Every published workbook has at least one embedded data source.
Managing data sources

As a best practice, administrators should manage data sources on the server. However, both administrators and data source owners can perform management tasks on published data sources.

To perform these tasks (described below), do the following:

1. Sign in to the site or server as an administrator or owner of the data source you want to work with.

   **Note:** Some tasks are available only to administrators, as described below.

2. Go to the Data Sources page, select the check box next to the data source, and in the upper-left of the Data Sources page, select Actions.

- **Edit and view permissions:** Permissions can specify which users or groups can connect to, modify, or download data sources. For information, see Set Permissions for a Data Source on page 851.

- **Edit connection information:** Update embedded credentials or other metadata that describes the connections to the original data.

- **Create a new workbook or download:** You can start a new workbook in the browser environment by connecting to a Tableau data source. Or download the data source to use locally.

- **Change the data source owner**
• **View the data source’s revision history**

• **Refresh extracts:** If a data source includes an extract, you can assign the extract to a refresh schedule. For information, see [Refresh Data on a Schedule on page 885](#).

• **Add or remove keyword tags:** Tags can contain a single word or multiple words, delimited by a comma.

• **Delete:** Deleting a data source affects workbooks that connect to the data source. Before you delete a data source, ensure that there are no workbooks that connect to the data source or edit the workbooks to use another data source.

• **Move:** Only administrators can move a data source from one project to another.

In addition, for data sources that are proxy connections, administrators can stay aware of how users authenticate to the database, and whether the appropriate drivers are installed. For information, see [Database Drivers on page 702](#) and [Data Security on page 958](#).

**See also**

[Best Practices for Published Data Sources](#) in the Tableau Desktop Help

**About Tableau Data Sources**

The Tableau Server data server is a server component that lets you centrally manage and store Tableau Server data sources. A data source is a reusable connection to data. The data can be located either in Tableau’s data engine, as an extract, or in a live relational database. For relational database connections, the information stored in the data source is used for a pass-through connection to the database. The data source can also include customizations you’ve made at the field-level in Tableau Desktop, such as calculations, dimension aliases, groups, or sets.

For administrators, there are many advantages to using Tableau Server data sources. Because one data source can be used by many workbooks, a data source that includes an extract means you save on server space and processing time. Extract refreshes can be scheduled per-extract instead of per-workbook, and when a workbook using a Tableau Server data source is downloaded, the data extract stays on the server, resulting in less network traffic. Finally, if a database driver is required for a connection, you only have to install the driver once, on Tableau Server, instead of multiple times, on all your users’ desktops.

To use the data server, authors connect to data in Tableau Desktop, either by creating an extract or using a connection to a live relational database, and publish the data source to Tableau Server. Once published, these data sources and the server contain everything workbook authors need to quickly connect to data and start authoring. To change a published data source, you download it to Tableau Desktop, make your changes, then republish, overwriting your original. Note that any new members you add to a parameter or any changes
you make to the default sort order are not part of the data source (they are part of the workbook).

If you are running a distributed installation of Tableau Server and expect data sources to be heavily used, there are several ways you can optimize your server deployment. See Distributed Environments on page 694 for more information.

**Note:** To use published multidimensional (cube) data sources, you must download them to Tableau Desktop, so many of the above advantages do not apply. For more information, see *Cube Data Sources* on page 893.

**View Data Sources or Connections**

You can filter the view to data sources or connections.

**The Difference Between Published Data Sources and Embedded Data Sources**

Published data sources contain connection information that is independent of any workbook and can be used by multiple workbooks. An embedded data source contains connection information and is associated with a workbook. Every workbook has one or more embedded data sources. If a workbook uses a published data source, an embedded data source is listed for the workbook.
Identifying Types of Data Sources

The list of data sources gives you information about the data sources and what they are connected to:

![Data Sources Table]

Data sources are distinguished by a number of characteristics in the list:

- **Icon/Name**—The data source icon next to the Name lets you know whether the data source is published (🔗) or embedded in a workbook (🗄).
  - Published data source names are links. Clicking the name of a published data source opens the data source page, with tabs for viewing the data source connections (if any) and connected workbooks.
  - Embedded data source names link to their workbooks. Clicking the name of an embedded data source opens the workbook associated with the data source, with tabs for viewing its data sources.

- **Connection Type** —The connection type gives you information about the type of connection the data source is making. A connection type of **Tableau Server** indicates that the connection is to a published data source. A Tableau Data Extract connection type means that the data source has an extract which is stored in Tableau Server.

- **Connects To**—The Connects To list tells you what the data source is connecting to. This could be a database outside of Tableau Server, an extract, or a published data source.

- **Live or Last Extract**—This column tells you whether the connection to the data is live, or, if it is a connection to an extract, when the extract was last updated.

Identifying Types of Connections

The list of connections gives you information about connections, including the data source they are connected to, connection type, and authentication:
- Connects To—Indicates what the connection’s data source is connecting to. This could be a database outside of Tableau Server, an extract, or a published data source.

- Connection Type —The connection type gives you information about the type of connection the data source is making. A connection type of Tableau Server indicates that the connection is to a published data source. A Tableau Data Engine connection type means that the data source has an extract which is stored in Tableau Server.

### Embedded Data Sources

Every workbook that is published to Tableau Server contains at least one embedded data source. These embedded data sources contain the connection information for the workbook and are listed on the Data Sources page:
By default the list of data sources is filtered to only display published data sources. To view embedded data sources, under **Data Source**, click the drop-down menu and select **Embedded in workbook** to change the filter:
The Difference Between Published Data Sources and Embedded Data Sources

Embedded data sources are different from published data sources in that each embedded data source is associated with a single workbook and describes the attributes required for connecting to a data source (e.g., server name, database name, etc.). That means if you have three workbooks that connect to the same data source, you will still have three embedded data sources listed on the Data Sources page.

Searching for Embedded Data Sources

The Filter area on the left side of the Data Sources page helps you find embedded data sources by connection type, database server name, port, username, password status (whether or not the database password is embedded) and whether or not there is an extract:
Which Connections Can I Edit?

You can edit connection information for live database connections and for extracts that need to be refreshed by Tableau Server. For example, you may have a large number of workbooks that connect to a database on a specific database server. If the name of the server changes, you can update all of the workbooks at once so they reference the new server name. Another example is if a workbook connects to a database using a specific user name and password. You can quickly update all of the workbooks to use a different set of credentials.

For details on how to edit data connections, see Edit Connections on page 896.

Cube Data Sources

Cube (multidimensional) data sources have certain characteristics that make them unique in Tableau.

Cube data sources do not support pass-through connections. This means that when a cube data source is published, you cannot make a connection from Tableau Server using the data source. It also means you cannot create a workbook using the data source in Tableau Server.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally. To download a published data source you need:

- The Download/Web Save As permission for the data source. For more information, see Manage Permissions on page 834 and Set Permissions for a Data Source on
Correct drivers installed and ports opened on computer running Tableau Desktop.

Connect to Published Data Sources

You can publish data sources to Tableau Server or Tableau Online from Tableau Desktop. Publishing data sources to Tableau Server or Tableau Online enables sharing data among colleagues; including those who don't use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

You can connect to these published data sources on Tableau Server or Tableau Online if you have permissions to create and edit views.

**Note:** Many of the topics in this section use the Sample-Superstore data source that comes with Tableau Desktop. To use this data source, you can publish it to Tableau Server or Tableau Online from Tableau Desktop. Follow the procedure in the Publish a Data Source topic in the Tableau Desktop Help to learn more.

If you do not have access to the Sample-Superstore data source, you can connect to your own published data and follow the procedures using similar measures and dimensions from it.

Connect to a published data source on the web

At any time while you're creating or editing a view on Tableau Server or Tableau Online, you can connect to one or more published data sources.

1. Sign in to Tableau Server or Tableau Online and select a view to edit.

2. In editing mode, click the New Data Source icon.

3. In the Connect to Data dialog box, select a published data source from the list, and then click Add.

**Note:** By default, only data sources that have been published to the server are displayed in the list.

You can also connect to a published data source when you’re creating a new workbook on Tableau Server or Tableau Online.
1. Sign in to Tableau Server or Tableau Online.

2. Navigate to the **Content** page and select **Data Sources**.

3. In the list of data sources, select the check box next to the one you want to use, and then click **Actions** and select **New Workbook**.

**Connect to a published data source from Tableau Desktop**

1. On the Connect to Data page in Tableau Desktop, click **Tableau Server**, and then provide the server name and your credentials.

2. Select a data source you want to use.

   The data source opens in the Data pane in the workbook. Tableau Server data sources
show a Tableau icon instead of a database icon.

See also

Edit Connections

Administrators and data source owners can manage the information that describes how a published data source connects to the original data. This information includes the database server, the server port, the database user name, and whether or not the password is embedded in the connection.

Note: From the permissions perspective, whether you can edit connections is evaluated separately from your permissions for editing data sources. Even if you have the Edit capability on a data source, to edit its connections, your site role must be Server Administrator, Site Administrator, or Publisher. If your role is Publisher, you also must be the data source owner.

1. Sign in to the site that has the data sources you want to modify, and open the Data Sources page.

2. Select the name of the data source with the connection you want to update.

   Use the search box or filters on the left to narrow the data source list. The values you type into the Server and Database username fields are treated as regular expressions.

3. In the Connections view, select the check box for the connection, and then click Actions > Edit Connection.
4. Update the connection information.

For **Server name**, if you want to use an IP address, make sure the database or its driver supports that type of connection. If it doesn’t, enter the database server name.

For connections to Google, Salesforce, and web data connector (WDC) data, see [Authentication types for Google, Salesforce, and WDC data](#) on the next page later in this topic.

5. Click **Save**.

6. Refresh the browser page for your changes to take effect.
Authentication types for Google, Salesforce, and WDC data

Google BigQuery, Google Analytics, Salesforce.com, and many web data connector (WDC) connections use the OAuth authentication standard, which uses secure access tokens instead of “raw” user name and password credentials. Database credentials do not need to be stored in Tableau, and all users connect through this access token, including Tableau Desktop users who want to create or edit workbooks that connect to this data source.

The following sections describe Google and Salesforce connection options. Web data connector options vary, but all involve signing in through the provider’s web-based sign-in form to establish the access token.

Google authentication options

When you edit Google BigQuery or Google Analytics connections, select either of the following options in the Edit Connection dialog box:

- Select Embed Google BigQuery (or Google Analytics) credentials in the connection to authenticate through a designated account, and then select an existing account from the list or select authenticate account now... to add a new one.

  When you add a new account, the Google sign-in page appears. After you provide your database credentials, Google prompts you to confirm Tableau access to the data. When you click Accept, Google returns an access token to use for connecting to the data.

  Note: If you create extracts of your Google data source, select this first option, so that you can schedule refresh tasks.

- Select Prompt user for Google BigQuery/Analytics credentials to require users to connect through their own individual access tokens or sign in each time they connect.

Salesforce.com authentication options

When you edit Salesforce.com connections, you can select any of the following options in the Edit Connection dialog box:

- Select Embed a Salesforce username and password to use a traditional authentication method.

- Select Embed Salesforce credentials in the connection to use an OAuth connection and schedule refresh tasks, and then select an existing account from the list or click Add a Salesforce Account to add a new one.

  When you add a new account, the Salesforce.com sign-in page appears. After you provide your database credentials, Salesforce.com prompts you to confirm Tableau access to the data. When you allow Tableau access, Salesforce.com creates an access token through which it connects to the data.
• Select **No Salesforce authentication** to require users to sign in to Salesforce.com each time they connect. (This option does not allow scheduled extract refreshes.)

**Monitor progress**

When you save your changes in the Edit Connection dialog box, the dialog displays the progress. If you close the dialog box, the modifications continue to run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped, but they will not impede other changes. For example, if you try to change the server name and add a password to several connections, the server names will be changed, and the passwords on workbooks will be changed. However, because you cannot add a password to a data source, the passwords for the data sources will not be changed.

For information about checking the progress of these tasks, see *Background Tasks for Extracts* on page 1103.

**Troubleshoot Data Sources**

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

• **Permissions for the data source**: Anyone connecting to a data source must have the **Connect** and **View** permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be
Multidimensional (cube) data sources have to be downloaded and used in Tableau Desktop, so they require **Download/Web Save As** permission. For more information about cubes in Tableau, see **Cube Data Sources** on page 893.

- **Ability to authenticate to the database**: There are several ways you can connect to data in Tableau and control who has access to what. Basically, whichever entity is connecting to the database must be able to authenticate. The entity could be Tableau Server performing an extract refresh. It could be a Tableau Desktop user connecting to a data source that then connects to a live database. It could also be a Tableau Server user who’s accessing a view that connects to a live database. Refer to **Data Security** on page 958 to learn more about your options.

- **Database drivers**: If the person who created and published the data source in Tableau Desktop needed to install additional database drivers, you may need to install them on Tableau Server as well. If you are running a distributed installation of Tableau Server where, for example, the data server process is running on a worker server, any required database drivers must be installed there as well as on the primary server. Other processes require drivers as well. See **Database Drivers** on page 702 for more information.

### Data Source Error Messages

Here are some errors that workbook authors and other users may encounter as they work with data sources and views:

- **Permission to access this Tableau Server data source denied**: Connecting to a data source requires the Connect permission. See **Manage Permissions** on page 834 and **Set Permissions for a Data Source** on page 851 for more information.

- **Data source not found**: Someone working with a view may see this error if a data source is removed from Tableau Server or if their Connect to Data page needs to be updated. To update the Connect to Data page in Tableau Desktop, click the Refresh icon:
Unable to connect to this Tableau Server data source: This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct settings.

Unable to list Tableau Server data sources: This error may occur if a user is trying to access Tableau Server data sources and there are connectivity issues between Tableau Server and Tableau Desktop.

Can’t connect with a cube data source: To use a published multidimensional (cube) data source, you must download the data source and use it in Tableau Desktop. Verify that you have the Download/Web Save As permission for the data source. For more information about cubes in Tableau, see Cube Data Sources on page 893.

Refresh Data on a Schedule

As a server administrator, you can allow users to subscribe to views published to the server, or to schedule tasks for refreshing published data extracts or the subscription email deliveries.

You can also specify which other users are allowed to set schedules. Otherwise, non-administrator users can work with schedules in the following ways:

- Tableau Desktop publishers can set scheduled refresh tasks when they publish a data source or a workbook with a data extract.
- Tableau Server users can subscribe to views that are delivered by email on a schedule.

Changes to an existing schedule, as well as new schedules you create on the server, are reflected in the publishing steps in Tableau Desktop the next time the author publishes content. Similarly, changes to a subscription schedule are reflected in the choices a server user has when subsequently subscribing to a view.

Quick Start: Refresh Extracts on a Schedule

For published workbooks that connect to data extracts, you can set up the server to refresh the data on a recurring schedule, so all workbooks connected to them always show the most up-to-date data.

To schedule refreshes you need to have administrator or data owner permissions.

1 Set up a schedule on the server

Sign in to the server, go to the Schedules page, and click New Schedule.
Tableau provides a few refresh schedules. You create additional schedules you need.

2 Enable scheduled extract refreshes and failure emails

As a server or site administrator, you can enable schedules, as well as email notification when extract refreshes fail.

Select **Settings**, and then go to the **General** page.

- Under Email Notification, select **Send email to data source and workbook owners when scheduled refreshes fail**.
- Under **Embedded Credentials**, select both check boxes to allow publishers to embed credentials and schedule extract refreshes.
3 Publish a workbook with an extract

In Tableau Desktop, select **Server > Publish Workbook**. Sign in to the server if you’re not already. In the **Publish Workbook to Tableau Server** dialog box, click **Schedules & Authentication**. Under **Extract Schedule**, select the schedule from the list.

If the original data requires authentication, you will also need to select how you want people to access it.

4 Monitor refresh performance

You can monitor scheduled tasks by viewing **Background Tasks for Extracts** on the **Status** page.
About Extracts and Schedules

Tableau Desktop authors can create data extracts, which are copies or subsets of data from the original data sources. Workbooks that use data extracts are generally faster than those that use live database connections because the extracted data is imported into the Tableau data engine. Extracts can also increase functionality. After an author publishes a workbook or a data source with an extract, the extract resides on Tableau Server.

Refreshing extracts on Tableau Server

You can use Tableau Server to refresh extracts on a schedule. Both server and site administrators can create, change, and reassign schedules. However, only a server administrator can enable scheduling. Any scheduling changes made in Tableau Server are reflected in the Schedule dialog box in Tableau Desktop when the workbook or data source is published again.

Schedules that you create have the following options:

Priority

The priority is a number which determines the order in which refresh tasks are run, where 0 is the highest priority and 100 is the lowest priority. The priority is set to 50 by default.

Execution mode

The execution mode determines how schedules are run by the Tableau Server backgrounder processes and can be set to parallel or serial. When you run a schedule in parallel, it runs on all available backgrounder processes, even if the schedule only contains one refresh task. When you run a schedule serially, it only runs on one backgrounder process. By default, the execution mode is set to parallel so that refresh tasks finish as quickly as possible. However, you may want to set the execution mode to serial if you have a very large schedule that is preventing other schedules from running because it is using all the available backgrounder processes.

Frequency.

The frequency determines how often a schedule is run. You can set the frequency to hourly, daily, weekly, or monthly. To run a schedule immediately, select the schedule and click Actions > Run Now.

Note: You can also refresh extracts from the command line using the tabcmd refreshextracts command. For more information, see tabcmd Commands on page 1319.

Refreshing extracts from Tableau Desktop

- At publish time: When an author publishes a workbook or data source that uses an extract, that author can assign it to a recurring refresh schedule on Tableau Server. The
refresh can be a full refresh or an incremental refresh. Incremental refreshes reference a column in the extract that has a data type of date, date/time, or integer; such as a timestamp. Tableau uses this column to identify new rows that need to be added to your extract. See Refreshing Extracts and Schedules in the Tableau Desktop help for more information.

- **User interface:** You can use the Refresh from Source, Add Data From File, and Add Data From Data Source options in Tableau Desktop to upload an addition to or refresh an extract on Tableau Server. You may want to do this if Tableau Server doesn't have sufficient credentials to refresh data from the original data source. See Updating Extracts on Tableau Server in the Tableau Desktop online help for details on how to upload.

- **Data Extract command line utility:** The Data Extract command line utility installs with Tableau Desktop. You can use it to upload an addition to an extract on Tableau Server or refresh it. See Tableau Data Extract Command Line Utility in the Tableau Desktop online help for more information on how to upload.

**See also**

Enable Extract Refresh Scheduling and Failure Notification on page 911

Automate Refresh Tasks on page 922.

**Manage Refresh Tasks**

Administrators can change the priority or schedule of a scheduled refresh, a refresh manually, or delete schedules. You can do this on the Tasks page:

1. Sign in to the site that has the schedules you want to manage, and then click Tasks.
2. On the Tasks page, do any of the following:
   - Click **Change Schedule**, and select a new schedule from the list.
   - Select **Run Now** to initiate the refresh manually.
     
     **Note:** If an extract does not have a scheduled refresh, you can refresh it on demand from the Data Connections page.
   - Select **Change Priority**, and enter a new number between 1 and 100 to move the extract up or down in the priority list.
Select **Delete** to completely remove the schedule for the selected data sources.

See also

*Enable Extract Refresh Scheduling and Failure Notification* on page 911

**Quick Start: Manage Incremental Extracts**

When you publish a workbook that has an incremental extract, you can associate it with up to two refresh tasks that Tableau Server will handle for you: An incremental refresh of the extract and a full refresh. After you publish the workbook, you or a Tableau Server administrator can modify any tasks that are associated with the workbook. You can also delete tasks or add more.

1 **Publish and Assign a Schedule**

In Tableau Desktop, after you create a workbook that uses an extract, go to **Server > Publish Workbook**, and click **Scheduling & Authentication**. Next, choose schedules for your refreshes and click OK.
After you publish in Tableau Desktop and choose your refresh schedules, Tableau Server handles the refresh tasks for you.

2 Select the Workbook

To modify a workbook’s scheduled task, sign in to Tableau Server and on the **Workbooks** page, select the workbook:
3 Access the Refresh Schedule

Click Refresh Schedule.

Select the check box for the refresh task you want to modify:
4 Edit, Delete, or Add More Tasks

Select the action you want to take—for example, Change Schedule—and make your selection. You can also delete the task, change its priority, or add more refresh tasks.

![Change Schedule]

Add a Workbook or Data Source to a Schedule

You can set scheduled refresh tasks for published data source extracts and published workbooks that connect to data extracts.

1. When you’re signed in to Tableau Server, display Content > Data Sources or Content > Workbooks, depending on the type of content you want to refresh.
2. Select the check box for the data source or workbook you want to refresh, and then
select **Actions > Extract Refresh**.

3. In the Refresh Extracts dialog, select **Schedule a Refresh**, and complete the following steps:
   - Select the schedule you want.
   - If available, specify whether you want a full or incremental refresh.
     By default, and if this option is not shown, a full refresh is run. Incremental refresh is available only if you configured it in Tableau Desktop before publishing the extract. For information, see **Refreshing Extracts** in the Tableau Desktop Help.
   - Click the **Schedule Refresh** button.

   ![Refresh Extracts dialog]

   **Note:** If you want to add a new schedule, you can do so on the **Schedules** page.

**Automate Refresh Tasks**

You can associate extract refresh tasks with schedules in Tableau Server to automate refreshing data extracts. You can also automate extract refreshes using tabcmd, a command line utility that comes with Tableau Server and can be installed on a separate computer from Tableau Server. In particular, you can use the `refreshextracts` command in combination with other commands in your own script. For example:

```
tabcmd login - http://mytabserver -u jsmith -p P@ssw0rd!
refreshextracts --datasource salesq4
```
Handle Extract Refresh Alerts

When Tableau Server cannot complete a scheduled refresh, an alert appears to indicate that the refresh has failed. If a scheduled refresh fails five consecutive times, Tableau Server suspends the refresh. When a refresh is suspended, Tableau Server does not try to run it again until someone takes an action that attempts to correct the cause of the failure.

**Note:** The number of consecutive failures for a refresh is set to five by default, but can be changed by a Tableau Server administrator, using the `tabadmin set backgrounder.failure_threshold_for_run_prevention` command. For more information, see `tabadmin set options` on page 1294.

You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn’t be refreshed
- The author of a workbook that connects to a data source that couldn’t be refreshed

When you open the Alerts menu you can see more information about the refresh failure(s):

When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.
When a data source name or workbook name is listed as the **Data source** (for example, **Data source: sales_data**), it means that the data source is a Tableau Server data source. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon:

<table>
<thead>
<tr>
<th>Data</th>
<th>Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚫ Population</td>
<td></td>
</tr>
<tr>
<td>⮞ Sales by Region</td>
<td></td>
</tr>
</tbody>
</table>

**Resolving Extract Refresh Problems**

To resolve refresh issues, you can take any of these actions, based on the cause indicated in the alert:

- **Errors related to access token validation or user credentials**
  
  You can resolve some extract refresh problems by clicking the **Connection Details** in the alert. Select the check box next to the problematic data source, click **Actions > Edit Connection**, and then enter the missing information. Click **Save** when you're done. After you update the connection information, Tableau Server restarts the refresh schedule.
  
  If you originally embedded the credentials or other data connection information when you published the workbook or data source from Tableau Desktop, you can also republish the workbook or data source. As part of the publishing process, you can choose to set a new refresh schedule. If you don’t choose a new schedule, Tableau Server restarts the existing schedule.

- **Errors that indicate the database was unreachable**
  
  Confirm that the database is online and that you can sign in to access the data. You can use the **Try again** link in the alert to restart the refresh schedule.
  
  If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

**Tip:** Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections.
Background Task Prioritization

Note: This topic only covers prioritization of background tasks for extract refreshes and schedules.

Scheduled extract refreshes and subscriptions are run in this order:

1. Any task that is already in process is completed first.
2. Any task that is manually **Run Now** will start when the next backgrounder process becomes available.
3. Tasks with the highest priority (the lowest number) start next, independent of how long they have been in the queue. For example, a task with a priority of 20 will run before a task with a priority of 50, even if the second task has been waiting longer.
4. Tasks with the same priority are executed in the order they were added to the queue. The first task added to the queue will be started first and the second task added will be started next.
5. When multiple tasks with the same priority are scheduled to run at the same time, they are started in the order they were created or enabled. There is no distinction between extract refreshes and email subscriptions.

The following limitations also impact when scheduled tasks run:

- Tableau Server can only run as many concurrent tasks as there are backgrounder processes configured.
- Separate extract refreshes for the same data cannot run at the same time.
- Tasks associated with a schedule that uses serial execution must run one at a time.

Maintain a History of Revisions

Revision history allows you to keep copies of content (workbooks and data sources) that have been changed. When revision history is enabled, each time a user saves updates to the content, either from Tableau Desktop or on Tableau Server, Tableau Server creates a new version of that content and stores it with older versions. You and authorized users can view and restore older versions of the content. By enabling revision history, you give users (and yourself) the confidence to experiment with the content, knowing that older versions are always available.

When revision history is enabled on a site and users publish or save workbooks, or publish data sources, a revision of each workbook and data source is saved, and users with the necessary permissions can access revision history.
• **Workbooks.** Users can preview, delete, or download previous versions. Published workbooks can be restored online, in Tableau Server, or can optionally be downloaded and then republished. For more information, see *Manage Workbook Revisions* on page 945.

• **Data Sources.** Users can delete or download previous versions. To restore a previous version of a published data source, users view the revision history for the data source in Tableau Server, download a revision, and then republish the data source to the same location, using the same name. For more information, see *Manage Data Source Revisions* on page 950

**Required permissions**

To access revision history, a user must have a site role of **Publisher**, plus the following permissions:

- Project: **View** and **Save**
- Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**
- Data sources in the project: **View**, **Save**, and **Download Data Source**

**Administrator control of revision history**

Server administrators can enable revision history for all workbooks and data sources in a site, on a per-site basis. Revision history is enabled by default, and the default number of revisions saved in history for each resource is 25.

Server administrators can set a limit the number of versions stored in revision history, and can also clear all revisions for every workbook and data source on a site. The most recent revision of each published workbook and data source is always retained.

When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if you set the limit to 15, the 15 most recent versions of the workbook or data source are saved.
More about revision history

- If a different author publishes over a workbook or data source with the same name, the most recent author becomes the owner of the content and can see its entire revision history.

- Workbooks and data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook or data source.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a workbook or data source is deleted from a site, all previous versions are also deleted.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

Security for previewing and restoring workbooks

When users click Restore or Preview for workbook revisions, user passwords are exchanged between the Tableau Server browser client and the server back end. Tableau Server encrypts these passwords using public/private key encryption, but to ensure these public keys are provided by Tableau Server, you must configure Tableau Server to use SSL (HTTPS). For more information, see SSL on page 969 in the Server Administrator Guide.

About restoring workbooks that require credentials

When you restore a workbook that uses a live connection and prompts for a username and password, you have the option to embed the credentials for the connection. If the workbook uses a data source with multiple connections, you may need to provide a user name and password for each connection that prompts for credentials.

Extracts with embedded credentials and scheduled refreshes

When you restore an extract that uses embedded credentials and scheduled extract refreshes, you will need to edit the data source connection and provide your credentials as part of the restore process. This ensures that the workbook has the credentials it needs when the refresh task runs.

1. Restore the workbook that uses embedded credentials and has scheduled extract refreshes.

2. Go to the Data Source page for data source used by the workbook.
3. Click the **Connections** tab.
4. For each connection in the data source, select the connection and click **Actions > Edit Connection**.
5. Enter (or keep) the connection information, and then click **Save**.

**Workbooks with OAuth connections**

When you restore a workbook that uses an OAuth connection, you cannot preview the workbook. If you are not able to restore the workbook in Tableau Server, download the workbook and then republish it to make it the current revision.

**Quick Start: Keep Content Revisions**

Every time you publish a workbook or data source to Tableau Server, Tableau can save a version of that content in its revision history. If you want to revert to a previous version, you can go to the workbook or data source in Tableau Server, view its revision history, and restore that version.

**Note:** A server administrator must enable **Revision History** in **Site Settings** to make this feature available.

**1 Publish your content**

In Tableau Desktop, click **Server > Publish Workbook** or **Server > Publish Data Source**. Make changes to the workbook or data source, and then publish it again to the same project, with the same name. Workbook revisions are also saved when a content owner edits and save a workbook in a project on Tableau Server.
Because your content has the same name, you need to confirm that you want to overwrite the workbook or data source when you publish the workbook or data source.

2 View revision history

Sign in to Tableau Server. Select the workbook or data source, and then in the actions menu (...), click Revision History.
Workbooks include the option to preview past versions.

**3 Restore a previous version of a workbook**

In the revision history for the workbook, select the revision, and then click **Restore**.
The restored version becomes the current version.

4 Restore a previous version of a data source

Select and then download the data source. Open the downloaded file in Tableau Desktop, and then republish it with the same name, to the same location in Tableau Server. This makes it the current revision of that data source.
More about revision history

- To access revision history, a user must have a site role of Publisher, plus the following permissions:
  - In the project: View and Save
  - Workbooks in the project: View, Save, and Download Workbook/Save As
  - Data sources in the project: View, Save, and Download Data Source

- Workbooks and data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you might need to make changes in the downloaded workbook or data source.

- You can delete specific revisions from revision history. Server administrators can clear all revisions for every workbook or data source in a site. The most recent revision of each published workbook and data source is always retained.

- When a workbook or data source is deleted from a site, all previous revisions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the data source are saved.

- If revision history has been turned on and then turned off, saved revisions are still retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

- Published workbooks can be previewed and restored online, in Tableau Server, or can optionally be downloaded, opened in Tableau Desktop, and then republished to the same location, using the same name. For more information, see Manage Workbook Revisions on page 945.

  Published data sources must be downloaded, opened in Tableau Desktop, and then republished to be restored. For more information, see Manage Data Source Revisions on page 950.

Manage Workbook Revisions

When revision history is enabled on your site and you publish a workbook from Tableau Desktop or save a workbook on Tableau Server, Tableau Server saves a version of that content in its revision history. Each time you publish or save that workbook again in the same location, with the same name, another revision is saved.

You can restore previous revisions of workbooks online, in Tableau Server. Optionally, you can download a revision, open it in Tableau Desktop, and then republish the workbook to the same location in Tableau Server, using the same name.
Permissions for revision history

To access revision history, you must have a site role of Publisher, plus the following permissions:

- In the project: View and Save
- Workbooks in the project: View, Save, and Download Workbook/Save As

View revision history

- In Tableau Server, select a workbook, and then click Revision History in the actions menu (…).

Preview a revision of a workbook
1. Select a workbook, and then click **Revision History** in the actions menu (. . .).
2. In the revision history, click **Preview** in the actions menu (. . .).

If a preview is available, it is displayed from Tableau Server in a new tab in the browser.

3. Click **Close Preview** in the preview page to return to the **Revision History** dialog box.

**Note:** If a workbook cannot be previewed online, you can download the workbook, and open it in Tableau Desktop to preview it.

**Restore a revision**

1. Select a workbook, and then click **Revision History** in the actions menu (. . .).
2. In the revision history, select a revisions, and then click **Restore**.
The restored version becomes the current version.

**Note**: If the revision can't be restored online, you can download the workbook, open it in Tableau Desktop, and then republish it to make it the current version.

**About restoring workbooks that require credentials**

When you restore a workbook that uses a live connection and prompts for a user name and password, you have the option to embed the credentials for the connection. If the workbook uses a data source with multiple connections, you might need to provide a user name and password for each connection that prompts for credentials.

**Extracts with embedded credentials and scheduled refreshes**

When you restore an extract that uses embedded credentials and scheduled extract refreshes, you will need to edit the data connection and provide your credentials as part of the restore process. This ensures that the workbook has the credentials it needs to complete the refresh successfully.

1. Restore the workbook that uses embedded credentials and has scheduled extract refreshes.
2. Go to the Data Source page for data source used by the workbook.
3. Click the **Connections** tab.
4. For each connection in the data source, select the connection and click **Actions > Edit Connection**.
5. Enter (or keep) the connection information, and then click **Save**.

**Workbooks with OAuth connections**

When you restore a workbook that uses an OAuth connection, you cannot preview the workbook. If you are not able to restore the workbook in Tableau Server, download the workbook and then republish it to make it the current revision.

**Download a revision**

1. Select a workbook, and then click **Revision History** in the actions menu (…).
2. In the revision history, click **Download** in the actions menu (…).
3. Open the workbook file in Tableau Desktop.
   You can also republish the with the same name, in the same location in Tableau Server, to make it the most current version.

**Note:** When you publish the workbook from Tableau Desktop, because your content has the same name, you will have to confirm that you want to overwrite the workbook. Proceed by clicking Yes.

**Delete a revision from history**

You can delete any previous version of a workbook from its revision history. You cannot delete the most current revision except by deleting the entire workbook.

1. Select a workbook, and then click Revision History in the actions menu (…).
2. In the revision history, click Delete in the actions menu (…).
The revision history list updates to indicate a revision has been deleted.

![Revision History]

**More about revision history**

- If a different author publishes over a workbook with the same name, the most recent author becomes the owner and can see its entire revision history.

- Workbooks are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a workbook is deleted from a site, all previous versions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the workbook are saved.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

**Manage Data Source Revisions**

When revision history is enabled on your site and you publish a data source, Tableau Server saves a version of the data source in its revision history.

To restore a published data source, you download it, open it in Tableau Desktop, and then republish it to the same location in Tableau Server, with the same name.
Permissions for revision history

To access revision history, you must have a site role of Publisher, plus the following permissions:

- In the project: View and Save
- Data sources in the project: View, Save, and Download Data Source

View revision history

- Select a data source, and then click Revision History in the actions menu (..).

Download and restore a revision

1. Select a workbook or data source, and then click Revision History in the actions menu (..).
2. In the revision history, click Download in the actions menu (..).
3. Open the data source file in Tableau Desktop, and then republish it with the same name, to the same location in Tableau Server. The uploaded version becomes the most current version.

**Note:** When you publish from Tableau Desktop, because your content has the same name, you will have to confirm that you want to overwrite the data source. Proceed by clicking **Yes**.

**Delete a revision from history**

You can delete any previous version of a data source from its revision history. You cannot delete most current revision except by deleting the entire data source.

1. Select a data source, and then click **Revision History** in the actions menu (…).
2. In the revision history, click **Delete** in the actions menu (…).
The revision history list updates to indicate a revision has been deleted.

More about revision history

- If a different author publishes over a data source with the same name, the most recent author becomes the owner and can see its entire revision history.

- Data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook or data source.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a data source is deleted from a site, all previous versions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the data source are saved.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

Administrative Views

The Status page contains an embedded Tableau workbook with various administrative views. These views help you to monitor different types of server or site activity.

- Shows server and site activity for Tableau Server.

- Shows site activity for Tableau Online.
Navigating to administrative views

To see administrative views, click **Status**. Site administrators can see administrative views for their site. Administrators of multiple sites can see views for the current site.

On a multi-site server, server administrators can see views for the entire server. Click the site menu, and then click **Manage All Sites** to access the server menus.

To see views for individual sites on a multi-site server, click the site menu, select the site name, and then click **Status**.

Traffic to Views

The Traffic to Views view gives you the ability to see how much of your user traffic goes to views.

You can filter what information is displayed and the time frame it comes from by selecting the view, the workbook, and the time range. Server administrators can specify the site.
Two time lines at the top of the view show you how views are being used over a time range you specify (the default is the last 7 days):

- **What is the Total View Count by Day**—This shows total view count by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count of views. Select the point to update the other sections of the view based on your selection.

- **What is the Total View Count by Time**—This shows the view count by time of day. The filters and any selection impact this graph.

Two bar graphs at the bottom of the view show results that are filtered by the **Min View Count** filter at the top of the view. These show you the views that are most often accessed, and the users who most frequently access views Only those views and users with counts greater than or equal to the minimum view count value are displayed:

- **What Views are Seen the Most**—This is a list of the most visited views. Like the other sections of the view, the information is limited by filters and any selection you make.

- **Who Accesses Views Most Often**—This shows the users who most often access the views and is limited by filters and any selection you make.

**Traffic to Data Sources**

The Traffic to Data Sources view gives you the ability to see usage of data sources on your Tableau Server installation. This can help you determine which data sources are most heavily used and those that are less often used. You can filter the information you see by selecting the data source, the action taken on that data source, and the time range. Server administrators can specify the site.
A time line at the top of the view shows you how data sources are being used over a time range you specify (the default is the last 7 days):

- **What is the Total Data Source Usage by Day**—This shows total data source usage by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count. Select the point to update the other sections of the view based on your selection.

Two bar graphs at the bottom of the view show results that are filtered by the **Min Interactions** filter at the top of the view. These show you which data sources are most used, and who uses data sources most often. Only those data sources and users with interaction counts greater than or equal to the minimum interactions value are displayed:

- **What Data Sources are Used Most**—This is a list of the most used data sources. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Uses Data Sources Most Often**—This shows the users who most often use the data sources. This is impacted by filters and any selection you make.

**Actions by All Users**

The Actions by All Users view gives you insight into how your Tableau Server installation is being used. You can filter the view by actions and by time range. Server administrators can filter by site. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of active users who have performed one of the selected actions.
Up to three separate groups of time lines show you how users are using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or Ctrl + Click on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

**Actions by Specific User**

The Actions by Specific User view gives you insight into how individual users are working in your Tableau Server installation. You can filter the view by user name, actions, and time range. Server administrators on multi-site installations can filter by site.
Up to three separate groups of time lines show you how a selected user is using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, or if no actions were taken, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

A bar graph at the bottom of the view shows which items the selected user is using.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

**Actions by Recent Users**

The Actions by Recent Users view shows you which signed-in users have been active on Tableau Server recently. This can be useful if you need to perform some maintenance activity and want to know how many and which users this will affect, and what they are doing on Tableau Server.

The view **Active, Recently Active**, and **Idle** users that are currently signed in to Tableau Server. For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last
took an action more than 30 minutes ago. The actions are displayed in the lower section of the view.

Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.

**Background Tasks for Extracts**

The Background Tasks for Extracts view displays extract-specific tasks that run on the server.
A table lists the extracts that ran in the time period specified in Timeline. Click Success or Error to filter the table based on status. Click a specific task to update the How Much Time did Extracts Take graph for the selected task. The How Many Extracts Succeeded or Failed table updates for the status (success or failure) of the task, but the count of extracts that succeeded or failed does not change.

Tasks can have a status of successful or error:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Error" /></td>
<td>Error—Server was unable to complete the task.</td>
</tr>
<tr>
<td><img src="image" alt="Success" /></td>
<td>Success—Server completed the task.</td>
</tr>
</tbody>
</table>

For details on a task, hover over its icon:
Background Tasks for Non Extracts

The Background Tasks for Non Extracts view displays tasks that the server runs that are not related to refreshing extracts. For example, edited OAuth connections, subscription notifications, and so on.

A table lists the tasks that ran in the time range specified. Click Success or Error to filter the table based on status. Select a specific task in the How Many Tasks Succeeded or Failed on this Site table to update the What Background Tasks Ran on this Site graph for the selected task.

Tasks can have a status of successful or error.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error—Server was unable to complete the task.</td>
<td></td>
</tr>
<tr>
<td>Success—Server completed the task.</td>
<td></td>
</tr>
</tbody>
</table>

For details on a task, hover over its icon.

Embed Views into Webpages

You can embed interactive Tableau views and dashboards into web pages, blogs, wiki pages, web applications, and intranet portals. Embedded views update as the underlying data changes, or as their workbooks are updated on Tableau Server. Embedded views follow the
same licensing and permission restrictions used on Tableau Server. That is, to see a Tableau view that’s embedded in a web page, the person accessing the view must also have an account on Tableau Server. As an alternative, if you have a core-based license you can select Enable Guest account, which allows users to load the view without signing in.

You can embed views the following ways:

- **Use the Share embed code:** The Share button at the top of each view provides embed code that you can copy and paste into your webpage. (If you don’t see the Share button, ask your administrator if they’ve used the `tabadmin` command to change the `vizqlserver.showshare` option.)

- **Write your own embed code:** You can enhance the embed code that Tableau provides, or you can build your own code. Either way you can use parameters that control the toolbar, tabs, and more.

**Note:** For users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

**Writing Embed Code**

If you’re writing your own embed code, you can take one of two approaches:

- **Use Tableau JavaScript:** This is the preferred approach. Use the embed code that Tableau generates as the starting point for your own code, adding or editing object parameters that control the toolbar, tabs, and more. The default embed code, which relies on a Tableau JavaScript file, is also the only way to control the load order of multiple embedded views.

- **Specify the View URL:** Embed a view using an Iframe or Image tag, where the source is the URL from the Link box of the Share dialog box. You may want to do this if you can’t use JavaScript on your website. There may also be situations when all you can specify is an URL—such as if you’re embedding a view using the SharePoint Page Viewer Web Part.

When you embed a view, you should define a width and height that the view will be displayed in. If you do not do this, the client browser will arbitrarily pick a width and height.

**Tableau JavaScript**

The following code shows an example of embed code that is generated when you click Share on a published view. Special characters in the `host_url` parameter are URL encoded, and those in the `site_root` and `name` parameters are notated as HTML numeric character references.
The source for the `<script>` tag is the URL for the Tableau Server JavaScript file, `viz_v1.js`. The JavaScript file handles assembling the full URL of the view that’s displayed for your users. The `name` and `site_root` object parameters are the only required parameters; all other parameters are optional.

**View URL as the Source**

Here’s an example of embedding the same view using an IFrame, where the source is the URL from the **Link** box of the **Share** dialog box:

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

The `embed` URL parameter is required, and you can optionally include parameters that control the toolbar and revert options, among others. You can also add filters to the URL that control the specific data that shows when a view is loaded.

**See also**

For examples, see the **Parameters You Can Use to Embed a View** below and search for "Script Tag Examples" in the Tableau Server Help.

**Parameters You Can Use to Embed a View**

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.
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
</table>
| customViews      | :customViews | no     | Hides the Remember my changes option. | <param name='customViews' value='no'/>
|                  |              |        |             | http://tabserver/views/Date-Time/DateCalcs?:embed=yes&amp;:customViews=no |
| device           | :device     | desktop; tablet; phone | If a dashboard has layouts for mobile devices, it displays a specific layout, regardless of the device type. If this parameter isn't set, Tableau Server detects screen size and loads the appropriate layout. See Embed Dashboards on page 519 for examples. | <param name='device' value='phone'/>
<p>|                  |              |        |             | <a href="http://tabserver/views/sales/sales_dashboard?:device=tablet">http://tabserver/views/sales/sales_dashboard?:device=tablet</a> |
| -                | :embed      | yes    | Required for URL parameter. Hides the top nav- | <a href="http://tabserver/views/Date-Time/DateCalcs?:embed=yes">http://tabserver/views/Date-Time/DateCalcs?:embed=yes</a> |</p>
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter</td>
<td>-</td>
<td>string</td>
<td>Customizes what data is displayed when the view opens. You can also filter by URL parameters.</td>
<td>&lt;param name='filter' value='Team=Blue'/&gt;</td>
</tr>
<tr>
<td>:format</td>
<td>pdf; png</td>
<td></td>
<td>Displays a view as a PDF or .png file.</td>
<td><a href="http://tableau-server/views/Sales/Q2?:format=pdf">http://tableau-server/views/Sales/Q2?:format=pdf</a></td>
</tr>
<tr>
<td>:highdpi</td>
<td>false</td>
<td></td>
<td>Renders a view using standard DPI (dots per inch) for high-resolution displays and devices.</td>
<td><a href="http://tableau-server/views/Sales/Q2?:highdpi=false">http://tableau-server/views/Sales/Q2?:highdpi=false</a></td>
</tr>
<tr>
<td>:original_view</td>
<td>yes</td>
<td></td>
<td>If the name parameter refers to a workbook</td>
<td>&lt;param name='filter' value=':original_view=yes'/&gt;</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| host_url         | -             | string | The server name as it appears in the URL. | \(<\text{param name='host_url' value='http://myserver.exampleco.com/'}>\)  
\(<\text{param name='host_url' value='http://localhost/'}>\) |
| link-target      | :link-target  | string | The target window name for external hyperlinks. | \(<\text{param name='linktarget' value='_blank'}>\)  
http://tabserver/views/Date-Time/DateCalcs?embed=yes&:linktarget=_blank |
<p>| load-order       | -             | number | When multiple views are embedded, the default load order is the order in | (&lt;\text{param name='load-order' value='2'}&gt;) |</p>
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>-</td>
<td>string</td>
<td>which the views are listed. Use this setting to override that order. Negative numbers are allowed.</td>
<td><code>&lt;param name='name' value='ExampleCoSales/Sales'/&gt;</code>&lt;br&gt;<code>&lt;param name='name' value=&quot;ExampleCoSales/Sales/jsmith@example.com/EastCoastSales'/&gt;</code></td>
</tr>
</tbody>
</table>

Required for object parameter, Workbook and sheet name and optionally, a custom view (user-name@domain/ [custom view name]). If you refer to the Tableau Server URL to confirm the value of name, exclude the session ID (:iid=<n>) at the end of the URL.
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>path</td>
<td>-</td>
<td>string</td>
<td>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 Trusted Authentication examples.</td>
<td><code>&lt;param name='path' value='trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4'/&gt;</code>&lt;br&gt;<a href="http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes:&amp;:tabs=yes">http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes:&amp;:tabs=yes</a></td>
</tr>
<tr>
<td>-</td>
<td>:record_performance</td>
<td>yes</td>
<td>Starts a performance recording for a view. Add this at the end of the URL, immediately before the session ID (<code>:iid=&lt;n&gt;</code>). See Create a Performance Recording</td>
<td><a href="http://tabserver/#/views/Sales2013/MarginsByAreaCode?:record_performance=yes&amp;:iid=1">http://tabserver/#/views/Sales2013/MarginsByAreaCode?:record_performance=yes&amp;:iid=1</a></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>on page 1139 for more information.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>:render</td>
<td>true; false; number</td>
<td>If client-side rendering is enabled (which is the default), setting this to false forces server-side rendering for the session. If client-side rendering is disabled, setting this to true enables it for the session. A number can be used to test a com-</td>
<td><a href="http://tabserver/views/DateTime/DateCalculs?:render=false">http://tabserver/views/DateTime/DateCalculs?:render=false</a></td>
<td></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>-</td>
<td>:revert</td>
<td>all; filter; sorts; axes; shelves</td>
<td>Returns the item to its original state.</td>
<td><a href="http://tabserver/views/DateTime/DateCalc-s?:embed=yes&amp;:revert=all">http://tabserver/views/DateTime/DateCalc-s?:embed=yes&amp;:revert=all</a></td>
</tr>
<tr>
<td>showShareOptions</td>
<td>-</td>
<td>true; false</td>
<td>Controls whether the Share options are displayed in an embedded view. The default is true.</td>
<td>&lt;param name='showShareOptions' value='true' /&gt;</td>
</tr>
</tbody>
</table>

Note: If Share
<table>
<thead>
<tr>
<th>Object Parameter</th>
<th>URL Parameter</th>
<th>Values</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>options still appear when you change this parameter to false, ask your Tableau Server administrator to change the tabsadmin set optio</td>
<td></td>
<td>vizql serv</td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>site_root</td>
<td></td>
<td>string</td>
<td>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="http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/">Trusted Authentication examples</a>.</td>
<td><code>&lt;param name='site_root' value='/#/Sales'/&gt;</code>&lt;br&gt;<code>&lt;param name='site_root' value=''/&gt;</code></td>
</tr>
<tr>
<td>tabs</td>
<td>:tabs</td>
<td>yes; no</td>
<td>Displays or hides tabs.</td>
<td><code>&lt;param name='tabs' value='yes'/&gt;</code></td>
</tr>
<tr>
<td>ticket</td>
<td>-</td>
<td>number</td>
<td>For trusted authentication only, cannot be used</td>
<td><code>&lt;param name='ticket' value='Etdpsm_Ew6rJY-9kRrALjauU'/&gt;</code>&lt;br&gt;<code>http://tableauserver/trusted/Etdpsm_Ew6rJY-</code></td>
</tr>
<tr>
<td>Object Parameter</td>
<td>URL Parameter</td>
<td>Values</td>
<td>Description</td>
<td>Examples</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------</td>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with the path object parameter. Must be used with name object to construct the trusted ticket redemption URL. See the Trusted Authentication examples.</td>
<td>9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&amp;:tabs=yes</td>
<td></td>
</tr>
</tbody>
</table>
| toolbar          | :toolbar      | yes; no; top | The toolbar is displayed by default on the bottom when this parameter is not set. When no the toolbar is excluded from the embedded view. When top, the toolbar is placed above the view. | <param name='toolbar' value=top'/>
|                  |               |        |             | http://tabserver/views/DateTime/DateCalcs?:embed=yes&:toolbar=no |
### Examples

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

#### Add Filters

You can pass filter values so the view opens showing just the data you want. For example, you may want to include a hyperlink from another part of your web application to an embedded sales performance view that only shows a specific region.

**Script Tag Example**

```html
<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='host_url' value='http://myserver/' />
  <param name='site_root' value='' />
  <param name='name' value='Superstore/Product' />
  <param name='filter' value='Region=East' />
</object>
```

To pass through multiple filters, just separate each value with a comma. For example:

```html
<param name='filter' value='Region=East,West' />
```
Iframe Tag Examples

```html
<iframe src="http://myserver/views/Superstore/Product?:embed=y&Region=East" width="800" height="600"></iframe>

<iframe src="http://myserver/views/Superstore/Product?:embed=yes&Region=East,West" width="800px" height="600px"></iframe>
```

For more information, see Filter on Multiple Fields below.

Filter on Multiple Fields

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

Script Tag Example

```html
<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='host_url' value='http://myserver/' />
    <param name='site_root' value='' />
    <param name='name' value='Superstore/Product' />
    <param name='filter' value='Region=Central,South&Customer Segment=Consumer,Home Office' />
</object>
```

Iframe Tag Example

```html
<iframe src="http://myserver/views/Superstore/Product?:embed=y&Region=Central,South&Segment=Consumer,Home Office" width="800" height="600"></iframe>
```

The first box below shows an example of the URL you might get when you click Share on a view and copy the link in the Link field.

The second box shows how you might modify the URL and add it to an Iframe by deleting the showShareOptions and display_count parameters, adding filter parameters for
Region and Segment, and adding width and height parameters, to create an embed link that displays only Consumer and Home Office products from the Central and South regions.

Note: If a filter value contains a special character, such as a comma, replace the character with the URL encoding sequence for \ (backslash, %5c) followed by the URL encoding sequence for the special character. The backslash is needed to escape the special character. For example, the URL encoding sequence for \, (backslash, comma) is %5c%2c.

Filter Dates and Times

If you want to filter on a Date/Time field, include the value using the default Tableau format shown below:

```
yyyy-mm-dd hh:mm:ss
```

The time part uses a 24-hour clock. Many databases store all date values as Datetime fields, so you may need to pass a time value along with your date.

Script Tag Example
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/Sales-Performance?:embed=yes&Date=2008-12-01%2022:18:00" width="800" height="600"></iframe>
```

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

**Filter Measures**

You can filter measures by including one or more values. There is no support for greater than, less than, or ranges. The example below filters to show only $100 and $200 sales.

**Script Tag Example**

```
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style='display:none;';>
    <param name='host_url' value='http://myserver/' />
    <param name='site_root' value='' />
    <param name='name' value='Sales/Sales-Performance' />
    <param name='filter' value='Profit=100, 200' />
</object>
```

**Iframe Tag Example**

```
<iframe src="http://myserver/views/Sales/Sales-Performance?:embed=yes&Profit=100,200" width="800" height="600"></iframe>
```
Control the Load Order of Multiple Views

You can control the order in which multiple views load for the people working with your views. This feature can only be accessed using embed code that relies on the Tableau JavaScript file. In the following example, two views are embedded. The second view loads first, followed by the top view. If you embed multiple views and give them all the same load order value, or if you don’t specify load order parameters, they are loaded in the order in which they appear on the page.

**Script Tag Example**

```html
<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='host_url' value='http://myserver/' />
  <param name='site_root' value='' />
  <param name='name' value='MyCoSales/TopPerformers' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
  <param name='filter' value='Salesperson=Top 5' />
  <param name='load-order' value='0' />
</object>
<script type='text/javascript' src='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='600' height='400' style='display:none;'>
  <param name='host_url' value='http://myserver/' />
  <param name='site_root' value='' />
  <param name='name' value='MyCoSales/SalesScoreCard' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
  <param name='filter' value='Salesperson=Top 5' />
  <param name='load-order' value='-1' />
</object>
```

**Embed Dashboards**

You can embed a Tableau dashboard in your own web site or wiki page and you can have it display in different layouts based on screen size.

- To automatically make the right layout appear for your users, regardless of what device they are using, create device-specific layouts for your dashboard in Tableau Desktop. Tableau Server automatically displays the correct layout based on screen size.
To always make a certain layout appear, regardless of screen size, use the device parameter in your embed code.

In the following example, the embed code displays a dashboard. If the dashboard uses device-specific layouts, the correct layout displays, that is, no additional parameters are needed:

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder'>
<object class='tableauViz' width='100%' height='100%' style='display:none;'
    <param name='host_url' value='http://mysite.myserver.com' />
    <param name='site_root' value=''/>
    <param name='name' value='ProfitAnalysis/Sales_Dashboard' />
    <param name='tabs' value='yes' />
    <param name='toolbar' value='yes' />
    <param name='filter' value=':original_view=yes' />
    </object></div>
```

The following example assumes that the embedded dashboard has device-specific layouts. The device parameter is set to phone. This means that, no matter which device displays the dashboard, the layout created for phones will be the one that's displayed.

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder'>
<object class='tableauViz' width='100%' height='100%' style='display:none;'
    <param name='host_url' value='http://mysite.myserver.com' />
    <param name='site_root' value=''/>
    <param name='name' value='ProfitAnalysis/Sales_Dashboard' />
    <param name='device' value='phone' />
    <param name='tabs' value='yes' />
    <param name='toolbar' value='yes' />
    <param name='filter' value=':original_view=yes' />
    </object></div>
```

**Screen Dimensions and Device Layouts**

If a dashboard used device-specific layouts, Tableau uses the largest dimension of the device screen, whether it's the height or width, to determine which one to load.

<table>
<thead>
<tr>
<th>If the largest dimension is...</th>
<th>This dashboard device layout is displayed...</th>
</tr>
</thead>
<tbody>
<tr>
<td>500 px or less</td>
<td>Phone</td>
</tr>
</tbody>
</table>
For details on how to create a dashboard that uses device-specific layouts, see Create Dashboards in the Tableau Desktop help.

Embed Code for Custom Views

When you embed a view of a workbook or sheet that has custom views available:

- If the embed code URL for the view explicitly refers to a custom view, that custom view will be displayed by default.
- If the embed code URL does not explicitly refer to a custom view, and a Default custom view has been defined, the Default custom view will be displayed in the embedded view by default.
- If no Default custom view has been defined, the original view will be displayed in the embedded view by default.

Note: To ensure the original view will be displayed by default in an embedded view, make sure the embed code URL for the name parameter does not explicitly refer to a custom view, and include the following filter parameter in the embed code: <param name='filter' value=':original_view=yes'/>

In the following example, the embed code will always display the original view of the Profit Analysis sheet in the Profit Analysis workbook, because the filter parameter is set to :original_yes, and the name parameter does not refer to a specific custom view in the URL for the sheet.

```html
<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width: 1496px; height: 749px;'></div>
<object class='tableauViz' width='1496' height='749' style='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
  <param name='name' value='ProfitAnalysis/ProfitAnalysis' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
  <param name='filter' value=':original_view=yes' />
</object>
```
In this example, the setting for the name parameter in this example specifically refers to the URL for a custom view named Furniture (in the Profit Analysis sheet in the Profit Analysis workbook).

<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>

<iframe width='1496' height='749' style='display:none;'></iframe>

In this example, the name parameter does not refer to a specific custom view in the URL for the sheet, and the original_view parameter has not been specified. The embed code here will display the custom view that has been set to Default in the Profit Analysis sheet in the Profit Analysis workbook. However, if the original view is still the Default (no other custom view has been set to Default), then the original view will be displayed as the default view.

<script type='text/javascript' src='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>

<iframe width='1496' height='749' style='display:none;'></iframe>

**Embed Views into Wikis**

You can easily embed a view into a wiki or other web page simply by putting the view inside an `<iframe>` tag.

1. Navigate to the wiki page you want to embed a view into.

2. Edit the page and add an `<iframe>` where the source is the URL from the Email box of the Share View dialog box. For example:
<iframe src="http://myserver/views/Date-Time/DateCalculations.png?embed=yes&:toolbar=no" width="800" height="600"></iframe>

3. Save your changes.

The view is embedded into the wiki page.

If both Tableau Server and the wiki are configured to use Microsoft SSPI, users accessing an embedded view on the wiki will be automatically signed in so they can see the view.

If the server and the wiki are not using the same method for authentication, users will first be asked to sign in to the server before they can see the view.

### Embed Images

In addition to embedding a view into a `<script>` or `<iframe>` tag you can also embed the view as an image. When you embed an image the view is not interactive, however, it is updated every time the page fully reloads. That way the image shows the latest data even if the underlying data changes.

1. Navigate to the page where you want to embed the image.

2. Edit the page and add an `<img>` tag where the source is the URL from the Email box of the Share View dialog box for the view, plus the .png file extension. For example:

   ```html
   <img src="http://tableauserver/views/Date-Time/DateCalculations.png" width="900" height="700">
   ```

**Note:** This approach will only work if the user accessing the embedded image also has
an active web browser session with Tableau Server and is signed in to Tableau Server using Microsoft SSPI.

**Embed Views into SharePoint (Microsoft SSPI)**

You can embed a Tableau Server view in a SharePoint page. To automatically authenticate Tableau Server users who access the embedded view you have two choices, both of which depend on which user authentication method was selected during Tableau Server Setup. You can use either Active Directory with Enable automatic logon to authenticate Tableau Server users (also known as using Microsoft SSPI), or you can use Local Authentication—and then also configure Tableau Server for trusted authentication.

This topic applies to the first option, where both Tableau Server and SharePoint are using Microsoft SSPI. If your Tableau Server is using Local Authentication, see Embed Views into SharePoint (Local Authentication) on page 527 for steps.

**Requirements**

**Licensed users:** Anyone who accesses an embedded view must be a licensed user on Tableau Server.

**SharePoint version:** Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server’s security requirements.

**TableauEmbeddedView web part:** You must have a TableauEmbeddedView web part deployed to your SharePoint server before you can embed Tableau views in a SharePoint page. For sample SharePoint code and instructions for how to create a web part and deploy it to your SharePoint server, see C:\Program Files\Tableau\Tableau Server\<version>\extras\embedding\sharepoint. **Note:** The sample SharePoint code is provided as an example, and may require modification to work in your SharePoint deployment.

**Embedding a View into SharePoint**

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.
2. In the section of the page where you want to embed the view, on the Insert tab, click Web Part.
3. Under Categories, in the Custom (or Miscellaneous) folder, select TableauEmbeddedView, and then click Add in the lower-right corner.
4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select Edit Web Part.

5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
   - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
   - In **View Path**, enter the path to the view you want to embed.
- Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.

- In the Appearance section you can specify a Title for the web part, the Height, Width, Chrome State, and Chrome Type. In general you should specify a fixed height (for example, 700 Pixels) and adjust the width to fit the zone.

6. Click OK to apply the changes and exit edit mode.
The view will be embedded into the web part that you just created. Your users will not need to log in to Tableau Server to see the embedded view, rather they will be automatically authenticated using Microsoft SSPI.

**Embed Views into SharePoint (Local Authentication)**

You can embed a Tableau Server view in a SharePoint page. If Tableau Server is using Local Authentication for user authentication, there are some extra steps you need to take before you start embedding views.

This topic describes how to complete the following steps:

- Edit the security permissions for the TableauEmbeddedView.dll file.
- Install and deploy the TableauEmbeddedView.wsp file.
- Verify the web part’s deployment.
- Embed a view in SharePoint using the Tableau web part.

**Note:** If your Tableau Server installation is using Active Directory for user authentication, you can start embedding views right away. For more information, see *Embed Views into SharePoint (Microsoft SSPI)* on page 524.

**Requirements**

**Users:** To access an embedded view, users must be licensed Tableau Server users and their user name on SharePoint must be the same as their user name on Tableau Server.

**SharePoint version:** Starting with Tableau Server 8.1, you must use SharePoint 2013 to embed Tableau Server views in SharePoint pages. SharePoint 2013 uses Microsoft .NET Framework version 4.5, which meets Tableau Server’s security requirements.

**Edit Security Permissions for TableauEmbeddedView.dll**

Edit the security permissions for TableauEmbeddedView.dll so that all users of the operating system can use it.

1. Locate the TableauEmbeddedView.dll and TableauEmbeddedView.wsp files that install with Tableau Server. If Tableau Server is installed on drive C, the files will be in the following directory:
   
   C:\Program Files\Tableau\Tableau Server\10.0\extras\embedding\sharepoint\
   
2. Copy the files to the root directory of your SharePoint server. The root directory is usually located at C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin, for example:
   
   C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin
3. To edit the security permissions on TableauEmbeddedView.dll, right-click **TableauEmbedded.dll** and then select **Properties > Security**.

4. Under **Group or user names**, select **Everyone**, and then click **Edit**.

5. Under **Permissions for Everyone**, for the **Full control** permission, select **Allow**.
6. Click **OK**.

**Install and Deploy TableauEmbeddedView.wsp**

The TableauEmbeddedView.wsp file gives SharePoint more information about what to do with the .dll file. You copied the TableauEmbeddedView.wsp file to the SharePoint root directory in the previous procedure. To install and deploy the .wsp file, follow these steps:

1. Open SharePoint 2013 Management Shell and enter the following command:

   ```
   Add-SPSolution -LiteralPath "C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin\TableauEmbeddedView.wsp"
   ```

2. On the SharePoint Central Administration home page, click **System Settings**.

3. In the **Farm Management** section, click **Manage farm solutions**.

4. On the Solution Management page, click the solution that you want to deploy.

5. On the Solution Properties page, click **Deploy Solution**.

6. On the Deploy Solution page, in the **Deploy When** section, select one of the following options:
- Now
- At a specified time. Specify a time by using the date and time boxes.

7. In the **Deploy To?** section, in the **A specific web application** list, click **All web applications** or select a specific Web application, and then click **OK**.

8. Open your SharePoint site. Click the settings icon, and then select **Site settings**.

9. Under Site Collection Administration, click **Site collection features**.

10. Scroll to the TableauEmbeddedView feature and then click **Activate** to activate the feature.

**Verify the Web Part's Deployment**

In the following procedure, you will verify that the Tableau web part is installed.

1. Open your SharePoint site in a web browser.
   It may take a few moments for the site to appear.

2. Click the settings icon, and then select **Site settings**.

3. Under **Web Designer Galleries**, click **Web parts**.
4. Confirm that `TableauEmbeddedView.webpart` is listed.

**Embed a View Using the Tableau Web Part**

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.
2. In the section of the page where you want to embed the view, on the **Insert** tab, click **Web Part**.
3. Under Categories, in the **Custom** (or **Miscellaneous**) folder, select `TableauEmbeddedView`, and then click **Add** in the lower-right corner.
4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select Edit Web Part.

5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
   - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
   - In **View Path**, enter the path to the view you want to embed.
- Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.

- In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed height (for example, 700 Pixels) and adjust the width to fit the zone.

6. Click **OK** to apply the changes and exit edit mode.

Now the view is embedded in the page and users who access it will be automatically signed in based on their user name and password for SharePoint.
This is an example of embedding views into SharePoint using the provided .dll file. You can also embed views into other types of web application. For more information, see JavaScript API on the Tableau Developer Portal.

**tabcmd**

The tabcmd utility is one of the two command line tools that installs with Tableau Server (the other is tabadmin on page 1255). The commands provided through tabcmd can help you automate common tasks, such as publishing workbooks in batches and administering users and groups. The tabcmd utility installs in the Tableau Server bin folder (C:\Program Files\Tableau Server\10.0\bin), but you can install and run tabcmd on another machine as well. For more information, see the following topics.

**Install tabcmd**

By default, the tabcmd command line utility installs with Tableau Server to the server’s bin folder (for example, C:\Program Files\Tableau\Tableau Server\10.0\bin). You can run it from there. For administrative flexibility, you can also install it on other computers.

If you installed the tabcmd command line utility on computers that are not running Tableau Server and you are upgrading Tableau Server to a new major version (version 9.3 to version 10.0 for example), Tableau recommends you also upgrade standalone installations of tabcmd to avoid any potential incompatibilities between versions.

To install tabcmd on another machine:

1. Navigate to the extras folder on Tableau Server:

   C:\Program Files\Tableau\Tableau Server-\10.0\extras\TabcmdInstaller.exe

2. Copy TabcmdInstaller.exe to the computer where you want to install it.

3. Double-click TabcmdInstaller.exe to run it.

4. Follow the prompts to install tabcmd.

   Because tabcmd is a command line tool, and due to some limitations with the Windows operating system, Tableau recommends that you install tabcmd in a folder named tabcmd at the root of the C:\drive (C:\tabcmd).

**Note:** Running the tabcmd Setup program does not automatically add tabcmd to the Windows PATH variable, you will need to either explicitly call tabcmd using its full path or add its directory to the PATH variable.
How to Use tabcmd

The basic steps for using tabcmd are as follows:

1. Open the Command Prompt as an administrator.
2. Change to the Tableau Server bin folder.
   
   For example: cd C:\Program Files\Tableau\Tableau Server\10.0\bin
   
   Or you can include the location in the command.
3. Run the tabcmd command.

When you use tabcmd, you must establish an authenticated server session. The session identifies the Tableau Server and the Tableau Server user running the session. You can start a session first, and then specify your command next, or you can start a session and execute a command all at once. If you are using tabcmd to perform more than one task, you must run each task one after the other (serially), rather than in parallel.

Commands (such as login) and the options (such as -s, -u, etc.) are not case sensitive, but the values you provide (such as p@ssw0rd or User@Example.com) are case sensitive.

Examples

The following command demonstrates starting a session with the Tableau Server named tabserver.myco.com:

```
tabcmd login -s http://tabserver.myco.com -u admin -p p@ssw0rd!
```

The next example shows a command that deletes a workbook named Sales_Workbook:

```
tabcmd delete "Sales_Workbook"
```

Here's how to accomplish all of the above with one command—note that you do not need login here:

```
tabcmd delete "Sales_Workbook" -s http://tabserver.myco.com -u admin -p p@ssw0rd!
```

A Tableau Server can run multiple sites. When a workbook is on the Default site of a multi-site server you don't need to specify Default, the above command is sufficient. However, if the command applies to something on a site other than Default, you need to specify the site ID for that site (see login on page 1338). Here's the same command for a workbook that's on the West Coast Sales site (site ID wsales):

```
tabcmd delete "Sales_Workbook" -s http://tabserver.myco.com -t wsales -u admin -p p@ssw0rd!
```

The options -s, -t, -u, and -p are among the tabcmd global variables, which can be used with any command.
For more information, see **tabcmd Commands on page 1319**.

**Status messages and logs**

When a command is successful, tabcmd returns a status code of zero. A full error message for non-zero status codes is printed to **stderr**. In addition, informative or progress messages may be printed to **stdout**.

A full log named **tabcmd.log** that includes debugging, progress, and error messages is written to **C:\Users\<username>\AppData\Local\Tableau**.

**tabcmd Global Options**

The table below shows the options that are used by all commands. The **--server**, **--user**, and **--password** options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td>--help</td>
<td></td>
<td>Displays the help for the command.</td>
</tr>
<tr>
<td>-c</td>
<td>--use-certificate</td>
<td></td>
<td>Use client certificate to sign in. Required when mutual SSL is enabled. For more information, see <strong>Configure External SSL on page 972</strong>.</td>
</tr>
<tr>
<td>-s</td>
<td>--server</td>
<td>Tableau Server URL</td>
<td>Required at least once to begin session.</td>
</tr>
<tr>
<td>-u</td>
<td>--user</td>
<td>Tableau Server username</td>
<td>Required at least once to begin session.</td>
</tr>
<tr>
<td>-p</td>
<td>--password</td>
<td>Tableau Server password</td>
<td>Required at least once to begin session. You can alternatively use the <strong>-P</strong> option.</td>
</tr>
<tr>
<td></td>
<td>--password-file</td>
<td>filename.txt</td>
<td>Allows the password to be stored in the given file rather than the com-</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-t</td>
<td>--site</td>
<td>Tableau Server site ID</td>
<td>Indicating that the command applies to the site specified by the site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.</td>
</tr>
<tr>
<td>-x</td>
<td>--proxy</td>
<td>Host:Port</td>
<td>Uses the specified HTTP proxy.</td>
</tr>
<tr>
<td>--no-prompt</td>
<td></td>
<td></td>
<td>When specified, the command will not prompt for a password. If no valid password is provided the command will fail.</td>
</tr>
<tr>
<td>--no-proxy</td>
<td></td>
<td></td>
<td>When specified, an HTTP proxy will not be used.</td>
</tr>
<tr>
<td>--no-cert-check</td>
<td></td>
<td></td>
<td>When specified, tabcmd (the client) does not validate the server’s SSL certificate.</td>
</tr>
<tr>
<td>--[no--]cookie</td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>--timeout</td>
<td>seconds</td>
<td></td>
<td>Waits the specified number of seconds for the server to complete pro-</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>cessing the command. By default the process will timeout in 30 seconds.</td>
</tr>
<tr>
<td>--</td>
<td></td>
<td></td>
<td>Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where --430105/Sheet1 is a required value for the export command.</td>
</tr>
</tbody>
</table>

**tabcmd Commands**

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

- **addusers** (to group)
- **creategroup**
- **createproject**
- **createsite**
- **createsiteusers**
createusers
delete **workbook-name** or **datasource-name**
deletegroup
deleteproject
deletesite
deletesiteusers
deleteusers
editdomain
editsite
export
**get url**
initialuser
listdomains
listsites
login
logout
publish
refreshextracts
removeusers
runschedule
set
syncgroup
version

**addusers group-name**

Adds users to the specified group.

**Example**

tabcmd addusers "Development" --users "users.csv"
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--users</td>
<td>filename.csv</td>
<td></td>
<td>Add the users in the given file to the specified group. The file should be a simple list with one user name per line. User names are not case sensitive. The users should already be created on Tableau Server. See also CSV Import File Guidelines on page 810.</td>
</tr>
<tr>
<td>--[no-]complete</td>
<td></td>
<td></td>
<td>When set to complete this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.</td>
</tr>
</tbody>
</table>

**creategroup group-name**

Creates a group. Use addusers (for local groups) and syncgroup (for Active Directory groups) commands to add users after the group has been created.

**Example**

tabcmd creategroup "Development"

**createproject project-name**

Creates a project.

**Example**

tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing quarterly sales reports."

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>--name</td>
<td>name</td>
<td>Specify the name of the project that you want to create.</td>
</tr>
<tr>
<td>-d</td>
<td>--description</td>
<td>description</td>
<td>Specify a description for the project.</td>
</tr>
</tbody>
</table>
createsite site-name

Creates a site.

Examples

Create a site named West Coast Sales. A site ID of WestCoastSales will be automatically created, the site will have no storage quota limit, and site administrators will be able to add and remove users:

```
tabcmd createsite "West Coast Sales"
```

Create a site named West Coast Sales with a site ID of wsales:

```
tabcmd createsite "West Coast Sales" -r "wcoast"
```

Prevent site administrators from adding users to the site:

```
tabcmd createsite "West Coast Sales" --no-site-mode
```

Set a storage quota, in MB:

```
tabcmd createsite "West Coast Sales" --storage-quota 100
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>--url</td>
<td>site ID</td>
<td>Used in URLs to specify the site. Different from the site name.</td>
</tr>
<tr>
<td>--user-quotta</td>
<td>number of users</td>
<td></td>
<td>Maximum number of users that can be added to the site.</td>
</tr>
<tr>
<td>--[no-]site-mode</td>
<td></td>
<td></td>
<td>Allow or deny site administrators the ability to add users to or remove users from the site.</td>
</tr>
<tr>
<td>--storage-quotta</td>
<td>number of MB</td>
<td></td>
<td>In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.</td>
</tr>
</tbody>
</table>

createsiteusers filename.csv

Adds users to a site, based on information supplied in a comma-separated values (CSV) file. If the user is not already created on the server, the command creates the user before adding that user to the site.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, role, administrator level, publisher (yes/no), and email address. For
information about the format of the CSV file, see **CSV Import File Guidelines on page 810.** As an alternative to including role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the **--role** option.

By default, users are added to the site that you are logged in to. To add users to a different site, include the global **--site** option and specify that site. (You must have permissions to create users on the site you specify.)

If the server contains multiple sites, you cannot assign the ServerAdministrator role to a user by using the `createsiteusers` command. (Use `createusers` instead.) If you specify the ServerAdministrator role for the role option, the command returns an error. If the CSV file includes System as value for administrator, the value is ignored and the user is assigned the Unlicensed role. However, if the server contains only one site (the default site), you can assign the ServerAdministrator role or specify system for the administrator value; in that case, the `createsiteusers` command works like the `createusers` command.

By default, this command creates users using a synchronous operation (it waits for all operations to complete before proceeding). You can use the **--no-wait** option to specify an asynchronous operation.

**Local authentication**

If the server is configured to use local authentication, the information in the CSV file is used to create users.

**Active Directory authentication**

If the server is configured to use Active Directory authentication, user information is imported from Active Directory to the server. In that case, any password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not guaranteed to be unique across domains, therefore you must include the domain as part of the user name (for example, `example\Adam` or `adam@example.com`).

While these can be sent either as `domain/username` or `username@domain.com`, we recommend using the `domain/username` format. See **User Management in Active Directory Deployments on page 1251** for more information.

**Example**

tabcmd createsiteusers "users.csv" --role "Interactor"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--admin-type</td>
<td>Site or None</td>
<td>(Deprecated. Use the <strong>--role</strong> option)</td>
<td></td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>option instead.) Assigns or removes the site administrator right for any user who does not already have an administrator setting in the CSV file. The default is None for new users and unchanged for existing users. If the server contains multiple sites; system administrators cannot be created or demoted using createSiteUsers. (Use createusers instead.)</td>
</tr>
<tr>
<td>--complete</td>
<td></td>
<td></td>
<td>Requires that all rows be valid for any change to succeed. This is the default setting.</td>
</tr>
<tr>
<td>--license</td>
<td>Interactor, Viewer, or Unlicensed</td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Specifies the license level for any user who does not already have a license level setting in the CSV file. The default is Unlicensed for new users and unchanged for existing users.</td>
</tr>
<tr>
<td>--no-complete</td>
<td></td>
<td></td>
<td>Specifies that the command should make changes on the server even if not all rows contain valid information. Rows</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>that contain invalid information are skipped.</td>
</tr>
<tr>
<td>--no-publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Disallows publishing rights for any users who do not already have a publisher setting in the CSV file. This is a default value for new users.</td>
</tr>
<tr>
<td>--nowait</td>
<td></td>
<td></td>
<td>Do not wait for asynchronous jobs to complete.</td>
</tr>
<tr>
<td>--publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Assigns publishing rights for any user who does not already have a publisher setting in the CSV file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.</td>
</tr>
<tr>
<td>-r</td>
<td>--role</td>
<td>ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed</td>
<td>Specifies a site role for any user who does not already have a role specified in the CSV file. The default is Unlicensed for new users and unchanged for existing users. If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-</td>
<td>--silent-progress</td>
<td>Do not display progress messages for the command.</td>
<td></td>
</tr>
</tbody>
</table>

**createusers filename.csv**

Create users in Tableau Server, based on information supplied in a comma-separated values (CSV) file.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, role, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see [CSV Import File Guidelines](#) on page 810. As an alternative to including role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the `--role` option.

If the server has only one site (the default site), the user is created and added to the site. If the server has multiple sites, the user is created but is not added to any site. To add users to a site, use `createsiteusers`.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

**Note:** You cannot assign the ServerAdministrator role if the server has more than one site. In that case, use the `createuser` command.

**Note:** If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.
Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory to the server. In that case, any password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not guaranteed to be unique across domains, therefore you must include the domain as part of the user name (for example, example\Adam or adam@example.com).

While these can be sent either as domain/username or username@domain.com, we recommend using the domain/username format. See User Management in Active Directory Deployments on page 1251 for more information.

Example

tabcmd createusers "users.csv" --role "ServerAdministrator"

tabcmd createusers "users.csv"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--admin-type</td>
<td>Site or None</td>
<td>(Deprecated. Use the --role option instead.) Assigns or removes the site administrator right for any user who does not already have an administrator setting in the CSV file. The default is None for new users and unchanged for existing users.</td>
<td></td>
</tr>
<tr>
<td>--complete</td>
<td></td>
<td>Requires that all rows be valid for any change to succeed. This is the default setting.</td>
<td></td>
</tr>
<tr>
<td>--license</td>
<td>Interactor, Viewer, or Unlicensed</td>
<td>(Deprecated. Use the --role option instead.) Specifies the license level for any user who does not already have a</td>
<td></td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>license level setting in the CSV file. The default is Unlicensed for new users and unchanged for existing users.</td>
</tr>
<tr>
<td>--no-complete</td>
<td></td>
<td></td>
<td>Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.</td>
</tr>
<tr>
<td>--no-publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Disallows publishing rights for any users who do not already have a publisher setting in the CSV file. This is a default value for new users.</td>
</tr>
<tr>
<td>--nowait</td>
<td></td>
<td></td>
<td>Do not wait for asynchronous jobs to complete.</td>
</tr>
<tr>
<td>--publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Assigns publishing rights for any user who does not already have a publisher setting in the CSV file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| -r            | --role       | ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed | Specifies a role for any user who does not already have a role specified in the CSV file. The default is Unlicensed for new users and unchanged for existing users.

On a multi-site server, the command does not assign the user to a site. Therefore, the only roles that the command will assign are ServerAdministrator and Unlicensed. In that case, if you specify a different role (like Publisher or Viewer), the command assigns the Unlicensed role.

On a single-site server, the user is created and added to the default site using the role that you specify.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

**Note:** If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator.
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>options.</td>
</tr>
<tr>
<td>--silent-progress</td>
<td></td>
<td></td>
<td>Do not display progress messages for the command.</td>
</tr>
</tbody>
</table>

**delete workbook-name or datasource-name**

Deletes the specified workbook or data source from the server.

This command takes the name of the workbook or data source as it is on the server, not the file name when it was published.

**Example**

```
tabcmd delete "Sales_Analysis"
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>--project</td>
<td>Project name</td>
<td>The name of the project containing the workbook or data source you want to delete. If not specified, the “Default” project is assumed.</td>
</tr>
<tr>
<td>--workbook</td>
<td></td>
<td>Workbook name</td>
<td>The name of the workbook you want to delete.</td>
</tr>
<tr>
<td>--data-source</td>
<td></td>
<td>Data source name</td>
<td>The name of the data source you want to delete.</td>
</tr>
</tbody>
</table>

**deletegroup group-name**

Deletes the specified group from the server.

**Example**

```
tabcmd deletegroup "Development"
```

**deleteproject project-name**

Deletes the specified project from the server.

**Example**
tabcmd deleteproject "Designs"

**deletesite site-name**
Deletes the specified site from the server.

**Example**

```bash
tabcmd deletesite "Development"
```

**deletesiteusers filename.csv**
Removes users from the site that you are logged in to. The users to be removed are specified in a file that contains a simple list of one user name per line. (No additional information is required beyond the user name.)

By default, if the server has only one site, or if the user belongs to only one site, the user is also removed from the server. On a Tableau Server Enterprise installation, if the server contains multiple sites, users who are assigned the role of Server Administrator are removed from the site but are not removed from the server.

If the user owns content, the user's role is change to Unlicensed, but the user is not removed from the server or the site. The content is still owned by that user. To remove the user completely, you must change the owner of the content and then try removing the user again.

If the user was imported from Active Directory, the user is removed from the site and possibly from the server. However, the user is not deleted from Active Directory.

**Example**

```bash
tabcmd deletesiteusers "users.csv"
```

**deleteusers filename.csv**
Deletes the users listed in the specified comma-separated values (CSV) file. The CSV file should contain a simple list of one user name per line.

**Example**

```bash
tabcmd deleteusers "users.csv"
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--[no--complete</td>
<td></td>
<td>When set to --complete this option requires that all rows be valid for any change to succeed. If not specified, --</td>
</tr>
</tbody>
</table>
### editdomain

Changes the nickname or full domain name of an Active Directory domain on the server. A domain "nickname" is the Windows NetBIOS domain name.

You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to sign in. However, if the user name that you are currently signed in with exists in both the current domain and the new domain, you can modify the full name for the current domain.

Review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain name mapping, and user names interact with Tableau Server.

To see a list of domains, use `listdomains`.

**Examples**

```
tabcmd editdomain --id 2 --nickname "new-nickname"
tabcmd editdomain --id 3 --name "new-name"
```

### editsite site-name

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

**Examples**

```
tabcmd editsite wc_sales --site-name "West Coast Sales"
tabcmd editsite wc_sales --site-id "wsales"
tabcmd editsite wsales --status ACTIVE
```
tabcmd editsite wsales --user_quota 50

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--site-name</td>
<td>Name to change the site to</td>
<td>The name of the site that's displayed.</td>
</tr>
<tr>
<td>--site-id</td>
<td>The site ID to change the site to</td>
<td>Used in the URL to uniquely identify the site.</td>
</tr>
<tr>
<td>--user-quota</td>
<td>Number of users</td>
<td>Maximum number of users who can be members of the site.</td>
</tr>
<tr>
<td>--[no-]site-mode</td>
<td></td>
<td>Allow or prevent site administrators from adding users to the site.</td>
</tr>
<tr>
<td>--status</td>
<td>ACTIVE or SUSPENDED</td>
<td>Activate or suspend a site.</td>
</tr>
<tr>
<td>--storage-quota</td>
<td>Number of MB</td>
<td>In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.</td>
</tr>
</tbody>
</table>

**export**

Exports a view or workbook from Tableau Server and saves it to a file. This command can also export just the data used for a view.

Note the following when you use this command:

- **Permissions**: To export, you must have the Export Image permission. By default, this permission is Allowed or Inherited for all roles, although permissions can be set per workbook or view.

- **Exporting data**: To export just the data for a view, use the --csv option. This exports the summary data used in a view to a .csv file.

- **Specifying the view, workbook, or data to export**: You specify this using the "workbook/view" string as it appears in the URL for the workbook or view, not using its "friendly name," and excluding the :iid=<n> session ID at the end of the URL. For example, to export the Tableau sample view Investment Growth from the Finance workbook, you would use the string `Finance/InvestmentGrowth`, not `Finance/Investment Growth`, or `Finance/InvestmentGrowth?iid=1`. Use --t <site_id> if the server is running multiple sites and the view or workbook is on a site other than Default.
To export a workbook, you still include a valid view in the string you use. Using the above example, to export the *Finance* workbook, you would use the string Finance/InvestmentGrowth. Finally, to export a workbook, it must have been published with **Show Sheets as Tabs** selected in the Tableau Desktop Publish dialog box.

- **The saved file's format**: Your format options depend on what's being exported. A workbook can only be exported as a PDF using the `--fullpdf` argument. A view can be exported as a PDF (`--pdf`) or a PNG (`--png`).

- **The saved file's name and location** (optional): If you don't provide a name, it will be derived from the view or workbook name. If you don't provide a location, the file will be saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

  **Note**: You must include a file name extension such as `.csv` or `.pdf`. The command does not automatically add an extension to the file name that you provide.

- **Dashboard web page objects not included in PDF exports**: A dashboard can optionally include a web page object. If you are performing an export to PDF of a dashboard that includes a web page object, the web page object won't be included in the PDF.

- **Non-English characters and PDF exports**: If you are exporting a view or workbook with a name that includes a non-English characters you need to URL encode the character.

  For example if your command includes the city Zürich, you need to URL encode it as Z%C3%BCrich:

  ```shell
tabcmd export "/Cities/Sheet1?locationCity=Z%C3%BCrich" --fullpdf
```

### Clearing the Cache to Use Real-Time Data

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using tabcmd with your own scripting and the `refresh` URL parameter is being used a great deal, this can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

### Examples

**Views**

```shell
tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report.csv"
tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f
```
"C:\Tableau_Workbooks\Weekly-Reports.pdf"

tabcmd export "Finance/InvestmentGrowth" --png

tabcmd export "Finance/InvestmentGrowth?:refresh=yes" --png

**Workbooks**

tabcmd export "Q1Sales/Sales_Report" --fullpdf

tabcmd export #/Sales "Sales/Sales Analysis" --fullpdf --pagesize tabloid -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>--filename</td>
<td>The name and extension to use for the saved file</td>
<td>Saves the file with the given filename.</td>
</tr>
<tr>
<td>--csv</td>
<td></td>
<td></td>
<td>View only. Export the view's data (summary data) in CSV format.</td>
</tr>
<tr>
<td>--pdf</td>
<td></td>
<td></td>
<td>View only. Export as a PDF.</td>
</tr>
<tr>
<td>--png</td>
<td></td>
<td></td>
<td>View only. Export as an image in PNG format.</td>
</tr>
<tr>
<td>--fullpdf</td>
<td></td>
<td></td>
<td>Workbook only. Export as a PDF. The workbook must have been published with Show Sheets as Tabs enabled.</td>
</tr>
<tr>
<td>--pagelayout</td>
<td>landscape, portrait</td>
<td></td>
<td>Sets the page orientation of the exported PDF. If not specified, its Tableau Desktop setting will be used.</td>
</tr>
<tr>
<td>--pagesize</td>
<td>unspecified, letter, legal, note folio, tabloid,</td>
<td></td>
<td>Sets the page size of the exported PDF. Default is letter.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ledger,</td>
<td>Sets the width. Default is 800 px.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>statement,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>executive,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>a3, a4, a5,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>b4, b5,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>quarto</td>
<td></td>
</tr>
<tr>
<td>--width</td>
<td>Number of pixels</td>
<td>Sets the height. Default is 600 px.</td>
<td></td>
</tr>
<tr>
<td>--height</td>
<td>Number of pixels</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

get url

Gets the resource from Tableau Server that's represented by the specified (partial) URL. The result is returned as a file.

Note the following when you use this command:

- **Permissions**: To get a file, you must have the Download/Web Save As permission. By default, this permission is allowed or inherited for all roles, although permissions can be set per workbook or view.

- **File extension**: The URL must include a file extension, for example, "/views/Finance/InvestmentGrowth.csv". The extension (.csv) determines what's returned. A view can be returned in PDF, PNG, or CSV (summary data only) format. A Tableau workbook is returned as a TWB if it connects to a published data source or uses a live connection, or a TWBX if it connects to a data extract.

To figure out the correct extension, you can use a web browser to navigate to the item on Tableau Server and add the file extension to the end of the URL.

When you type the URL for the GET request, exclude the session ID (:iid=<n>) that appears at the end of the file name. For example, use "/views/Finance/InvestmentGrowth.pdf" instead of "/views/Finance/InvestmentGrowth?:iid=3.pdf".

**Note**: If you are downloading a view to a PDF or PNG file, and if you include a --filename parameter that includes the .pdf or .png extension, you do not have to include a .pdf or .png extension in the URL.

- **The saved file's name and location** (optional): The name you use for --filename should include the file extension. If you don't provide a name and file extension, both will
be derived from the URL string. If you don't provide a location, the file is saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

- **PNG size** (optional): If the saved file is a PNG, you can specify the size, in pixels, in the URL.

**Clearing the cache to use real-time data**

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using `tabcmd` with your own scripting, using the `refresh` parameter a great deal can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

**Examples**

**Views**

```bash
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename "Weekly-Report.png"
```

```bash
tabcmd get "/views/Finance/InvestmentGrowth.pdf" -f "Q1Growth.pdf"
```

```bash
tabcmd get "/views/Finance/InvestmentGrowth" -f "Q1Growth.pdf"
```

```bash
tabcmd get "/views/Finance/InvestmentGrowth.csv"
```

```bash
tabcmd get "/views/Finance/InvestmentGrowth.png?:size=640,480" -f growth.png
```

```bash
tabcmd get "/views/Finance/InvestmentGrowth.png?:refresh=yes" -f growth.png
```

**Workbooks**

```bash
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Workbooks\Weekly-Reports.twb"
```

**initialuser**

Create the initial administrative user on a server that does not have an initial administrative user defined.
**Note:** The `tabcmd initialuser` command does not require authentication to Tableau Server, but you must run the command on the primary server node.

**Examples**

```
tabcmd initialuser --username "admin" --password "P@ssword!"

tabcmd initialuser --username "admin" --password "P@ssword!" --friendly "Tableau Admin"
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>--friendly</td>
<td>Display name for the user</td>
<td>Creates the initial administrative user with the display name.</td>
</tr>
</tbody>
</table>

**listdomains**

Displays a list of the Active Domain domains that are in use on the server, along with their nicknames and IDs. If the server is configured to use local authentication, the command returns only the domain name `local`.

**Example**

```
tabcmd listdomains
```

**listsites**

Returns a list of sites to which the logged in user belongs.

**Example**

```
tabcmd listsites --username adam --password P@ssword!
```

**login**

Logs in a Tableau Server user.

Use the `--server, --site, --username, --password` global options to create a session.

**Note:** When you use the `tabcmd login` command, you cannot use SAML single sign-on (SSO), even if the server is configured to use SAML. To log in, you must pass the user name and password of a user who has been created on the server. You will have the permissions of the Tableau Server user that you're signed in as. For more information, see [Site Roles for Users on page 788](#) and [Manage Permissions on page 834](#).
If you want to log in using the same information you've already used to create a session, just specify the --password option. The server and user name stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify the port.

You need the --site (-t) option only if the server is running multiple sites and you are logging in to a site other than the Default site. If you do not provide a password you will be prompted for one. If the --no-prompt option is specified and no password is provided the command will fail.

Once you log in, the session will continue until it expires on the server or the logout command is run.

Example

Logs you in to the Tableau Server running on your local machine:

tabcmd login -s http://localhost -u jsmith -p p@ssW0rd!

Logs you in to the Sales site on sales-server:

tabcmd login -s http://sales-server -t Sales -u administrator -p p@ssW0rd!

tabcmd login -s http://sales-server:8000 -t Sales -u administrator -p p@ssW0rd!

Logs you in to the Sales site on sales-server using SSL but does not validate the server’s SSL certificate:

tabcmd login --no-certcheck -s https://sales-server -t Sales -u administrator -p p@ssW0rd!

Establishes a forward proxy and port for localhost:

tabcmd login --proxy myfwdproxyserver:8888 -s http://localhost -u jsmith -p p@ssW0rd!

Logs you in to the reverse proxy using SSL:

tabcmd login -s https://myreverseproxy -u jsmith -p p@ssW0rd!

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s</td>
<td>--server</td>
<td>server URL</td>
<td>If you are running the command from an on-premises Tableau Server computer, you can use <a href="http://localhost">http://localhost</a>. Otherwise, specify the computer's URL, such as</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-t</td>
<td>--site</td>
<td>site ID</td>
<td>Include this option if the server has multiple sites, and you are logging in to a site other than the Default site. The site ID is used in the URL to uniquely identify the site. For example, a site named West Coast Sales might have a site ID of west-coast-sales.</td>
</tr>
<tr>
<td>-u</td>
<td>--username</td>
<td>user name</td>
<td>The user name of the user logging in. For Tableau Online, the user name is the user's email address.</td>
</tr>
<tr>
<td>-p</td>
<td>--password</td>
<td>password</td>
<td>Password for the user specified for --username. If you do not provide a password you will be prompted for one.</td>
</tr>
<tr>
<td></td>
<td>--password-file</td>
<td>filename.txt</td>
<td>Allows the password to be stored in the given file rather than the command line, for increased security.</td>
</tr>
<tr>
<td>-x</td>
<td>--proxy</td>
<td>Host:Port</td>
<td>Use to specify the HTTP proxy server and port for the tabcmd request.</td>
</tr>
<tr>
<td></td>
<td>--no-prompt</td>
<td></td>
<td>Do not prompt for a password. If no password is specified, the login command will fail.</td>
</tr>
<tr>
<td></td>
<td>--no-proxy</td>
<td></td>
<td>Do not use an HTTP proxy server.</td>
</tr>
<tr>
<td></td>
<td>--cookie</td>
<td></td>
<td>Saves the session ID on login. Subsequent commands will not require a login. This value is the default for the command.</td>
</tr>
<tr>
<td></td>
<td>--no-cookie</td>
<td></td>
<td>Do not save the session ID information after a successful login. Subsequent com-</td>
</tr>
</tbody>
</table>


For Tableau Online specify the URL https://online.tableau.com.
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--timeout SECONDS</td>
<td>Number of seconds</td>
<td>The number of seconds the server should wait before processing the login command. Default: 30 seconds.</td>
</tr>
</tbody>
</table>

**logout**

Logs out of the server.

**Example**

tabcmd logout

**publish filename.twb(x), filename.tds(x), or filename.tde**

Publishes the specified workbook (.twb(x)), data source (.tds(x)), or data extract (.tde) to Tableau Server.

If you are publishing a workbook, by default, all sheets in the workbook are published without database user names or passwords.

The permissions initially assigned to the workbook or data source are copied from the project that the file is published to. Permissions for the published resource can be changed after the file has been published.

If the workbook contains user filters, one of the thumbnail options must be specified.

**Example**

tabcmd publish "analysis.twbx" -n "Sales_Analysis"
--db-username "jsmith" --db-password "p@ssw0rd"

tabcmd publish "analysis_sfdc.tde" -n "Sales Analysis"
--oauth-username "username" --save-oauth

If the file is not in the same directory as tabcmd, include the full path to the file.

**Example**

tabcmd publish "C:\Tableau Workbooks\analysis.twbx" -n "Sales_Analysis"
--db-username "jsmith" --db-password "p@ssw0rd"

tabcmd publish "C:\Tableau Workbooks\analysis_sfdc.tde" -n "Sales Analysis"
--oauth-username "username" --save-oauth
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>--name</td>
<td>Name of the workbook or data source on the server</td>
<td>If omitted, the workbook, data source, or data extract will be named after filename.</td>
</tr>
<tr>
<td>-o</td>
<td>--overwrite</td>
<td></td>
<td>Overwrites the workbook, data source, or data extract if it already exists on the server.</td>
</tr>
<tr>
<td>-r</td>
<td>--project</td>
<td>Name of a project</td>
<td>Publishes the workbook, data source, or data extract into the specified project. Publishes to the “Default” project if not specified.</td>
</tr>
<tr>
<td>--db-username</td>
<td></td>
<td></td>
<td>Use this option to publish a database user name with the workbook, data source, or data extract.</td>
</tr>
<tr>
<td>--db-password</td>
<td></td>
<td></td>
<td>Use this option to publish a database password with the workbook, data source, or data extract.</td>
</tr>
<tr>
<td>--save-db-password</td>
<td></td>
<td></td>
<td>Stores the provided database password on the server.</td>
</tr>
<tr>
<td>--oauth-username</td>
<td>Email address of the user account</td>
<td></td>
<td>Connects the user through a pre-configured OAuth connection, if the user already has a saved access token for the cloud data source specified in --name. Access tokens are managed in user preferences. For existing OAuth connections to the data source, use this option instead of --db-username and --db-password.</td>
</tr>
<tr>
<td>--save-oauth</td>
<td></td>
<td></td>
<td>Saves the credential specified by --oauth-username as an embedded credential with the published workbook or data source. Subsequently, when the publisher or</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>server administrator signs in to the server and edits the connection for that workbook or data source, the connection settings will show this OAuth credential as embedded in the content. If you want to schedule extract refreshes after publishing, you must include this option with <code>--oauth-username</code>. This is analogous to using <code>--save-db-password</code> with a traditional database connection.</td>
</tr>
<tr>
<td>--thumb-nail-user-name</td>
<td></td>
<td></td>
<td>If the workbook contains user filters, the thumbnails will be generated based on what the specified user can see. Cannot be specified when <code>--thumbnail-group</code> option is set.</td>
</tr>
<tr>
<td>--thumb-nail-group</td>
<td></td>
<td></td>
<td>If the workbook contains user filters the thumbnails will be generated based on what the specified group can see. Cannot be specified when <code>--thumb-nail-user-name</code> option is set.</td>
</tr>
<tr>
<td>--tabbed</td>
<td></td>
<td></td>
<td>When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.</td>
</tr>
<tr>
<td>--append</td>
<td></td>
<td></td>
<td>Append the extract file to the existing data source.</td>
</tr>
<tr>
<td>--replace</td>
<td></td>
<td></td>
<td>Use the extract file to replace the existing data source.</td>
</tr>
<tr>
<td>--disable-uploader</td>
<td></td>
<td></td>
<td>Disable the incremental file uploader.</td>
</tr>
<tr>
<td>--restart</td>
<td></td>
<td></td>
<td>Restart the file upload.</td>
</tr>
</tbody>
</table>
refreshextracts **workbook-name or datasource-name**

Performs a full or incremental refresh of extracts belonging to the specified workbook or data source.

This command takes the name of the workbook or data source as it appears on the server, not the file name when it was published. Only an administrator or the owner of the workbook or data source is allowed to perform this operation.

**Examples**

tabcmd refreshextracts --datasource sales_ds

tabcmd refreshextracts --workbook "My Workbook"

tabcmd refreshextracts --url SalesAnalysis

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--incremental</td>
<td></td>
<td></td>
<td>Runs the incremental refresh operation.</td>
</tr>
<tr>
<td>--synchronous</td>
<td></td>
<td></td>
<td>Adds the full refresh operation to the queue used by the Backgrounder process, to be run as soon as a Backgrounder process is available. If a Backgrounder process is available, the operation is run immediately. The refresh operation appears on the Background Tasks report. During a synchronous refresh, tabcmd maintains a live connection to the server while the refresh operation is underway, polling every second until the background job is done.</td>
</tr>
<tr>
<td>--workbook</td>
<td>Name of a workbook</td>
<td></td>
<td>The name of the workbook containing extracts to refresh. If the workbook has spaces in its name, enclose it in quotes.</td>
</tr>
<tr>
<td>--datasource</td>
<td>Name of a data source</td>
<td></td>
<td>The name of the data source containing extracts to refresh.</td>
</tr>
</tbody>
</table>
| --project      | Name of a project |          | Use with --workbook or --datasource to identify a workbook or data source in a project other than Default. If not
### removeusers group-name
Removes users from the specified group.

**Example**

```bash
tabcmd removeusers "Development" --users "users.csv"
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| --url          | URL name of a workbook | The name of the workbook as it appears in the URL. A workbook published as “Sales Analysis” has a URL name of “SalesAnalysis”.

### runschedule schedule-name
Runs the specified schedule.

This command takes the name of the schedule as it is on the server.

For Tableau Online, the command can be run within the scope of a single site, using site administrator permissions.

**Example**

```bash
tabcmd runschedule "5AM Sales Refresh"
```

### set setting
Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server.
Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- `allow_scheduling`
- `embedded_credentials`
- `remember_passwords_forever`

Example

```
tabcmd set embedded_credentials
```

**syncgroup group-name**

Synchronizes a Tableau Server group with an Active Directory group. If the Tableau Server group does not already exist, it is created and synchronized with the specified Active Directory group.

If the group name itself includes an "@" (other than as the domain separator) you need to refer to the symbol using the hex format "\x40".

Example

```
tabcmd syncgroup "Development"
```
```
tabcmd syncgroup "Dev\x40Fremont"
```

**Note:** If you synchronize a group that you are a member of, changes that you make using this command do not apply to your user. For example, if you use this command to remove the administrator right from users in a group that you are a member of, you are still an administrator when the command finishes.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>--administrator</td>
<td>System, Site, or None</td>
<td>(Deprecated. Some operations may no longer work. Use the --role option instead.) Assigns or removes the administrator right for users in the group. The</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td></td>
<td>None option removes the administrator right from all users in the group (except you, if you are a member of the group that you are synchronizing). If you do not include this option, users who are added to the group after you run the command are not assigned the administrator right.</td>
</tr>
</tbody>
</table>
| --license     | Interactor, Viewer, or Unlicensed |          | (Deprecated. Some operations may no longer work. Use the --role option instead.) Specifies the license level for users in the group.  

**Note:** License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting in Tableau Server 9.0. |
<p>| --no-publisher |              |          | (Deprecated. Some operations may no longer work. Use the --role option instead.) Disallows publishing rights for users in the group. |</p>
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--overwritesiterole</td>
<td></td>
<td></td>
<td>Allows a user's site role to be overwritten with a less privileged one when using --role. By default, a user site role can be promoted when using --role, but cannot be demoted. Because the --overwritesiterole option will demote user site roles, use it with caution.</td>
</tr>
<tr>
<td></td>
<td>--publisher</td>
<td></td>
<td>(Deprecated. Some operations may no longer work. Use the --role option instead.) Assigns publishing rights to users in the group.</td>
</tr>
<tr>
<td>-r</td>
<td>--role</td>
<td>ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed</td>
<td>Specifies a role for users in the group. The default is Unlicensed. <strong>Note:</strong> If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>--silent-progress</td>
<td>Do not display progress messages for the command.</td>
</tr>
</tbody>
</table>
version

Displays the version information for the current installation of the tabcmd utility.

Example

tabcmd version
Server Administrator Guide

The Server Administrator Guide is your complete reference for handling administrative tasks on Tableau Server.
Before you install...

Note: You can find additional information about technical specifications for Tableau Server on the Tableau web site, here.

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


- **Supported browsers**—Tableau Server 10 supports Internet Explorer 11 in native mode, and the latest versions of Chrome, Firefox, and Safari.

This has potential to impact:

- Customers installing Tableau Server for the first time on Windows 8 or Windows Server 2012 (non-R2). For more information, see Internet Explorer Support.

- Customers accessing embedded Tableau views in web pages that force Internet Explorer into compatibility mode. For more information, see Internet Explorer Compatibility Mode.

- **Minimum requirements**—The computer you install Tableau Server on must meet or exceed the minimum hardware requirements. Tableau Server will not install if your computer does not meet the minimum requirements.

  - Minimum *requirements* are appropriate for testing and prototyping.

  - For production environments your computers should meet or exceed the minimum *recommendations*.

  For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server on page 674.

- **Administrative account**—The account under which you install Tableau Server must have permission to install software and services.

- **Optional: Run As Account**—A Run As User account for the Tableau Server service to run under is useful if you’re using NT Authentication with data sources or if you’re planning on doing SQL Server impersonation. For more information, see Run As User on page 577 and SQL Server Impersonation on page 1042.

- **IIS and port 80**—Tableau Server’s gateway listens on port 80, which is also used by Internet Information Services (IIS) by default. If you are installing Tableau Server on a machine that’s also running IIS, you should modify the Tableau’s gateway port number to
avoid conflict with IIS. See Tableau Server Ports on page 1244 and Edit the Default Ports on page 598 for details.

- Static IP addresses—Any computer running Tableau Server, whether it's a single server installation or part of a cluster, must have a static IP address. For more information, see Hostname Support in Tableau Server on page 698.

Configuration Information

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Your Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Account</td>
<td>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.</td>
<td>Username: Username: Password: Domain:</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Instead of using Tableau’s built-in user management system, you can authenticate through Active Directory. If so, you’ll need the fully-qualified domain name.</td>
<td>Active Directory Domain:</td>
</tr>
<tr>
<td>Open port in Windows firewall</td>
<td>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.</td>
<td>_ _ - Yes _ _ - No</td>
</tr>
</tbody>
</table>

Ports

By default Tableau Server requires several TCP/IP ports to be available to the server. See the topic Tableau Server Ports on page 1244 for the full list, including which ports must be available for all installations vs. distributed installations or failover-ready installations. The default ports can be changed if there is a conflict. See Edit the Default Ports on page 598 to learn how.

Drivers

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

What's New and What's Changed

Find out about the new and changed features in Tableau Server:
• See the What’s New in Tableau Server topic in the Tableau Server online help for information about key new features.
• See What’s Changed - Things to Know Before You Upgrade for information about changes that may impact your users.

Minimum Hardware Requirements and Recommendations for Tableau Server

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

• **Minimum requirements** are the minimum hardware your computer must have in order to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping.

• **Minimum recommendations** are higher than minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the setup program will warn you but you can continue the installation.

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

| Note: If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see Server Process Limits on page 654 |

Minimum Hardware Requirements

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

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

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>RAM</th>
<th>Free Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Hardware Requirements</td>
<td>2-core</td>
<td>8 GB</td>
<td>15 GB</td>
</tr>
</tbody>
</table>
For the requirements:

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

Note: For Tableau Server 10.0, you need a minimum of 2 physical cores. If you are installing on an Amazon EC2 instance, this means 4 vCPUs. For more information, see Amazon EC2 Instances.

Minimum Hardware Recommendations

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process.

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

Determining Computer Hardware

To determine how many physical cores a computer has, the Tableau Server setup program queries the operating system. To view hardware information that the setup program detected on your computer, open the tabadmin.log file in the following folder on the computer where you are installing Tableau Server:

<install directory>\ProgramData\Tableau\Tableau Server-\logs\tabadmin.log
In the `tabadmin.log` file, look for lines similar to the following. These lines provide information about the physical and logical cores that the setup program detected and that it used to determine the core count that is being used for licensing.

```
2015-04-09 14:22:29.533 -0700_DEBUG_10.36.2.32:<machine name>_: _
    pid=21488_0x2cd83560__user=___request=___ Running hardware check
```

```
2015-04-09 14:22:29.713 -0700_DEBUG_10.36.2.32:<machine name>_: _
    pid=21488_0x2cd83560__user=___request=___ Detected 12 cores and
    34281857024 bytes of memory
```

```
2015-04-09 14:22:29.716 -0700_DEBUG_10.36.2.32:<machine name>_: _
    pid=21488_0x2cd83560__user=___request=___ Hardware meets recom-
    mended specifications. Default values will be used.
```

Manually determining the number of cores on your computer

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

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

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

   The output will display the device ID or IDs and the number of physical cores the computer has.

   ![Command Prompt Output](image)

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

   The following command shows a longer version that lists the logical processors as well as the physical cores.
WMIC CPU Get
DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesignation

In the above example, the server has a total of twelve physical cores, resulting in 24 logical cores.

**Domain Trust Requirements**

When you run Tableau Server in an Active Directory environment across multiple domains (either in the same Active Directory forest or in different forests), some Tableau functionality is dependent on the trust relationship between the domains. For example, some administrators manage users in domains that are separate from where they deploy server applications, such as Tableau Server. In other organizations, a Tableau Server deployment might be shared with external partners or with different partners in the organization. Finally, Windows-authenticated data sources, such as SQL Server, MSAS, or Oracle, that Tableau Server connects to may also be in other domains.

If it’s feasible, we recommend configuring two-way trust between all domains that interact with Tableau Server. If this is not possible, Tableau Server can be configured to support user authentication where a one-way trust has been configured. In this case, a one-way trust between domains is supported when the domain in which Tableau Server is installed is configured to trust the domain where user accounts reside.

The following illustration shows one-way trust between the domain where Tableau Server is installed and the domain where user accounts reside:
In this scenario, Tableau Server is in the dev.local domain, and users from the users.lan Active Directory domain are imported into Tableau Server. A one-way trust is required for this scenario; specifically, the dev.local domain is configured to trust the users.lan domain. Users in the users.lan domain can access Tableau Server in the dev.local with their normal Active Directory credentials. However, you may need to update the domain nickname on Tableau Server before users log on with the nickname. Refer to the Tableau Knowledge Base for more information.

Kerberos single sign-on is supported in this one-way trust scenario.

Review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

**Connecting to live data in one-way trust scenarios**

In the one-way trust scenario, users connecting to Tableau Server can connect to live data that’s hosted in the cloud or on any other data source on premises that does not rely on Windows authentication.

Data sources that require Windows-authentication might have additional authentication requirements that complicate the scenario, or that can even prevent Tableau Server users from connecting. This is because Tableau Server uses the Run As User account for authentication with such data sources. If you are running Tableau Server in a different domain than data sources that use Windows authentication, verify that the Run As User account that is used for Tableau Server can access the data source.
Run As User

The Run As User is a Windows account that Tableau Server uses ("runs as") when it access resources. For example, Tableau Server reads and writes files on the computer where Tableau Server is installed. From the perspective of Windows, Tableau Server is doing this as the Run As User. In some cases, Tableau Server may use the Run As User account to access data from external sources, such as databases or files on a shared network directory.

As you plan your Tableau Server deployment, you need to determine if the default Run As User, configured to run under the context of the local Network Service account (NT Authority\Network Service), will suffice for your needs. If it does not, then you will need to update the Run As User to run under a domain account that has access to the resources in your Active Directory domain(s).

In either case, it’s important to understand the security implications of the account that Tableau Server uses for the Run As User. Specifically, if Tableau Server needs to access other servers, file shares, or databases that use Windows authentication, then the account that is configured for Run As User will be used to access those resources. The account that is configured for Run As User must also have elevated permissions to the local Tableau Server. A general best security practice is to limit the scope of all user accounts to the minimum required permissions. We make the same recommendation to you as you plan Run As User.

You set or update the Run As User account in the Tableau Server Configuration utility. The utility sets permissions for the Run As User, but if you are unsure if the account you want to use for Run As User satisfies the requirements, or if you have changed the Run As User and are getting permission errors, see Required Run As User Account Settings on page 1232.

![Tableau Server Configuration](image)

Default Run As User account: Network Service

The Network Service account is a predefined local account with limited permissions that exists on all Windows computers. While it has limited administrative access to the local computer on which it runs, it does have more access to resources than members of the Active Directory default Users group. For example the Network Service group can write to the registry, the event log, and has special rights to log on for application services.

By default, the Run As User is set to a local account called Network Service. Use the default Network Service account when:
You are using local authentication for Tableau Server.
All users in your organization include extracted data in the workbooks that they are uploading to Tableau Server.
You are running Tableau Server in a single-server deployment.
External data sources that your users access through Tableau Server do not require Windows NT integrated security or Kerberos. In most data-access scenarios, Microsoft SQL Server, MSAS, Teradata, and Oracle databases require Windows NT integrated security.

While the Network Service account can be used to access resources on remote computers within the same Active Directory domain we do not recommend using the default account for such scenarios. Instead, configure a domain account for Run As User if Tableau Server must connect to data sources in your environment. See Create and Update the Run As User Account below.

Run As User account: Domain user
For all Active Directory scenarios, we recommend updating the Tableau Server Run As User with a domain user account. Update the Run As User to a domain user account when data sources accessed through Tableau Server require Windows NT integrated security or Kerberos.

If you have deployed a distributed deployment of Tableau Server, then you can update the Run As User account with either a domain user or a Windows workgroup user. In either case, you must use the same user account for all server nodes. See Distributed Requirements on page 696 for more information.

To configure your environment to use a domain account, see Create and Update the Run As User Account below.

Create and Update the Run As User Account
If you are operating in an environment where a majority of your data sources are authenticated in the context of Active Directory (Windows NT integrated security) then you will need to configure the Run As User to use a domain account, not the local account (Network Service) that’s the default.

There are two steps:

1. Create the Run As User account in Active Directory
2. Update Tableau Server to use the Run As User account

Creating the Run As User account
Follow these best practices:

- Create a dedicated account in Active Directory for the Tableau Server Run As user account. In other words, don’t use an existing account. By using a dedicated account you
can be sure that the data resources that you permission for Tableau Server are only accessible by Tableau Server Run As User.

- Do not use an account with any kind of domain administrative permissions. Specifically, when you create an account in Active Directory, create an account in the domain User Group. Do not add the account that you create to any Active Directory security groups that needlessly elevate the permissions for the account.
- Permission the data sources in your directory for this one account. The account that you’ll use for Run As User only needs Read access to the appropriate data sources and network shares.

**Updating the Run As User in Tableau Server**

After you have created the Run As User account in Active Directory, configure Tableau Server to use that account as the Run As User. See [Configure General Server Options](#) on page 608 for information on how to update the Run As User account. After you update the Run As User, Tableau Server (tabadmin) will automatically configure permissions on the local computer for the Run As User that you have entered.

If you have installed Tableau Server on a drive other than the system drive, then you will need to configure the system drive to allow the Run As User additional permissions. The system drive is the drive where Windows is installed. For example, if you have installed Windows on the C:/ drive, then C:/ is your system drive. If you install Tableau Server on any other drive (D:/, E:/, etc), then you will need to configure permissions to allow the Run As User to read, execute, and modify the system drive.

**Related tasks**

The Run As User is central to many operations on Tableau Server, especially those that are involved with remote data access. To avoid access errors, review the tasks here and follow the links for those that apply to your scenario.

- If you are running Tableau Server in an organization with multiple Active Directory domains, see [Domain Trust Requirements](#) on page 575.
- Enabling Kerberos single sign-on requires additional configuration related to the Run As User. To enable Kerberos single sign-on with Tableau Server, see [Kerberos](#) on page 987.
- Enabling impersonation requires additional configuration related to Run As User. To deploy and enable impersonation with Microsoft SQL Server, see [Impersonate with Embedded SQL Credentials](#) on page 1046.
- If you have installed Tableau Server onto the non-system drive, then you will need to manually set some permissions for the Run As User. See [Required Run As User Account Settings](#) on page 1232 for more information.

**Configuring Proxies for Tableau Server**

In most enterprises, Tableau Server needs to communicate with the internet. Communications between your network and the internet should be mediated using proxy servers. Forward proxy
servers mediate traffic from inside the network to targets on the internet. Reverse proxy servers mediate traffic from the internet to targets inside the network.

**Who should read this article?**

This article is for IT professionals who are experienced with general networking and gateway proxy solutions. The article describes how and when Tableau requires internet access, and describes how to configure your network and Tableau to use forward and reverse proxy servers for access to and from the internet. There are many third-party proxy solutions available, so some of the content in the article is necessarily generic.

In this article:

- How Tableau communicates with the internet
- Configure a forward proxy server
- Configure a reverse proxy server

**How Tableau communicates with the internet**

Tableau Server requires outbound access to the internet for these scenarios:

- Working with maps. Tableau uses map data that is hosted externally. By default, Tableau uses OpenStreetMaps for map data.
  
  Tableau Server needs to connect to maps.tableausoftware.com using port 443. If it cannot make this connection, maps may fail to load.

- Licensing. Tableau products connect to the internet to activate license keys. Unless you activate Tableau software with the [Offline Activation Tool](https://tableausoftware.com/activation), all Tableau products must have continuous access to the internet to validate their licenses.

  Tableau Server needs to connection to the following internet locations for licensing purposes:
  - licensing.tableau.com:443
  - o.ss2.us
  - ocsp.rootg2.amazontrust.com
  - ocsp.rootca1.amazontrust.com
  - ocsp.sca1b.amazontrust.com
  - crt.sca1b.amazontrust.com
  - crt.rootca1.amazontrust.com
  - ocsp.sca0a.amazontrust.com
  - crt.sca0a.amazontrust.com
  - ocsp.sca1a.amazontrust.com
Requests to the above domains may be on port 80 or 443.

If Tableau Server cannot make a connection while attempting to activate its license, you will be prompted to do an offline activation.

- Working with external or cloud-based data.

Tableau Server can run without internet access, but in most organizations, the scenarios in the list require Tableau to be able to access the internet.

To configure access to the internet from Tableau Server, you should use a forward proxy.

**Note:** Both Tableau Desktop and Tableau Server need to communicate with the internet for mapping, licensing, and external data. In this article, we focus on Tableau Server, which has specific requirements for configuring internet access. Do not set up Tableau Server on the computer that's acting as your organization's internet gateway.

In many enterprises, users also need to access Tableau Server from outside the network (that is, from the internet). For example, in many enterprises, users want to be able to reach Tableau Server from their mobile devices in order to interact with views that are stored on the server. To configure access to Tableau Server from the internet or from mobile devices, you should use a reverse proxy.

**Configure a forward proxy server**

To enable communication from Tableau Server to the internet, deploy Tableau Server behind a forward proxy server. When Tableau Server needs access to the internet, it doesn't send the request directly to the internet. Instead, it sends the request to the forward proxy, which in turn forwards the request. Forward proxies help administrators manage traffic out to the internet for tasks such as load balancing, blocking access to sites, etc.

If you use a forward proxy, you must configure the computers that run Tableau Server inside the network to send traffic to the forward proxy.

**Note:** If you know that none of your users need access to map data or online data sources in the workbooks that they'll be publishing to Tableau Server, and if you are
configuring Tableau Server for offline licensing, you can skip this section. Otherwise, you'll need to configure Tableau Server to connect to the internet.

Configuring Tableau Server to work with a forward proxy

The steps for configuring internet options on the Tableau Server computer depend on which of these scenarios describes your enterprise:

- **Your organization doesn't use a forward proxy solution.** If your organization is not running a proxy solution and the computer where you are installing Tableau Server can communicate with the internet, you don’t need to follow the procedures here.

- **A proxy solution is deployed, and automatic configuration files define connection settings.** If your organization uses automatic configuration files (such as PAC or .ins files) to specify internet connection information, you can use this information in the Local Area Network (LAN) Settings dialog box in Windows. For more information, see Automatic Detection and Configuration of Browser Settings on the Microsoft support site.

- **A proxy solution is deployed, but automatic configuration files are not deployed.** For this scenario, you must configure LAN settings so that connections to your proxy server are run under the security context of the Run As User account. You must also configure localhost and other internal Tableau Server instances as exceptions.

The following procedure describes the steps for the last scenario—a proxy solution without automatic configuration files.

**Note:** If you are using a distributed installation of Tableau Server, perform the following procedures on the primary server and on each worker node.

**Step 1: Add the Run As User account to the Local Administrators group**

To perform this procedure, you must log onto the Tableau Server computer as the Run As User. By default, the "log on locally" policy is not applied to the Run As User account. Therefore, you must temporarily add the Run As User account to the Local Administrators group.

If you haven't installed Tableau Server on the computer yet, see Run As User for more information about creating the Run As User account. If you already installed Tableau Server and set the Run As User setting, you can determine the Run As User account name by logging onto Tableau Server. The Tableau Server Run As User is listed on the General tab of the Tableau Server Configuration window. To access the configuration utility, in the Windows Start menu, search for Configure Tableau Server.
Add the Run As User to the Local Administrators group using steps in Add a member to a local group on the Microsoft website. When you've finished configuring the forward proxy information, you'll remove the Run As User account from the Local Administrators group.

Step 2: Configure the proxy server in Windows LAN Settings

1. Using the Run As User account, log onto the computer where Tableau Server is installed or will be installed.

2. Open the Local Area Network (LAN) Settings dialog box. (A quick way to get to this dialog box is to search for Internet Options in the Windows Start menu. In the Internet Properties dialog box, click the Connections tab, and then click LAN settings.)

3. Under Proxy server, select Use a proxy server for your LAN, enter the proxy server address and port, and then select Bypass proxy server for local addresses.

![Local Area Network (LAN) Settings](image)

Leave this dialog box open and continue to the next step.

Step 3: Add exceptions to bypass the proxy server

You add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the
In the LAN settings dialog box, click **Advanced**. (This button is available only if you've selected the option to use a proxy server for your LAN.)

2. In the **Proxy Settings** dialog box, enter **localhost** in the **Exceptions** field. In addition, enter the server names and IP addresses of other Tableau Server computers in the same cluster. Use semicolons to separate items.

3. Close the proxy settings dialog box and the Local Area Network (LAN) Settings dialog box.

4. In the **Internet Properties** dialog box, click **OK** to apply the settings.

Stay logged onto the computer and continue to the next step.

**Step 4: Test the proxy configuration**

To test the new configurations, while still logged on as the Run As User on the Tableau Server computer, open a web browser and test the following Tableau mapping URL:

**Miami and Havana (blue water)**

This is the URL:

https://maps.tableausoftware.com(tile/d/mode=named|from=tableau1_2_base/mode=named|from=tableau1_2_admin0_borders/mode=named|from=tableau1_2_place_labels/ol/6/17/27.png?apikey=ttab56540ba691a909b0f7d2af0f6fe7"

If the configuration is working, you see a map of Miami and Havana. This indicates that the Tableau Server computer is able to access the internet through the proxy.

**Step 5: Remove the Run As User account from the Local Administrator group**

After you have tested the proxy settings, remove the Run As User account from the Local Administrators group. Leaving the Run As User in the administrator group unnecessarily elevates the permissions of the Run As User group and is a security risk.

Restart Tableau Server to ensure that all changes are implemented.

**Configure a reverse proxy server**

A reverse proxy is a server that receives requests from external (internet) clients and forwards them to Tableau Server. Why use a reverse proxy? The basic answer is security. A reverse proxy makes Tableau Server available to the internet without having to expose the individual IP address of that particular Tableau Server to the internet. A reverse proxy also acts as an authentication and pass-through device, so that no data is stored where people outside the company can get to it. This requirement can be important for organizations that are subject to various privacy regulations such as PCI, HIPAA, or SOX.
How a reverse proxy works with Tableau Server

The following diagram illustrates the communication path when a client makes a request to Tableau Server that is configured to work with a reverse proxy server.

1. An external client initiates a connection to Tableau Server. The client uses the public URL that's been configured for the reverse proxy server, such as https://tableau.example.com. (The client doesn't know that it's accessing a reverse proxy.)

2. The reverse proxy maps that request in turn to a request to Tableau Server. The reverse proxy can be configured to authenticate the client (using SSL/TLS) as a precondition to passing the request to Tableau Server.

3. Tableau Server gets the request and sends its response to the reverse proxy.

4. The reverse proxy sends the content back to the client. As far as the client is concerned, it just had an interaction with Tableau Server, and has no way to know that the communication was mediated by the reverse proxy.

Proxy servers and SSL

For better security, you should configure reverse proxy servers to use SSL for any traffic that's external to your network. This helps to ensure privacy, content integrity, and authentication. Unless you've deployed other security measures to protect traffic between your internet gateway and Tableau Server, we also recommend configuring SSL between the gateway proxy and Tableau Server. You can use internal or self-signed certificates to encrypt traffic between Tableau Servers and other internal computers.

Reverse proxy and user authentication

Tableau Server will always authenticate users. This means that even if you are authenticating inbound connections at the gateway for your organization, Tableau Server will still authenticate the user. Therefore, we recommend a transparent scenario where Tableau Desktop, Tableau Mobile, or browser user requests are not prompted for authentication at the gateway. This recommendation doesn't prohibit using SSL for client/server system-level authentication at the gateway proxy, in fact, we strongly recommend SSL system-level authentication.

You can use SAML, OpenID Connect, or Trusted Tickets with a reverse proxy.

If your organization is authenticating with Active Directory:
- Active Directory with Enable automatic logon (SSPI) is not supported with a reverse proxy.
- Tableau Server must be configured for reverse proxy before configuring Tableau Server for Kerberos. For more information, see Configure Kerberos on page 993.

**Configure Tableau Server to work with a reverse proxy server**

Before you configure Tableau Server, you'll need to collect the following information about the proxy server configuration. To configure Tableau Server, you use the `tabadmin` utility. The information you need to collect corresponds to options you'll need when you run `tabadmin`.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Corresponding <code>tabadmin</code> option</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address or CNAME</td>
<td>You can either enter an IP address or a CNAME for this option. The public IP address or addresses of the proxy server. The IP address must be in IPv4 format, such as 203.0.113.0, and it must be a static IP. If you are unable to provide a static IP, or if you are using cloud proxies or external load balancers, you can specify the CNAME (Canonical Name) DNS value that clients will use to connect to Tableau Server. This CNAME value must be configured on your reverse proxy solution to communicate with Tableau Server.</td>
<td>gateway.trusted</td>
</tr>
<tr>
<td>FQDN</td>
<td>The fully qualified domain name that people use to reach Tableau Server, such as <code>tableau.example.com</code>. Tableau Server doesn't support a FQDN with information beyond the domain name, such as <code>example.com/tableau</code>.</td>
<td>gateway.public.host</td>
</tr>
<tr>
<td>Non-FQDN</td>
<td>Any subdomain names for the proxy server. In the example of <code>tableau.example.com</code>, the subdomain name is <code>tableau</code>.</td>
<td>gateway.trusted_hosts</td>
</tr>
<tr>
<td>Aliases</td>
<td>Any public alternative names for the proxy server. In most cases, aliases are designated using CNAME values. An example would be a proxy server <code>bigbox.example.com</code> and CNAME entries of <code>ftp.example.com</code> and <code>www.example.com</code>.</td>
<td>gateway.trusted_hosts</td>
</tr>
<tr>
<td>Ports</td>
<td>Port numbers for traffic from the client to the server.</td>
<td>gateway.public.port</td>
</tr>
</tbody>
</table>
If you are using a distributed installation of Tableau Server, then run the following procedure on the primary node in your cluster.

1. **Open a command prompt and navigate to the Tableau Server bin directory.**

   1. Open a command prompt as an administrator:

   2. Enter the following to change to the folder where `tabadmin.exe` is located:

   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   ```

   2. Enter the following command to stop Tableau Server:

   ```
   tabadmin stop
   ```

   3. Enter the following command to set the FQDN that clients will use to reach Tableau Server through the proxy server, where `name` is the FQDN:

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

   For example, if Tableau Server is reached by entering `https://tableau.example.com` in the browser, enter this command:

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

   4. Enter the following command to set the address or the CNAME of the proxy server,
where `server_address` is the IPv4 address or CNAME value:

```
tabadmin set gateway.trusted "server_ip_address"
```

If your organization uses multiple proxy servers, enter multiple IPv4 addresses, separating them with commas. IP ranges are not supported. To improve start up and initialization of Tableau Server, minimize the number of entries for `gateway.trusted`.

5. Enter the following command to specify alternate names for the proxy server, such as its fully qualified domain name, any not fully qualified domain names, and any aliases. If there’s more than one name, separate the names with a comma.

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

For example:

```
tabadmin set gateway.trusted_hosts "proxy1.example.com, proxy1, ftp.example.com, www.example.com"
```

6. If the proxy server is using SSL to communicate with the internet, run the following command, which tells Tableau that the reverse proxy server is using port 443 instead of port 80:

```
tabadmin set gateway.public.port "443"
```

**Note:** If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server. See [Configure External SSL](#) on page 972.

7. Enter the following command to commit the configuration change:

```
tabadmin config
```

8. Enter the following command to restart the server:

```
tabadmin start
```

**Configure the reverse proxy server to work with Tableau Server**

When a client accesses Tableau Server through a reverse proxy, specific message headers have to be preserved (or added). Specifically, all proxy servers in the message chain must be represented in the `gateway.trusted` and `gateway.trusted_hosts` settings.

The following graphic shows example headers for a single-hop message chain, where the proxy server is communicating directly with Tableau Server:
The following graphic shows example headers for a multiple-hop message chain, where the message traverses two proxy servers before connecting to Tableau Server:

The following table describes what these headers are and how they relate to the configuration settings on Tableau Server:

<table>
<thead>
<tr>
<th>Headers</th>
<th>Description</th>
<th>Related Tableau Server settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMOTE_ADDR and X-FORWARDED-FOR (XFF)</td>
<td>Tableau Server needs these headers to determine the IP address of origin for requests. X-FORWARDED-FOR header must present IP address chain to Tableau Server in the order the connections have occurred.</td>
<td>The IP address that you set in gateway.trusted must match the IP presented in REMOTE_ADDR. If you sent multiple addresses in gateway.trusted, one of them must match the IP presented in REMOTE_ADDR.</td>
</tr>
<tr>
<td>HOST and X-FORWARDED HOST (XFH)</td>
<td>These headers are used to generate absolute links to Tableau Server when it replies to the client. X-FORWARDED-HOST header must present host names to Tableau Server in the order the connections have occurred.</td>
<td>The host names that are presented in X-FORWARDED-HOST header must be included in the host names that you specify in gateway.trusted.hosts.</td>
</tr>
<tr>
<td>X-FORWARDED-PROTO (XFP)</td>
<td>This header is required if SSL is enabled for traffic from the client to Port configuration on reverse proxy (inbound connections)</td>
<td></td>
</tr>
</tbody>
</table>
the proxy, but not for traffic from the proxy to Tableau Server. The `X-FORWARDED-PROTO` headers are important for scenarios where HTTP or HTTPS is not maintained along each hop of the message route. For example, if the reverse proxy requires SSL for outside requests, but traffic between the reverse proxy and Tableau Server is not configured to use SSL, `X-FORWARDED-PROTO` headers are required. Some proxy solutions add the `X-FORWARDED-PROTO` headers automatically, while others do not. Finally, depending on your proxy solution, you might have to configure port forwarding to translate the request from port 443 to port 80. from client and outbound connections to Tableau Server) must be specified in the corresponding parameter: `gateway.public.port`, which is the port clients use to connect to the proxy.

If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server. See Configure External SSL on page 972.

### Validate reverse proxy setup

To validate your reverse proxy setup, perform the following tasks from a computer on the internet.

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log in to Tableau Server from Tableau Desktop.</td>
<td>Sign in to Tableau Server or Online</td>
</tr>
<tr>
<td>Publish to Tableau Server.</td>
<td>Publish a Workbook</td>
</tr>
<tr>
<td>Open workbook from Tableau Server.</td>
<td>Opening Workbooks from the Server</td>
</tr>
<tr>
<td>Log out Server (with Desktop).</td>
<td>Sign in to Tableau Server or Online</td>
</tr>
<tr>
<td>Log into Tableau Server from a web browser.</td>
<td>Sign in</td>
</tr>
<tr>
<td>Download workbook from a web browser.</td>
<td>Download Workbooks</td>
</tr>
<tr>
<td>Check to make sure tabcmd (from a non-server client)</td>
<td>How to Use tabcmd on</td>
</tr>
</tbody>
</table>
Tableau Server Ports

The following table lists the ports that Tableau Server uses by default, and which must be available for binding. If you install multiple instances of a process (Cache Server for example) on a node, consecutive ports are used, starting at the base port. If Windows Firewall is enabled, Tableau Server will open the ports it needs for internal communication between processes. (There are circumstances when you may need to take action in addition. If you are making an external connection to the Tableau Server database you may need to open ports manually. If you have a distributed installation with a worker running Windows 7, see the Tableau Knowledge Base.)

Dynamic port remapping

When dynamic port remapping is enabled (the default), Tableau Server first attempts to bind to the default ports, or to user-configured ports if they are defined. If the ports are not available, Tableau Server attempts to remap most processes to other ports, starting at port 8000. When next restarted, Tableau Server will revert to using the default or configured ports.

The gateway port and SSL port are not dynamically remapped. If port 80 is not available when Tableau Server is first installed, the installation program will choose a different gateway port (usually 8000). This value will display on the General tab of the Configuration utility. Tableau Server will always use the port shown in the Configuration utility for the gateway process.

When dynamic port remapping is disabled, Tableau Server does not attempt to remap processes and if a conflict is detected, Tableau Server will not start.

**Note:** Port conflicts can affect how JMX ports are determined. For more information, see Enable the JMX Ports on page 600.

You can disable dynamic port remapping using the `tabadmin set service.port_remapping.enabled` command. For more information, see `tabadmin set options` on page 1294.

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP</td>
<td>Gateway</td>
<td>X</td>
</tr>
</tbody>
</table>

- 591 -
<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port.</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-------------</td>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>3729</td>
<td>TCP</td>
<td>Tableau Server setup</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>373-0-3731</td>
<td>TCP</td>
<td>Tableau worker servers in distributed and highly available environments (the primary Tableau Server does not listen on these ports).</td>
<td>X X</td>
<td>--</td>
</tr>
<tr>
<td>5000</td>
<td>UDP</td>
<td>Server Worker Manager process (tabadmwkrk.exe) that is used for auto-discovery of worker servers in a distributed environment.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>6379</td>
<td>TCP</td>
<td>Cache Server process (redis-server.exe). Base port 6379. Consecutive ports after 6379 are used, up to the number of processes.</td>
<td>X</td>
<td>workerX.cacheserver.port</td>
</tr>
<tr>
<td>8060</td>
<td>TCP</td>
<td>PostgreSQL database</td>
<td>X</td>
<td>psql.port</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/U-DP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
</tr>
<tr>
<td>------</td>
<td>----------</td>
<td>-------------</td>
<td>----------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>8061</td>
<td>TCP</td>
<td>PostgreSQL database. Used for verifying integrity of database for restoring.</td>
<td>All</td>
<td>Distributed</td>
</tr>
<tr>
<td>8062</td>
<td>TCP</td>
<td>PostgreSQL database</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8080</td>
<td>TCP</td>
<td>Solr, Tomcat HTTP, and Repository processes</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8085</td>
<td>TCP</td>
<td>Tomcat HTTP</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8250</td>
<td>TCP</td>
<td>Background tasks</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8350</td>
<td>TCP</td>
<td>Background tasks</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8600</td>
<td>TCP</td>
<td>Application Server process (vizportal.exe). Base port 8600. Consecutive ports after 8600 are used, up to the number of processes.</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>8700</td>
<td>TCP</td>
<td>Application Server process (vizportal.exe)</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>All</td>
<td>Distributed</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>-----</td>
<td>-------------</td>
</tr>
<tr>
<td>8755</td>
<td>TCP</td>
<td>Tableau Administrative process</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>910-0-9199</td>
<td>TCP</td>
<td>VizQL Server process (base port 9100). Consecutive ports after 9100, up to the number of processes, are also used. By default, Tableau Server installs with two VizQL Server processes (ports 9100 and 9101).</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>9200, 9400</td>
<td>TCP</td>
<td>VizQL Server process</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>9345</td>
<td>TCP</td>
<td>File Store service</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>9346</td>
<td>TCP</td>
<td>File Store status service</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>970-0-9899</td>
<td>TCP</td>
<td>Data Server process (base port 9700). Consecutive ports after 9700, up to the number of pro-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>-------------</td>
<td>----------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9800, 1000-0</td>
<td>TCP</td>
<td>Data Server process</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>1100-0</td>
<td>TCP</td>
<td>Search server</td>
<td>X  X</td>
<td>workerX.search-server.port</td>
</tr>
<tr>
<td>1110-0</td>
<td>TCP</td>
<td>Search server</td>
<td>X  X</td>
<td>workerX.search-server.startup.port</td>
</tr>
<tr>
<td>1200-0</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) client port</td>
<td>X</td>
<td>workerX.zookeeper.port</td>
</tr>
<tr>
<td>1201-2</td>
<td>TCP</td>
<td>Cluster Controller process</td>
<td>x  x</td>
<td>cluster.status.port</td>
</tr>
<tr>
<td>1300-0</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) leader port</td>
<td>x</td>
<td>zookeeper.config.leaderPort</td>
</tr>
<tr>
<td>1400-0</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) leader election port</td>
<td>x</td>
<td>zookeeper.config.leaderElectPort</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>TCP</td>
<td></td>
<td><strong>Workers and primary server to communicate licensing information in distributed and highly available environments.</strong></td>
<td><strong>All Distributed High Availability</strong></td>
<td></td>
</tr>
<tr>
<td>2700-0–2700-9</td>
<td>TCP</td>
<td>X</td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>TCP</td>
<td></td>
<td><strong>One additional port is dynamically chosen for workers and the primary server to communicate licensing information in distributed and highly available environments. Instead, you can specify a fixed port (27010 is recommended). See the Tableau Knowledge Base for details.</strong></td>
<td>X</td>
<td>--</td>
</tr>
<tr>
<td>TCP</td>
<td></td>
<td><strong>Data Engine process. Tableau Server</strong></td>
<td>X</td>
<td>dataengine.port</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------------------</td>
<td>----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>installs with one Data Engine process. There can be up to two Data Engine processes per node.</td>
<td>All Distributed High Availability Parameter</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Edit the Default Ports**

Tableau Server processes are configured to use certain ports on the computer where the server is installed. For more information, see Tableau Server Ports on page 1244.

In general, you do not need to make changes to the port assignments for the server processes. However, if the computer that's running Tableau Server is also running other software that uses ports (this is not recommended), it's possible that the port assignments for Tableau Server processes conflict with ports used by the other software. In that case, you can assign different ports to Tableau Server processes.

To modify the ports used by Tableau Server processes, you use the command line administrative tool (tabadmin on page 1255). For example, the default port for the application server process (vizportal.exe) is 8000. You can use the tabadmin parameter workerX.vizportal.port to change it to a different port.

*Note:* Changing ports requires a restart of Tableau 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 or schedule this maintenance during non-business hours.

Follow the steps below to change the Tableau Server port configuration. If you are enabling the server's JMX ports, see Enable the JMX Ports on page 600

1. Open a command prompt as an administrator and type the following:
   
   `cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   `tabadmin set <workerX>.<parameter> <new port value>`

2. Modify a port value by typing one of the following commands:
tabadmin set <parameter> <new port value>

where:

- `<workerX>` indicates which machine in a cluster you want to change the process port for. The placeholder X refers to the worker number—worker0 is the primary server (or the only server if you are not running a distributed server), worker1 is the first worker server, worker2 is the second worker server, and so on. If you are running a distributed server and you want to change the default port for a process on all machines in the cluster, you need to run the command (from a command prompt on the primary) once for each machine in the cluster.

- `<parameter>` is the server process that you are setting the port for, such as vizportal.port.

- `<new port value>` is the new port number you want the server process to use.

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

```bash
tabadmin set worker0.vizportal.port 8020
```

The following example sets the port for a 3-machine cluster (one primary and two workers) to 9200 for the VizQL server process.

```bash
tabadmin set worker0.vizqlserver.port 9200
tabadmin set worker1.vizqlserver.port 9200
tabadmin set worker2.vizqlserver.port 9200
```

You can use the following parameters to modify the corresponding ports—see Tableau Server Ports on page 1244 for a complete list of tabadmin parameters that can be set.

<table>
<thead>
<tr>
<th>Port to Change</th>
<th>Parameter</th>
<th>Multiple workers?</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>gateway.public.port</td>
<td>No</td>
</tr>
<tr>
<td>80</td>
<td>workerX.gateway.port</td>
<td>Yes</td>
</tr>
<tr>
<td>6379</td>
<td>workerX.cacheserver.port</td>
<td>Yes</td>
</tr>
<tr>
<td>8060</td>
<td>psql.port</td>
<td>Yes</td>
</tr>
<tr>
<td>8600</td>
<td>vizportal.port</td>
<td>Yes</td>
</tr>
</tbody>
</table>
3. After you make the necessary port configuration changes, restart Tableau Server by typing the following:

```
tabadmin restart
```

### Enable the JMX Ports

To help you work through a problem with Tableau Server, Tableau Support may ask you to enable the server's JMX ports. These ports can be useful for monitoring and troubleshooting, usually with a tool like JConsole.

To enable the JMX ports on Tableau Server:

1. **Stop the server.**
2. Enter the following command:

   ```
tabadmin set service.jmx_enabled true
   ```
3. Enter the configure command:

   ```
tabadmin configure
   ```
4. **Start the server.**

**Important** Enabling JMX ports can introduce some security risk. To mitigate this risk, it is important to limit access to the JMX ports to the fewest number of clients that's practical for your scenario. You typically limit access using the host's firewall rules, an external security device, or routing rules.
JMX Port List

Here’s the list of JMX ports, all of which are disabled by default. When these ports are enabled, they are used for all types of installations: single-server, distributed, and highly available:

<table>
<thead>
<tr>
<th>Port</th>
<th>Used by this server process ...</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>8300 - 8359</td>
<td>Application server JMX. Determined by the application server port(s) + 300.</td>
<td>--</td>
</tr>
<tr>
<td>8550</td>
<td>Background monitor JMX. Determined by the background port of 8250 + 300.</td>
<td>--</td>
</tr>
<tr>
<td>9095</td>
<td>Service monitor JMX.</td>
<td>svcmonitor.jmx.port</td>
</tr>
<tr>
<td>9400 - 9499</td>
<td>VizQL server JMX. Determined by the VizQL server port(s) + 300.</td>
<td>--</td>
</tr>
<tr>
<td>10000 - 10299</td>
<td>Data server JMX. Determined by the data server port(s) + 300.</td>
<td>--</td>
</tr>
</tbody>
</table>

How the JMX Ports Are Determined

By default, the JMX ports for the application server (8300 - 8359), backgrounder (8550), VizQL server (9400 - 9599), and the data server (10000 - 10299) are assigned using the formula “base port + 300”. (See Tableau Server Ports on page 1244 for a list of the default base ports.) In addition, if there are multiple instances of a process, each will have a JMX port. For example, if you configure Tableau Server to run four instances of the application server process, ports 8000 (default base port), 8001, 8002, and 8003 are used. Application server JMX ports 8300 (base port + 300), 8301, 8302, and 8303 are then bound to their respective process instances.

If dynamic port remapping is enabled (which is the default) and if a port conflict is detected, JMX ports are not determined using the "base port + 300" formula. Instead, both base ports and JMX ports are assigned to available ports starting at port 8000. No offset is used for JMX ports; they are assigned the next available port, just like base ports are. If it’s important that you have a fixed JMX port, you can disable port remapping or change the base ports so that there are no port conflicts.

Even though they’re not directly used by Tableau Server, if a JMX port is being used by another application, Tableau Server processes won’t run. In addition, JMX ports cannot be edited directly using tabadmin. You change a JMX port by changing the base port for its process. In other words, if port 10000 isn’t available for the data server JMX process, you use tabadmin (as described in Edit the Default Ports on page 598) to change the data server base port from 9700 to 9800. This will move the data server JMX port to 10100.
**Restore the Default Value for a Port**

You can restore the default value for a port by following the procedure below:

1. Open a command prompt as an administrator and type the following:
   
   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.0\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.vizqlserver.port --default
   ```

3. Update the server’s configuration by typing the following:

   ```
   tabadmin config
   ```

4. Restart Tableau Server by typing the following:

   ```
   tabadmin restart
   ```

**Install and Configure**

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

**Everybody’s Install Guide**

Installing Tableau Server is about as easy as it gets with server software. Still, if you're new to it, you can use someone to help you figure out what to prepare and how to go through it. And now we’ve got you covered.

Take a look at the brand-new Tableau Server: Everybody’s Install Guide.

Everybody’s Install Guide explains how to plan for, install, and manage a single-machine instance of Tableau Server.

**Run Server Setup**

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

1. Double-click the installation file.
2. Follow the on-screen instructions to complete Setup and install the application.
The default installation path is \Program Files\Tableau\Tableau Server. You can choose a different location, including a different drive, either by browsing to or typing in a new path.

**Note:** When you upgrade a Tableau Server that's been installed to a non-default location, you need to navigate to that non-default path during the upgrade. For details, see Upgrade Tableau Server to a Non-Default Location on page 689.

3. After the installation completes, click **Next** to open the Product Key Manager window. If you need to support characters that are not the Latin-1 set, install the Windows Language Packs via **Control Panel > Regional and Language Options**. The language packs will need to be installed on the primary server as well as any worker machines.

**Activate Tableau**

Tableau Server requires at least one product key that both activates the server and specifies the number of license levels you can assign to users. You can access your product keys from the Tableau Customer Account Center. After installing and configuring the server, the product key manager automatically opens so you can enter your product key and register the product.
If you need to activate the product on a computer that is offline, see **Activate Tableau Offline below.** If you need to activate additional product keys to add capacity to an existing Tableau Server installation, see **Add Capacity to Tableau Server** on page 1170.

If you are activating Tableau Server as part of the install process, the Product Key Manager opens automatically. If you need to open it, in Windows, click **Start > All Programs > Tableau Server <version> > Manage Product Keys.**

**Note:** You can also find instructions for activating and registering Tableau Server on the download help page.

1. Select **Activate the product:**

   ![Activate Tableau](image)

   - **Start trial now**
     
     Begin using the product right away. You will be able to use the product for up to 14 days without restriction.
   
   - **Start trial later**
     
     Do not enter the trial period or start using the product at this time.
   
   - **Activate the product**
     
     Enter a product key to activate the product.

2. Enter or paste your license key and click **Activate.**
3. Click **Continue.**
4. Enter the fields to register Tableau and click **Register.**
5. Restart Tableau Server after registration is complete.

**Activate Tableau Offline**

If you are working offline you can follow the steps below to complete offline activation.
1. When the product key manager opens click **Activate the product.**
   Paste your server product key into the corresponding text box and click Activate. You can get your product key from the **Tableau Customer Portal.**

2. When you are offline, activation will fail and you are given the option to save a file that you can use for offline activation. Click **Save.**

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

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

5. From a computer that has Internet access, open a web browser and visit the **Product Activations** page on the Tableau website. Complete the instructions to submit your offline.tlq file.
   After you submit your offline.tlq file online, while your browser is still displaying the Product Activations page, a file called **activation.tlf** is created, and Tableau prompts you to save the file to your computer.

6. Save the activation.tlf file and move it to the computer where you are installing Tableau Server.

7. On the computer where you are installing Tableau Server, open a command prompt as an administrator and run the following command:
   ```bash
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   ```

8. Next, type **tabadmin activate --tlf <path>\activation.tlf,** where `<path>` is the location of the response file you saved from the Product Activations page. For example:
   ```bash
   tabadmin activate --tlf \Desktop\activation.tlf
   ```
   Keep the command prompt window open.

9. After the license is initialized, you are prompted to activate the product again. On Tableau Server, click **Start > All Programs > Tableau Server 10.0**

10. Right-click **Manage Product Keys** and select **Run as Administrator.**
    Even if you are logged into the Tableau Server computer as an administrator, you need to do this to avoid a potential registration error.

11. Click **Activate the product.**

12. Enter your product key again (the same one you entered in step 1).

13. Save the .tlq file.

14. From a computer that has Internet access, open a web browser and visit the **Product Activations** page again on the Tableau website. Complete the instructions.
    Tableau will again create a file called **activation.tlf** and prompt you to save it.
15. Save the file and move it to the computer where you are installing Tableau Server.

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

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

Tableau Server is now activated. If you need additional assistance, contact Tableau Customer Service.

**Add Capacity to Tableau Server**

You may need to add capacity to your Tableau Server installation to allow you to increase the number of users (if you have a user-based license) or the number of cores (if you have a core-based license).

Tableau Software will provide you with a new product key that adds capacity to your existing Tableau Server installation. You need to activate this key and use it together with your existing product key(s) to get the combined capacity you are licensed for.

Follow the steps below to add a product key to Tableau Server.

**Note:** This process requires a restart of Tableau Server.

1. Start the Product Key Manager:
   - In Windows, select **Start > All Programs > Tableau Server <version> > Manage Product Keys**.

2. Click **Activate** in the Manage Product Key dialog box:

   ![Manage Product Key dialog box](image)

3. Enter or paste your new product key and click **Activate**:
4. Restart Tableau Server after registration is complete.

Configure Tableau Server

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

You can also run the utility after installing Tableau Server by selecting All Programs > Tableau Server 10.0 > Configure Tableau Server on the Windows Start menu. You need to stop the server before making any configuration changes. See Reconfigure the Server on page 641 for steps.

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

- **Settings are system-wide**: The settings you enter apply to the entire server. If the server is running multiple sites, these settings affect every site.

- **User Authentication is "permanent"**: The User Authentication setting (on the General tab) can only be set when you are installing Tableau Server for the first time. You can change all of the other settings after installation by stopping the server and rerunning the Configuration utility.

See the topics below for details on the different Configuration tabs:
Configure General Server Options

Use the following sections to help you configure options on the General tab:

- Server Run As User
- User Authentication
- Gateway
- Firewall
- Sample data

Server Run As User

By default, Tableau Server runs under the Network Service account. To use an account that will accommodate NT authentication with data sources, specify a user name and password. The user name should include the domain name. See Run As User on page 577 to learn more about using a specific user account.

![Server Run As User](image)

User Authentication

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.

Tableau Server supports several types of SSO solutions: OpenID, SAML, and Kerberos. It’s important to understand how the decision about whether to use Active Directory or local authentication affects SSO:

- OpenID requires local authentication.
- Kerberos requires Active Directory authentication.
- SAML works with either Active Directory or local authentication. However, if you plan to configure Tableau Server for site-specific SAML authentication, you must select local authentication.
If you use Active Directory:

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

Be sure to type the fully qualified domain name (FQDN) and nickname (NetBIOS name).

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

The default port for web access to Tableau Server (via HTTP) is port 80. If the installation program determines that port 80 is in use when you first install Tableau Server, an alternate port (for example 8000) is used and shown in the Port number box.

You may need to change the port for other networking needs, for example, if you have a hardware firewall or proxy in front of the Tableau Server host, this might make running a back-end system on port 80 undesirable.

**Gateway**

**Firewall**

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.
Sample data

Select whether to include sample data and users. The **Include sample data and users** option installs several sample workbooks and data, which can help you get familiar with Tableau Server (especially if you are installing a trial version of the product). If you select **Include sample data and users**, the first user created in Tableau Server will be assigned as the owner of sample workbooks and data. To change the assigned owner, see **Manage Ownership** on page 784.

You can continue to the next page to configure Caching and Initial SQL options. If you do not want to configure these options click **OK**.

**Configure Data Connections**

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

**Caching**

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

- **Refresh Less Often**—Data is cached and reused whenever it is available regardless of when it was added to the cache. This option minimizes the number of queries sent to the database. Select this option when data is not changing frequently. Refreshing less often may improve performance.

- **Balanced**—Data is removed from the cache after a specified number of minutes. If the data has been added to the cache within the specified time range the cached data will be used, otherwise new data will be queried from the database.

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

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

When connecting to some data sources, you can specify an initial SQL command to run when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. If your data source supports running an initial SQL statement, an Initial SQL link appears in the lower-left corner of the Server Connection dialog box in Tableau Desktop.

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 check box. Workbooks created with initial SQL statements will still open but the initial SQL commands will not be sent.

Servers

Use the options on the Servers tab to do the following:

- Adjust the number of processes running on Tableau Server,
- Configure a distributed Tableau Server environment,
- Select the preferred active repository for failover situations.

You can also use the Server tab to add computers on which to run Tableau Server processes.

![Tableau Server Configuration](image)

**Number of processes per server**

Tableau Server deployments run multiple processes. You can choose to run the processes one computer, or to distribute them across multiple computers. To improve performance, you can adjust the number of processes that run on each computer, for each process type.

For more information on changing the number of server processes for a single-server environment, see [Reconfigure Processes on page 642](#).

For more information on how many processes to run in order to improve performance, see [Performance Tuning Examples on page 1135](#).

For more information on setting up a multi-server, or distributed, environment, see [Distributed Environments on page 694](#).

**Preferred active repository**

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

![Preferred Active Repository](image)

Configure a preferred active repository if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active...
repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see Collect Data with the Tableau Server Repository on page 1117

Install Tableau Server on a Two-Node Cluster

When you install Tableau Server on a two-node cluster, you can install server processes on one or both nodes. A two-node cluster can improve the performance of Tableau Server, because the work is spread across multiple machines.

Note the following about two-node clusters:

- A two-node cluster does not provide failover or support for high availability.
- You can't install more than one instance of the repository on a two-node cluster, and the repository must be on the primary node.

If you need failover or high availability, or want a second instance of the repository, you must install Tableau Server on a cluster of at least three computers. In a cluster that includes at least three nodes, you can configure two instances of the repository, which gives your cluster failover capability.

Primary Server Installation Defaults

By default, the Tableau Server installer configures the number of process instances that Tableau Server runs based on the hardware detected by the installer. The default configuration applies to single-server installations and to the primary server of a multi-server installation.

You can calculate the default configuration based on the following rules for each process, where the number of cores refers to the number of logical CPU cores:

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Number of Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>VizQL Server</td>
<td>Equal to the number of cores divided by four, up to a maximum of four process instances.</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>Set to two unless the number of cores is fewer than eight.</td>
</tr>
<tr>
<td>Cache Server</td>
<td>Set to two unless the number of cores is fewer than eight.</td>
</tr>
<tr>
<td>Data Server</td>
<td>Set to two unless the number of cores is fewer than eight.</td>
</tr>
</tbody>
</table>

For all other process types, the number of process instances is set to one, regardless of the hardware.

Here’s an example default configuration for a computer with 16-cores:
<table>
<thead>
<tr>
<th>Process Name</th>
<th>Number of Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>VizQL Server</td>
<td>4</td>
</tr>
<tr>
<td>Application Server</td>
<td>1</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>2</td>
</tr>
<tr>
<td>Cache Server</td>
<td>2</td>
</tr>
<tr>
<td>Data Server</td>
<td>2</td>
</tr>
<tr>
<td>Data Engine</td>
<td>1</td>
</tr>
</tbody>
</table>

**Distributed Installation Recommendations**

When you add computers (workers) to a Tableau Server installation, you must decide how many processes to run on each computer. This page provides recommendations based on the number of computers that you plan to use and on each computer's hardware.

These recommendations are intended only as a starting point. To determine the best configuration for your installation, you should do the following:

- Understand how your organization uses Tableau Server and tune your configuration for your use case—for example, whether you want to optimize for user response or for extract refreshes.
- Perform thorough performance testing to identify the best places to adjust process configuration.

For more information on tailoring a Tableau Server installation to your organization's needs, see *Performance Tuning Examples* on page 1135.

For more information on the requirements for a distributed installation and for information on configuring workers, see *Distributed Environments* on page 694.

**Recommendations for all installations**

Although the computers that make up a Tableau Server cluster do not need to have identical hardware, they must all meet the same minimum system requirements. All of the recommendations on this page assume that the computers where you install Tableau Server have eight cores or more.

The following recommendations apply to all server configurations:

- Run Backgrounder processes on a dedicated computer, especially if you plan on refreshing extracts frequently. Backgrounder processes are generally the most CPU intensive and can slow down other processes on the same computer.
- Run Data Engine processes on a different computer than Backgrounder processes.
Because Data Engine processes are also CPU intensive, you can prevent CPU bottlenecks by hosting the Data Engine processes and the Backgrounder processes on separate machines.

- If you plan to refresh extracts frequently or if you plan to refresh large extracts, increase the number of processes for Backgrounder and Data Engine processes.

**Recommendations for two computers**

The following table shows recommendations for process configuration if you're running two computers (one primary server and one worker) in your cluster. As noted earlier, these are a starting point. In the table, \( n \) refers to the number of cores for the computer.

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Primary: Number of Processes</th>
<th>Worker: Number of Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gateway</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Application Server</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>( n/4 )</td>
<td></td>
</tr>
<tr>
<td>Cache Server</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Search and Browse</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Backgrounder</td>
<td></td>
<td>( n/2 )</td>
</tr>
<tr>
<td>Data Server</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Data Engine</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>File Store</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Repository</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Configuration Notes**

For light extract usage, decrease the number of Backgrounder processes to \( n/4 \). Because this decreases the load on the worker, you can move all the data engine processes to the worker as well.

**Recommendations for three computers**

In the table below, \( n \) corresponds to the number of cores for the machine.
<table>
<thead>
<tr>
<th>Process Name</th>
<th>Primary: Number of Processes</th>
<th>Worker 1: Number of Processes</th>
<th>Worker 2: Number of Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Gateway</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Application Server</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>VizQL Server</td>
<td>n/4</td>
<td>n/4</td>
<td></td>
</tr>
<tr>
<td>Cache Server</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Search and Browse</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Backgrounder</td>
<td></td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>Data Server</td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Data Engine</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>File Store</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Repository</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**Configuration Notes**

Because worker two is not running other CPU-intensive processes like the Data Engine process, you can increase the number of Backgrounder processes. We recommend a maximum of 2n for Backgrounder processes.

This configuration assumes that the primary computer runs the active repository and worker one runs the passive repository.

**Note:** In a distributed installation with three or more nodes, you can have a maximum of two repository instances (active and passive). You can also run Tableau Server with one repository, but doing this means there is no failover available for the repository. For more information, see Tableau Server Repository on page 653.

**Configure SMTP Setup**

Tableau Server can send email to alert system administrators if there is a system failure and can email subscriptions (snapshots of selected views) to system users. For this functionality to work, you need to first configure the SMTP server that Tableau Server uses to send email.
1. In the Tableau Server Configuration utility, click the SMTP Setup tab.

2. Under SMTP Server:
   a. Enter the name of your SMTP server.
   b. (Optional) If your account requires it, enter a user name and password for your SMTP server account.
   c. If you are not using the default SMTP port 25, change the SMTP port value.
   d. Leave the Enable TLS box cleared so the connection to your mail server is unencrypted.

   Encrypted SMTP connections are not supported for alerts or subscriptions.

3. For Send email from, enter the email address that will send an alert if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.com or noreply@mycompany), but it does not have to also be an actual email account on Tableau Server.

   **Note:** Some SMTP servers may require this to be an actual email account. You can override the system-wide Send email from address on a per-site basis for subscriptions. For more information, see What is a Site? on page 737.

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

5. For Tableau Server URL, enter http:// or https://, followed by the name or IP
address of the Tableau server. This value will be used for the footer of subscription emails.

6. Click **OK**.

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

**Configure Alerts and Subscriptions**

On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, you can configure the following email alerts and subscriptions:

- Email subscriptions to views
- Email alerts for system failures
- Disk space usage:
  - Recording usage history
  - Email alerts when space crosses or remains below pre-configured thresholds

**Note:** You need to configure SMTP before you can configure subscriptions or alerts. For more information, see **Configure SMTP Setup on page 617**.

**Subscriptions to views**

Tableau Server can be configured to send email subscriptions (snapshots of selected views) to system users.

When you enable subscriptions, Tableau Server users have the option to subscribe to views. For more information, see **Manage Subscriptions on page 925**.

**To enable email subscriptions**

1. On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, select **Enable email subscriptions**.
Alerts for system failures

Tableau Server can send email alerts to server administrators when there is a system failure. When you configure alerts, Tableau Server sends an email to the recipients listed in **Send email to** on the **SMTP Setup** tab any time that the data engine, repository, or gateway server processes stop or restart, or any time the primary Tableau Server stops or restarts. If you are running a single-server installation (all processes on the same machine), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover (see Configure for Failover and Multiple Gateways on page 720), a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

To configure email alerts for system failures

1. On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, select **Send email alerts for server component up, down, and failover events**.

2. Click **OK**.
Disk space monitoring

If Tableau Server is configured to monitor free disk space and send alerts about low disk space, when space on any node in a server installation drops below the configured thresholds, Tableau Server sends an email to the recipients listed in Send email to on the SMTP Setup tab.

Disk space usage

When you configure Tableau Server to record disk space usage, information about free disk space is saved in the Repository and you can view the usage history using the Administrative Views.

To configure Tableau Server to record disk space usage

1. On the Alerts and Subscriptions tab of the Tableau Server Configuration utility, select Record disk space usage information, including threshold violations.

2. Click OK.

You can configure Tableau Server to send email alerts when disk space usage on any node crosses a threshold, or remains below the threshold.

To configure email alerts for low disk space

1. On the Alerts and Subscriptions tab of the Tableau Server Configuration utility, select Send alerts when unused drive space drops below thresholds.
2. In **Warning threshold**, enter the percentage of free disk space that Tableau Server should use as a warning threshold.

   If free disk space on any node in your Tableau Server cluster drops below this percentage, Tableau Server sends a warning alert email. Alerts continue until free disk space rises above the threshold. To configure the frequency of alerts, see Step 4 below.

3. In **Critical threshold**, enter the percentage of free disk that Tableau Server should use as a critical threshold.

   If free disk space on any node in your Tableau Server cluster drops below this percentage, Tableau Server sends a critical alert email. Alerts continue until free disk space rises above the threshold. To configure the frequency of alerts, see Step 4 below.

4. In **Send email alert every**, enter the number of minutes for how often Tableau Server should send an alert.

5. Click **OK**.

**Configure External SSL**

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the web browser and the server or Tableau Desktop and the server is protected. Steps on how to configure the server for SSL are described in the topic below; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server. If you are running a Tableau Server cluster and you want to use SSL, see Configure SSL for a Cluster on page 624, below, for recommendations.

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

**Note:** Be sure to use a SHA-2 (256 or 512 bit) certificate. All major browsers will display warnings when connecting to SHA-1 certificates. By the end of 2017, it's likely that most browsers will no longer connect to servers that are presenting SHA-1 certificates.

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 10.0 folder. For example:

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

   This location gives the account that's running Tableau Server the necessary permissions for the files. You may need to create this folder.

3. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 10.0 > 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 has an embedded passphrase, and is not password protected with the file extension .key.

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

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

   - Select **Use mutual SSL and automatic login with client certificates**.

     **Note:** Tableau Server does not support mutual SSL and SAML together.

   - In **SSL CA certificate file**, browse to the location for the certificate file. The SSL
CA certificate file must be a valid PEM-encoded x509 certificate with the extension .crt.

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

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

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

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

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

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

**Configure SSL for a Cluster**

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

**SSL and Multiple Gateways**

A highly available Tableau Server cluster can include multiple gateways, fronted by a load balancer ([learn more](#)). If you are configuring this type of cluster for SSL, you have two choices:
Configure your load balancer for SSL. Traffic is encrypted from the client web browsers to the load balancer. Traffic from the load balancer to the Tableau Server gateway processes is not encrypted. No SSL configuration in Tableau Server is required, it's all handled by your load balancer.

Configure Tableau Server for SSL: Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. See the procedure below for details.

Configure a Server Cluster for SSL

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

1. Configure the external load balancer for SSL passthrough. Refer to your load balancer's documentation for assistance.
2. Make sure that the SSL certificate you use was issued for the load balancer's host name.
3. Configure the primary Tableau Server node as described in the procedure above.
4. Place the same SSL certificate and key file that you used for the primary on each Tableau Server worker node that is running a gateway process. Use the same folder location on the workers that you used on the primary.
   
   If you are using mutual ssl, place the SSL CA certificate file you used for the primary on each worker node that is running a gateway process. Use the same folder location that you used on the primary.
   
   You do not need to do any additional configuration on the workers.
   
   For example, say you have a cluster that includes a primary Tableau Server node and three worker nodes with gateway processes are running on the primary, Worker 2 and Worker 3. In this situation, you configure the primary Tableau Server for SSL, then copy the same SSL certificate and key files to Worker 2 and Worker 3. Because these files are in C:\Program Files\Tableau\Tableau Server\SSL folder on the primary, they are in that same location on Worker 2 and Worker 3.
   
   You can configure a Tableau Server cluster to use SSL. If the primary Tableau Server computer is the only node that is running the gateway process (which it does by default), then that's the only place where you need to configure SSL. See the procedure above for steps.

Configure Internal SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository.
1. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 10.0 > Configure Tableau Server**.

2. In the Tableau Server Configuration dialog box, click the **SSL** tab.

3. Select one of the following options:
   - **Required for all connections**
     When this option is selected, Tableau Server uses SSL for communications between the repository database and other server components. In addition, direct connections to Tableau Server (connections using the "tableau" or "readonly" users) must use SSL.
   - **Optional for direct user connections**
     This option configures Tableau Server to use SSL between the repository and other server components and supports but does not require SSL for direct connections by "tableau" or "readonly" users.
   - **Off for all connections** (the default)
     This option disables SSL for internal communications and direct connections.

4. Click **OK**.

For more information on downloading the public certificate for direct connections, see **Configure SSL for Direct Connections** on page 976.

**Configure SSL for Direct Connections**

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

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

1. Generate the SSL certificate file using the **regenerate_internal_tokens** command.

2. Locate the SSL cert file by looking in the workgroup.yml file on the primary Tableau Server node.
   
   The workgroup.yml file is located on the primary Tableau Server node in the \ProgramData\Tableau\Tableau Server\data\tabsvc\config folder.
   
   The location of the SSL certificate and key files are listed in the file. For example:
   
   `pgsql.ssl.cert.file: C:/ProgramData/Tableau/Tableau Server/data/tabsvc/config/pgsql/server.crt`
pgsql.ssl.key.file: C:/ProgramData/Tableau/Tableau Server-data/tabsvc/config/pgsql/server.key

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

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

**Configure Server-Wide SAML**

Configure server-wide SAML when you want users on Tableau Server to authenticate with a single SAML identity provider (IdP). For information about authenticating users with different IdPs for different sites on Tableau Server, see Configure Site-Specific SAML on page 1022.

Before you configure Tableau Server for SAML, make sure you meet the SAML Requirements on page 1014.

To configure Tableau Server to use server-wide SAML:

1. Place the certificate files in a folder named SAML, parallel to the Tableau Server 10.0 folder. For example:

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

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

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

   If you are configuring SAML after you installing Tableau Server, open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server) and then click the SAML tab.

3. On the SAML tab, select **SAML authentication for the server** and provide the location for each of the following:

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

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

   **SAML certificate file**—A PEM-encoded x509 certificate with the file extension .crt. This file is used by Tableau Server, not the IdP.
**SAML certificate key file**—An RSA or DSA private key file that is not password protected, and that has the file extension .key. This file is used by Tableau Server, not the IdP.

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

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

5. Save the XML file with the name of your choice.

6. On your IdP’s website or in its application:
   - Add Tableau Server as a Service Provider. Refer to your IdP’s documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the file you saved in step 5.
   - Confirm that your IdP uses **username** as the attribute element to verify.

7. Still within your IdP, export your IdP’s metadata XML file.

   It’s a good idea to verify that the metadata XML you get from the IdP includes a **SingleSignOnService** element in which the binding is set to HTTP-POST, as in the following example:

   ```xml
   ```

8. Copy your IdP’s metadata XML file to the following folder on the computer where Tableau Server is installed:

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

9. On the SAML tab in the Tableau Server Configuration dialog box, enter the location to the file in the **SAML IdP metadata file** text box:
10. Click **OK**. Tableau Server is now configured for SAML authentication.

**Configure a Server Cluster for SAML**

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

1. Configure the primary Tableau Server as described in the procedure above.

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

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

**Test Your Configuration**

Test your SAML configuration by opening a new web browser instance and typing the Tableau Server name in the URL window:
You should note that the sign in prompt that appears is from your IdP and not Tableau Server:

Configure Kerberos

You can configure Tableau Server to use Kerberos. This allows you to provide a single sign-on experience across all the applications in your organization. Before you configure Tableau Server for Kerberos make sure you meet the Kerberos Requirements on page 992.

1. Open a command prompt as an administrator and change directories to the location of Tableau Server's bin directory. The default location is \Program Files\Tableau\Tableau Server\10.0\bin.
2. Type the following command to stop Tableau Server:
   `tabadmin stop`
3. Open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server), and then click the Kerberos tab.
4. Select Enable Kerberos for single sign-on.
5. Click Export Kerberos Configuration Script. The generated script configures your
Active Directory domain to use Kerberos with Tableau Server. For more information, see Kerberos Configuration Script on page 995.

**Note:** Verify the host names in the setspn lines of the script. If you are using an external load balancer or a reverse proxy, the host names should match the name you used when you configured Tableau Server for the load balancer or proxy. If you have not configured Tableau Server for your proxy or external load balancer, do that and then re-export the Kerberos configuration script to ensure it has the correct host names. See Add a Load Balancer on page 730 and Configuring Proxies for Tableau Server on page 579.

6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the .keytab file. The domain administrator should do the following:

   - Review the script to verify it contains correct values.
   - Run the script at a command prompt on any computer in the domain by typing the script name (not by double-clicking the script in Windows Explorer).

   The script creates a file, kerberos.keytab, in a `\keytabs` folder in the location that the script was run.

7. Save a copy of the .keytab file created by the script to the Tableau Server computer. In Step 3, enter the path to the .keytab file, or click the browse button to navigate to the file. The keytab file will be copied to all the gateway nodes in your Tableau Server installation when you click OK in the Configuration utility.

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

8. (optional) Click Test Configuration to confirm that your environment is configured correctly to use Kerberos with Tableau Server.
If you have not configured any data sources for Kerberos delegation, 0 is shown for the **Number of services configured for delegation**.

9. Click **OK** to save your Kerberos configuration.

10. Start Tableau Server.

**Confirm Your SSO Configuration**

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

You should be automatically authenticated to Tableau Server.

**Configure SAP HANA SSO**

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

**Note**: You do not need to use SAML sign on with Tableau Server in order to use HANA SSO. You can sign in to Tableau Server using whatever method you choose.

With SSO for SAP HANA, Tableau Server functions as an Identity Provider (IdP) and this configuration allows you to provide a single sign-on experience for users making SAP HANA connections. As part of the configuration, you need to acquire a SAML certificate and key file for Tableau Server (these should be a public key certificate and private key). You need to also install the signed certificate in HANA. You can generate the certificate and key yourself, or get them from a Certificate Authority. For more information on generating a certificate/private key and configuring SAP HANA, see the [Tableau Knowledgebase](#).
Note: The SAP HANA driver version 1.00.9 or later must be installed on Tableau Server in order to use SSO for SAP HANA. The driver cannot encrypt the SAML assertion, so you may want to enable encryption for the SAML connections. For more information, see the Tableau Knowledgebase.

Configure SSO for SAP HANA

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 10.0 folder. For example:
   
   C:\Program Files\Tableau\Tableau Server\SAML

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

2. After you install Tableau Server, run the Configuration utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server), and then click the SAP HANA tab.

3. Select Use SAML to enable single sign-on for SAP HANA and provide the location for each of the following:

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

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

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

**Configure Tableau Server for OpenID Connect**

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

- **OpenID Connect** on page 1050
- **Configure the Identity Provider (IdP) for OpenID Connect** on page 1053
- Configure Tableau Server for OpenID Connect (you are here)
- **Signing In to Tableau Server Using OpenID Connect** on page 1056
- **Changing IdPs in Tableau Server for OpenID Connect** on page 1058

**Note:** Before you perform the steps described here, you must configure the OpenID identity provider (IdP) as described in **Configure the Identity Provider (IdP) for OpenID Connect** on page 1053.
Important notes

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

- You can use OpenID Connect with Tableau Server only if the server is configured to use local authentication. OpenID Connect is not available if the server is configured to use Active Directory authentication. For more information, see Configure General Server Options on page 608.

- We recommend that you configure Tableau Server to use SSL for external communications. This helps to maintain secure communications between Tableau Server and the IdP during the exchange of authentication information. For details, see Configure External SSL on page 972.

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

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

Configure the server

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

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

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

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

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

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

9. Copy the URL in the box labeled **Configure the OpenID provider using the following redirect URL for Tableau Server**. You'll use this value in the next procedure to finish configuring the IdP.

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

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

**Add the redirect URL to the IdP configuration**

After you configure Tableau Server, you finish the IdP configuration using the server's redirect URL.
1. Return to the IdP portal where you set up the project or application.
2. Edit the project configuration and find the redirect URL.
3. Enter the redirect URL that you copied in the previous procedure.

**Add an Administrator Account**

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

**Active Directory**

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

**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**.
Managing Licenses

The following topics describe how to manage Tableau licenses in your enterprise or organization.

Enterprise Desktop Licensing Overview

If you are responsible for deploying or managing Tableau Desktop installations in your organization, then managing and tracking licensing entitlement, and understanding desktop usage are essential tasks.

Tableau provides two main tools that will help you track Tableau Desktop licensing and usage in your organization:

- Tableau Software customer portal: the online portal is where you purchase, manage, and view registration information for licenses assigned to your users. The portal is also where you manage your Tableau account on behalf of your organization.
- Desktop license reporting: you can configure Tableau Desktop to report usage information to an instance of Tableau Server running in your organization.

This topic describes how you can use these two tools to manage Desktop licenses and track Desktop usage in your organization.

Customer Portal: asset and account management

The Tableau Software customer portal is where you manage all elements of licensing entitlement. The portal provides you with access to your purchased license keys along with a platform upon which you can track license key assignments.
Use the portal for the following tasks:

- View your purchased license entitlements
- Track the assignment of license entitlements to specific departments and assigned users
- Monitor and compare user registration received by Tableau to your purchased and assigned licenses
- Open support cases and manage current and prior case interactions with Tableau Software
- Download Tableau installation packages
- Manage your organizational account and invoicing

The portal is not intended to provide detailed usage data, however, you can determine desktop installations in your organization according to the user registration records that are housed in the portal.

**Activation, Deactivation, and Registration**

It's important to understand that the Tableau Customer portal only logs registration events from installations in your organization. This means that as the asset manager for your organization, you must manage deactivating and reactivating licenses, as these events are not logged in the customer portal. Desktop licenses are perpetual, which means that as long as they are registered and active for an authorized user, the license key will not expire.

The reason it’s important to understand these details is for scenarios where a computer with a desktop license is no longer accessible. For example, if a computer is lost, stolen, or formatted before the license has been deactivated, the license key may not always be able to be reassigned.

Therefore, it's important for users to deactivate their licenses before decommissioning a computer where Tableau Desktop is installed.

Users can deactivate licenses in Tableau Desktop by opening Manage Product Keys (Help > **Manage product keys**...), selecting the Product Key and then clicking **Deactivate**. Other command line tools are available for bulk deactivate and silent deactivate. Contact your account representative for more information.
Be sure that the license key-registration pairs that are shown in the portal match the activated Desktop-user pairs in your organization. For larger organizations, use Desktop Reporting and Tableau Server to identify activated Desktop-user pairs.

**Desktop Reporting: Monitoring usage in your organization**

Deploying Tableau Desktop in your organization shows a commitment to data analysis as a core business requirement. For many organization, quantifying the return on software investments is an important business need. Understanding how often and to what extent your users are utilizing Tableau Desktop can be important as you plan asset allocation.

After you configure desktop reporting in your organization, you can view usage reports on Tableau Server to answer questions like the following:

- What types of licenses are installed in your organization.
- Which users have Tableau licenses.
- Which licenses are used most and least often.
- Whether trial licenses need to be converted to paid licenses.
- Which licenses are expired or might soon expire.
- When maintenance renewals are due in your organization.

Desktop reporting is enabled by configuring each Tableau Desktop installation with a pointer to at least one Tableau Server in your organization. You can configure each Desktop during the install process with a command line option, or you can deploy a registry update to existing Desktop clients. For more information see [Configure Tableau Desktop License Reporting on page 1081](#).

**Overview of Tableau Server Licenses**

Tableau Server can be licensed under two models: user-based or core-based.

User-based licenses lets you deploy Tableau Server on a single computer or on multiple computers in a cluster. The license restricts how many users can work on your installation of Tableau Server.

For core-based licensing, you can install Tableau Server on a single-node or multi-node cluster, as long as the total number of cores for all of the nodes does not exceed the number of cores that you have licensed. Core-based licensing imposes no constraints on the number of user accounts in the system.

The following topics describe how to manage Tableau licenses in your enterprise or organization.

**View Server Licenses**

Server administrators can view the license and product key information for Tableau Server.
Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.

To view server licenses

- In a site, click Settings > Licenses.

If you have a user-based Tableau Server license, you can review how these levels have been distributed.

If you have a core-based Tableau Server license, the Licenses page shows how many cores are allowed, how many have been licensed, and how many are in use (and on what server computers).

Also see:

- Overview of Tableau Server Licenses on the previous page
- Handle an Unlicensed Server on page 1198.

Reconfigure the Server

When you install Tableau Server for the first time, you do initial configuration of the server as part of the installation. You can run the Tableau Server Configuration utility after installing Tableau Server to make additional configuration changes. Some configuration options are only available when you run the configuration utility after installation. You can also use the tabadmin on page 1255 command line tool to make configuration changes. Configuration setting changes are written to the tabsvc.yml file located in the <install drive>:\ProgramData\Tableau\Tableau Server\config directory.

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

To change a Tableau Server configuration setting:

1. Stop the server by selecting All Programs > Tableau Server 10.0 > Stop Tableau Server on the Windows Start menu.
2. Select Configure Tableau Server on the Windows Start menu.
3. If you are using an Active Directory account for the server’s Run As User account, enter
its password on the General tab.

4. Make your configuration change.

5. Click OK.

6. Start the server by selecting All Programs > Tableau Server 10.0 > Start Tableau Server on the Windows Start menu.

Reconfigure Processes

To change how processes are configured for a single server installation, follow the steps below. If you are changing how processes are configured for a worker, refer to Install and Configure Worker Nodes on page 699.

1. You will need to stop Tableau Server to make this configuration change. From the Start menu, click All Programs > Tableau Server 10.0 > Stop Tableau Server.

2. Open the Tableau Server Configuration dialog box from the Start menu by navigating to All Programs > Tableau Server 10.0 > Configure Tableau Server.

3. Enter your Password, if necessary, on the General tab then click the Servers tab:

4. Highlight This Computer and click Edit:

5. The Edit Tableau Server dialog box is where you change the number of processes:
You can run up to eight instances of the VizQL, application server, data server, or background processes—although this limit can be changed if necessary. See Server Process Limits on page 654 for more information. You need to have at least one instance of backgrounder installed. Also, for Tableau Server to function, there must always be one active instance of the data engine (and associated file store) and the repository. For steps on how to move them to another machine, see Move the Data Engine and File Store Processes on page 707. For steps on how to configure additional instances of them, refer to High Availability on page 709.

After you make your changes, click OK.

6. If you want to designate a specific computer as the preferred active repository, select the computer from the Select host list. If you add workers, you need to save the configuration and restart the Configuration utility for the workers to display in the list. For more information about the repository, see Tableau Server Repository on page 653.

7. Click OK to close the Configuration utility.

8. Start Tableau Server again. From the Start menu, click All Programs > Tableau Server 10.0 > Start Tableau Server.
Tableau Server Processes

There are Tableau Server processes whose default configuration you can change to achieve different results. The topics Performance Tuning Examples on page 1135 and High Availability on page 709 describe some of the approaches you can take. High-level status for each process is displayed on the server’s Status page and more detailed information related to some of the processes—such as the background process—is in the Administrative Views on page 1097 topic.

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

For information on log files generated by these processes, see Server Log File Locations on page 1190.

<table>
<thead>
<tr>
<th>Process</th>
<th>File Name</th>
<th>Purpose</th>
<th>Multi-Threaded?</th>
<th>Performance Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>vizportal.exe</td>
<td>Handles the web application, REST API calls, supports browsing and searching</td>
<td>Yes</td>
<td>Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Background</td>
<td>backgrounder.exe</td>
<td>Executes server tasks, including extract refreshes, subscriptions, ‘Run Now’ tasks,</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn’t consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------</td>
<td>----------------------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Cache Server</td>
<td>redis-server.exe</td>
<td>Query cache</td>
<td>No</td>
<td>Refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>clustercontroller.exe</td>
<td>Responsible for monitoring various components, detecting failures, and executing failover when needed</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
</tr>
<tr>
<td>Coordinatio Service</td>
<td>zookeeper.exe</td>
<td>In distributed installations, responsible for ensuring there is a quorum for</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server installations of more than five nodes, also installed on the first four worker nodes.</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Process</th>
<th>File Name</th>
<th>Purpose</th>
<th>Multi-Threaded?</th>
<th>Performance Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Engine</td>
<td>tdeserver64.exe</td>
<td>Stores data extracts and answers queries</td>
<td>Yes</td>
<td>The data engine’s workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory consumption is primarily based on the size of the data extracts being loaded. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.</td>
</tr>
<tr>
<td>Data Server</td>
<td>dataserver.exe</td>
<td>Manages connections to Tableau Server data sources</td>
<td>Yes</td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.</td>
</tr>
<tr>
<td>File Store</td>
<td>filestore.exe</td>
<td>Automatically replicates extracts across data engine nodes</td>
<td>n/a</td>
<td>Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Repository</td>
<td>postgres.exe</td>
<td>Tableau Server database, stores workbook and user metadata</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository on page 653.</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>searchserver.exe</td>
<td>Handles fast search, filter, retrieval, and display of content metadata on the server</td>
<td>Yes</td>
<td>The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>vizqlserver.exe</td>
<td>Loads and renders views, computes and executes queries</td>
<td>Yes</td>
<td>Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.</td>
</tr>
</tbody>
</table>

Tableau Server Coordination Service

Tableau Server uses the Coordination Service to coordinate activities on the server, including for high availability installations. The Coordination Service is built on Apache ZooKeeper, an open-source project.

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

- Memory. The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.
- Disk speed. Because the service stores state information on disk, it benefits from fast
disk speed on the individual node computers.

- Connection speed between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Configuration for the Coordination Service

The Coordination Service is installed automatically as a part of Tableau Server. The number of nodes with the Coordination Service installed depends on the total number of nodes in the Tableau Server installation. On a one- or two-node system, one instance of Coordination Service is installed. On a three- or four-node installation, three instances of Coordination Service are installed. For Tableau Server installations that have five or more nodes, a total of five Coordination Service instances are installed. (The Coordination Service is installed on the first <n> nodes in the cluster, so on a one- or two-node cluster it is installed on the first node, on a three- or four-node cluster it is installed on the first three nodes, and on a cluster of five or more nodes it is installed on the first five nodes.)

You do not have to explicitly configure the coordination service, and there are no settings you can make for the service. As a consequence, when you add a node to your cluster, you do not see the configuration service listed as process—for example, you do not see the coordination service listed in the Add Tableau Server dialog box:
The Base Install option includes the Coordination Service and Cluster Controller. As you can see, this option is disabled, because you cannot choose when to install those services.

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a quorum—a minimum number of instances of the service. This means that the number of nodes you have in your installation impacts how many instances of the Coordination Service are running.

If you reduce the number of nodes

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

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

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

Viewing Coordination Service Status

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

`tabadmin status --verbose`

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

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

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

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

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

This command will perform a normal cleanup on page 1263 as well as removing Coordination Service files.

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

Tableau Server File Store

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

<table>
<thead>
<tr>
<th>Process</th>
<th>File Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name</td>
<td>filestore.exe</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the File Store process is visible on the Status Page. For more information, see View Server Process Status on page 1154</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs are located in \logs\filestore. For more information, see Server Log File Locations on page 1190</td>
</tr>
</tbody>
</table>

The decommission Command

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

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

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

![Edit Tableau Server](image)

**Port assignment**

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

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

**Log files for the gateway process**

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

- Activity logs. The name for these log files has the format `access.yy_mm_dd hh_mm_ss.log`.

- Error logs. All errors are logged in a single file named `error.log`.

For more information, see [Archive Log Files](https://example.com) on page 1184.

**Gateway processes in a cluster**

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

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

If the Tableau Server is configured to use SSL, you must make sure that the certificate for SSL support is in the same location on each computer in the cluster that has the gateway process running. For more information about using SSL, see Configure External SSL on page 972.

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

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

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

Additional information

**Configuring Proxies for Tableau Server** on page 579

**Add a Load Balancer** on page 730

**Configure for Failover and Multiple Gateways** on page 720

Tableau Server Repository

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

<table>
<thead>
<tr>
<th>Process</th>
<th>Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name</td>
<td>postgres.exe</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the Repository is visible on the Status Page. For more information, see View Server Process Status on page 1154</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the repository are located in \logs\repository. For more information, see Server Log File Locations on page 1190</td>
</tr>
</tbody>
</table>

Preferred active repository

When you configure Tableau Server after the initial installation, you have the option to specify a **Preferred Active Repository**. This is an optional step, and if you do not specify a preferred active repository, Tableau Server will select the active repository on startup.
Configure a preferred active repository if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see Collect Data with the Tableau Server Repository on page 1117

The failoverrepository Command

If failover occurs and your passive repository becomes the active repository, it remains the active repository until either Tableau Server restarts or you use the failoverrepository command to switch back. Specify the repository you want to be the active one, or specify that the preferred active repository (if configured) should be made active again. For more information, see failoverrepository on page 1274.

Server Process Limits

When you reconfigure processes for Tableau Server, there is a limit to the amount that you can increase the number of process instances. By default, the limit is set to eight. If your machine has enough RAM and CPU cores, and you want to go above this limit, you can change the limit using the service.max_procs tabadmin setting. For each process instance, Tableau recommends that the machine running the process have at least 1 GB of RAM and 1 logical CPU core.

To change the maximum number of processes allowed:

1. After Setup, stop the server.
2. In the Tableau Server bin directory, type the following command, where number is the maximum number of process instances you want to allow:
   `tabadmin set service.max_procs <number>`
   For example:
   `tabadmin set service.max_procs 10`
3. Still in the bin directory, type:
tabadmin config

4. Start the server so the changes can take effect.

Customize the Server

You can customize how Tableau Server looks in order to personalize it for your company or group. You can perform these customizations:

- Change the server name that appears in the browser tab, tooltips, and messages.
- Change the logos that appear in different server page contexts.
- Control the language used for the server user interface and the locale used for views.
- Install custom fonts on Tableau Server and client computers that connect to Tableau Server.

Administrators and project leaders can also add images for projects in thumbnail view. For more information, see Add a Project Image.

See the following topics for more information:

Change the Name or Logo

You can customize the Tableau Server look and feel to brand it for your company by changing the server name and by using a custom logo.

Your custom name appears on browser tabs and in a tooltip when users hover over the home logo in the upper left corner of the main page. The customizable logo appears in the sign-in page, the server page header, and in web authoring pages. (Note that some references to Tableau Server cannot be changed, such as the logo on browser tabs and the phrase "Tableau Server" in the copyright notice.)
A - The Tableau logo for the browser window tab cannot be changed.
B - The server name can be changed using `tabadmin customize name`.
C - The header logo can be changed using `tabadmin customize header_logo`.

**Change the server name**

The server name is displayed in tooltips, messages, and on the browser window tab. The following example shows a custom name displayed as a tooltip and on the browser window tab.
1. Open a command prompt as an administrator and type the following:
   
   `cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   
2. Type the following command:
   
   `tabadmin customize name "new_name"
   
   Replace `new_name` with your custom name, as in the following example:
   
   `tabadmin customize name "My Company"
   
   **Note:** To change to a name that includes Unicode characters, identify the hex encoding for each Unicode character and add `\u` before each hex value. For example, for the two-character string 测试, type the command `tabadmin customize name "\u6D4B\u8BD5"`
   
3. Type the following command to restart the server so that the change takes effect:
   
   `tabadmin restart`

**Change the logo**

You can customize the logo that appears on the Tableau Server sign-in page, the header logo in server pages, and the small logo that appears in the upper left when a view is being edited in web authoring. The name "Tableau" is part of this logo. It cannot be changed independently of the logo.

The image files you use should be in GIF, JPEG, or PNG format.

The `header_logo` image can be up to 160 by 160 pixels, but not smaller than 32 by 32 pixels. For best results use an image that's 125 by 35 pixels. If the image is larger than 160 by 160 pixels, it is clipped.

The `sign_in_logo` image can be a maximum of 3000 by 3000 pixels.

**Note:** The background colors differ in these locations, so your logo might look different depending on where it appears in the server interface.
Header logo

![Default header logo](image1)

This is the default header logo.

This is what a custom header logo might look like.

Sign-in logo

![Default sign-in logo](image2)

This is the default sign-in logo.
This is what a custom sign-in logo might look like.

Small logo

This is the default small logo for web authoring.
This is what custom small logo might look like for web authoring.

**Customize a logo**

1. Open a command prompt as an administrator and type the following command:
   ```bash
cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   
2. Type one of the following commands, depending on which logo you want to set. Substitute your own image file for C:\My Pictures\logo.png.
   ```bash
   tabadmin customize header_logo "C:\My Pictures\logo.png"
   tabadmin customize sign_in_logo "C:\My Pictures\logo.png"
   tabadmin customize smalllogo "C:\My Pictures\logo.png"
   ```

   **Note:** If an image for the logo or the header logo is larger than 160 by 160 pixels, it is clipped.

3. Type the following command to restart the server so that the change takes effect:
   ```bash
   tabadmin restart
   ```

**Restore the default name or logo**

1. Open a command prompt as an administrator and type the following command:
   ```bash
cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   
2. Type the following command:
   ```bash
   tabadmin customize parameter -d
   ```
   **For parameter, specify** name, logo, header_logo, sign_in_logo, or smalllogo.

3. Type the following command to restart the server so that the change takes effect:
   ```bash
   tabadmin restart
   ```
Language and Locale

Tableau Server is localized into several languages and has language and locale settings. The Language setting controls user interface (UI) items such as menus and messages. The Locale setting controls items in views such as number formatting and currency.

Administrators can configure language and locale on a server-wide basis (see Server Settings (General) on page 1177, and individual users can configure their own settings (search for "Your Account Settings" in the Tableau Server Help). If a user configures their own language and locale, their settings override the server settings.

Default Settings

The default language for Tableau Server is set during Setup. If the host computer is configured for a language Tableau Server supports, it installs with that language. If it’s not a supported language, Tableau Server installs in English.

How Language and Locale are Determined

Another influence on which language and locale display when a user clicks a view is the user’s web browser. If a server user has not specified a Language setting on their User Account page, and their web browser is set to a language that Tableau Server supports, the browser’s language will be used—even if Tableau Server itself is set to a different language.

Here’s an example: Assume that Tableau Server has a system-wide setting of English as the Language for all users. Server user Claude does not have a language specified on his Tableau Server User Account page. Claude’s browser uses German (Germany) for its language/locale.

When Claude signs in to Tableau Server, the server UI displays in German and when he clicks View A, it’s using the Germany locale for numbers and currency. If Claude had set his user account Language and Locale to French (France), the UI and view would have been displayed in French. His user account setting supercedes those of his web browser, and both of those have precedence over Tableau Server’s system-wide setting.

Another setting to be aware of is the Locale setting in Tableau Desktop (File > Workbook Locale). This setting determines the locale of the data in the view, such as which currency is listed or how numbers are formatted. By default, Locale in Tableau Desktop is set to Automatic. However, an author can override that by selecting a specific locale. Using the above example, if the author of View A set Locale to Greek (Greece), certain aspects of the data in View A would display using the Greek (Greece) locale.

Tableau Server uses these settings, in this order of precedence, to determine language and locale:

1. Workbook locale (set in Tableau Desktop)
2. Tableau Server User Account language/locale settings
3. Web browser language/locale
4. Tableau Server Maintenance page language/locale settings
5. Host computer’s language/locale settings

**Use Custom Fonts**

You can use custom fonts with Tableau Server. When you do this the safest way to guarantee that users have the experience you intend is to keep the following in mind:

- The fonts need to be installed on the computer where Tableau Server is running.
- The fonts need to be installed on any client computers that will connect to Tableau Server. You need to have the fonts installed locally in order for your browser to properly display them.
- As a best practice, use "web safe" fonts that are installed by default on all major browsers. This increases the likelihood that the fonts will display properly on client machines.
- Different browsers render the same fonts differently, so even when a client browser has the custom font installed, it may look different when viewed in different browsers. This can be especially noticeable with comments or titles where specific spacing is used for an intentional effect.

**Note:** For more information about installing fonts in Windows, see the Microsoft Knowledgebase.

**Navigate Server Admin Pages**

As a server administrator, you can access all of the menus and pages in Tableau Server for server and site management. If your server is configured for multiple sites, the site menu is available for navigation. Click **Manage All Sites** in the site menu to access server administration pages.

The server administrator pages include server-wide settings that you will use to configure, monitor, and maintain Tableau Server.

For information on navigating content pages, "Navigate Tableau Server" and "Access and Manage Your Content" in Tableau Server help.

**Server Administrator Pages**

In a single-site deployment, all server and site menus are available to you in the main menu. To create a site, click **Settings > Add a Site**.
On a multi-site server, when a site is selected, you will see these menus:

To access server administration pages, click the site menu, and then select **Manage All Sites**:

On a multi-site server, these are the server administration menus. The site menu text changes to **All Sites** to let you know you are managing server-wide settings.
To return to the site administration menus, click All Sites, and the select the site you want to manage.

Server administrators can:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.

To manage site-related settings, you must first navigate to a specific site. Within each site, you can:

- Manage content and assign permissions.
- Manage schedules for extract refreshes and subscriptions.
- Monitor site activity and record workbook performance metrics.
- Manage storage space limits for content published by users.
- Allow web authoring.
- Enable revision history.
- Allow site administrators to add and remove users.
- Allow users to subscribe to workbooks and views, and allow content owners to subscribe others to workbooks and views.
- Enable offline snapshots for favorites (iOS only).

**Upgrade Tableau Server**

The topics in this section help you upgrade Tableau Server. The topics describe planning, testing, and actually upgrading your existing server installation. We include information about best practices, as well as steps for upgrading a single node server and a multi-node installation. Where possible, we call out possible pitfalls and help you to avoid these.

**Research the Upgrade**

Before you upgrade Tableau Server, we recommend that you plan the upgrade.

- Learn about the new version of Tableau Server, including what’s new and what’s changed.
  - **What's Changed - Things to Know Before You Upgrade** below

  **Note:** As of version 10.0 there are significant changes in the Tableau Server Setup program. Read about the **Tableau Server Setup changes on the next page** before you upgrade to version 10.0.

- Make sure the computers you are going to upgrade (both for the test environment and the production environment) meet the minimum hardware requirements. Minimum requirements and recommendations can change from version to version.
  - **Minimum Hardware Requirements and Recommendations for Tableau Server** on page 674
- Understand how version compatibility might impact your installation of Tableau Server.
  - **Desktop and Server Compatibility**

The following topics guide you through planning steps.

**What's Changed - Things to Know Before You Upgrade**

**What Changed in Version 10.0**

Version 10.0 includes some changes you should know about before upgrading.

The following sections summarize the significant changes to Tableau Server 10.0 and provide links for additional information.

- **Tableau Server Setup changes**
- Two-node installations are limited to a single instance of the repository on the next page
- Domain change from tableausoftware.com to tableau.com on the next page
- Minimum hardware requirements adjusted on the next page
- Tableau Server is no longer available as a 32-bit application on the next page
- Tableau Server no longer supports Windows Vista or Windows Server 2008 on the next page
- Tableau Server no longer supports older versions of Microsoft Internet Explorer on the next page
- Tableau Server does not support Microsoft Internet Explorer 11 and higher in compatibility mode on page 668
- API Server process (wgserver) has been removed on page 668

**Tableau Server Setup changes**

The changes described in this section have a significant impact on the upgrade process.

**Manual uninstall of previous version is no longer required**

Starting with version 10.0, you can upgrade Tableau Server without first manually uninstalling your previous version (when the previous version is 64-bit 8.2 or later). When you run the setup program, the existing version of Tableau Server is recognized and is uninstalled during the upgrade process.

**Note:** If you are upgrading to version 10.0 and want, you can manually uninstall the existing version before you upgrade, following the same upgrade process as you would in versions earlier than 10.0.

**Setup gives you a backup option**

If you follow the new workflow and let the Setup program uninstall your existing version of Tableau Server, you are prompted during the setup process create a full backup of your Tableau installation. This backup is a safety measure and is created for use in the event of an unexpected issue during upgrade. If you already have a backup of the current state of your installation, you can skip the backup during the upgrade to save time. For more information, see Tableau Server Upgrade Backup Options on page 688.
Two-node installations are limited to a single instance of the repository

If you configure a two-node installation of Tableau Server, you are limited to a single repository. If you are upgrading from a two-node installation that has two repositories, you will be prompted to remove one instance. For more information, see Install Tableau Server on a Two-Node Cluster on page 614.

Domain change from tableausoftware.com to tableau.com

As of version 10.0, the licensing server is located at tableau.com. In versions earlier than 10.0, this was located at tableausoftware.com. Any firewall rules or proxy configurations that specify tableausoftware.com for the licensing server must be updated for version 10.0. For more information on proxy settings, see Configuring Proxies for Tableau Server on page 579.

Note: Earlier versions of Tableau Server will continue to access the licensing server on the tableausoftware.com domain. If your organization is running versions of Tableau Server prior to 10.0, continue to use tableausoftware.com for proxy and firewall settings.

Minimum hardware requirements adjusted

With version 10.0, Tableau Server can be installed on a 2-core computer. (Previously, Tableau Server required at least 4 cores.) The 2-core configuration allows you to test Tableau Server on constrained hardware and is intended only for trials and prototyping. For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server on page 674.

Tableau Server is no longer available as a 32-bit application

With version 10.0, Tableau Server is available only as a 64-bit application. For information about upgrading a 32-bit version of Tableau Server to version 10, see Upgrade from 32-bit to 64-bit Tableau Server on page 689.

Tableau Server no longer supports Windows Vista or Windows Server 2008

With version 10.0, Tableau Server no longer supports Microsoft Windows Vista or Windows Server 2008 (Windows Server 2008 R2 is still supported).

Tableau Server no longer supports older versions of Microsoft Internet Explorer

With version 10.0, Tableau Server no longer supports Microsoft Internet Explorer 8, 9, or 10. This change impacts customers doing an initial install of Tableau Server on Windows 8.0 or Windows Server 2012 (non-R2). Neither of these operating systems supports Internet Explorer 11.

To complete configuration of Tableau Server you must use a browser on the server computer to add an administrator account. This requires browser that is supported by Tableau Server. This means that on Windows 8.0 and Windows Server 2012 non-R2, you must use the latest
version of Chrome, Firefox or Safari. You can uninstall this browser after the initial installation is complete.

**Note**: This only affects an initial installation and configuration, because no browser is required when upgrading or restoring Tableau Server.

For more information on the Microsoft policy for supporting Internet Explorer, see Microsoft Support Lifecycle.

**Tableau Server does not support Microsoft Internet Explorer 11 and higher in compatibility mode**

Version 10.0 of Tableau Server does not support legacy compatibility modes in Internet Explorer 11 and higher.

This change impacts you if your users view web pages that have Tableau views embedded in them and that set Internet Explorer to compatibility mode with HTML DOCTYPE values. This can be an issue with SharePoint configurations that force compatibility mode. To avoid having users view web pages that put their browser into compatibility mode, either adjust configurations so that Internet Explorer is not put into compatibility mode, or use another supported browser, such as Chrome, Firefox, or Safari.

**API Server process (wgserver) has been removed**

The API Server process (wgserver) has been removed from Tableau Server. The process was available but disabled by default in version 9.3. The API Server process was formerly used to support the REST API, but as of version 9.3, this functionality was moved to the Application Server process. The API Server process is now removed, even if you explicitly enabled the process in a previous version of Tableau Server. For information about changes to the REST API, see What's New in the REST API.

**Note**: For historical reasons, some tabadmin configuration options will continue to use "wgserver" in the option name, but this does not refer to the old API server.

**What Changed in Version 9.3**

Version 9.3 includes some changes you should know about before upgrading.


The updates to Tableau Server 9.3 have the following impact:

**New default configurations based on hardware**

The Tableau Server installer detects your computer's hardware and creates an optimal default configuration for a single-server installation and for the primary server in a multi-server
installation. The new default configuration determines the number of processes to run for each Tableau Server process type.

When you upgrade from a single-server or multi-server installation in which you previously accepted the default configuration, the upgrade process changes the configuration to the new hardware-based default configuration. However, if you upgrade a server where you configured a custom number of server processes, the upgrade preserves the custom configuration—both for single-server and multi-server upgrades.

If after you upgrade you want to revert to the previous default configuration, use the following table to determine the number of processes to set in the Tableau Server Configuration utility based on the number of CPU cores on the primary server:

### Number of Processes

<table>
<thead>
<tr>
<th>Number of Processes</th>
<th>VizQL Server</th>
<th>Data Server</th>
<th>Backgrounder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Before 9.3</td>
<td>In 9.3</td>
<td>Before 9.3</td>
</tr>
<tr>
<td>8 Cores</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>12 Cores</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>16 Cores or more</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

**Note:** If the computer where you installed Tableau Server has fewer than eight CPU cores, the default configuration has not changed from running one of each process.

For more information on the defaults for 9.3, see Primary Server Installation Defaults on page 614.

For more information on setting the number of processes for Tableau Server, see Reconfigure Processes on page 642.

**High Availability Postgres Repository - faster failover**

Improvements to the failover process now mean that processes do not need to be restarted after the passive repository is made active. This means that the downtime for a repository failover is significantly reduced.

**Distributed installation - manual worker upgrades**

Due to an update in third-party software, an upgrade to version 9.3 requires manual upgrade of worker nodes. A prompt during installation of 9.3 will let you know that worker nodes cannot be upgraded automatically. For more information on upgrading, see Perform the Upgrade on page 684.
API Server (wgserver) deprecated

The API Server process has been deprecated. In version 9.3, the process is still available in Tableau Server, but it is disabled by default for new installations. The API Server process was formerly used to support the REST API, but as of version 9.3, this functionality has been moved to the Application Server process. If you explicitly enabled the API Server process in a previous version of Tableau Server, the process will still be enabled in 9.3.

What Changed in Version 9.2

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


The updates to Tableau Server 9.2 have the following impact:

Assign Permissions to Contents setting

Because content permissions can be locked to the project, the **Assign Permissions to Contents** button has been removed and is no longer available for projects and workbooks. For more information, see [Quick Start: Lock Project Permissions](#) and [Lock Content Permissions to the Project](#) on page 869.

Schedules Run in Parallel by Default

When you create a schedule in Tableau Server, the schedule runs in parallel, that is, it runs on all available backgrounder processes at the same time. Schedules finish more quickly when they are run in parallel, but you have the option of running schedules serially as well. For example, you may want to run a very large schedule in serial to allow other schedules to run at the same time. For more information, see [About Extracts and Schedules](#) on page 909.

What Changed in Version 9.1

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

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

The updates to Tableau Server 9.1 have the following impact:

SAML authentication - logout

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

If your pre-9.1 Tableau Server is configured for SAML authentication, the logout functionality will not work until you reconfigure the metadata for SAML. You must re-export the
SAML metadata file and re-import it into your IDP. For more information about configuring SAML metadata, see Configure Server-Wide SAML on page 1019.

Hidden fields in published data sources - unavailable for workbooks

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

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

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

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

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

For information on unhiding fields in the Data pane, see Hide or Unhide Fields in the Tableau Desktop help.

Clickjack protection - enabled by default

Starting with version 9.1, clickjack protection is enabled by default on Tableau Server. The protection has been available for several releases, but had been off by default. For more information on clickjack protection and how it impacts embedded views, see Clickjack Protection on page 964.

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

What Changed in Version 9.0

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

The updates to Tableau Server 9.0 have the following impact:

Customizations

Default start page

Any user-defined default start page will be reset to the Tableau Server default start page. Users will need to reset their default start page after the upgrade. For more information about setting a default start page, search for "Manage Your Content Page and Account Settings" in the Tableau Server Help.

Custom logos

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

- The background for large custom logos is different based on logo location. On the navigation bar the background is black and on the sign-in screen the background is white. For more information, see Change the Name or Logo on page 655.
- The small logo option has been deprecated. There are no locations in Tableau Server where the small logo is displayed, so the option does not do anything.

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

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

- **64-bit Tableau Server**—At minimum you must have 4 cores, 8 GB of RAM, and 15 GB of free disk space to install the 64-bit version of Tableau Server.
- **32-bit Tableau Server**—At minimum you must have 2 cores, 4 GB of RAM, and 15 GB of free disk space to install the 32-bit version of Tableau Server.

For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server on page 674.

**Note:** If you are upgrading Tableau Server on a computer that does not meet the minimum hardware requirements, you will not be able to install Tableau Server 10.0. If you cannot upgrade 64-bit Tableau Server because of hardware requirements but your computer meets the minimum hardware requirements for 32-bit Tableau Server, you may be able to upgrade to 32-bit Tableau Server.
High availability and failover

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

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

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

The Tableau Software user

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

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

Internal PostgreSQL database password regeneration

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

tabadmin restore - Doesn’t automatically restart Tableau Server

Starting with version 9.0, a tabadmin restore command will not automatically start Tableau Server. If you want the server to start after doing a restore, use the --restart option. For more information, see restore on page 1285.

"Remember me" option

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

Session ID in URLs

With version 9.0 of Tableau Server, the session ID at the end of server URLs is now indicated by an "iid" parameter, :iid=<n>. For example,
This parameter replaces the hash symbol "#<n>" used for the session ID in 8.x versions of Tableau Server.

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

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

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

- **Minimum requirements** are the minimum hardware your computer must have in order to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping.

- **Minimum recommendations** are higher than minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the setup program will warn you but you can continue the installation.

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

**Note:** If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see Server Process Limits on page 654

Minimum Hardware Requirements

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

These minimum requirements are appropriate for a computer that you use for prototyping and testing of Tableau Server. They apply to single-node installations and to each computer in a distributed installation.
## Minimum Hardware Requirements

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>RAM</th>
<th>Free Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Hardware</td>
<td>2-core</td>
<td>8 GB</td>
<td>15 GB</td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For the requirements:

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

**Note:** For Tableau Server 10.0, you need a minimum of 2 physical cores. If you are installing on an Amazon EC2 instance, this means 4 vCPUs. For more information, see [Amazon EC2 Instances](#).

## Minimum Hardware Recommendations

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process.

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

## Determining Computer Hardware

To determine how many physical cores a computer has, the Tableau Server setup program queries the operating system. To view hardware information that the setup program detected on your computer, open the `tabadmin.log` file in the following folder on the computer where you are installing Tableau Server:
In the `tabadmin.log` file, look for lines similar to the following. These lines provide information about the physical and logical cores that the setup program detected and that it used to determine the core count that is being used for licensing.

```
2015-04-09 14:22:29.533 -0700_DEBUG_10.36.2.32:<machine name>_: _pid=21488_0x2cd83560__user=__request=__ Running hardware check

2015-04-09 14:22:29.713 -0700_DEBUG_10.36.2.32:<machine name>_: _pid=21488_0x2cd83560__user=__request=__ Detected 12 cores and 34281857024 bytes of memory

2015-04-09 14:22:29.716 -0700_DEBUG_10.36.2.32:<machine name>_: _pid=21488_0x2cd83560__user=__request=__ Hardware meets recommended specifications. Default values will be used.
```

**Manually determining the number of cores on your computer**

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

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

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

   The output will display the device ID or IDs and the number of physical cores the computer has.

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

In this example, there are two CPUs, each with six cores, for a total of twelve physical cores. This computer would satisfy the minimum hardware requirements for installing Tableau Server.
The following command shows a longer version that lists the logical processors as well as the physical cores.

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

In the above example, the server has a total of twelve physical cores, resulting in 24 logical cores.

**Prepare for the Upgrade**

To properly prepare for a Tableau Server upgrade, gather information about your existing installation, key files related to your installation and the upgrade, and complete the pre-upgrade tasks.

**Gather information and required files**

**Credentials**

You will need the following credentials in order to upgrade and configure Tableau Server:

- Credentials for a Windows user account that has administrative access on the Tableau Server computer. You need these credentials in order to run the setup program for upgrading and to run `tabadmin`. You also need administrative credentials for worker node computers if your upgrade requires you to upgrade the workers manually.

- Credentials for the Run As User account on the Tableau Server computer.

**Custom configuration information**

If you are upgrading on the existing hardware your configuration will be preserved, but it's a good practice to collect this configuration information about your existing installation for several reasons: you need this information when you configure a test environment, you need this information if you are migrating to new hardware migration as part of the upgrade, and you can use the information to confirm that the upgraded Tableau Server is configured as expected if you notice something unexpected after upgrading.

Collect this information and any associated files and save them to a location that is not on any of the Tableau Server computers.
The following list includes examples of the type of information you should gather:

- **Customizations** This includes non-default ports, timeout values, custom logo images, and fonts. Also make a note of Windows path environment variables that affect Tableau Server.

- **SMTP** configuration. You can see your current SMTP configuration on the SMTP Setup tab of the Configuration utility. For more information, see Configure SMTP Setup on page 617.

- **SSL** configuration and certificates. You can see your SSL configuration on the SSL tab of the Configuration utility. This tab also lists the location of the certificate and certificate key files. You should copy and save these files in a safe location. For more information, see Configure External SSL on page 972.

- **SAML** configuration, certificates and any IdP metadata files. You can view your current SAML configuration on the SAML tab of the Configuration utility, including the certificate, key, and metadata files. Save copies of these files to a safe location. For more information, see Configure SAML Setup on page 617.

- **Kerberos** configuration. You can see your current Kerberos configuration on the Kerberos tab of the Configuration utility, including the location of the keytab file you should copy and save. For more information, see Configure Kerberos on page 993.

- **OpenID** configuration. Find your current OpenID configuration details on the OpenID tab of the Configuration utility. For more information, see Configure Tableau Server for OpenID Connect on page 1054.

- **Worker** configurations. Collect the configurations of your worker nodes, including any certificates or other supporting files that you have needed to copy to the worker nodes. You can find detailed information about the number of processes configured on each node on the Servers tab of the Configuration utility. See Reconfigure Processes on page 642 for more information.

- **Other** values. Note the number of projects, groups, workbooks, views, data sources, and users you have in your production environment. Having this information makes it easy to do a quick check after the upgrade to make sure everything was restored as expected.

**Environment configuration**

The steps you need to take during the upgrade process depends on whether you're installing the upgrade on the same hardware or you're migrating to new hardware. Upgrading on the same hardware is straightforward and requires a minimum of manual steps (the steps you need to take depend on what version you are upgrading from, what version you are migrating to, and whether or not your existing installation is in the default location). Migrating to new hardware requires you to manually restore your Tableau Server data and reconfigure your settings after you install the new version.
Install location

By default Tableau installs to C:\Program Files\Tableau\Tableau Server. If your current Tableau Server installation is to a non-default path, you need to note the current location in case you are prompted to provide it during upgrade. Depending on your existing version and the version you are upgrading to, the setup program may not find your existing data and configuration unless you specify it.

Setup files

You will need the following setup files before you upgrade Tableau Server:

- The setup program for your existing version of Tableau Server; if you are upgrading a distributed installation, the Tableau Worker Software setup program.

  You might not need these. However, we recommend that you have them available in case there's a problem during the upgrade. That way you can use these and your server backup to restore your installation to its pre-upgrade state.

  **Note:** If you do not have the install program for your existing version, you can download it from the Alternate Downloads Site. If you have a distributed installation of Tableau Server, you should also download the corresponding installer for worker nodes. Save the setup programs in a safe location that is not part of your production or test version of Tableau Server. You will need these if you need to go back to your existing version after upgrading.

- The setup program for the new version of Tableau Server, including the Worker Software Setup program, if you are upgrading a distributed installation.

**tabcmd**

A new version of tabcmd is released with every release of Tableau Server. If you install tabcmd on computers that are not part of your Tableau Server installation, you need to update tabcmd on those computers. The latest version of tabcmd is installed in the **extras** folder when you upgrade Tableau Server. For more information, see Install tabcmd on page 1315.

**Pre-Upgrade Tasks**

Perform these tasks before starting your upgrade.

**Back up Tableau Server data**

We recommend that you make a backup of your installation of Tableau Server before beginning the upgrade process. This provides data that you’ll need to set up a test version of the upgraded environment. It also lets you recover if the upgrade process fails.
Note: We recommend you disable subscriptions and scheduling in your production environment immediately before taking the backup, and reenable them after the backup is complete. Doing this will help avoid having your users receive duplicate subscriptions and email messages when you restore your backup in your test environment.

If you already have a backup that lets you restore the current state of the server, you can skip this step.

1. On the Tableau Server computer, open a command window as an administrator.
2. Change to the `\bin` folder where the `tabadmin.exe` file is. For example to change the `\bin` directory in a default installation:
   ```bash
   cd "C:\Program Files\Tableau\Tableau Server\version\bin"
   ```
3. Type the following command to remove unneeded files from the server:
   ```bash
   tabadmin cleanup
   ```
4. Type the following command to back up the server, substituting your own path and filename for `myserver.tsbak`:
   ```bash
   tabadmin backup myserver.tsbak
   ```
   If you're running Tableau Server version 9.3 or later, run the backup command using the `--verify` option to verify the backup:
   ```bash
   tabadmin backup --verify myserver.tsbak
   ```

For more information, see Remove Unneeded Files on page 1152 and Back Up Tableau Server Data on page 1144.

Check your product maintenance status

If you attempt to upgrade a Tableau Server installation that has a product key with expired maintenance, your upgraded Tableau Server will be unlicensed. Before upgrading, make sure that the server’s maintenance hasn't expired.

1. In Windows, select Start > All Programs > Tableau Server > Manage Product Keys.
2. In the Manage Product Keys dialog box, look for the expiration date under the
If your maintenance has expired, select the product key and then click **Refresh**. If the maintenance date isn’t updated, contact Tableau Technical Support. Reactivating the product key will be part of the upgrade process.

For more information, see *Activate Tableau* on page 603. If your server doesn’t have internet access, see *Activate Tableau Offline* on page 604.

**Test the Upgrade**

The best way to learn what impact a Tableau Server upgrade will have to your current environment is to test it. Knowing how an upgrade will affect your users and your server helps you plan and communicate before the actual upgrade, ensuring that your users will not be caught by surprise.

If you have a Tableau Server test environment this is a great place to test out the upgrade.

We recommend the following sequence for testing a Tableau Server upgrade:

1. **Prepare a test environment**
2. **Upgrade the test environment**
3. **Confirm that existing functionality works**
4. **Test new features**

**Prepare a test environment**

To start, create a test environment that mirrors your production environment as closely as possible. The closer your test environment is to the actual environment you will be upgrading, the more accurate a representation you will have of how the upgrade will impact you. This includes identical or similar hardware and operating systems, as well as the same authentication options and network access.

When you’ve got a test computer or virtual machine ready, follow these steps for creating a test environment.
**Note:** To perform these steps, you must be signed into the Tableau Server computer as a Windows administrator. This applies both to the existing server and to the test server environment.

1. On the existing production environment, create a backup of Tableau Server using the `tabadmin backup` command. If you're using version 9.3 or later, include the `--verify` option.

   For more information, see [Creating a Pre-Upgrade Backup on page 1146](#).

2. On your test environment, install a copy of the same version of Tableau Server as you have in your production environment.

   **Note:** You can download the setup program for your current version (and the Worker Setup Program for nodes in a cluster) from the [Alternate Downloads Site](#).

3. Restore your existing database *without* configuration data using the `tabadmin restore` command and the `--no-config` option.

   You don't want the existing configuration data because your test server will have different IP addresses, and you don't want the test environment to conflict with the existing server.

   For more information, see [Restore from a Backup on page 1150](#).

   If you are not the administrator of the production Tableau Server installation and do not have those credentials, you might need to reset the Tableau Server administrator password using the `tabadmin reset` command.

4. Manually replicate your existing Tableau Server configuration.

   You need to manually configure certain aspects of your environment because you restore the Tableau database using the `--no config` option. You also need to manually configure some customizations.

**Upgrade the test environment**

Follow the appropriate steps for upgrading the test environment, depending on your configuration:

- [Upgrade a Single-Node Tableau Server on page 684](#)
- [Upgrade a Multi-Node (Distributed) Tableau Server on page 685](#)

**Confirm that everything works as expected**

After you have the new version of Tableau Server installed and configured in your test environment, you are ready to test. You should test basic functionality, along with any special
aspects of server that your organization relies on. For example, if there are key subscriptions that your organization relies on, make sure that you test those.

These are some areas of testing to consider:

- **Server processes.** Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected (including on all worker nodes if this is a distributed installation).

- **User access.** Confirm that Tableau Server users can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.

- **Publishing workbooks and data sources.** Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect.

- **Viewing published workbooks.** Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views embedded in web pages (for example, in SharePoint pages).

- **Subscriptions and extract refreshes.** Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.

- **Permissions.** Confirm that permissions are still set as expected for users and content.

- **Command-line utilities and APIs.** If applicable, test the command line utilities (tabadmin and tabcmd) and programmatic access via APIs.

**Test new features**

Take a look at the new features that come with the version you are upgrading to, and at any features that were added between the version you currently have and the new version. Think about how to help your users understand the benefits of the features that apply to your environment.

For more information on new features, see What's New in the Tableau Server Help.

**Communicate about the upgrade**

The best way to make an upgrade go smoothly is by letting your organization know ahead of time about the upgrade and how it might impact them. If you’ve had users help test, take advantage of their experience by having them help communicate the changes they saw while testing. You can also provide user access to the test environment if there are key people who should see the upgraded version before the actual upgrade.
Perform the Upgrade

After you’ve completed the Prepare for the Upgrade on page 677, upgrade your existing Tableau Server installation to version 10.0 by following one of the topics listed at the bottom of this page. (If you are migrating to new hardware as part of your upgrade, refer to Migrate to New Hardware on page 690 instead.)

When you install the newer version of Tableau Server, use the same drive and directory that the earlier version used. This way, data and configuration settings from your earlier version can be automatically imported. If your installation location is not the default, see Upgrade Tableau Server to a Non-Default Location on page 689.

Before upgrading, read through the list of pre-upgrade tasks to make sure you have followed the recommended best practices. For details, see Prepare for the Upgrade on page 677.

If you are upgrading from 32-bit Tableau Server to 64-bit Tableau Server you must uninstall your existing version before installing the new version. For more information, see Upgrade from 32-bit to 64-bit Tableau Server on page 689.

As a best practice, you should always make a backup of your Tableau Server data before upgrading. This backup is necessary in the event that something unexpected happens during the upgrade, or if you need to roll back to your previous version of Tableau Server. If you create this backup yourself, you can do it while Tableau Server is running. If you are upgrading to version 10.0 or later you can choose to have the Setup program create the backup while upgrading Tableau Server, but the backup is created while the server is stopped so this extends the length of time the server is unavailable. For more information, see Prepare for the Upgrade on page 677, Tableau Server Upgrade Backup Options on page 688, and Back Up Tableau Server Data on page 1144.

Upgrade a Single-Node Tableau Server

The instructions in this topic explain how to perform a Tableau Server upgrade on a single-node server, without changing the hardware or other configuration as part of the upgrade. If this is not your situation, see the following topics for instructions:

- If you are upgrading a distributed (multi-node) Tableau Server installation, see Upgrade a Multi-Node (Distributed) Tableau Server on the next page.
- If you are upgrading to a non-default location, see Upgrade Tableau Server to a Non-Default Location on page 689.
- If you are upgrading to new hardware, see Migrate to New Hardware on page 690.
- If you are upgrading from a 32-bit version of Tableau Server to a 64-bit version, see Upgrade from 32-bit to 64-bit Tableau Server on page 689.

Before you upgrade

Make sure you’ve prepared for the upgrade by reviewing changes in the new version and by gathering required information. For more information, see Research the Upgrade on
page 665 and Prepare for the Upgrade on page 677. Also make sure you’ve downloaded the Tableau Server setup program and Worker Software setup program from the download site. Be sure to download the same version for both primary and workers. You will need the worker software if it turns out you must manually upgrade worker nodes. Optionally (but recommended), test the upgrade on a test environment that mimics your production environment. For more information, see Test the Upgrade on page 681.

Perform the upgrade

1. Run the Tableau Server setup program on the Tableau Server computer.
   
The setup program will determine that you have a previous version installed and prompt you for a backup option.

2. Specify whether you want the setup program perform a full backup before it uninstalls the existing version. This backup is only needed if something unexpected happens during the upgrade. The server will be unavailable while the backup is being created. If you have a recent backup you can choose the option to not do the full backup. For more information, see Tableau Server Upgrade Backup Options on page 688.

3. After uninstalling the existing version (and creating a backup, if you selected that option), the installation process prompts you for an install location. If you are upgrading a version that was installed to the default location, you can accept the default. If you are upgrading an installation that was installed to a non-default location, navigate to the original location.

   If you do not navigate to the original location, your data and configuration settings from the original installation will not be found and used for your upgraded installation.

   **Note:** If you are upgrading from a version of Tableau Server that is earlier than 9.x, your existing extracts will be migrated to the new File Store during upgrade. This process may take a long time (up to several hours if you have a large number of extracts or extracts that are large in size). While this takes place a message displays: "Migrating extracts to File Store This process may take up to several hours." For more information, see Troubleshoot Tableau Server Install and Upgrade on page 1221

Upgrade a Multi-Node (Distributed) Tableau Server

In a multi-node (distributed) installation of Tableau Server, you need to upgrade all the nodes (primary and all workers) so they are running the same version of Tableau.

Start by upgrading the primary node. For some upgrade paths (for example, from version 9.3 to version 10.0), the upgrade process will automatically push an upgrade from the primary node to the worker nodes. If the workers cannot be automatically upgraded, usually when the upgrade includes updates to PostgreSQL drivers or other third-party software, the Setup program will let you know that you must manually upgrade the worker nodes.
Important: You should not upgrade a worker node before upgrading the primary node. Start the upgrade on the primary node, and if prompted by the setup process, move to the worker nodes and manually upgrade them before returning to the primary node to complete the upgrade. If you upgrade a worker node before upgrading the primary node, you will need to completely uninstall Tableau Server, reinstall, and restore from your backup.

- If you are upgrading to a non-default location, see Upgrade Tableau Server to a Non-Default Location on page 689.
- If you are upgrading to new hardware, see Migrate to New Hardware on page 690.
- If you are upgrading from a 32-bit version of Tableau Server to a 64-bit version, see Upgrade from 32-bit to 64-bit Tableau Server on page 689.

Upgrading a high availability installation of Tableau Server requires some additional steps to upgrade the backup primary, as explained later in this topic.

Before you upgrade

Make sure you’ve prepared for the upgrade by reviewing changes in the new version and by gathering required information. For more information, see Research the Upgrade on page 665 and Prepare for the Upgrade on page 677. Also make sure you’ve downloaded the Tableau Server setup program and Worker Software setup program from the download site. Be sure to download the same version for both primary and workers. You will need the worker software if it turns out you must manually upgrade worker nodes. Optionally (but recommended), test the upgrade on a test environment that mimics your production environment. For more information, see Test the Upgrade on page 681.

Upgrade the primary node and worker nodes

1. Run the Tableau Server setup program on the primary server node.
   The setup process will determine that you have a previous version installed and prompt you for a backup option.

2. Specify whether you want the setup program perform a full backup before it uninstalls the existing version. This backup is only needed if something unexpected happens during the upgrade. The server will be unavailable while the backup is being created. If you have a recent backup you can choose the option to not do the full backup. For more information, see Tableau Server Upgrade Backup Options on page 688.

3. After uninstalling the existing version (and creating a backup, if you selected that option), the installation process prompts you for an install location. If you are upgrading a version that was installed to the default location, you can accept the default. If you are upgrading an installation that was installed to a non-default location, navigate to the original location.
If you do not navigate to the original location, your data and configuration settings from the original installation will not be found and used for your upgraded installation.

4. With most upgrades, the worker nodes will be automatically updated.

   **Note:** When worker nodes are upgraded automatically, the Windows registry on the worker nodes will not reflect the upgrade, so the old version will still show in the program list of Control Panel. This is expected.

If the upgrade process requires you to upgrade the worker nodes manually, you will be prompted during installation of the primary with a message: "One or more workers could not be upgraded automatically."

   a. If you see this message, sign in to each worker node and run the Tableau Worker setup program.
   
   b. Setup uninstalls the existing version of the worker node software and installs the new version.
   
   c. Return to the primary server.

5. Follow the prompts in the Setup program to complete the installation.

   **Note:** If you are upgrading from a version of Tableau Server earlier than 9.x, your existing extracts will be migrated to the new File Store during upgrade. This process may take a long time (up to several hours if you have a large number of extracts or extracts that are large in size). While this takes place a message displays: "**Migrating extracts to File Store This process may take up to several hours.**" For more information, see Troubleshoot Tableau Server Install and Upgrade on page 1221

If you are upgrading a distributed installation that includes a backup primary, you need to upgrade the backup primary in a separate step, after upgrading the rest of the cluster.

### Upgrade a backup primary server

If you have configured your Tableau Server installation for high availability, after you've finished the upgrade for the primary node and the worker nodes, upgrade the backup primary computer.

1. On the backup primary, run the Tableau Server Setup program.
   
   Setup will determine that you have a previous version installed and prompt you for a backup option. Choose to not create a backup.

2. If you changed hardware or any configuration values on the primary or worker nodes
during the upgrade, reconfigure the backup primary by following the instructions in Create a Backup Primary on page 733.

**Tableau Server Upgrade Backup Options**

When you upgrade Tableau Server, the setup program can make a backup as part of the upgrade process. This can be helpful in case something unexpected happens while you are upgrading, or if you need to go back to your previous version.

**Note:** Beginning with version 10.0 of Tableau Server, the Setup program gives you the option to skip making a full backup.

You might choose instead to create a backup before you start the upgrade process and then skip the backup during the upgrade. This lets you create a backup while Tableau Server is running and available to your users. If the backup occurs during the setup process, the server is stopped, and it increases the length of time your users cannot access their Tableau content.

For more information on backing up Tableau Server, see Back Up Tableau Server Data on page 1144.

If the setup process detects an existing installation of Tableau Server, it offers you the following options:

- **Full backup.** When you select this option, the setup process performs a complete backup of your Tableau data and configuration before it uninstalls the existing version.

  Choose this option if you haven't recently made a backup and it is acceptable for Tableau Server to be unavailable to your users during the backup. A full backup can take a significant amount of time—up to several hours if you have a large installation or a lot of stored data (extracts). Because the server is unavailable during the period the backup is being created, choose this option only if you do not already have an up-to-date backup.

- **Without full backup.** When you select this option, the upgrade process uninstalls the previous version without making a backup.

  Choose this option only if you have a recent backup. This option can save you a significant amount of time during the upgrade (for data-heavy installations, you can save hours).

**Note:** Any changes made between the time you took the backup and the time you do the upgrade are lost because they aren't included in the backup.
Special Installation Scenarios

The bulk of the upgrade instructions cover typical upgrade situations. Not everyone is doing a straightforward or default upgrade, and the following topics cover some of the most common non-standard situations.

Upgrade from 32-bit to 64-bit Tableau Server

Starting with version 10.0, Tableau Server is available only as a 64-bit application. In previous versions, it was also available as a 32-bit application.

A distributed installation of Tableau Server must run the same bit version (all 32-bit or all 64-bit) and release version (10.0 for example) on all nodes. When you upgrade a distributed installation of Tableau Server to the 64-bit version, you need to manually upgrade each worker node by uninstalling the 32-bit version on each worker before you install the 64-bit version of the worker software.

1. Uninstall the 32-bit version on your primary Tableau Server computer.

   **Note:** Uninstalling removes the server software but leaves your data and configuration settings intact. If your existing 32-bit version was installed to the default location (C:\Program Files (x86)\Tableau\Tableau Server\<version>) the 64-bit Setup program will find the data and configuration and use it in the upgraded installation. If your existing 32-bit version was installed to a non-default location, see Upgrade Tableau Server to a Non-Default Location below.

2. Install 64-bit Tableau Server on the primary Tableau Server node.
3. Uninstall the 32-bit version on each worker node.
4. Install 64-bit Tableau Server Worker software on each worker node.
5. Return to the primary server and complete the configuration of 64-bit Tableau Server.

Upgrade Tableau Server to a Non-Default Location

You need to install Tableau Server to the same location as your existing Tableau Server so the setup process can locate and upgrade the existing configuration and data.

By default, the Tableau Server setup program installs Tableau Server in the following location:

C:\Program Files\Tableau\Tableau Server\version

and stores data and logs in this location:

C:\ProgramData\Tableau\Tableau Server
If your existing installation is in the default location, you can accept the defaults in the Setup program, and the install process finds your associated data and configuration settings during upgrade.

If your existing installation is not in the default location, you need to tell the Setup program where to install the new version of Tableau Server. How you specify a non-default upgrade location when prompted by the Setup program depends on whether you browse to the location or type the path.

- If you browse to the install location, the Setup program takes that path and appends \Tableau Server to it, and then installs to a \<version> folder immediately below the \Tableau Server folder. You can see the path that Tableau Server will be installed to when you are browsing, and you can edit that path if it is not correct (if, for example, you do not have a \Tableau Server folder).

- If you type the install location, the Setup program accepts the path you type and installs to a \<version> folder at the end of that path.

Don’t include the \<version> folder when browsing or typing the path.

For example, to upgrade an existing 9.0 Tableau Server is installed to D:\Tableau\9.0, you can either

- Browse to D: (the Setup program will populate the path as D:\Tableau Server) and edit the path so it reads D:\Tableau

  or

- Type the path as D:\Tableau

Migrate to New Hardware

Use the following procedure to migrate Tableau Server from one computer to another. Specifically, these steps describe how to move Tableau Server data and configuration settings from your in-production computer to a new computer where Tableau Server is installed. Before you start, make sure you have followed the steps in Prepare for the Upgrade on page 677, including creating a backup. The backup is what you use to restore your Tableau Server data.

1. Install Tableau Server on the new computer. For details, see Run Server Setup on page 602.

2. Copy your backup file .tsbak to the bin folder on your new Tableau Server (for example, C:\Program Files\Tableau\Tableau Server\<version>\bin).

   **Note:** The file does not need to be in the bin directory but if it is not, you will need to include a full path to the file when you run the restore command.
3. Next, stop Tableau Server.

4. Restore your in-production data without configuration information to your new Tableau Server installation using the `tabadmin restore` command:

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

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

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

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

   For more information about restoring Tableau Server data, see Restore from a Backup on page 1150.

5. Start the server.

6. **Distributed installations only**: Run the Tableau worker installer on the new worker computers you want to add to your new Tableau Server cluster. See Install and Configure Worker Nodes on page 699 for steps.

7. The same Tableau Server product key can be activated three times: once for a production environment, once for a test environment, and once for a QA environment. After you have tested your new Tableau Server installation and confirmed that it’s ready for production, you must deactivate your earlier production version of Tableau Server, and then you must uninstall it. To deactivate the earlier version:

   a. Select Start > All Programs > Tableau Server > Manage Product Keys.

   b. For each product key, select the product key and click Deactivate.

   **Note**: If you do not have an internet connection, you are prompted to create an offline activation file to complete the deactivation process. See Activate Tableau Offline on page 604 for steps.

## Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes on page 1212.

### General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:
1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

2. Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tabadmin restart` command. This will stop all the processes associated with Tableau Server and then restart them.

3. Clean up files associated with the Coordination Service (ZooKeeper). To clean up Coordination Service files, use the `tabadmin cleanup --reset-coordination` command.

Starting Tableau Server

Tableau Server cannot determine if it fully started

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

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

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

Tableau Server doesn’t start

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

Installing Tableau Server

Install fails due to hardware requirements

Starting with version 9.0, Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to both primary server computers and worker computers. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server on page 674.

Install or upgrade generates an error when PostgreSQL ODBC driver does not install correctly

In certain circumstances (when a system reboot is pending, or another program is being installed or updated, the Tableau Server PostgreSQL ODBC driver does not install correctly. When this happens, this message displays:
PostgreSQL ODBC driver (64-bit) version 09.03.0400 did not install properly.

**Note:** The version may be different, depending on what version of Tableau Server you are installing.

If this occurs, follow these steps to correct the issue:

1. Check to see if the driver shows as installed in Control Panel.
2. If the driver is not installed, download it from the Tableau Drivers page and install it.
3. If the driver is installed, uninstall it from Control Panel, restart the computer, download the driver, and install it again.

**Upgrading Tableau Server**

**Extract migration is slow**

Tableau Server 9.0 introduced a more reliable storage mechanism for data extracts called the File Store. Upgrading from a previous version requires migration of the extracts. This can take a long time (up to several hours) if you have a large number of extracts or extracts that have a lot of data. During migration a message displays:

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

If the migration progress appears to be stalled or stuck, you can verify that migration is continuing by watching the `tabadmin.log`. An entry is written to this log for each extract that is migrated. You can periodically copy the log and open your copy in a text editor like Notepad to verify that entries are being written to it.

**Upgrading fails due to lack of disk space**

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

To free up disk space:

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

After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.
2. Clean up unnecessary files using the `tabadmin cleanup` command. For more information, see Remove Unneeded Files on page 1152.

Reindexing Tableau Server Search & Browse

Problems that can be solved by reindexing Search & Browse

Symptoms of an index that needs to be rebuilt include:

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

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

Distributed Environments

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

Quick Start: Distributed Server

Increase the scalability of your Tableau Server environment by distributing the server components across several machines. Install Tableau Server on your primary node, then use the Worker Installer to install the software onto one or more “worker” nodes. With the worker nodes installed, configure the primary node to use those workers.

1 Install Tableau Server on Worker Nodes

Download the Tableau Server Worker installer from the download site and install it on all of the computers (other than the primary server) that you want to include in the distributed installation of Tableau Server. Be sure to download the same version of the worker software that you installed on the primary node.

Install Tableau Server on the primary node before you install on the other nodes in the distributed installation.

2 Open Configuration Utility

On your primary node, open the configuration utility by selecting Tableau Server 10.0 > Configure Tableau Server.
3 Add Worker Nodes

In the Configuration Utility, select the Servers tab and click Add. In the Add Tableau Server dialog box, type the IP address or the name of the worker node in the Computer box and specify the number of processes to allocate to the node. Repeat this for each machine you want to include in the distributed installation.

You can click Discover to automatically add any worker computers you installed in step 1 above.

4 Configure & Update Primary Node

After you set up the worker nodes, make all configuration changes and updates on the primary node. Use the command line tools and the Tableau Server Configuration utility on the primary
node. Updates will be pushed to the workers automatically.

**Distributed Requirements**

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

**Hardware**

While the computers you use in your cluster must meet the requirements described in *Before you install... on page 570*, they do not need to be identical.

**Hardware Guidelines for High Availability**

Here are some guidelines for the systems you use for failover and high availability:

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

  **Note**: If you install Tableau Server on a two-node cluster, a message displays to let you know that you are limited to one instance of the repository, and that high availability and failover are not available in a two-node configuration. You can add a third node but are not required to do so. In a two-node cluster, if one of the two nodes goes down, Tableau Server may not function correctly.

- **Failover & multiple gateway support—three computers and a load balancer**: To configure a cluster that provides the above plus support for multiple gateways, you need at least three computers or VMs, and a load balancer to front the cluster.

- **High availability—four computers and a load balancer**: To configure for high availability, you need the resources described above plus an additional computer to be the backup primary for your primary Tableau Server.

- **Primary computers**: If you configure for high availability, the primary Tableau Server and the backup primary may be running few or no Tableau Server processes. Therefore, the computers that run the primary and backup primary do not need as many cores as the ones running your worker servers. You will, however, need adequate disk space for backups because the primary computer is used during the database backup and restore processes. In addition to the amount of space needed for the backup file, you need temporary disk space roughly 10 times the size of the backup file (so if your backup is 4 GB, you should have about 40 GB of temporary disk space available).

**Software**

As of version 10.0, Tableau Server is only available in a 64-bit version.
Earlier versions were offered in both 32-bit and 64-bit versions. If you are running a version of Tableau Server that was available in both 32-bit and 64-bit, be aware that each computer must run the bit version—either all 64-bit or all 32-bit. For example, if the primary Tableau Server is running the 64-bit version of Tableau Server, the workers in the cluster must run the 64-bit version of Tableau Server Worker.

**Networking and Ports**

- **Ports**: As with any distributed system, the computers or VMs you use need to be able to communicate with one another. See Tableau Server Ports on page 1244 for a list of ports that must be available on the gateways and workers.

- **Same domain**: If Tableau Server is installed in a Windows Active Directory environment, then all computers in a cluster must be members of the same domain.

- **Service account**: The server’s Run As User on page 577 account, which is specified on the primary Tableau Server, must be the same on each computer in the cluster. If you are not running in an Active Directory environment, then we recommend updating the Run As User with a Windows workgroup user. You must specify the same user account and password on each node in the cluster. While you can leave the default NetworkServices account on each node in the cluster, we do not recommend this as a best security practice.

- **Static IP addresses**: Any computer running Tableau Server, whether it’s a single server installation or part of a cluster, must have a static IP address (learn more).

- **Discoverable**: Each node in the cluster must be discoverable from other node computers using DNS or a local host file.

**Best Practices**

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

- **IP addresses or computer names**: Note the IPv4 addresses or computer names of each computer or VM you’ll be working with. You will need to provide them during Tableau Worker Setup and configuration. As mentioned above, each computer in the cluster must use a static IP address, even if you use the computer’s name to identify it during configuration.

- **CNAME record**: If you’re configuring for high availability and you are not using a load balancer, make sure your primary Tableau Server and backup primary have the same CNAME record so that your Tableau Server users have a smooth experience if one primary fails and you configure the other to take over. If you are using a load balancer, it’s the load balancer’s name that users will be using as the Tableau Server URL, regardless of the gateway that’s actually handling the request.

- **User account credentials**: For each computer, you need credentials for a user account with local admin permissions. If you’re configuring for high availability, the Run
As account you use for your primary Tableau Server must be the same as the one you use for your backup primary Tableau Server.

- **Backup**: It's a best practice to create a backup prior to making significant system changes. See [Back Up Tableau Server Data](#) on page 1144 for steps.

**SSL**

If you are planning to configure SSL for a highly available Tableau Server cluster with multiple gateways and a load balancer ([learn more](#)), make sure that the SSL certificate you use was issued for the load balancer's host name. See [Configure SSL for a Cluster](#) on page 974 for other details.

**Hostname Support in Tableau Server**

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

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

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

**To update the Tableau Server configuration:**

1. On the primary Tableau Server, open a command prompt as an administrator.
2. Type the following:
   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   ```
3. Stop the server:
   ```
   tabadmin stop
   ```

**Note**: If the IP address on the primary has already changed, the `tabadmin stop` command will not work because the worker nodes will not accept connections from the new IP address. If you cannot successfully stop Tableau
Server, restart the worker nodes and then try again to stop the server. When the server stops, you can skip to step 6 to update the configuration.

4. Update the IP address for the worker node using the Tableau Server Configuration utility on the primary:
   a. From the Windows Start Menu, search for "Configure Tableau Server" and run the utility.
   b. On the Servers tab, select the worker whose hostname you want to change, and click Edit.
   c. Type the name for the worker node and click OK.
   d. Click OK again to close the Configuration utility.

5. Start the server:
   ```
   tabadmin start
   ```

**Install and Configure Worker Nodes**

After you complete the initial configuration, you can set up Tableau Server to run on multiple computers. This is called a distributed installation, or cluster.

Running a distributed installation uses additional ports on the primary Tableau Server and requires that certain ports be available for binding during Setup on the Tableau Worker Servers. See Tableau Server Ports on page 1244 for more information. There are also additional requirements to be aware of when you run a distributed installation. See Distributed Requirements on page 696 for details.

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

To install Tableau Server worker nodes:

1. Make sure you’ve installed Tableau Server on the primary computer.
2. Stop Tableau Server on the primary node (see Tableau Server Monitor on page 1176 to learn how).
3. Download the Tableau Server Worker software from the Tableau Customer Account Center.
4. Run Tableau Server Worker Setup on each computer you want to add to the Tableau Server cluster.

5. During installation you will be asked to provide the IPv4 addresses or computer name of the primary server. Using a computer name is recommended.

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

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

**Note:** If you configured SAML on the Tableau Server primary node, you need to copy the SAML certificate, SAML key, and SAML IdP metadata files to each node that's running a Tableau application server process (vizportal.exe). For more information, see Configure a Server Cluster for SAML on page 1021.

To configure Tableau Server for worker nodes:

1. Once the Worker software is installed on worker computers, and with the primary Tableau Server still stopped, return to the primary server and open the configuration utility by selecting **Tableau Server 10.0 > Configure Tableau Server** on the Start menu.

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

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

With the 64-bit version of Tableau Worker Server, you can run up to two instances of each process. In rare cases and if the server's hardware allows, that limit can be changed. See Server Process Limits on page 654 and Tableau Server Performance on page 1086 for more information.

**Note:** You can only add a second repository if you have three or more nodes. If you attempt to add a repository to a two-node cluster, a message displays to let you know that you are limited to one instance of the repository, and that high availability and failover are not available in a two-node configuration.

By default, the data engine and file store, repository, and gateway are hosted on the primary server. Running these processes on an additional server, or moving them off of the primary server, is part of configuring for high availability. See High Availability on page 709 for more information.

4. Click **OK**. It may take several minutes for the updates to complete.

5. Repeat these steps for each computer you want to add to the distributed environment.
When you're finished adding workers, click **OK** to save the changes and close the Configuration utility, then start Tableau Server on the primary node.

**Database Drivers**

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

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

<table>
<thead>
<tr>
<th>Server process</th>
<th>Requires database driver?</th>
</tr>
</thead>
<tbody>
<tr>
<td>VizQL server</td>
<td>yes</td>
</tr>
<tr>
<td>Application server</td>
<td>yes</td>
</tr>
<tr>
<td>Data server</td>
<td>yes</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>yes</td>
</tr>
<tr>
<td>API server</td>
<td>yes</td>
</tr>
<tr>
<td>Data engine (extract storage)</td>
<td>no</td>
</tr>
<tr>
<td>Repository</td>
<td>no</td>
</tr>
<tr>
<td>Gateway</td>
<td>no</td>
</tr>
<tr>
<td>Cluster controller</td>
<td>no</td>
</tr>
<tr>
<td>Cache server</td>
<td>no</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>no</td>
</tr>
<tr>
<td>File store</td>
<td>no</td>
</tr>
</tbody>
</table>

**Reinstall and Configure Worker Node**

You might need to reinstall one of your Tableau worker nodes. To do so, follow one of these procedures. The specific steps you take depend on whether or not the worker you are reinstalling has data engine or repository components on it and whether or not these are duplicated on any other node in the installation.
Use the following procedure to help you reinstall and configure a worker node that is hosting the only data engine or repository in the distributed installation. Every Tableau Server installation requires at least one data engine and one repository. If you are reinstalling the worker node that hosts either of these processes, you must first add the process to a second node.

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

1. Create a full backup of Tableau Server. For more information, see Back Up Tableau Server Data on page 1144.

2. Stop Tableau Server on the primary by selecting Tableau Server 10.0 > Stop Tableau Server on the Windows Start menu, or by running the `tabadmin stop` command from the command line.

3. On the Start menu, select Tableau Server 10.0 > Configure Tableau Server.

4. In the Configuration Utility:
   - On the General tab, enter your password.
   - On the Servers tab, add the data engine and/or repository components that the worker is hosting to another worker or to the primary, and then save your changes. For example, if the worker you are reinstalling currently hosts the data engine, add the data engine to another node.

5. Start the primary Tableau Server node so that synchronization completes between the existing data engine or repository on the worker you will be reinstalling and the newly added instances of those processes.

6. Open the Status page in Tableau Server and check on the components you added:
   - If you added a data engine/file store, wait until the new file store status no longer says "Syncing".
   - If you added a repository, wait until the new repository status says "Passive".

7. Stop Tableau Server.

8. If you are removing a node that hosts data engine, decommission the file store you are removing:

   From the Windows command line, in the C:\Program Files\Tableau\Tableau Server\10.0\bin directory, run:
   
   `tabadmin decommission <worker_node>`

   where `<worker_node>` is the name or ip address of the worker you are going to remove, as it appears in the list of servers on the Servers tab of the Configuration utility.
9. In the Configuration Utility:
   - On the **General** tab, enter your password.
   - On the **Servers** tab, select the worker you want to reinstall and then click **Delete**.
   - Save your changes.

10. Start Tableau Server and verify that everything is working as expected.

11. On the worker:
   - Uninstall the Tableau Server worker software from Windows Control Panel.
   - Delete (or rename) the following folders: `C:\Program Files\Tableau` and `C:\ProgramData\Tableau`. `ProgramData` is a hidden folder so may not be visible.
   - Install the updated worker software.

12. On the Tableau Server primary, stop Tableau Server, add the worker back into the configuration, and then save the changes.

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


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

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

1. Create a full backup of Tableau Server.

2. Stop Tableau Server on the primary by selecting **Tableau Server 10.0 > Stop Tableau Server** on the Start menu or by running the `tabadmin stop` command at a command prompt.

3. If you are removing a node that includes a data engine/file store pair, decommission the file store on that node:

   From the Windows command line, in the `C:\Program Files\Tableau\Tableau Server\10.0\bin` directory, run:

   `tabadmin decommission <worker_node>`

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

4. Open the configuration utility by selecting **Tableau Server 10.0 > Configure Tableau Server** on the Start menu.
5. In the Configuration Utility:
   - On the General tab, enter your password.
   - On the Servers tab, select the worker you want to reinstall and then click Delete.
   - Save your changes.

6. Start Tableau Server and verify that everything is working as expected.

7. On the worker:
   - Uninstall the Tableau Server Worker software from Control Panel.
   - Delete (or rename) the following folders: C:\Program Files\Tableau and C:\ProgramData\Tableau. \ProgramData is a hidden folder so may not be visible.
   - Install the updated worker software.

8. On the primary node, stop Tableau Server, use the configuration utility to add the worker back into the configuration, and then save the configuration.

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


**Maintain a Distributed Environment**

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

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

You can monitor the status of the Tableau Server cluster on the server Maintenance page. See **Server Maintenance on page 1154** to learn more about maintaining the server.
Move the Repository Process

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

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

1. Create a full backup of Tableau Server. For more information, see *Back Up Tableau Server Data* on page 1144.

2. If you haven’t done so already, stop Tableau Server and run the Tableau Server Configuration utility (*Start > Tableau Server 10.0 > Configure Tableau Server*) on the primary Tableau Server node.

3. On the **Servers** tab, select the computer (IP address or computer name) onto which you want to move the repository and click **Edit**. It can be another worker or the primary (*This Computer (Primary)*).

4. In the **Edit Tableau Server** dialog box, select the **Repository** check box and click **OK** to close the dialog box.

**Note:** Beginning with version 10.0, you cannot have more than one instance of the repository unless you have at least three nodes. For more information, see *Two-node installations are limited to a single instance of the repository* on page 667.

5. Click **OK** in the Tableau Server Configuration utility to save your changes and close the utility.
6. **Start the primary Tableau Server node** so that synchronization completes between the existing repository and the newly added repository.

7. Open the Status page in Tableau Server and wait until the new repository status no longer says "Setting up". When the repository status is "Passive" the synchronization is complete.

8. **Stop the server** and open the Tableau Server Configuration utility.

9. On the **Servers** tab, highlight the computer from which you are removing the process and click **Edit**.

10. Remove the processes you are moving: clear the **Repository** check box and click **OK**.

11. Click **OK** again to save your changes and close the utility.

12. **Start the primary server** so that the changes can take effect.

If you are performing this procedure as part of deleting a worker node from the Tableau Server configuration (as described in **Remove a Worker Node on the next page**), stop Tableau Server again before proceeding.

**Move the Data Engine and File Store Processes**

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

1. Create a full backup of Tableau Server. For more information, see **Back Up Tableau Server Data** on page 1144.

2. If you haven’t done so already, **stop the primary Tableau Server node** and run the Tableau Server Configuration utility (**Start > Tableau Server 10.0 > Configure Tableau Server**) on the primary Tableau Server node.

3. On the **Servers** tab, highlight the computer (IP address or computer name) onto which you want to move the processes and click **Edit**. It can be another worker or the primary (**This Computer (Primary)**).

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

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

5. Click **OK** in the Tableau Server Configuration utility to save your changes and close the utility.
6. **Start the primary Tableau Server node** so that the changes can take effect.
7. Open the Status page in Tableau Server and wait until the new file store status no longer says "Synchronizing".
8. **Stop the server.**
9. **Decommission** the file store on the worker:
   From the Windows command line, in the C:\Program Files\Tableau\Tableau Server\10.0\bin directory, run:
   ```
tabadmin decommission <worker_node>
   ``
   where `<worker_node>` is the name or ip address of the worker you are going to remove, as it appears in the list of servers on the **Servers** tab of the Configuration utility.
10. Open the Tableau Server Configuration utility and on the **Servers** tab, highlight the computer from which you are removing the process and click **Edit.**
11. Remove the processes you are moving: enter 0 for **Data Engine** and click **OK.** The File Store check box will be cleared automatically.
12. Click **OK** again to save your changes and close the utility.
13. **Start the primary server** so that the changes can take effect.

If you are performing this procedure as part of deleting a worker node from the Tableau Server configuration (as described in **Remove a Worker Node** below) stop Tableau Server again before proceeding.

**Remove a Worker Node**

To **delete a worker from your Tableau Server configuration:**

1. **Stop the server** on the primary Tableau Server.
2. On the primary server, open the configuration utility by selecting **Tableau Server <version> > Configure Tableau Server** on the Start menu.
3. In the configuration utility, select the **Servers** tab.
4. If the worker is hosting the data engine or the repository, move those processes onto another machine before continuing. See **Move the Data Engine and File Store**
Processes on the previous page for steps.

5. Next, highlight the worker and click **Delete**.
6. Click **OK**.
7. Start the server.

**High Availability**

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

**Quick Start: Configuring Failover & Highly Available Gateways**

Extracts and repository data can change rapidly and even regular backups may not help you fully recover from a system failure. Another vulnerability is having a single entry point, or gateway, for your Tableau Server cluster. To help with this, distributed Tableau Server deployments provide real-time content replication and failover support, as well as the ability to run multiple gateways.

**1 Install the Servers**

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

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

Stop the primary server and open its Configuration utility.

**3 Set Up Email Alerts**

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

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

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

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

2 Configure the Distributed System

1. On the Servers tab, click Add to add a worker server. Enter its IPv4 address or computer name. Enter 1 for each process. Select Repository, Gateway, and Search & Browse. Click OK:
2. Click **Add** to add a second worker server. Enter its IPv4 address or computer name. Enter 1 for every process except the **Data Engine** (set that to 0). Leave **Repository** cleared but select **Gateway**. Click **OK**.

3. Click **OK** to close the Configuration utility, then start Tableau Server on the primary server so your changes can take effect.

4. Stop the primary server and open the Configuration utility.

5. On the **Servers** tab, select the second worker and click **Edit**. Set **Data Engine** to 1 and select the **Repository** check box. Click **OK**, then **OK** again to close the Configuration utility. Start Tableau Server.

6. Still on the **Servers** tab, select **This Computer (Primary)** and click **Edit**. Set every process to 0, clear the **Repository** check box but keep **Gateway** selected. Click **OK**.

**4 Load Balance the Gateways**

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

In your load balancer, enter the IP address for each computer that’s running a gateway process (the primary and the two workers), and configure the load balancing method, such as Fastest or Round Robin.
**Quick Start: Creating a Backup Primary**

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

Before beginning, make sure you have configured your environment for failover and highly available gateways, using the *Quick Start: Configuring Failover & Highly Available Gateways* on page 709 as your guide. You should have two worker servers and a primary Tableau Server. To help ensure a smooth transition for your Tableau Server users, assign the same common name to both your current and backup primary servers.
Configuring Primary Failover

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

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

*computer1* is the current primary’s IPv4 address or computer name. *computer2* is the backup primary’s IPv4 address or computer name.

2 Copy the Primary’s Config to the Backup
Copy the primary’s tabsvc.yml file (located in ProgramData\Tableau\Tableau Server\config) to a temporary location on the backup primary. In the file, replace all occurrences of the IPv4 address or computer name for the primary with the IPv4 address or computer name for the backup primary.

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

```
tabadmin autostart off
```

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

After the Primary Fails

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

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

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

5 Start the Backup Primary
Run the following command:

```
tabadmin autostart on
```

Then start the server. Your backup primary is now your primary.
6 View Status

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

Understanding High Availability

If you’re configuring a Tableau Server system for high availability, the steps you perform are all designed to build in redundancy, thus reducing your potential downtime. The four areas that require redundancy are the data engine, repository, and gateway processes, and the primary Tableau Server, which runs the server’s licensing component. Because there must always be one active of the repository process, configuring the cluster is a multi-phased procedure that requires the primary Tableau Server to be stopped and restarted at certain points so that settings can take effect. For exact steps, see Configure for Failover and Multiple Gateways on page 720 and Use a Backup Primary on page 732. See Distributed Requirements on page 696 as well.

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

A Single Server System

After you install the primary Tableau Server, it is running at least one instance of all server processes. This is the most basic configuration of Tableau Server. It has no redundancy.
Here’s what the Process Status table on the Server Status page typically looks like for a single-server system:
To build in redundancy, you need to add additional servers to host copies of the repository and data engine/file store processes. In addition, to reduce the system’s vulnerability, you can run multiple gateways, and the primary should be isolated on its own node, ideally running as few of the server processes as possible. The fewest number of computers required to achieve this is three (see A Three-Node System below).

A Three-Node System

A three-node system helps you reduce the primary's vulnerability:
This configuration would look like the following Process Status table on the Server Status page.
In a three-node cluster, the Data Engine and Repository processes have been moved from the primary to a worker, and the primary is only running the Gateway and Search & Browse processes. In this configuration, if your active worker fails, the passive worker automatically becomes active. Exactly how to create this three-node cluster, including how to add the workers and remove the processes from the primary, is described in Configure for Failover and Multiple Gateways on page 720. (Licensing functionality is integral to the primary and cannot be removed, so it is not displayed on the Status page. Cluster Controller and Coordination Service are installed on all nodes as part of the "base install" and are not configurable. Coordination Service does not show on the Status page and Cluster Controller only displays if there are two or more nodes in the cluster.)

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

Add a Load Balancer

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

Add a Backup Primary

Adding a backup primary provides a safeguard for your system. The backup primary is an additional server added to the system to be ready if your primary fails. While it is not an active server, after you complete the first set of steps in Use a Backup Primary on page 732, it is
ready to be activated. While the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

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

The Process Status table for the configuration shown above looks the same as for a three-node system. If the primary fails and you perform the steps for the backup primary to take over, your system is back online using the new primary:
The licensing service only runs on the primary Tableau Server node. When a server process starts or restarts, the process checks with the licensing service to verify that the process is licensed. If the license is confirmed, the process is fully functional and able to respond to requests from other licensed Tableau Server processes and does not need to reconfirm the license until 72 hours have passed, or until the process restarts. If the process is not able to verify that it is licensed (if the primary node is unavailable, for example) it continues to check for a valid license but cannot function as a part of Tableau Server until it confirms the license.

**Configure for Failover and Multiple Gateways**

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

1. Install Tableau Server on your primary computer.
2. After Setup completes, check the Status page. All the processes should have a green “active” status:
3. **Stop the server** on the primary computer.

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

   ![Worker Configuration](image)

   **Worker Configuration**
   
   Enter the computer name or IPv4 address of the primary Tableau Server. If the server has multiple IPv4 addresses, enter all of them, separating each with a comma.

   **IP Address or Name:**
   
   TABLEAU_SERVER

   **Note:** The primary computer must have a static IP address assigned to it, even if you are using the primary’s computer name to identify it. For more information, see [Hostname Support in Tableau Server](#) on page 698.

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

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

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

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

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

   **Note**: If you install Tableau Server on a two-node cluster, a message displays to let you know that you are limited to one instance of the repository, and that high availability and failover are not available in a two-node configuration. You can add a third node but are not required to do so. In a two-node cluster, if one of the two nodes goes down, Tableau Server may not function correctly.

8. Click **Add** on the **Servers** tab to add another worker.

9. Enter the IPv4 address or computer name of the second worker, enter at least 1 for every process but the **Data Engine** (set that to 0). Select **Gateway**.
Click OK to close the Add Tableau Server dialog box and click OK to save the configuration and close the Configuration utility.

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

11. **Important:** Allow several minutes for the server’s synchronization processes to copy data. This can take anywhere from 5 minutes to 15 minutes (or even much longer) depending on the size of your installation and the number of extracts.

12. Open the Status page in Tableau Server and check on the components you added:
   - If you added a data engine/file store, wait until the new file store status no longer says "Syncing".
   - If you added a repository, wait until the new repository status says "Passive".

13. After you’ve confirmed that the synchronization is complete, **stop the server** on the primary.

14. Open the Configuration utility. On the **General** tab enter the Run As account password, then click the **Servers** tab, select **This Computer (Primary)**, and click **Edit**.

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

   With a core-based license, the gateway and search & browse processes consume no cores. Configuring the primary Tableau Server to run nothing but the gateway and search & browse is a useful strategy if, for example, you have a 16-core server license and two 8-core workers. You can run three nodes (the primary plus two workers) and only the worker nodes are consuming cores.
Click **OK**.

16. On the **Servers** tab, select the first worker, click **Edit**, and select the **Gateway** check box. Leave the other settings unchanged. Click **OK**.

17. Still on the **Servers** tab, select the second worker and click **Edit**.

18. Set **Data Engine** to 1 (**File Store** will be automatically selected) and select the **Repository** check box.
19. Click OK.

The Servers tab should now look similar to this:

20. You can also set up email alerts so that you’re notified of server failures or changes in status for your data engine and repository processes. To do this, click the Alerts and Subscriptions tab in the Configuration utility and follow the steps in To configure email alerts for system failures on page 620.

21. Click OK to close the Configuration utility.
22. If you are removing a data engine/file store (step 14 above), a message appears to let you know that the file store was not decommissioned, and asking if you want to decommission it. Click **Yes** to decommission the file store.

23. **Start the server** on the primary computer (it may take a few minutes for your changes to take effect). Your system is now configured to provide failover support for the repository process. It is also configured for multiple gateways. You can now **use a load balancer** to ensure the cluster's availability in the event of a gateway failure—and to distribute the cluster's workload.

The Status page should look similar to this:

![Server Status](image)

**Configure Tableau Server for High Availability with Coordination Service-Only Nodes**

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

**To run Tableau Server with Coordination Service-only nodes**

1. Install Tableau Server on the primary computer (primary node).
2. Install Tableau Server worker software on additional computers.
You need at least three nodes to run Coordination Service, plus the nodes on which you want to run Tableau Server as part of your distributed installation. In the example below, a total of six nodes are used.

3. On the primary node, run the Configuration utility, and add the nodes on which you installed the worker software.

4. In the Configuration utility, edit each server that will run Tableau Server, and specify the processes that should be installed. For more information on how to configure a distributed installation, see Install and Configure Worker Nodes on page 699.

5. In the Configuration utility, edit each server that will run only the Coordination Service process, and configure the node so it is not running any other Tableau Server processes. These nodes are considered "external" to the Tableau Server configuration and will only run the only Coordination Service.

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

6. Close the Configuration utility.

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

**Example**

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

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

   After Setup completes, check the Server Status page. All the processes should have a green “active” status.
2. **Stop the server** on the primary computer.

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

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

   ![Worker Configuration Screenshot](image)

   **Note**: The primary computer must have a static IP address assigned to it, even if you are using the primary’s computer name to identify it. For more information, see **Hostname Support in Tableau Server** on page 698.

4. Start the Tableau Server Configuration utility: **Start > All Programs > Tableau Server**
> **Configure Tableau Server.** On the **General** tab, enter the Run As account password.

5. On the **Servers** tab, click **Discover** to add the five worker nodes.

6. Select the first worker node, and then click **Edit**. Enter 1 for **Data Engine (File Store** will be automatically selected), and then select the **Repository** check box.

   Click **OK** to close the **Edit Tableau Server** dialog box.

7. Select the second worker node, and repeat step 6.

8. For each of the next three computers:

   a. Select the computer from the **Servers** list, click **Edit**, and then enter 0 for every process. Clear the options for **Repository**, **Gateway**, and **Search & Browse**. **Base Install** will be selected and you cannot change this. These nodes will run only Coordination Service.

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

   ![Edit Tableau Server dialog box](image)

   b. Click **OK** to close the **Edit Tableau Server** dialog box.

9. Click **OK** to close the **Edit Tableau Server** dialog box, and then click **OK** to save the
configuration and close the Configuration utility.

10. At a command prompt on the primary computer, remove Coordination Service from the primary node and the two worker nodes that are running Tableau Server:

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

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

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

### Add a Load Balancer

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

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

   **Note:** If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see Configure Kerberos on page 993.

### Guidelines

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

- **Tested load balancers:** Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

  If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.

- **Tableau Server URL:** When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the
primary Tableau Server.

- **Trusted host settings:** The computer running the load balancer must be identified to Tableau Server as a trusted host. See the procedure below for how to configure Tableau Server.

- **Proxy server configurations:** The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. For more information, see Configure a reverse proxy server on page 584.

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

  **Note:** You can use persistence with Kerberos enabled.

### Configure Tableau Server to Work with a Load Balancer

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

1. **Stop the server.**

2. In the Tableau Server bin directory, enter the following command, where `name` is the URL that will be used to reach Tableau Server through the load balancer:

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

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

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

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

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

   For example, if your load balancer is configured for SSL and listening on port 443, enter the following command:

   ```bash
tabadmin set gateway.public.port "443"
   ```

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

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

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

or

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

5. In the next command, you will provide any alternate names for the load balancer, such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases. These are the names a user might type in a browser. Separate each name with a comma:

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

For example:

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

6. Run the config command:

```
tabadmin config
```

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

**Use a Backup Primary**

Before you follow the procedures in the topics below, follow the steps in *Configure for Failover and Multiple Gateways* on page 720. After going through those steps, you have two worker servers that are providing failover support. Each server is also running a gateway, for which a load balancer can be configured. The primary Tableau Server is running a gateway process and licensing, which is not exposed or assignable as a process. Now that you have redundancy for the data engine, repository, and gateway, you need to build in redundancy for your primary Tableau Server. You do this by creating a backup of it. While the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

The licensing service only runs on the primary Tableau Server node. When a server process starts or restarts, the process checks with the licensing service to verify that the process is licensed. If the license is confirmed, the process is fully functional and able to respond to requests from other licensed Tableau Server processes and does not need to reconfirm the license until 72 hours have passed, or until the process restarts. If the process is not able to verify that it is licensed (if the primary node is unavailable, for example) it continues to check for a valid license but cannot function as a part of Tableau Server until it confirms the license. To see when the last licensing check occurred, look at the log files in the
ProgramData\Tableau\Tableau Server\data\tabsvc\logs\licensing folder.

Removing a Backup Primary

If you have a backup primary node that you no longer need, you can easily remove it from your Tableau Server installation. To remove a backup primary, all you need to do is use Windows Control Panel to uninstall Tableau Server.

Create a Backup Primary

Do the following to create a backup primary:

1. On the primary, open a command prompt as an administrator and navigate to the Tableau Server bin directory:
   
   C:\Program Files\Tableau\Tableau Server\10.0\bin

2. Stop the server:
   
   tabadmin stop

3. Issue the failoverprimary command.

   Enter the following command, using either the computer names for the current and backup primaries (recommended) or all the IPv4 addresses for the current and backup primaries. Separate multiple computer names or IPv4 addresses with a comma and no spaces:

   tabadmin failoverprimary --primary "primary1_name,primary2_name"

   or

   tabadmin failoverprimary --primary "primary1_IP,primary2_IP"

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

   tabadmin failoverprimary --primary "TABLEAU_SERVER,TABLEAU_SERVER2"

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

   tabadmin failoverprimary --primary "10.32.139.22,10.32.139.26"

   If the primary and backup primary have multiple IPv4 addresses, enter them all. For example:
tabadmin failoverprimary --primary
"10.32.139.22,10.32.139.23,10.32.139.26,10.32.139.27"

4. **Next**, copy the `tabsvc.yml` file from the primary node (located in `ProgramData\Tableau\Tableau Server\config`) and save it copy in a temporary location on your backup primary computer. You need to use this file if you switch to your backup primary (failover).

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

5. On your backup primary, edit the copy of the `tabsvc.yml` file and replace all occurrences of the IP address(es) or computer name for the primary with the IP address (es) or computer name for the backup primary (the computer you’re currently on). If the primary is only running the gateway, as described in this procedure, the only line you’ll need to edit is `worker.hosts`. 


- 734 -
6. Install Tableau Server on the backup primary. Use the same product key or keys that you used for license activation, and the same Run As account and configuration settings that you used on your primary.

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

7. After Setup completes, stop the server on the backup primary.

8. Still on your backup primary, enter the following command to disable the automatic
starting of the Tableau Server service:

```bash
tabadmin autostart off
```

9. Commit the configuration change:

```bash
tabadmin config
```

You’ve finished creating a backup primary. See Switch to Backup Primary for what to do if your current primary fails.

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

Switch to Backup Primary

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

1. On your backup primary, find the tabsvc.yml file you copied and edited in step 5 of Create a Backup Primary on page 733. Copy this from your temporary location to ProgramData\Tableau\Tableau Server\config and replace the existing tabsvc.yml file on the backup primary. You need to do this so the backup primary has the same configuration settings as the primary did.

2. If web data connectors were imported to the primary server, copy them to the following folder on the backup primary:

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

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

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

   ```
   C:\Program Files\Tableau\Tableau Server\10.0\bin
   ```

4. Enter the following command, using either the computer name of your backup primary (soon to be your new primary) or the IPv4 addresses of the backup primary (soon to be your new primary) and the primary (soon to be your backup primary). Separate multiple computer names or IPv4 addresses with a comma and no spaces.

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

   or
tabadmin failoverprimary --primary "primary2_IP,primary1_IP"

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

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

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

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

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

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

5. Enter the following command:

```
tabadmin autostart on
```

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

```
tabadmin config
```

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

![Server Status](image)

8. Your former primary is now your backup primary.

**What is a Site?**

You might be used to using the term site to mean "a collection of connected computers," or perhaps as the short form of "website." But in Tableau-speak, we use *site* to mean a collection of content (workbooks, data sources, users, etc) that's walled off from any other content on that
instance of Tableau Server. (Another way to say this is that Tableau Server supports multitenancy by allowing server administrators to create multiple sites on the server for different sets of users and content.)

Each site has its own URL and its own set of users, and each site has completely segregated content, projects, and data sources. You can set permissions per user or group on a project, workbook, view, or data source. All server content is published, accessed, managed, and controlled on a per-site basis.

**What is a site administrator?**

A site administrator is in charge of creating and maintaining the framework on Tableau Server that enables Tableau Desktop users in the organization to publish, share, manage, and connect to data sources and workbooks. Their duties can include creating and managing users and groups, creating *projects* to organize content on the site, assigning permissions to allow users to access the content they need, scheduling extract refreshes, and a few other tasks.

**Site administrators and server administrators**

In addition to a site administrator, there's also a *server administrator*. The server administrator sets up Tableau Server—they install and upgrade it, configure the services that run on Tableau Server, back it up, and perform other tasks that pertain to running Tableau Server as a whole. Server administrators also create sites as needed. (Site administrators don't have permissions to create sites.)

In some organizations, the same person might be both a server administrator and the site administrator for one or more sites. Even so, the tasks performed by a site administrator and a server administrator are distinct.

**About this guide**

This guide tells you, a site administrator, how to plan, create, and manage sites on an instance of Tableau Server. Note the following:

- We don't cover the duties of a server administrator. We have a separate guide that covers those tasks.
- We don't discuss how to publish content to the server. Users do this from Tableau Desktop. However, we do discuss how to set up users on the site and give them permissions to publish and view the content that they need. For information about how to publish to Tableau Server, see Publish Data Sources and Workbooks in the Tableau Desktop documentation.
Work with Sites

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

Authentication and sign-in credentials

All sites on a server use the same Run As User account and user authentication mode. You choose both of these settings when you install Tableau Server. See Configure General Server Options on page 608 for more information.

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

The Default site

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

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

Site administrator and server administrator site roles

There are two types of administrators in Tableau Server, server administrators and site administrators. For each site, server administrators can control whether site administrators can add and remove users for the sites they manage (select Site <name> > Settings).
If **Only server administrators** is selected, site administrators cannot add or remove site users. However, they can still manage groups, projects, workbooks, and data connections within their site. If **Server and site administrators** is selected (the default), site administrators can do all of the above, and add or remove users.

**Licensing and user limits**

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

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

**Add or Edit Sites**

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

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

To edit a site, select the site you want to modify, and then select **Edit Settings**. In a single-site deployment, click **Settings**, and then click the **General** tab.
2. Enter a **Site name** and **Site ID** for the site (if you are editing the Default site, you cannot change the **Site ID**):

![Sites page](image)

**Name**

<table>
<thead>
<tr>
<th>Site Name:</th>
<th>Enter Site Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site ID:</td>
<td>Enter Site ID</td>
</tr>
<tr>
<td>URL:</td>
<td><a href="http://10.32.139.28/#">http://10.32.139.28/#</a></td>
</tr>
</tbody>
</table>

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

3. Workbooks, extracts, and data sources all consume storage space on the server. For **Storage**, select either **Server Limit** or **GB**, and enter the number of GB you want as a limit.
If you set a server limit and the site exceeds it, publishers will be prevented from uploading new content until the site is under the limit again. Server administrators can track where the site is relative to its limit using the Max Storage and Storage Used columns on the Sites page.

<table>
<thead>
<tr>
<th>Name</th>
<th>Users</th>
<th>Site Admins</th>
<th>Max Users</th>
<th>Storage Used</th>
<th>Max Storage</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>10</td>
<td>1</td>
<td>Server limit</td>
<td>12.9 MB</td>
<td>Server limit</td>
<td>Active</td>
</tr>
<tr>
<td>MyFirstSite</td>
<td>0</td>
<td>0</td>
<td>Server limit</td>
<td>0 B</td>
<td>Server limit</td>
<td>Active</td>
</tr>
<tr>
<td>MySecondSite</td>
<td>0</td>
<td>0</td>
<td>Server limit</td>
<td>0 B</td>
<td>Server limit</td>
<td>Active</td>
</tr>
</tbody>
</table>

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

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

- **Server limit**: For a server with user-based licensing, the limit is the number of available server seat licenses. For a server with core-based licensing, there is no limit to the number of users that can be added. For more information, see View Server Licenses on page 1169 and Handle an Unlicensed Server on
page 1198.

- **<n> users**: Allows a site administrator to add users up to a limit you specify. See *Work with Sites* for information on licensing and user limits.

5. Leave **Allow users to use web authoring** selected or clear it to disable authoring for content in the site (not server-wide).

Disabling web authoring means that users cannot edit published workbooks from the server web environment. To update a workbook published to the server, a Tableau Desktop user must re-publish it. For more information, see *Disable Web Authoring* on page 884.

6. For **Subscriptions**, keep **Allow users to subscribe to workbooks and views** selected if you want site users to be able to subscribe to views. Keep **Allow content owners to subscribe other users to workbooks and views** to allow administrators, project leaders, and content owners to set up subscriptions for other users. These options are visible only if you have also configured subscription settings in the Configuration dialog box.

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

For **Email footer**, select **Custom footer** and enter the text you want to display above the Tableau Server URL in subscription footers.
The email footer will look similar to the following:

7. Select **Record workbook performance metrics** to permit your site users to collect metrics on how workbooks perform, such as how quickly they load

   In addition to having this check box selected for the site, to initiate recording, users must add a parameter to the workbook's URL. For more information, see **Create a Performance Recording** on page 1139.

8. Click **Create** or **Save**.

   **Note:** As a server administrator, when you add your first site to Tableau Server, the site menu becomes available and shows the name of the current site. When **All Sites** is selected in the top menu bar, the Users page displays the label **Server Users**, because it pertains to all users on the server. When a site is selected in the top menu bar, the Users page displays the label **Site Users**. As a server administrator, you can add users to the server, or to individual sites. For more information, see **Users** on page 781 and **What is a Site?** on page 737.

---

**Import or Export a Site**

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

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

**Before you export**

Before you export a site, note the following:

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

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

**Create user accounts on the target server.** The site import process assigns users to a target site. The users must already have user accounts on the target server. If you are exporting one site into another on the same Tableau Server, you will have all the user accounts you need. If you are exporting a site from Tableau Online or from a different Tableau server, you must create user accounts on the target server before you can perform the import.

**Check user authentication.** User authentication is a server-wide setting and all sites on a server must use the same setting. You can export from and import to servers that are using different user authentication methods, but you will need to modify the mapping files used for the import. This step is built into the import process and described in [Verify the site mappings on page 749](#). Because Tableau Online sites use a custom user authentication method, exporting from a Tableau Online site requires edits to the user-specific mapping files. This ensures a clean import, regardless of how the target server is configured.

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

![Schedules screenshot](#)

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

**Configure the target server to deliver subscriptions.** Subscriptions will be imported to the new site, but you must configure the target server to deliver the subscriptions, if it isn't already configured. For more information, see Configure Alerts and Subscriptions on page 619.

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

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

If your source site has workbooks that use published data sources, the target site name must match the source site name. The data connections for the workbooks will continue to refer to the source site name and can't be updated on the new site.

**Locate site IDs.** The commands you use to export or import a site require a site ID (also known as the content URL) as a parameter. A site ID uniquely identifies a site on Tableau Server. When you are signed in to a site, the site ID is displayed after the # character in the URL.

If the server is not running multiple sites, the web browser URL includes #, but not the word site or the site ID. If you see a URL like that in the following picture, you are using the built-in site, which is named Default.

If the server is running multiple sites, the web browser URL includes #site/ followed by the site ID for your site, as in the following example:
Tips for importing to a target with fewer users or schedules than the source site

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

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

Export a Site

You don't need to stop Tableau Server during the export or import process. Run the `tabadmin exportsite` command to export the site.

The site ID for the Tableau Server Default site is " " (double quotation marks, no space). If you are using Windows PowerShell to run the command, enclose the double quotes for the Default site within single quotes (' " ').

1. Open a command prompt as an administrator and navigate to the bin directory on Tableau Server. For example:
   
   C:\Program Files\Tableau\Tableau Server\10.0\bin

2. Type the following command:
   
   `tabadmin exportsite <site ID> --file <filename or path>`.

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

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

   For examples of other options you can use with the `exportsite` command, see `exportsite` on page 1271.

   During the export, Tableau Server locks the site.

Import a Site

If you don’t already have a target site for the import, create one. See Add or Edit Sites for steps.
Importing a site is a three-step process. First, run the `tabadmin importsite` command to generate the files that will be imported. Next, verify files that show how the site will be imported. Finally, run the `tabadmin importsite_verified` command to finish the import.

Before you begin, you will need the exported site file and the site ID for the target site.

The site ID for the Tableau Server Default site is " " (double quotation marks, no space). If you are using Windows PowerShell to run the command, enclose the double quotes for the Default site within single quotes ('""').

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

**Start the site import process**

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

   C:\Program Files\Tableau\Tableau Server\10.0\bin

2. Type the following command:

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

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

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

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

   For examples of other options you can use with the `importsite` command, see `importsite` on page 1275.

3. After you enter the command, the mapping files for you to verify are placed in `ProgramData\Tableau\Tableau Server\data\tabsvc\temp\import_<site ID>_ <datetime>\mappings`. Note this location for the next procedure.

**Verify the site mappings**

The mapping files that are generated after you initiate a site import with the `importsite` command show you how the site's resources will be assigned once the import is complete. Items that Tableau Server was unable to map, and which need editing, are marked in the CSV files with question marks (???). Before you can run the final `importsite_verified` command you must change the question marks so that they represent valid assignments on the target site.
**Note:** You can’t add or remove users as part of your changes. All user names for the users that you import must already exist on the target server.

If your source site has workbooks that use published data sources, the target site name must match the source site name. The data connections for the workbooks will continue to refer to the source site name and can’t be updated on the new site.

To verify a site’s mapping files:

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

   ![Command Prompt](image)

   - `C:\ProgramData\Tableau\Tableau Server\data\<source site name>\import_<source site name>_20140513155544\mappings`

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

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

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

<table>
<thead>
<tr>
<th>CSV file name</th>
<th>Column title</th>
<th>Can it be edited?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mappingsDomainMapperForGroups</td>
<td>source_name</td>
<td>No</td>
<td>A user group name on the source site.</td>
</tr>
<tr>
<td>mappingsDomainMapperForGroups</td>
<td>source_domain</td>
<td>No</td>
<td>The user authentication type on the source site:</td>
</tr>
</tbody>
</table>

- 750 -
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Either <em>local</em> (for Local Authentication) or a domain name (for Active Directory).</td>
</tr>
</tbody>
</table>
| target_domain_name | Yes*  
  The user authentication type on the source site: either *local* for Local Authentication, or a domain name (such as example.com or example.lan) for Active Directory.  
  *Do not edit the target_domain_name value for All Users. Keep its value of local, even if your target server is configured for Active Directory user authentication. The All Users group is a special default user group that must exist on every Tableau Server. |
| mappingsScheduleMapper | No  
  The names of custom and default extract or subscription schedules on the source site.  
  The type of schedule, either *Extract*, for extract refreshes, or *Subscription*, for subscription deliveries on the source site.  
  The names of custom schedules on the target site. You can edit this value. For example, if the schedule is named *Friday Update* on the source site. |
You can rename it **Friday Refresh** on the target site.

The type of schedule, either **Extract**, for extract refreshes, or **Subscription**, for subscription deliveries on the target site.

*In rare cases, there may be question marks (???) in this column. If there are, replace them with either **Extract** or **Subscription**, matching the entry you see under **source_scheduled_action_type**.

<table>
<thead>
<tr>
<th>target_scheduled_action_type</th>
<th>No*</th>
</tr>
</thead>
</table>
| The type of schedule, either **Extract**, for extract refreshes, or **Subscription**, for subscription deliveries on the target site.

<table>
<thead>
<tr>
<th>mappingsSiteMapper</th>
<th>source_url_namespace</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>The site ID of the source site.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>target_url_namespace</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>The site ID of the target site.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>mappingsSystemUserNameMapper</th>
<th>source_name</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>The username of a user on the source site.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>source_domain_name</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>The user authentication type on the source site: either <strong>local</strong>, for Local Authentication, a domain name (such as example.com or example.lan) for Active Directory, or <strong>external</strong> (for a Tableau Online site).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>target_name</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usernames for users who will be assigned to the target site upon import.</td>
<td></td>
</tr>
</tbody>
</table>
Confirm that all the usernames listed exist on the target server system and replace any question marks (???) with a valid username from the target server.

You can’t create usernames by adding rows to the CSV file. Similarly, you can’t remove usernames by deleting rows.

You can edit a username in the target_name column to be different from its source username as long as it already exists on the target server system using that different name. For example, a user can have a source_name value of jsmith@myco.com and a target_name value of johnsmith@example.com as long as the username johnsmith@example.com exists on the target server.

You can’t map a user on the source site to more than one username on the target site.

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

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

Complete the site import

1. Open a command prompt as an administrator and navigate to the bin directory on Tableau Server. For example:
   
   C:\Program Files\Tableau\Tableau Server\10.0\bin

2. Type the following command:
   
   tabadmin importsite_verified <site ID> --importjobdir <PATH>

   where <site ID> is the site ID of the target site and <PATH> is the directory that’s one level up from the mappings directory you used in Verify the site mappings on page 749. For example:

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

   For examples of other options you can use with the importsite_verified command, see importsite_verified on page 1277.

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

Delete Sites

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

Note: The Default site cannot be deleted.
1. On the site menu, click Manage All Sites, and then click Sites.

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

**Site Availability**

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

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

**Note:** If a site becomes locked and you cannot access the Sites page through the Server interface, use the tabadmin sitestate on page 1286 command to change the state to active.

**To activate or suspend a site**

1. In the site menu, click Manage All Sites, and then click Sites.
2. Select the site, and then select Actions > Activate or Suspend.
Projects

As an administrator, you can create *projects* to collect and organize related content. *Content* in Tableau Server refers to workbooks, views, and data sources, and the projects that contain them.

You access projects from the Content page in Tableau Server.

As an administrator, you can do the following for projects:

- Create projects.
- Rename projects.
- Change project owners.
- Set permissions for projects and default permissions for their content.
- Lock content permissions.

**Note:** Only administrators can create and own projects.

Project Leader

Users who have the **Project Leader** permission in a project can:

- Control who has access to project content by setting default permissions for project content at any time, even when content permissions are locked to the project.
- Lock content permissions to the project.
- Move workbooks between projects.
- Change the name of a project.
- Change owners of workbooks or data sources in the project.
- Run, add, or remove extract refreshes for workbooks and data sources in the project.

**Note:** Users with a site role of Interactor or Publisher get full project leader permissions. A project leader does not have to be the project owner.

**Default project**

Tableau creates every site with a Default project. The Default project serves as a template for new projects in that site. It defines the default settings and permissions that are applied to new projects and to the workbooks and data sources within those projects.

When you create a new project, the new project uses a copy of the Default project permissions.

**Default permissions**

As an administrator or project leader, you can set permissions for every project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project.

Each project can have its own set of default permissions.

For more information, see **Set Default Permissions for a Project, and its Workbooks and Data Sources** on page 861.
Only administrators and Project Leaders can edit the default permissions for a project and its workbooks and data sources.

For information on using projects to control permissions for content, see Create Project-Based Permissions on page 881.

**Project content permissions**

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.

When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

**Note:** If a workbook or data source with editable permissions is moved to a locked project, the default permissions in the locked project are applied to the moved content and its permissions will then be locked.

For more information, see Lock Content Permissions to the Project on page 869.
Add Projects

1. Click Content > Projects, and then click New Project.

   ![New Project dialog]

2. Enter a name and description for the project, and then click New Project.

   You also can include formatting and hyperlinks in the project description. Click Show formatting hints for syntax. For information on adding a image for the project, see Add a Project Image on page 761.

   To edit a project, click the Project to open it, click Details, and then click Edit Description.

Move Workbooks into Projects

All workbooks must be in a project. By default, workbooks are added to the Default project. After you create your own projects, you can move workbooks from one project to another. You can move workbooks into projects if you are an administrator, or if you have the site role of Publisher or Interactor and at least one of the following is true:

- You have been given the Move permission for the project.
- You have been given Project Leader permission for the project.
To move a workbook into a project

1. Click **Content > Workbooks**. In the Workbooks page, select one or more workbooks, and then select **Actions > Move**.

![Image of Workbooks page]

2. Select a different project for the workbook, and then click **Move**.

![Image of Move to Project dialog]

- 760 -
Because all workbooks must be part of a project, you can remove a workbook from a project by moving it to the Default project. Each workbook can only be contained in a single project.

**Add a Project Image**

Projects can have images that are displayed in thumbnail view in Tableau Server.

To set a project image:

1. In a site, click **Projects**, and then open a project.
2. Click **Details**, and then click **Edit Description**.
3. Add the URL for your image in the About field. Click **Show formatting hints** for syntax examples that show how you can format the description text.

Type the URL using this syntax:

```
!http://www.example.com/image.png!
```
4. Click Save.

Set Default Permissions for a Project, and its Workbooks and Data Sources

As an administrator or project leader, you can set a project’s permissions and the default permissions for its workbooks and data sources.

Each project can have its own set of default permissions. The permissions that you set are the default permissions for all content in the project, including content that is being published to the project from Tableau Desktop.
Note: New projects are always created with the default permissions set for the Default project.

For additional information on working with permissions, see Manage Permissions on page 834 and Projects on page 756. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Notes on default permissions in locked projects

You can choose to have the default permissions apply to all workbooks and data sources in a project, and ensure that no one can change those settings, by locking content permissions to the project. For more information, see Lock Content Permissions to the Project on page 869.

- Workbooks and data sources in a locked project always use the default permissions set for content in that project. Views in a locked project always use the workbook permissions. This applies to workbooks and data sources when they are being published from desktop.
- Administrators and users with the Project Leader permission can always edit default permissions, even when a project is locked.
- Users, including content owners, cannot edit individual workbook, view, and data source permissions when content is locked to the project.

To set default permissions in a project

1. In the Content page of a site, click a project, and then click Permissions in the project place page.

2. Click Add a user or group rule, select Group or User, and then select the group or user name from the list.
For an existing user or group, click the actions menu (..), and then click **Edit**.

3. Select a permission role template for **Project**, **Workbooks**, or **Data Sources**, and then click **Save**.
Or, to create a custom set of capabilities, click the **Project**, **Workbooks**, or **Data Sources** labels to expand the permissions view. Click capabilities to set them to **Allowed**, **Denied**, or **Unspecified**. Click **Save**.

This example shows how to set project permissions. The same general steps apply for workbooks and data sources.

**Note:** To change the settings after saving, click the actions menu ( ...), and then click **Edit**.

4. **View the user permissions, which are the effective permissions.**

   Click a group name or user name in the permission rules to see the resulting user permissions.
Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.

Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
5. Follow the same steps to configure additional permission rules for more users or groups.

**Set Permissions for a Project**

Every project includes permissions that can be set for the project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the project, and each project can have its own set of default permissions. For more information, see *Set Default Permissions for a Project, and its Workbooks and Data Sources* on page 861.

Administrators and users with the Project Leader permission can lock content permissions to a project. For more information, see *Quick Start: Lock Project Permissions, Lock Content Permissions to the Project* on page 869.

For more information on working with permissions, see *Manage Permissions* on page 834 and *Projects* on page 756.

**Note:** When you create a new project, it initially will have the same permissions as the *Default* project in the site, which are the default permissions for the project, and its workbooks and data sources.
The three capabilities you can set specifically for a project are: **View, Save, and Project Leader.**

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</td>
</tr>
<tr>
<td>Save</td>
<td>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project.  When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining all permissions. Subsequently, the original owner's access to the workbook is determined by that user's group permissions and any further permissions the new owner might set. This permission also determines the user's or group's ability to overwrite a workbook after editing it on the server. For related information, see <a href="https://www.tableau.com">Grant Web Edit, Save, and Download Permissions</a> on page 878.</td>
</tr>
<tr>
<td>Project Leader</td>
<td>Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions.</td>
</tr>
</tbody>
</table>
To set permissions for the project

1. On the Projects page, select a project, and then select Actions > Permissions.

2. Click Add a user or group rule, select Group or User, and then select the group or user name from the list.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click Save.
The available permission role templates for projects are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>Allows the user or group to view the workbooks and views in the project.</td>
</tr>
<tr>
<td>Publisher</td>
<td>Allows the user or group to publish workbooks and data sources to the server.</td>
</tr>
<tr>
<td>Project Leader</td>
<td>Allows the user or group to set permissions for all items in a project.</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to <strong>Denied</strong>.</td>
</tr>
<tr>
<td>Data Source Connect</td>
<td>Allows the user or group to connect to data sources in the project.</td>
</tr>
<tr>
<td>Data Source Editor</td>
<td>Allows the user or group to connect to, edit, download, delete, and set permissions for a data source in the projects. They can also publish data sources, and as long as they are the owner of a data source they publish, can update connection information and extract refresh schedules. This permission is relevant for views when the view they access connects to a data source.</td>
</tr>
</tbody>
</table>

4. To further customize the rule, click the actions menu (….) next to the permission rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
6. Follow the same steps to configure additional permission rules on the content for more users or groups.

Quick Start: Lock Content Permissions to a Project

As an administrator or project leader, you can lock content permissions in a project to prevent users from changing the permissions of any content in the project. When permissions are locked to the project, the default permissions are applied to all workbooks and data sources in a project and cannot be modified by users (including the content owners).

Note: Content owners always get full access to the content they've published, but cannot change permissions for their workbooks and data sources when the parent project permissions are locked.

For related information on setting permissions, see Manage permissions. For more information on setting default permissions and locking content permissions to the project, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861 and Lock Content Permissions to the Project on page 869. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.
1 Set Default Permissions for the Project

Because the content inside locked projects always uses the default permissions, first verify that your default permissions are set appropriately. In a site, click Content > Projects. Open a project, and then click Permissions. Add a user or group and select a permission role template for that content type, or click Edit, and then set capabilities to Allowed, Denied, or Unspecified.

Administrators and Project Leaders can edit default permissions at any time.

2 Lock Content Permissions to the Project

In a project's permissions, click the Managed by the owner button. The button label indicates whether content permissions are currently locked to the project or managed by the content owner. Select Locked to the project, and then click Save.
When permissions are locked to the project, all content in the project uses the default permissions. No users can change permissions for individual workbooks (including views) or data sources in the project.

3 View Locked Permissions

Open a project, select a workbook or data source in the project, and then click **Actions > Permissions**. When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

In this example, the workbook owner has full permissions for the workbook, but cannot change the workbook permissions while they are locked to the project.
Unlock Content Permissions for the Project

In a site, click **Content > Projects**. Select a project, and then click **Actions > Permissions**. Click the **Locked to the project** button. Select **Managed by the owner**, and then click **Save**.

When a project's content permissions are **Managed by the owner**, individual workbooks, views, and data sources in the project start with the default permissions and can be modified by users.

Notes on project permissions:

- Only administrators and project leaders can lock content permissions, and set and edit default permissions in a project.
- Administrators and project leaders can edit default permissions for the project, its workbooks, and its data sources at any time, at the project level.
- Individual workbook, view, and data source permissions cannot be edited by users (including content owners) when a project is locked.
- Workbooks and data sources in a locked project always use the default permissions. Views in a locked project always use the workbook permissions.

Lock Content Permissions to the Project

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.
When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

**Note:** Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For information on default permissions, see *Set Default Permissions for a Project, and its Workbooks and Data Sources* on page 861. For a best-practice walkthrough on how to implement permissions, see *Projects and Content Permissions*.

**Note:** Administrators and project leaders can set and edit default permissions for the project, and its workbooks and data sources when it is locked.

1. In the Content page of a site, open a project, and then click **Permissions** in the project place page.

2. Click the **Managed by the owner** button.
   
   The padlock icon on the button label indicates whether content permissions are currently locked to the project or managed by the content owner.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.
When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

4. To unlock content permissions for the projects, open the project permissions again. Click the **Locked to the project** button. In the **Content Permissions in Project** dialog box, select **Managed by the owner**, and then click **Save**.
The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

Create Project-Based Permissions

As an administrator, you can organize a collection of related workbooks and data sources in a project. You can then control access to that content by creating permission rules for groups of users who need similar access levels to publish or interact with that content.

Note: For this scenario, you set the permission rule for the All Users group for the project to None, which means that permissions are Unspecified for the All Users group.

Preparation

Before you begin creating projects and project-based permissions, document the projects and permission levels that you want users to have in each project.

This roadmap exercise helps you organize permissions to be most efficient to manage over time, and can help you identify any user or permission gaps in your solution. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Also read the following topics in the Tableau Server Help:

- Manage Permissions on page 834 and permissions-related topics
- Projects on page 756 and projects-related topics
- Grant Web Edit, Save, and Download Permissions on page 878

Step 1: Create projects and user groups

1. Sign in to Tableau Server with your administrator user name and password.
2. On the Projects page, click New Project.
3. Click Groups, and then click New Group.
   Create groups that correspond to each project and access level. For example, for a project that allows users only to access the views, you might use a name similar to Project1 Viewer. For a project that allows interaction with the views, Project1 Interactor.
4. Click Users, and then click Add Users. Select one or more users in the list, select Actions > Group Membership, and then select a group for the users. Click Save to confirm the group membership.
   Repeat this step to add users to other groups.

Step 2: Assign permissions at the project level

After you set up your projects and user groups, you can start assigning permissions. Repeat these steps for each project. Also see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.
1. On the Projects page, select a project, and then select **Actions > Permissions**.

2. For the **All Users** group permission rule, set the permission role template to **None**.
   
   Click the actions menu (..) next to **All Users**, and then click **Edit**. Select **None** for **Project, Workbooks**, and **Data Sources**, and then click **Delete**. This means that all capabilities will be set to **Unspecified**.

3. Click **Add a user or group rule**, select **Group**, and then select the group name in the list.
   
   To edit an existing rule, click the actions menu (..) next to the permission rule name, and then click **Edit**.

4. Select a permission role template for **Project, Workbooks**, and **Data Sources** to specify a predefined set of capabilities for the group or user.

5. To further change capabilities included in the rule, click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**.
   
   Click **Save** when you are done.
   
   Repeat steps 3-5 for each group or user requiring project permissions.

   **Note:** You can optionally lock content permissions to the project to enforce the default permissions for all content in the project. This overwrites any previous permissions assigned to workbooks and views in the project. For more information, see **Lock Content Permissions to the Project** on page 869.

**Step 3: Check project permissions**

- View the resulting user permissions.

   Click a group name or user name in the permission rules list to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

When you publish workbooks to the project, the permissions are updated accordingly.

For information on granting Save permissions to users, see **Grant Web Edit, Save, and Download Permissions** on page 878.

**Delete Projects**

Only administrators can delete projects. When you delete a project, all of the workbooks and views that are part of the project are also deleted from the server.

1. Click **Content > Projects**. In the Projects page, select a project, and then select **Actions > Delete**.
2. Click **Delete** in the confirmation dialog box.

   The **Default** project cannot be deleted.

**Users**

Everyone who needs access to Tableau Server must be added as a user.

**Guest user**

A Guest user is available in Tableau Server (core-based licenses only) in each site to allow users who don’t have an account on the server to see and interact with embedded views. When enabled, the user can load a webpage that contains an embedded visualization without signing in. For more information, see **Guest User** on page 792.

**Server users and site users**

Server administrators can add users to the server, and server administrators and site administrators (if enabled under Guest Access in **Settings** for the server) can add users to individual sites. For details on allowing site administrators to add users to sites, see step 4 in **Add or Edit Sites** on page 740.

In a single-site environment, server and site administrators can add users on the **Users** page.
In a multi-site environment, server and site administrators can add users in the **Site Users** page.

Server administrators can add users in the **Server Users** page. When you add a user to the server, you can assign the site membership and site roles per site for the user.
If you add a user without assigning site membership and role, the user is assigned the Unlicensed role and won’t use a server license (user-based licensing only). The user will exist in Server Users, but will not be a member of any site until you add that user to the site.

**Note:** Every user who is added to a site is also automatically added to the server. Site administrators can remove users from their sites, but they cannot delete users from the server. Server administrators can delete users from the server.

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

**About Users and Groups**

Every user who wants to publish content to a site must have an account on that site. In addition, users who want to interact with content (not just view) must be able to sign in to the site.
Manage Ownership

When you publish a data source or workbook on Tableau Server or when you create a project, you become its owner. Ownership can be changed. For example, if an employee who is the original owner leaves, the administrator can reassign ownership to another user. After you change ownership, the original owner has no special connection to the item, and their ability to access it is determined by their Tableau Server permissions.

**Note:** You cannot delete a Tableau Server user if the user owns any items. When you attempt to delete the user, their site role is set to Unlicensed. You must first change the ownership of the items and then delete the user. For more information, see Deleting a User from Tableau Server.

If you change the ownership of a workbook or data source that has embedded credentials, the embedded credentials will be deleted. You will need to download the workbook or data source, update the embedded credentials for the new owner, and then re-upload the workbook or data source.

Your ability to change or be given ownership depends on your permissions and your relationship to the item, as described in the following table.

<table>
<thead>
<tr>
<th>Item type</th>
<th>Who can change ownership</th>
<th>Who can be given ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Projects</strong></td>
<td>Server administrator</td>
<td>Server administrator</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td>Site administrator</td>
</tr>
<tr>
<td><strong>Workbooks and Data Sources</strong></td>
<td>Server administrator</td>
<td>Server administrator</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td>Site administrator</td>
</tr>
<tr>
<td></td>
<td>Owner of the item</td>
<td>Member of the site that contains the item (Guest user excluded).</td>
</tr>
<tr>
<td></td>
<td>Project leader for the project that contains the item</td>
<td></td>
</tr>
</tbody>
</table>

Change a Workbook Owner

By default, the publisher of a workbook is its owner. Administrators, project leaders, and the current owner of the workbook can change ownership. The new owner must be a server administrator or a site administrator, or be any user other than Guest on the same site as the workbook.

**To change the owner for a workbook**
1. On the Content page for a site, select **Workbooks**.

2. Select one or more workbooks, and then select **Actions > Change Owner**.

3. Type the name of a user or select a user from the list.

4. Click **Change Owner**.

**Change a Data Source Owner**

By default, the publisher of a data source is its owner. Administrators, project leaders, and the current data source owner can change ownership. The new owner must be a server or site administrator, or be any user other than Guest on the same site as the data source.

**To change the owner for a data source**
1. On the Content page for a site, select **Data Sources**.
2. Select one or more data sources, and then select **Actions > Change Owner**.

3. Type the name of a user or select a user from the list.

4. Click **Change Owner**.

**Change a Project Owner**

By default, the creator of a project is its owner. Administrators can change project ownership. The new owner must be a server administrator or an administrator for the project’s site.

**To change the owner for a project**

1. On the Content page for a site, select **Projects**.
2. Select one or more projects, and then select **Actions > Change Owner**.
3. Type the name of a user or select a user from the list.
4. Click **Change Owner**.

**Site Roles for Users**

Every user added to Tableau Server must have an associated site role. The site role is assigned by the administrator. The site role determines the levels of permissions allowed for a user, including whether a user can publish, interact with, or only view content published to the server. Administrators are also defined based on the site role.

**Note:** Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.

Users are accounts on the server that can be associated with one or more sites, and with groups in those sites. Any user that is added to Tableau Server or to a site becomes member of the All Users group. The All Users group is present in every site and cannot be deleted.

**Who can publish content**

Users with the following site roles can publish to Tableau Server:

- Server Administrator
- Site Administrator
- Publisher
- Viewer (can publish)
- Unlicensed (can publish)
Note: A system change is required on computers that Unlicensed (can publish) users will use to publish. For more information, see the corresponding quick fix article.

Users with a site role of Interactor, Viewer, and Unlicensed cannot publish content to the server.

Site roles and permissions

Effective user permissions for a resource are determined by:

- The maximum capabilities allowed for a user's site role. The site role acts as the "ceiling" for what permissions are allowed.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

When you select a site role for a user, help is available to remind you of the general level of permissions for that site role.

![Table showing site roles and permissions](image)

When you set permissions for a content item, the User Permissions section in the Permissions window indicates when a permission capability is not allowed for that site role.

The User Permissions area of the Permissions window shows the effective permissions for each user. These are the actual permissions for each user, after the user's site role and permission rule has been evaluated.

For details, see Permission Rules and User Permissions on page 836.
General capabilities

- **Server Administrator**: The server administrator can access all server features and settings on the server and all sites. Server administrators can create sites, add users of any site role type, control whether site administrators can add users, create additional server administrators, and they can administer the server itself. This includes handling maintenance, settings, schedules, and the search index.

  Server administrators can perform operations on all content anywhere on the server, regardless of what permissions have been assigned to the content. Server administrators can also manage other users on the server.

- **Site Administrator**: Site administrators can manage groups, projects, workbooks, and data connections. By default, site administrators can also add users and assign site roles and site membership. This setting can be enabled or disabled by the server administrator (see step 4 in *Add or Edit Sites* on page 740).

  Site administrators have unrestricted access to content on a specific site. A user can be specified as a site administrator on multiple sites.

- **Publisher**: Publishers can sign in, browse the server, and interact with the published views. They also can connect to Tableau Server from Tableau Desktop in order to publish and download workbooks and data sources.

  Publishers can publish (upload) workbooks and data sources to the server. Publishers aren't allowed to manage other users.

- **Interactor**: Interactors can sign in, browse the server, and interact with the published views. It's important to note that specific views, workbooks, and projects may have been published with permissions that restrict a user's capabilities. Permission settings can be edited by the workbook author or an administrator.

  Interactors can view workbooks and can interact with views. They are not allowed to publish to the server.

- **Viewer**: Viewers can sign in and see published views on the server but cannot interact with the views. Users with this site role can only be given permission to view, add comments, and view comments. They cannot interact with filters in the view or sort data in a view.

- **Unlicensed**: Unlicensed users cannot sign in to the server. When you import server users from a CSV file, all are assigned a site role of Unlicensed.

  If an insufficient number of licenses are available when an administrator creates a user (though CSV import of a site user, or import from Active Directory, or when a local user is created) the user will be assigned the Unlicensed site role.

  Attempting to remove a user who owns content from a site will demote the user to Unlicensed. The user will still own the content.

- **Viewer (can publish)**: The user can connect to Tableau Server from Tableau Desktop
to publish and download workbooks and data sources, but cannot interact with content on the server.

- **Unlicensed (can publish).** The user can connect to Tableau Server Tableau Desktop to publish workbooks to the server, but cannot sign in to Tableau Server directly.

**Site roles and Active Directory import and synchronization**

When you import Active Directory users to a site, either as a single user or as member of a group, you can specify a site role for the user. If a user is not yet a member of any site on the server, the user is added to the site with the assigned role. When you synchronize Active Directory groups, the site role is applied through the **Minimum Site Role** setting on the **Groups - Details** page.

If a user already exists in a Tableau Server site, the site role assigned during the import or sync process will be applied if it gives the user more access in a site. Importing or synchronizing users and groups will promote a user’s site role, but not demote a user’s site role.

If a user already has the ability to publish, that ability will always be maintained. For example, if a user with the current site role of **Unlicensed (can publish)** is imported with the new site role of **Interactor**, that user’s site role will be promoted to **Publisher** on import.

To guarantee a user maintains a site role with equal or greater capabilities in server after an import, the following matrix shows the rules applied for site roles on import. **Bold** indicates that a site role was promoted to preserve the user’s ability to publish.

**Note:** The Import Site Role row headers indicate the site role specified for import. The Current Site Role column headers represent the current user site role. The table values represent the resulting site role. A **bold** site role in the table indicates a site role promotion that preserves the ability to publish.

<table>
<thead>
<tr>
<th>Import Site Role</th>
<th>Site Administrator</th>
<th>Publisher</th>
<th>Interactor</th>
<th>Viewer</th>
<th>Viewer (can publish)</th>
<th>Unlicensed</th>
<th>Unlicensed (can publish)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Site Administrator</td>
</tr>
<tr>
<td>Publisher</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Publisher</td>
</tr>
<tr>
<td>Inter-</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Inter-</td>
<td>Inter-</td>
<td>Pub-</td>
<td>Inter-</td>
<td>Pub-</td>
</tr>
<tr>
<td>Import Site Role</td>
<td>Current Site Role</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Interactor</td>
<td>Viewer</td>
<td>Viewer (can publish)</td>
<td>Unlicensed</td>
<td>Unlicensed (can publish)</td>
<td></td>
</tr>
<tr>
<td>actor</td>
<td></td>
<td>actor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viewer (can publish)</td>
<td>Site Administraotor</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>Viewer</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Interactor</td>
<td>Viewer</td>
<td>Viewer (can publish)</td>
<td>Viewer</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>Unlicensed (can publish)</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Publisher</td>
<td>Viewer (can publish)</td>
<td>Viewer (can publish)</td>
<td>Unlicensed (can publish)</td>
<td>Unlicensed (can publish)</td>
</tr>
<tr>
<td>Unlicensed</td>
<td>Site Administrator</td>
<td>Publisher</td>
<td>Interactor</td>
<td>Viewer</td>
<td>Viewer (can publish)</td>
<td>Unlicensed</td>
<td>Unlicensed (can publish)</td>
</tr>
</tbody>
</table>

**Guest User**

A Guest user is available in Tableau Server to allow users who don't have an account on the server to see and interact with an embedded view. When enabled, the user can load a webpage that contains an embedded visualization without signing in.

**Note:** The Guest user option is available only with a core-based license.

When you embed a Tableau Server view into an internal website page, every person who views that page will need a Tableau Server account (they'll be asked for a user name and password) unless you have purchased a core-based (hardware) license. In that case you can have as many accounts as you want, as well as the ability to enable Guest user access, which does not require log in or authentication.

Guest is a special account that is used only to allow users to see views. The Guest user cannot browse the Tableau Server interface and won't see server interface elements in the view, such as user name, account settings, comments, and so on.
Note: Enabling the Guest user for a site can increase the number of potential simultaneous viewers beyond the user list you might be expecting. The administrative view Status > Traffic to Views can help you gauge the activity.

A Guest user can have the following permissions

**Projects, Workbooks, and Views**: View, Export Image, Summary Data, View Comments, Filter, Full Data, Web Edit, Download (to save a local copy)

**Data Sources**: View and Download

When a Guest user is included in a group that has a permission rule set on a content item, Guest user permissions do not affect the permission-levels of other users in that group.

**To enable Guest access**

1. **Single-site**: Click Settings > General.
   - **Multisite**: In the site menu, click Manage All Sites and then click Settings > General.
2. For Guest Access, select **Enable Guest account** to allow people who are not signed into a Tableau Server account to see views with Guest access permissions.
3. Click **Save**.

**The Guest user is unique in the following ways**:

- The Guest user represents all unauthenticated users accessing content on the server.
- Tableau Server must use a core license for Guest to be available.
- Server administrators can enabled/disable Guest across the server; it is not controllable per site.
- The Guest user cannot be edited and can never own content.
- The Guest user can be made a member of one or more groups in a site.
- Only the server administrator can enable or disable Guest access (in **All Sites > Settings > General**).
- The Guest user, when enabled, is a member of the All Users group.
- The Guest user cannot be deleted; it must be disabled by the server administrator in **All Sites > Settings > General**.
- If the Guest user needs to be able to access a workbook that uses an extract data source, make sure Guest has the View permission for the data source. The Guest user is not allowed to connect to published data sources, unless the publisher embedded their credentials when publishing the content.
- The Guest user is not allowed to save customized views.
- The "Guest" user is not a user name. It cannot be used to log in, to request trusted tickets, or in a user filter.

**Add Users to the Server**

In a single-site environment, server administrators can add users on the **Users** page.

![Users page](image)

After you add a site to Tableau Server, it becomes a multi-site server with a **Server Users** page (all server users from every site appear here) and a **Site Users** page. Only server administrators can access the **Server Users** page, and both site administrators and server administrators can access the **Site Users** page.

![Site Users page](image)

The **Server Users** page is the only place where you can assign users to multiple sites, delete users from the server, and if the server is using local authentication, reset user passwords.
The following procedure describes how to add users to the server. There are two approaches you can take: One at a time (described below) or in batches using the Import command, which relies on a CSV file (described in Import Users on page 804 and CSV Import File Guidelines on page 810).

To add a user to the server

1. In the site menu, click Manage All Sites, click Users, and then click Add Users.

2. If you are using local authentication, click New User. If you are using Active Directory, click Active Directory User.

   Enter a user name.

   - **Local authentication**: If the server is using local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, jsmith@example.com instead of jsmith).

   - **Active Directory**: If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server domain will be assumed.
Before adding users, be sure to review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

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

3. If the server is using local authentication, provide the following:

- **Display Name**—Type a display name for the user (e.g., *John Smith*).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.
- **Selected users are Server Administrators**: Specify whether the user should be a server administrator.
- **Name (Site Membership) / Site Role**: If the user is not a server administrator, you can assign a user to zero or more sites, along with a site role for each site. You do not have to choose site membership and site role at this time. If you don't specify site membership and site role for a new server user, the user will be added as a Server User only, with a site role of Unlicensed. For details on site roles, see
4. Click Create.

Add Users to a Site

Administrators can add users to sites in the following ways:

- By adding a local user account or a user account from Active Directory, as described in this topic. You can also add users by importing an Active Directory group. For details, see Create a Group via Active Directory on page 822.

- By importing a CSV file that contains user information. For details, see Import Users on page 804 and CSV Import File Guidelines on page 810.

In a single-site environment, administrators can add users to a site on the Users page. In a multi-site environment, you will use the Site Users page. Server administrators must give site administrators the ability to add users to sites. This setting can be enabled or disabled by the server administrator (see step 4 in Add or Edit Sites on page 740).
Note: Users can be added to sites, or to the server. To add users to the server, see Add Users to the Server on page 794. The options available for adding users depends on the authentication method that you select when you first configure Tableau Server. If you are using local authentication, you cannot add Active Directory users. If you are using Active Directory, you cannot add local users.

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

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

Note: This screenshot is from a single-site environment. In a multi-site environment, this would be the Site Users page.
To add local users to a site

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

![Add Users to this Site](image)

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

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

![New User](image)

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

3. Select a site role. For details on site roles, see Site Roles for Users on page 788.

4. Click Add User.

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

To add Active Directory users to a site

Before adding users to a site, be sure to review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

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

2. Enter one or more user names (separated by semicolons). If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server’s domain will be assumed.
Note: Do not enter the user's full name in this field; it can cause errors during the importing process.

3. Select a site role. For details on site roles, see Site Roles for Users on page 788.

4. Click Import Users.

Assign Site Membership

Server administrators and site administrators with the ability to add site users can change a user's site role. For details on site roles, see Site Roles for Users on page 788 and Change Site Roles on page 820.

Only server administrators can change the site membership of users.

1. In the site menu, click Manage All Sites, and then click Users.

2. Select one or more users, and then select Actions > Site Membership.
3. Select one or more sites, and a role for each site, and then click **Save**.
Add Users to a Group

One way to simplify user management is to assign users to groups. For example, you can assign permissions to a group to apply them to all users in the group.

To add a user to a group, the group must already exist. For information, see Groups on page 821.

Add users to a group (Users page)

1. In a site, click Users.
2. Select the users you want to add to a group, and then click Actions > Group Membership.
3. Select the groups and then click Save.

Add users to a group (Groups page)

1. In a site, click Groups, and then click the name of the group.
2. In the group’s page, click Add Users.
3. Select the users to be added, and then click **Add Users**.

**Import Users**

To automate the process of adding users to a site, you can create a CSV file that contains user information, and then import the file. You can import users to a site, or, to the server (if you are a server administrator).

**Note:** This topic contains the steps for importing, assuming that you have already created the CSV file. If you have not created the file yet, see [CSV Import File Guidelines](#) on page 810 for a list of file format requirements and import options.
Add users from a CSV file to a site

1. In a site, click **Users**, click **Add Users**.

![Add Users to this Site](image)

2. Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

![Import users from file](image)

The results of the import are displayed.
For a single-site server, the site roles assigned to the users during the import process will be imported with the users. If a user already exists in the Tableau Server site, the site role assigned during the import process will be applied only if it gives the user more access to the server. Importing users and groups will promote a user’s site role, but not demote a user’s site role.

3. Click **Done**.

**Add users from a CSV file to a server**

1. **Single-site**: Click **Users**, and then click **Add Users**.
   
   **Multisite**: In the site menu, click **Manage All Sites**, click **Users**, and then click **Add Users**.
2. Click **Import From File**, click **Browse** and navigate to the file, and then click **Import Users**.

The results of the import are displayed.

For a multi-site server, when you import users in the Server Users page, you are creating server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a server user can have is either Unlicensed or Server Administrator. When you assign site membership to a server user, you can specify the site role for that user per site. For details, see **Assign Site Membership** on page 801. If you import the users in the Site Users page, the users will be assigned the site roles you specify in the CSV file, for that site.

3. Click **Done**.
Multi-site environments

If the server is running multiple sites and you are a server administrator, you can import a CSV file from two different locations. Where existing user accounts are concerned, each location has different capabilities.

- The Server Users page appears in a multi-site environment. Only server administrators can access this page.

You can import the CSV file from here if you want to update existing user accounts in addition to adding new ones. For example, if you import a file that has a new password for each existing user, their passwords will be reset.

- The Site Users page.

Server administrators can add new user accounts with CSV imports and, if existing users are part of the import, the Password and Display Name fields must either match or be left blank. If new passwords or full names are used, the import will fail.

Single-site environments

Server and site administrators on a single-site server perform CSV user imports from the Users page in a site.
Multi-site versus single-site import

Users can belong to more than one site on the same server, but they must use the same credentials for each site. This becomes important when you’re adding users to a site and those users might already be members of a different site. If you try to import a user who already exists, and if the user’s credentials in the CSV file don’t match the existing credentials, the import fails for that user.

Note: The issue of credentials mismatch during import doesn't apply if the server is configured to use Active Directory for authentication. In that case, the CSV file should never contain a password, because user passwords are managed by Active Directory.

If you’re importing users into a site and you think that the users might already exist on the server, you can try leaving the Password column in the CSV file blank. When you import the users, if a user who is defined in the CSV already exists in another site, the user is added to the site where you’re importing. However, if the user doesn’t already exist on the server, the user is created, and the CSV import window alerts you that the new user doesn't have a password. You can then use the server environment to assign a password to any user who doesn't have one.

Multi-site

For a multi-site server, when you import users in the Server Users page, you are creating server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a server user can have is either Unlicensed or Server Administrator.

Single site

For a single-site server, the site roles assigned to the users during the import process will be imported with the user. If a user you are importing already exists in Tableau Server, the site role assigned during the import process will be applied only if it gives the user more access to the server. Importing users and groups will promote a user's site role, but not demote a user's site role.
CSV Import File Guidelines

You can automate adding users by creating a comma-separated values (CSV) file with user information and then importing the file. You can include attributes in the CSV file, such as site role and the ability to publish, to apply to the users at the same time you import them.

To import users, you can use the server administration pages or the tabcmd utility. For details, see Import Users on page 804 or createsiteusers filename.csv on page 1322.

Note: If you use the tabcmd utility to import users, you can pass options on the command line that can specify default values for the users' site roles. For more information, see the createsiteusers filename.csv on page 1322 documentation.

You can import users into a site or into the server. If you import users into a site, site roles are applied to the user. If you specify site roles, but importing users would exceed your license limits, users are imported as Unlicensed. If you import users into the server (not into a specific site), the user isn't assigned to a site, and site roles in the CSV file like Publisher and Interactor are treated as Unlicensed.

CSV File Format Requirements

When you create the CSV file for importing users, make sure that the file meets the following formatting requirements:

- The file does not include column headings. Tableau Server assumes that every line in the file represents a user.
- The file is in UTF-8 format, and includes the byte-order mark (BOM).
- Character encodings such as BIG-5 have been converted to UTF-8. You can do this by opening the file in a text editor and using the Save As command.
- If a name includes the "@" character other than as a domain separator, you need to refer to the symbol using the hex format: \0x40

  For example, user@fremont@myco.com should be user\0x40fremont@myco.com

Required Columns in the CSV File

The following values are required for each user:

- User name
- Password: If Tableau Server is configured to use Active Directory authentication, there must be a Password column, but the column itself should be empty. If the server is using local authentication, you must provide passwords for new users.
Additional Import File Options

The CSV file can contain the following fields, in the order shown here:

- **User name.** The user name. If the server is configured to use Active Directory, this value must match a user defined in Active Directory. If the user name is not unique across domains, you must include the domain as part of the user name (for example, example\Adam or adam@example). This is the only required field.

- **Password.** A password for the user. If the server is configured to use Active Directory, this value is not used.

- **Display name.** The display name is part of the information that's used to identify a user on the server. If the user's display name is already in use, Tableau Server updates the existing user information with the settings in the CSV file. If the server is configured using Active Directory, this value is not used.

- **License level** (Interactor, Viewer, or Unlicensed). This setting determines the role for a non-administrator user. If you are using the server administration pages to import users, the license level is set only if you are importing into an individual site. If you are using the server administration pages to import users while managing the server (not a specific site), and if the user is not set to be an administrator, the site role is set to Unlicensed. (You can change the site role later.)

**Note:** In Tableau Server 9.0, license levels have been replaced with site roles. If you create a user using the server UI, you select a site role like Site Administrator, Publisher, Interactor, and View (can publish). For information about site roles, see Site Roles for Users on page 788. For more information about how the license levels and other values in the CSV file are converted to site roles, see Settings and Site Roles later in this topic.

- **Administrator level** (System, Site, or None). This setting determines whether the user is imported as an administrator. If you are using the site administration pages, you can set the administrator role to System only if you are importing while managing the server. If you are using the server administration pages to import users while managing a site, and if the administrator role for a user in the CSV file is set to System, Tableau Server imports the user as a site administrator.

- **Publisher permissions** (yes/true/1 or no/false/0). This setting determines whether the user has publisher permissions. If you are using the site administration pages, the publisher setting is used only if you are importing into an individual site. If you are importing users while managing a server, this value isn't used.

- **Email address.** The email address is part of the information that's used to identify a user on the server. If the email address is already in use, Tableau Server updates the existing user information with the settings in the CSV file.
The order of the columns is significant. The first column is treated as the user name, the second as the password, the third as display name, etc., regardless of the content in the columns.

**Settings and Site Roles**

The license level, administrator, and publisher settings for a user are used during the import process to set a user’s site role. The following table shows how the settings are converted to site roles.

<table>
<thead>
<tr>
<th>CSV settings</th>
<th>Site role</th>
</tr>
</thead>
<tbody>
<tr>
<td>License level=(any)</td>
<td>Site administrator. This setting is valid only if you are importing users while managing a specific site. If you set a user to be a site administrator, the other values are ignored.</td>
</tr>
<tr>
<td>Administrator=System</td>
<td>Site administrator. This setting is valid only if you are importing users while managing a specific site. If you set a user to be a site administrator, the other values are ignored.</td>
</tr>
<tr>
<td>Publisher=(any)</td>
<td>Publisher</td>
</tr>
<tr>
<td>License level=Interator</td>
<td>Interactor</td>
</tr>
<tr>
<td>Administrator=None</td>
<td>Interactor</td>
</tr>
<tr>
<td>Publisher=true</td>
<td>Interactor</td>
</tr>
<tr>
<td>License level=Viewer</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>Administrator=None</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>Publisher=true</td>
<td>Viewer (can publish)</td>
</tr>
<tr>
<td>License level=Unlicensed</td>
<td>Unlicensed (can publish)</td>
</tr>
<tr>
<td>Administrator=Unlicense</td>
<td>Unlicensed (can publish)</td>
</tr>
<tr>
<td>CSV settings</td>
<td>Site role</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>censed</td>
<td></td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=true</td>
<td></td>
</tr>
<tr>
<td>License level=Unlic-</td>
<td>Unlicensed</td>
</tr>
<tr>
<td>ensed</td>
<td></td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=false</td>
<td></td>
</tr>
</tbody>
</table>

**Notes**

- If you are importing users while managing the server, you can create users with only two site roles: system (server) administrator and Unlicensed. All other settings are site specific. In that case, if the administrator level for a user in the CSV file is not `System`, the user's site role is set to Unlicensed.

- If you have a user-based server installation, and if adding users would exceed the number of users allowed by your license, the users are added as unlicensed users.

**Example**

The following example shows a CSV file that contains information for several users.

```plaintext
henryw,passw0rd,Henry Wilson,Interactor,None,yes,henryw@example.com
freds,pa$$word,Fred Suzuki,Viewer,None,no,freds@example.com
alanw,p@ssword,Alan Wang,Interactor,Site,yes,alanw@example.com
michellek,mypassword,Michelle Kim,Interactor,System,yes,michellek@example.com
```

If you import this file while managing a site, four users are added to that site. The Administrator mode for user Michelle is set to `System`. However, because you are importing the users into a site, Tableau Server sets user Michelle to be a site administrator, not a system administrator. Three of the users are allowed to publish.

If you import this file while managing the server, four users are added to the server, but they are not added to any site. The site roles in the CSV file (Interactor and Viewer) must be associated with site users, so the site role for the users who are not administrators is set to Unlicensed.
View, Edit, and Delete Users

View and edit site users

Sign in to a site as an administrator, and then click **Users**. In this page you can set group membership, set site role, or remove the user from the site.

Click a user name to see the content they own.

Click **Settings** in a user page to view their account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.
If Tableau Server is running multiple sites, **Server Users** lists all users on the server system, and **Site Users** displays all users for the current site.

If the server is configured to use the internal user management system (Local Authentication), you can edit the **Display Name**, **Email**, and **Password** for users after they have been added. If you are making many changes, you may find it easier to import the changes from a CSV file. For details, see [Import Users](#) on page 804 and [CSV Import File Guidelines](#) on page 810.

**For multi-site servers**: Site administrators can edit an existing user’s account as long as the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings. For example, if User Joe is a member of Site A and Site B and the site administrator is only an administrator of Site B, the site administrator cannot edit Joe’s Full Name or reset his password.

**View and edit server users**

Sign into Tableau Server as a server administrator. On the site menu, click **Manage All Sites**, and then click **Users**. In this page you can set site membership or delete the user from the server.
Click a user name to view account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.

**Search for users**

To search for a specific user, in the **Search** box on the left, type all or part of the user's name, and then press **Enter**.

The search operation checks the display name and user name attributes.

You can use the asterisk (*) character as a search wildcard. For example, searching for **John** will return all user names that start with **John**.
Remove users from a site

You can remove a user from a site only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but not removed.

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

1. In a site, click **Users**. Select one or more users to delete, and then select **Actions > Remove**.

2. Click **Remove** in the confirmation dialog.
Delete users from the server

You can delete a user from Tableau Server only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to delete a user who owns content, the user site role will be set to Unlicensed, but the user will not be deleted.

If a user is a member of multiple sites, and owns content in one or more of those sites, the user will be removed from the sites in which they don’t own content. The user will remain a member in sites where they do own content, but demoted to the Unlicensed site role.

1. In the site menu, click Manage All Sites, and then click Users. In a single-site environment, click Users.

   Select one or more users to delete, and then click Actions > Delete.

2. Click Delete in the confirmation dialog box.

Change passwords for users of a single site

To change the password for a user with membership to a single site, sign in to Tableau Server as a site administrator or a server administrator.

1. Ensure that the correct site is selected in the menu.
2. Click Users.
3. Click the display name of a user.
4. Click Settings.
5. Click the **Change Password** link, edit the password, and then click **Save Password**.

![Change Password link](image1)

**Change passwords for users of multiple sites**

To change the password of a user with membership to multiple sites, sign in to Tableau Server as a server administrator.

1. In the site menu, click **Manage All Sites**.
2. Click **Users**.
3. Click the display name of a user.
4. Click the **Change Password** link, edit the password, and then click **Save Password**.

![Change Password link](image2)
**Change Site Roles**

Server administrators and site administrators with the ability to add site users can change the site role of a user at any time. For details on site roles, see [Site Roles for Users on page 788](#).

Only server administrators can change the site membership of users. For details, see [Assign Site Membership on page 801](#).

1. In a site, click **Users**.
2. Select one or more users, and then select **Actions > Site Role**.

3. Select a site role, and then click **Change Site Role**.

![Change Site Role](image-url)

Choose a site role for user “Andrew Smith”.

- Publisher
- Server Administrator
- Site Administrator
  - Publisher
  - Interactor
  - Viewer
  - Unlicensed
  - Viewer (can publish)
  - Unlicensed (can publish)

![Site Role](image-url)
Groups

You can organize Tableau Server users into groups to make it easier to manage multiple users. You can either create groups locally on the server or import groups from Active Directory.

To keep Active Directory group membership up-to-date:

- Site administrators can synchronize selected groups on demand in a site. For more information, see Synchronize Active Directory Groups on a Site.
- Server administrators can synchronize all Active Directory groups on the server based on a schedule or on-demand. For more information, see Synchronize All Active Directory Groups on the Server.

You can also assign permissions to a group for a project, workbook, view, or data source. For details, see Manage Permissions on page 834.

The All Users group exists in every site by default. Every user added to the server becomes a member of the All Users group automatically. You cannot delete this group, but you can set permissions for it.

Create a Local Group

Local groups are created using the Tableau Server internal user management system. After you create a group you can add and remove users.

1. In a site, click Groups, and then click New Group.

2. Type a name for the group and click Create.
Create a Group via Active Directory

When you import Active Directory groups, a matching group is created on the server and a user is created on the server for each member of the group that is not already on the server.

Each user is assigned a site role as part of the import process. If a user already exists on the site without a group affiliation, the user is added to the group with the assigned site role, and the same permissions in the site.

Before importing groups, be sure to review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

**Note:** Importing users and groups will promote a user's site role, but never demote a user's site role. If any of the users to be imported already exist in Tableau Server, the site role assigned during the import process will be applied only if it gives the user more access to the server. For more information, see Site Roles for Users on page 788.

1. In a site, click **Groups**, and then click **Add Groups**
2. Type the name of the Active Directory group you want to import, and then select the group name in the resulting list.
3. Select the site role for the users.
4. Click **Import**.

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

**Synchronize Active Directory Groups in a Site**

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

1. In a site, click **Groups**.

   On the Groups page, select one or more groups.
2. Click **Actions > Synchronize**.

Set the minimum site role for users in an Active Directory group

In the **Groups - Details** page, administrators can set the minimum site role for group users to be applied during synchronization.

This setting does not run synchronization; it sets the minimum site role to be applied to the group every time synchronization runs. When you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role will be applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as **Unlicensed** by default.

**Note:** A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see **Site roles and Active Directory import and synchronization** on page 791.

1. In a site, click **Groups**.
2. On the Groups page, select a group.
   
   Click **Actions > Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.

What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see **Sync behavior when removing users from Active Directory** on page 1253.

What happens when an Active Directory group is removed from Tableau Server?

Many Tableau administrators use Active Directory groups to import and create users. After the users are imported into Tableau Server, administrators will then delete the group in Tableau Server. Deleting a group does not delete the users in it.
**Quick Start: Synchronize All Active Directory Groups on a Schedule**

After you import Active Directory groups in Tableau Server, you can make sure they stay synchronized in Tableau Server by setting up a schedule. You can also synchronize all Active Directory groups on the server on-demand, at any time. The minimum site role setting for the group is applied when users are synchronized.

**Note:** To use this feature, your Tableau Server installation must be set up for Active Directory.

1. **Set a minimum site role for synchronization**

   In a site, click **Groups**. Select a group, and then click **Actions** > **Minimum Site Role**. Select the minimum site role, and then click **Change Site Role**. Server and site administrators can set the minimum site role for group users to be applied during Active Directory synchronization. If you don’t set a minimum site role, new users are added as **Unlicensed**.

   ![Minimum Site Role](image)

   Synchronizing can promote a user's site role, but will never demote a user's site role.

2. **Set the schedule**

   Server administrators can enable synchronization for all Active Directory groups on the **General** tab of the **Settings** page for the server. Enable synchronization, select the
frequency settings, and then click **Save**.

All Active Directory groups on the server are synchronized according to the same schedule.

**3 Run synchronization on-demand (optional)**

On the **General** tab of the **Settings** page, click **Synchronize All Groups** to synchronize all Active Directory groups on Tableau Server immediately. Click this button at any time to ensure new users and changes are reflected in all Active Directory groups on the server.
Click **Synchronize All Groups** to synchronize all Active Directory groups on the server outside of a schedule.

4 View the status of synchronization tasks

Server and site administrators can view the results of Active Directory synchronization jobs in the **Background Tasks for Non Extracts** administrative view. On the server or in a site, click **Status**. Under **Analysis**, click **Background Tasks for Non Extracts** and filter on the **Queue Active Directory Groups Sync** and **Sync Active Directory Group** tasks.

The Queue Active Directory Groups Sync queues the Sync Active Directory Group tasks to be run.

**Synchronize All Active Directory Groups on the Server**

As a server administrator, you can synchronize all Active Directory groups on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.
The Last synchronized time indicates the time that synchronization most recently began.

Synchronize Active Directory groups on a schedule

1. **Single-site:** Click Settings > General.
   - **Multisite:** In the site menu, click Manage All Sites and then click Settings > General.

2. Scroll down the page to Active Directory Synchronization, and then select Synchronize Active Directory groups on a regular schedule.
3. Select the frequency and time of synchronization.

4. Click Save.

**Synchronize all Active Directory groups on demand**

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. **Single-site**: Click **Settings > General**.

2. **Multisite**: In the site menu, click **Manage All Sites**, and then click **Settings > General**.

![Active Directory Synchronization](image)

2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

**View synchronization activity**

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site**: Click **Status**.

2. **Multisite**: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.

3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.
Set the minimum site role for users in an Active Directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don’t set a minimum site role, new users are added as **Unlicensed** by default.

**Note:** A user’s site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see **Site roles and Active Directory import and synchronization** on page 791.

1. In a site, click **Groups**.
2. On the Groups page, select a group.
   - Click **Actions > Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.

What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user's content.
before removing the user’s account completely. For more information, see Sync behavior when removing users from Active Directory on page 1253.

Delete Groups

You can delete any group from the server (with the exception of the All Users group). When you delete a group, the users are removed from the group but they are not deleted from the server.

1. In a site, click Groups.
2. On the Groups page, select one or more groups to delete.
3. Select Actions > Delete.

Control Access to Published Content

Administrators can control access to Tableau Server content by assigning permissions to projects, workbooks, views, and data sources. They also can specify and change owners for projects, workbooks, and data sources.

Content owners have control over the permissions for the content that they publish to the server.
Manage Permissions

In Tableau Server, you set *content permissions* in order to specify who is allowed to work with what content in a site.

About content permissions

Content permissions ensure that only the right people can see and interact with your content. For example, you can tightly restrict who has access to your company’s financial information, but widely share organizational development content.

You assign content permissions to the following items:

- Projects
- Workbooks
- Views
- Data sources

About permission rules, site roles, and user permissions

You assign content permissions by setting *permission rules*. Permissions rules are the explicit capabilities you assign to a user or group for a given content item. A *capability* is a task that you want a user to be able to perform, such as editing a view. Every project, workbook, view, or data source can have a unique set of permission rules.

In addition to content permissions, a user’s *site role* and whether the user is a *content owner* also affects what tasks a user can perform and what actions are available to the user for each content type.

*User permissions* are the effective permissions that determine what a user can actually do with the content. They are the result of how Tableau evaluates each user or group permission rule that applies to a user for a given content item.

For more information, see *Site Roles for Users* on page 788 and *How Permissions are Evaluated* on page 839. Also see *Projects and Content Permissions* for a walkthrough that uses a best practice approach to setting up permissions.
Example: A permission rule set for the "Finance" group.

Who can set permissions

User who have the Set Permissions capability can change permissions for content items in projects that aren't locked. Administrators, content owners, and users with the Project Leader capability automatically receive the Set Permissions capability.

Note Project Leader is a permissions capability that you can set for a user or group at the project level.

Default permissions and projects

The permissions assigned to content when it is published or created on the server are the item's default permissions. Default permissions are set only at the project level, and only by administrators and users with the Project Leader capability.

- New projects get a copy of content permissions from the Default project in the site. These permissions include the permissions for the project, and the default permissions for its workbooks and data sources.
- New workbooks and data sources use the default permissions from their project. When content permissions are not locked, the individual workbook and data source permissions can be edited to differ from the defaults.
- New views use the default permissions from their workbook. When content permissions
are not locked and the views aren't shown as tabs in the workbook, the individual view permissions can be edited to differ from the defaults. Note that tabbed views always use their workbook permissions.

When the content permissions are locked to the project, workbooks and data sources in the project will always use the default permissions. Views in the workbooks will always use their workbook permissions. The default permissions can only be changed at the project level.

If you are new to using permissions in Tableau Server, see Projects and Content Permissions for a walkthrough that uses a best practice approach to setting up permissions.

For more information on the Default project, see Projects on page 756.

For more information on default permissions, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

For more information on locking content permissions, see Lock Content Permissions to the Project on page 869.

**Permission Rules and User Permissions**

When you specify permissions for a project, workbook, view, or data source, you specify who is allowed to work with that resource through a permission rule. Permission rules are the explicit capabilities that can be set for an individual user, or for a group—for each resource.

The Permissions window has two sections: Permission Rules (upper section) and User Permissions (lower section). You set permissions in Permission Rules, and you view the effective or resulting permissions in User Permissions.
Permission Rules

The permission rules you set up include the user or group and the set of capabilities you want users to have for on that content item (such as the ability to edit a view). Available capabilities vary depending on the type of content selected, and can be set to **Allowed**, **Denied**, or **Unspecified**.

For information about setting and viewing permissions, see **Quick Start: Permissions** on page 842, **Edit Permission Rules** on page 872, and **View Permission Rules and User Permissions** on page 873.

Click the ellipses next to the permission rule name.

Select a permission role template and edit capabilities (the actions allowed on the content).

- **User / Group**: Lists users or groups of users the rule applies to.

- **Permissions**: Lists available permission role templates for a specific project, workbook, view, or data source. Each permission role template (such as **Editor**, **Interactor**, **Viewer**) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to **Custom**. For more information about permission role templates and capabilities, see **Set Permissions for Workbooks and Views** on page 846, **Set Permissions for a**
Project on page 856, and Set Permissions for a Data Source on page 851.

- **View / Interact / Edit**: Categories for the sets of capabilities that can be set to *Allowed*, *Denied*, or *Unspecified*. *(Unspecified* results in *Denied* if no other permissions are specified for a user or group on the content.)*

**User Permissions**

The User Permissions area of the Permissions window shows the effective permissions for each user. These are the actual permissions for each user, after the user's site role and permission rules have been evaluated.

To view the user permissions for a group or user, click a user or group name in the permission rules list. The effective permissions for users in the group are displayed in the lower half of the Permissions window.

Effective user permissions for a resource are determined by:

- The maximum capabilities allowed for a user's site role. The site role acts as the "ceiling" for what permissions are allowed. For more information, see Site Roles for Users on page 788.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

For example, if a user is granted Editor-level permissions for a workbook (which allows all available capabilities), but has the site role of Viewer and does not own the workbook, the user will only be allowed the capabilities of *View*, *Export Image*, *Summary Data*, *View Comments*, *Add Comments*, and *Save*.

In the following example, a permission rule has been created for the Finance group. The permission role template of *Editor* was initially applied to the group, which granted all capabilities. The administrator then set the *Save* capability to *Denied*, so the name for the set of permissions applied to the group became *Custom*. The User Permissions section for the Finance group shows that most of the users in the group have all capabilities, except for the *Save* capability. One user has even fewer capabilities because that user has a site role of Viewer.
Note that the All Users group permission rule in this example has been set to None, which leaves all of the permissions as Unspecified for the All Users group. This approach requires the administrator to specifically assign permissions for only the groups or users that should see the content.

How Permissions are Evaluated

Permissions in Tableau Server are assigned to resources, also known as content—projects, workbooks, views, and data sources. You specify who can work with a resource using permission rules.

The views, workbooks, projects, and data sources on Tableau Server that users can access, and the actions available for these different content types, are affected by:

- **Site role**. A user’s site role determines whether a user can publish, interact with, or only view resources and the different levels of permission capabilities allowed for a user. The site role acts as the "ceiling" for what permissions are allowed. For more information, see Site Roles for Users on page 788.

- **Content permissions**. Every resource, that is, every project, workbook, view, or data source, can have a unique set of permission rules.

  A permission rule includes the user or group, and the set of capabilities you want to grant users for a resource (such as the ability to edit a view). Each permission role template
(such as Editor, Interactor, Viewer) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to Custom.

Available capabilities vary depending on the resource. Capabilities can be set to Allowed, Denied, or Unspecified. Denied always takes precedence over Allowed, and Unspecified results in Denied if no other permission rules allow a capability for a user.

- **Ownership.** Content owners always get full access to the content they've published. In projects with locked permissions, content owners cannot edit permissions for their workbooks and data sources.

Users with the Set Permissions capability can change permissions for content items in projects that aren't locked. Administrators, content owners, and users with the Project Leader capability automatically have the Set Permissions capability.

You can set permission rules for an individual user or group for each resource. This diagram illustrates how permission rules are evaluated in Tableau Server.

Effective user permissions are determined by:

- Maximum permissions allowed for a user’s site role. For more information, see Site Roles for Users on page 788.
- Whether the user owns the content item
- The evaluation of each user or group permission rule that applies to that user for that content item

**Notes on permissions**

- Server and site administrators can access all the resources in a site with full permissions.
- You cannot set permissions at the site level; permissions are assigned to resources only.
- Publishers (content owners) always get full access to their content. Content owners can change permissions on their workbooks and data sources, unless the parent project permissions are locked. For more information, see Lock Content Permissions to the Project on page 869.
- Individual user permissions on resources take precedence over group permissions on
resources. In other words, user permissions trump group permissions.

- Workbook permissions serve as templates for view permissions. When content permissions are locked to the project, and when a workbook uses tabbed views, views inherit their workbook permissions. When permissions are not locked, and when a workbook is saved without tabs, the workbook and view permissions can be edited independently.

- Project default permissions serve as templates for content in a project. When content permissions are locked to the project, the workbooks and data sources always use the default permissions. When permissions are not locked, workbook and data source permissions can be edited independently.

- For each content item, every site user is automatically included in the All Users group. As a result, the All Users permission rule affects how permissions are evaluated for users when you create additional group permission rules for that content item.

If you use Tableau Server in an environment where openly sharing knowledge and information across the organization is important, set the permission rule for the All Users group in the Default project to the Publisher permission template. Users can publish to and consume content from new projects.

If you use Tableau Server in an environment where restricting access is important, set the permission rule for the All Users group in the Default project to the role of None. Then, add explicit permissions for groups and users to allow them to publish and work with content in new projects.

Tableau Server evaluates permissions in the following order of precedence:

1. Server and Site Administrator: Administrators can access all site content with full permissions.

2. User - Unlicensed, Viewer license, or Guest: If a user is Unlicensed, has a Viewer license (different than Viewer site role), or is a Guest, there are certain capabilities they are never allowed to perform. If the capability is explicitly denied for the user because of licensing, they are denied.

3. Project Owner: If the user owns the project that contains the content, the capability is allowed. Otherwise,

4. Project Leader: If the user has the Project Leader capability, or is in a group that has the Project Leader capability, they are allowed. If the user is explicitly denied the Project Leader capability, they are denied. Otherwise,

5. User - Authorizable Owner: If the user is the owner of the content, they are allowed. Otherwise,

6. User - Capability Denied: If the user has been explicitly denied the capability for the content, they are denied. Otherwise,
7. **User - Capability Allowed:** If the user has been explicitly allowed the capability for the content, they are allowed. Otherwise,

8. **Group - Capability Denied:** If the user belongs to a group that has been explicitly denied the capability for the content, they are denied. Otherwise,

9. **Group - Capability Allowed:** If the user belongs to a group that has been explicitly allowed the capability for the content, they are allowed. Otherwise,

10. The user is denied access to the content.

**Quick Start: Permissions**

You can use permission rules to control access to specific content on a site. Every user has a set of allowed capabilities based on their site role. Each content type—projects, workbooks, views, and data sources—can have permission rules assigned to groups or to specific users. The easiest and most efficient way to manage permissions is to create permission rules for groups.

**1 Add Users to Groups**

Within a site, click **Groups**. Create groups for users who should have the same permissions, and then add the users to these groups. Click a group name, and then click **Add Users** to select the users to be included in the group.
2 Select the Content

On the Content page for a site, click Workbooks, Views, Projects, or Data Sources. Select an item in the page. Select Actions > Permissions to view the permission rules for that content.
A permission rule is a set of capabilities (such as the ability to edit a view) that are allowed or denied to a user or group of users. Available capabilities vary depending on the type of content selected.

3 Create a Permission Rule

Click Add a user or group rule, select Group, enter search text, and then select a name from the list. Select a permission role template to apply an initial set of capabilities for the group. Click a capability to set it to Allowed or Denied, or leave it Unspecified. Click Save when you are done.
Whether a user can set permissions is based on their site role and how their Set Permissions capability is set.

4 View User Permissions

After you save the permission rule for the group, you can view the effective permissions for that content.

Click a group name to see the group’s users and their permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
Custom indicates a user's capabilities have been changed from the initial settings for their site role or content role.

Site roles

A user's site role determines the maximum permissions allowed for that user.

- Server and site administrators can access all site content with full permissions.
- Owners always get full access to the content they've published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For more information, see Site Roles for Users on page 788.

Permissions evaluation

- **Denied** takes precedence over **Allowed**.
- **Unspecified** results in **Denied** if no other permissions are specified.
- Specific user permissions on content take precedence over group permissions on content. In other words, user permissions trump group permissions.

For more information on working with permissions, see Manage Permissions on page 834, How Permissions are Evaluated on page 839, Permission Rules and User Permissions on page 836, and Projects on page 756. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Set Permissions for Workbooks and Views

As an administrator or user with the Set Permissions capability, you can set permission rules for a workbook or a view.
For more details on working with permissions, see Manage Permissions on page 834 and Projects on page 756.

**Note:** When project content permissions are locked, permissions cannot be changed for individual workbooks and views in the locked project. For more information, see Lock Content Permissions to the Project on page 869 and Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Use permission rules to set these capabilities for workbooks:

![Permission rules for workbooks](image)

Use permission rules to set these capabilities for views:

![Permission rules for views](image)

To set permissions on a workbook or view

1. In the Content page of a site, click **Workbooks** or **Views**. Select a workbook or view, and then click **Permissions** to view the current permission rules.
Note: If you select multiple items and some of the items are read-only, you cannot view the permissions. Instead, select one view at a time.

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.
The list of capabilities and the available permission role templates vary depending on whether you are setting permissions for a workbook or a view. For more information on capability definitions, see Permissions Reference on page 874.

**Note:** For workbooks and views that contain confidential data, it is good practice to set the All Users group permissions to **None** (all permissions **Unspecified**). You can then add other group permission rules to allow access.

The available permission role templates for workbooks and views are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Applies to...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>workbooks views</td>
<td>Allows the user or group to view the workbook or view on the server.</td>
</tr>
<tr>
<td>Interactor</td>
<td>workbooks views</td>
<td>Allows the user or group to view the workbook or view on the server, edit workbook views, apply filters, view underlying data, export images, and export data. All other permissions are inherited from the user's or group's project permissions.</td>
</tr>
<tr>
<td>Editor</td>
<td>workbooks views</td>
<td>Sets all capabilities for the rule to <strong>Allowed</strong>.</td>
</tr>
<tr>
<td>None</td>
<td>workbooks views</td>
<td>Sets all capabilities for the rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>Denied</td>
<td>workbooks views</td>
<td>Sets all capabilities for the rule to <strong>Denied</strong>.</td>
</tr>
</tbody>
</table>

4. To further customize the rule, click the actions menu (…) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
5. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

6. Follow the same steps to configure additional permission rules on the content for more users or groups.

Note: Tabbed views are views in a workbook that is published to the server with Show Sheets as Tabs enabled. Tabbed views use the workbook permissions instead of the view permissions. When you view the permissions for a tabbed view in a workbook, you see the workbook’s permission rules in the Permissions window, not the view’s permission rules.

To edit tabbed view permissions, you must open the tabbed view's workbook permissions. The changes that you make to the workbook permissions affect all tabbed views in that workbook. When the workbook is saved again without tabs (or tabs are hidden), the default permissions are again applied to the workbook and views, and view
permissions can then be edited.

Views in a workbook in a project with locked permissions will also use the workbook permissions. For more information, see Lock Content Permissions to the Project on page 869.

**Set Permissions for a Data Source**

As an administrator or user with the Set Permissions capability, you can change permissions for a data source.

For information on how data source authentication interacts with data source permissions, see How "Embedded password" and "Prompt user" settings affect permissions for published data source connections on page 855.

For more information on permissions in general, see Manage Permissions on page 834 and Projects on page 756.

**Note:** When project content permissions are locked, permissions cannot be changed for data sources in the locked project. For more information, see Lock Content Permissions to the Project on page 869 and Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Use permission rules to set the following capabilities for a data source:
<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>View the data source on the server.</td>
</tr>
<tr>
<td>Connect</td>
<td>Connect to the data source. The <strong>Connect</strong> permission allows a user to connect to a data source from an editor (in Tableau Desktop or Tableau Server web editing). <strong>Note:</strong> If a workbook author embeds credentials in a workbook or view, users who also have the <strong>Web Edit</strong> permission will be able to access to the workbook’s data source regardless of their <strong>Connect</strong> permissions.</td>
</tr>
<tr>
<td>Save</td>
<td>Publish data sources to the server and overwrite data sources on the server.</td>
</tr>
<tr>
<td>Download Data Source</td>
<td>Download the data source from the server.</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete the data source.</td>
</tr>
<tr>
<td>Set Permissions</td>
<td>Specify permissions for the data source.</td>
</tr>
</tbody>
</table>

**To set permissions for a data source**

1. In the Data Sources page, select one or more data sources, and then select **Actions > Permissions**.
2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.

3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.
The permission role templates for data sources are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector</td>
<td>Allows the user or group to connect to the data source on the server.</td>
</tr>
<tr>
<td>Editor</td>
<td>Allows the user or group to connect to, download, delete, and set permissions on data sources on the server. They can also publish data sources, and as long as they are the owner of a data source they publish, they can update connection information and extract refresh schedules. (The latter two capabilities are no longer available if an administrator or project leader changes data source ownership.)</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to Unspecified.</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to Denied.</td>
</tr>
</tbody>
</table>

**Note:** Cube data sources, like those for Microsoft Analysis Services or Oracle Essbase connections, must be used locally. To download the published data source to Tableau Desktop, you need the Download permissions. You must explicitly grant the Download permissions because the Data Source Connector role does not provide these. For more information, see Cube Data Sources on page 893.

4. To further customize the rule, click the actions menu (…) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
5. Follow the same steps to configure additional permission rules on the content for more users or groups.

6. View the resulting permissions.

   Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

How "Embedded password" and "Prompt user" settings affect permissions for published data source connections

When a Tableau Desktop user publishes a workbook or data source to Tableau Server, the authentication mode (if used) affects how the Connect permission is evaluated.
- If a workbook author uses the **Embedded password** option when publishing a workbook, users will always be able to access the workbook, regardless of those users' unique **Connect** permissions on the published data source connection that is used by the workbook. In other words, the author is effectively giving users permission to access the workbook's published data source connection and those users' **Connect** permissions are irrelevant.

- In cases where a workbook author uses the **Prompt users** option when publishing a workbook, access to the workbook's published data source connection will follow their **Connect** permissions. Users will have access to the workbook to the data source connection when the published data source uses an embedded password and **Connect** is **Allowed**. Users will be prompted for the data source password when **Connect** is **Allowed**.

The following table summarizes how the **Connect** permission interacts with different modes of authentication for a workbook's published data source connection.

<table>
<thead>
<tr>
<th>Workbook</th>
<th>Published Data Source Connection</th>
<th>Connect capability</th>
<th>Access to data source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded password</td>
<td>Embedded password</td>
<td>Allowed</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td></td>
<td>Prompt user</td>
<td>Allowed</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Allowed (uses workbook author’s Connect permissions)</td>
</tr>
<tr>
<td>Prompt user</td>
<td>Embedded password</td>
<td>Allowed</td>
<td>Allowed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Denied</td>
</tr>
<tr>
<td></td>
<td>Prompt user</td>
<td>Allowed</td>
<td>Prompt user for credentials</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Denied</td>
</tr>
</tbody>
</table>

**Set Permissions for a Project**

Every project includes permissions that can be set for the project, and for its workbooks and data sources. These permissions become the default permissions settings for all content in the
project, and each project can have its own set of default permissions. For more information, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Administrators and users with the Project Leader permission can lock content permissions to a project. For more information, see Quick Start: Lock Project Permissions, Lock Content Permissions to the Project on page 869.

For more information on working with permissions, see Manage Permissions on page 834 and Projects on page 756.

**Note:** When you create a new project, it initially will have the same permissions as the Default project in the site, which are the default permissions for the project, and its workbooks and data sources.

The three capabilities you can set specifically for a project are: View, Save, and Project Leader.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</td>
</tr>
<tr>
<td>Save</td>
<td>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project.</td>
</tr>
</tbody>
</table>
When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining all permissions.

Subsequently, the original owner's access to the workbook is determined by that user's group permissions and any further permissions the new owner might set.

This permission also determines the user's or group's ability to overwrite a workbook after editing it on the server. For related information, see Grant Web Edit, Save, and Download Permissions on page 878.

| Project Leader | Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions. |

**To set permissions for the project**

1. On the Projects page, select a project, and then select *Actions > Permissions*.

2. Click *Add a user or group rule*, select *Group* or *User*, and then select the group or user name from the list.
3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click Save.

The available permission role templates for projects are:

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>Allows the user or group to view the workbooks and views in the project.</td>
</tr>
<tr>
<td>Publisher</td>
<td>Allows the user or group to publish workbooks and data sources to the server.</td>
</tr>
<tr>
<td>Project Leader</td>
<td>Allows the user or group to set permissions for all items in a project.</td>
</tr>
</tbody>
</table>
None | Sets all capabilities for the permission rule to **Unspecified**.
---|---
Denied | Sets all capabilities for the permission rule to **Denied**.
Data Source Connector | Allows the user or group to connect to data sources in the project.
Data Source Editor | Allows the user or group to connect to, edit, download, delete, and set permissions for a data source in the projects. They can also publish data sources, and as long as they are the owner of a data source they publish, can update connection information and extract refresh schedules. This permission is relevant for views when the view they access connects to a data source.

4. To further customize the rule, click the actions menu (· · ·) next to the permission rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

5. View the resulting permissions.
   Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
6. Follow the same steps to configure additional permission rules on the content for more users or groups.

**Set Default Permissions for a Project, and its Workbooks and Data Sources**

As an administrator or project leader, you can set a project’s permissions and the default permissions for its workbooks and data sources.

Each project can have its own set of default permissions. The permissions that you set are the default permissions for all content in the project, including content that is being published to the project from Tableau Desktop.

**Note:** New projects are always created with the default permissions set for the Default project.

For additional information on working with permissions, see Manage Permissions on page 834 and Projects on page 756. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.
Notes on default permissions in locked projects

You can choose to have the default permissions apply to all workbooks and data sources in a project, and ensure that no one can change those settings, by locking content permissions to the project. For more information, see Lock Content Permissions to the Project on page 869.

- Workbooks and data sources in a locked project always use the default permissions set for content in that project. Views in a locked project always use the workbook permissions. This applies to workbooks and data sources when they are being published from desktop.
- Administrators and users with the Project Leader permission can always edit default permissions, even when a project is locked.
- Users, including content owners, cannot edit individual workbook, view, and data source permissions when content is locked to the project.

To set default permissions in a project

1. In the Content page of a site, click a project, and then click Permissions in the project place page.

2. Click Add a user or group rule, select Group or User, and then select the group or user name from the list.
For an existing user or group, click the actions menu (...), and then click **Edit**.

3. Select a permission role template for **Project**, **Workbooks**, or **Data Sources**, and then click **Save**.
Or, to create a custom set of capabilities, click the **Project**, **Workbooks**, or **Data Sources** labels to expand the permissions view. Click capabilities to set them to **Allowed**, **Denied**, or **Unspecified**. Click **Save**.

This example shows how to set project permissions. The same general steps apply for workbooks and data sources.

**Note:** To change the settings after saving, click the actions menu ( . . . ), and then click **Edit**.

4. View the user permissions, which are the effective permissions.

   Click a group name or user name in the permission rules to see the resulting user permissions.
Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.

Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
5. Follow the same steps to configure additional permission rules for more users or groups.

**Quick Start: Lock Content Permissions to a Project**

As an administrator or project leader, you can lock content permissions in a project to prevent users from changing the permissions of any content in the project. When permissions are locked to the project, the default permissions are applied to all workbooks and data sources in a project and cannot be modified by users (including the content owners).

**Note:** Content owners always get full access to the content they've published, but cannot change permissions for their workbooks and data sources when the parent project permissions are locked.

For related information on setting permissions, see Manage permissions. For more information on setting default permissions and locking content permissions to the project, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861 and Lock Content Permissions to the Project on page 869. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

1 Set Default Permissions for the Project

Because the content inside locked projects always uses the default permissions, first verify that your default permissions are set appropriately. In a site, click Content > Projects. Open a project, and then click Permissions. Add a user or group and select a permission role template for that content type, or click Edit, and then set capabilities to Allowed, Denied, or Unspecified.
Administrators and Project Leaders can edit default permissions at any time.

2 Lock Content Permissions to the Project

In a project's permissions, click the Managed by the owner button. The button label indicates whether content permissions are currently locked to the project or managed by the content owner. Select Locked to the project, and then click Save.
When permissions are locked to the project, all content in the project uses the default permissions. No users can change permissions for individual workbooks (including views) or data sources in the project.

3 View Locked Permissions

Open a project, select a workbook or data source in the project, and then click **Actions > Permissions**. When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

In this example, the workbook owner has full permissions for the workbook, but cannot change the workbook permissions while they are locked to the project.
Unlock Content Permissions for the Project

In a site, click **Content > Projects**. Select a project, and then click **Actions > Permissions**. Click the **Locked to the project** button. Select **Managed by the owner**, and then click **Save**.

When a project's content permissions are **Managed by the owner**, individual workbooks, views, and data sources in the project start with the default permissions and can be modified by users.

Notes on project permissions:

- Only administrators and project leaders can lock content permissions, and set and edit default permissions in a project.
- Administrators and project leaders can edit default permissions for the project, its workbooks, and its data sources at any time, at the project level.
- Individual workbook, view, and data source permissions cannot be edited by users (including content owners) when a project is locked.
- Workbooks and data sources in a locked project always use the default permissions. Views in a locked project always use the workbook permissions.

Lock Content Permissions to the Project

As an administrator or project leader, you can prevent users from changing the permissions for workbooks and data sources in a project. To do so, you can lock content permissions for that project.
When permissions are *locked to the project*, the default permission settings are applied to all workbooks, views, and data sources in a project and cannot be modified by users (including content owners). When permissions are *managed by the owner* ("unlocked"), content permissions remain the same as when the project was locked, but the permissions become editable.

**Note:** Owners always get full access to the content they’ve published, but can only change permissions for their workbooks and data sources when the parent project permissions are not locked.

For information on default permissions, see *Set Default Permissions for a Project, and its Workbooks and Data Sources* on page 861. For a best-practice walkthrough on how to implement permissions, see *Projects and Content Permissions*.

**Note:** Administrators and project leaders can set and edit default permissions for the project, and its workbooks and data sources when it is locked.

1. In the Content page of a site, open a project, and then click **Permissions** in the project place page.

2. Click the **Managed by the owner** button.

   The padlock icon on the button label indicates whether content permissions are currently locked to the project or managed by the content owner.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.
When permissions are locked to the project, users can view workbook or data source permissions in the project, but they cannot modify them.

4. To unlock content permissions for the projects, open the project permissions again. Click the **Locked to the project** button. In the **Content Permissions in Project** dialog box, select **Managed by the owner**, and then click **Save**.
The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

**Edit Permission Rules**

1. In the Content page of a site, select a project, workbook, view, or data source, and then select **Actions > Permissions** to view the current permission rules.

To select an item in the page, select the checkbox for the item.

Example of permission rules for a workbook.

2. For the permission rule that you want to change, click the actions menu (..) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.

3. View the resulting permissions.
Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

![Permission Rules Table]

4. Follow the same steps to configure additional permission rules on the content for more users or groups.

**View Permission Rules and User Permissions**

At any time, you can view the permissions for a user or group, for a view, workbook, project, or data source. The permissions shown are specific to the view, workbook, data source, or project you have selected.

1. On the Content page for a site, click *Workbooks, Views, Projects, or Data Sources*. To select an item in the page, select the checkbox for the item.

2. Select **Actions > Permissions** to view the current permission rules.
3. Click a group or user name in the permission rules area to see the resulting permissions. Hover over a capability box in User Permissions to see a tooltip with details on whether a capability is allowed or denied.

Permissions Reference

Administrators and other authorized users can allow or deny permissions on resources in Tableau Server. Permissions can also be set in Tableau Desktop when publishing a workbook or data source to Tableau Server.

Administrators always have full control of all resources on Tableau Server, and site administrators have full control of all resources on a site. If you publish a workbook or data source to Tableau Server, you are the owner of that resource, and you retain full control over that resource, with the exception of setting permissions on resources in locked projects. For more information, see Lock Content Permissions to the Project on page 869.
The following table shows which permissions apply to which resources in Tableau Server, and describes the capabilities (that is, the actions users can perform) with each permission.

<table>
<thead>
<tr>
<th>Permission</th>
<th>Applies to...</th>
<th>When allowed, users can...</th>
</tr>
</thead>
</table>
| View             | workbooks, data sources, views, projects | View the item on Tableau Server.  
**Note:** When a workbook is configured to show sheets as tabs, all views use the workbook permissions, even if different permissions are specified on an individual view. |
| Web Edit         | workbooks, views | Edit views in workbooks. See Grant Web Edit, Save, and Download Permissions on page 878. |
| Save             | workbooks, data sources, views, projects | Overwrite the resource on the server. When allowed, the user can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining access to all permissions. Subsequently, the original owner's access to the workbook is determined by that user's group permissions and by any further permissions the new owner might set.  
This permission also determines the user's or group's ability to overwrite a workbook after editing it on the server. See Grant Web Edit, Save, and Download Permissions on page 878.  
Special consideration for the **All Users** group: To help protect an owner's content from being overwritten by another user (via publishing from Tableau Desktop or saving a web-edited workbook on Tableau Server), whenever a user publishes into a project where the **All Users** group has permissions, the Save permission for the **All Users** group is changed from **Allowed** to **Unspecified** by default. You can then manually modify this permission by following the steps in Set Permissions for Workbooks and Views on page 846 to change this from **Unspecified** to **Allowed**. |
<p>| Download Workbook/Save As | workbooks | Download a workbook from the server, and also save an edited workbook as a new workbook on the server. For more information, search for &quot;Download Workbooks&quot; in the Tableau Server Help and see Grant Web Edit, Save, and Download Permissions on page 878. |</p>
<table>
<thead>
<tr>
<th>Permission</th>
<th>Applies to...</th>
<th>When allowed, users can...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download Data Source</td>
<td>data sources</td>
<td>Download the data source from the server.</td>
</tr>
<tr>
<td>Delete</td>
<td>workbooks</td>
<td>Delete the resource.</td>
</tr>
<tr>
<td></td>
<td>data sources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>workbooks</td>
<td>Modify filters in the view, keep only filters, and exclude data.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Add Comments</td>
<td>workbooks</td>
<td>Add comments to views in a workbook. Search for &quot;Comment on Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>View Comments</td>
<td>workbooks</td>
<td>View the comments associated with the views in a workbook. Search for &quot;Comment on Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Download Summary Data</td>
<td>workbooks</td>
<td>View the aggregated data in a view, or in the user’s selection within the view, and download that data as a text file.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Download Full Data</td>
<td>workbooks</td>
<td>View the raw data behind each row in a view, as restricted by any marks the user has selected, and download the data as a text file.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Permission</td>
<td>Applies to...</td>
<td>When allowed, users can...</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Download Image/PDF</td>
<td>workbooks</td>
<td>Download each view as an image. For more information, search for &quot;Download Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td>Share Customized</td>
<td>workbooks</td>
<td>Make saved customizations to a view available for others to see. Users can create custom views using <strong>Custom Views</strong> in Tableau Server. For more information, Search for &quot;Custom Views&quot; in the Tableau Server Help.</td>
</tr>
<tr>
<td>Move</td>
<td>workbooks</td>
<td>Move workbooks between projects.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Only administrators can move data sources between projects.</td>
</tr>
<tr>
<td>Set Permissions</td>
<td>workbooks</td>
<td>Specify permissions for the resource. For workbooks, this permission extends to the views in a workbook.</td>
</tr>
<tr>
<td></td>
<td>data sources</td>
<td>Connect to the data source on the server. The <strong>Connect</strong> permission allows a user to connect to a published data source from an editor (in Tableau Desktop or Tableau Server web editing). If a workbook author embeds credentials in a workbook or view, users who also have the <strong>Web Edit</strong> permission will be able to access to the workbook's data source regardless of their <strong>Connect</strong> permissions. <strong>Note:</strong> If a workbook is configured to show sheets as tabs, all views use the workbook permissions, even if different permissions are specified on an individual view.</td>
</tr>
<tr>
<td>Project Leader</td>
<td>projects</td>
<td>Set permissions for all resources in a project and for the project itself. Can lock project permissions and edit default permissions.</td>
</tr>
<tr>
<td>Permission</td>
<td>Applies to...</td>
<td>When allowed, users can...</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>✔️</td>
<td></td>
<td>Also can change the owner, move content, and run refresh schedules.</td>
</tr>
</tbody>
</table>

**Note:** Tabbed views are views in a workbook that is published to the server with **Show Sheets as Tabs** enabled. Tabbed views use the workbook permissions instead of the view permissions. When you view the permissions for a tabbed view in a workbook, you see the workbook’s permission rules in the Permissions window, not the view’s permission rules. To edit tabbed view permissions, you must open the tabbed view’s workbook permissions. The changes that you make to the workbook permissions affect all tabbed views in that workbook. When the workbook is saved again without tabs (or tabs are hidden), the default permissions are again applied to the workbook and views, but view permissions can then be edited.

Views in a workbook in a project with locked permissions will also use the workbook permissions. For more information, see **Lock Content Permissions to the Project** on page 869.

**Grant Web Edit, Save, and Download Permissions**

For a user to be able to edit, save, and download workbooks, they must have a site role that allows those actions, and specific capabilities in a user or group permission rule.

The following capabilities control whether a user can edit, save, and download views:

- **Web Edit**—determines whether the user can edit workbook views in Tableau Server.
  
  To edit an existing workbook, a user must have a site role of **Interactor** or **Publisher**. The **Web Edit** capability must be set to **Allowed** in the workbook permissions.

  **Note:** Users with a site role of **Interactor** are not allowed to save or download workbooks.

- **Download/Save As**—determines whether users see the **Save As** command while they are editing a view, and whether they can save their changes to a new workbook. It also determines whether users can open a workbook on the server using Tableau Desktop.
  
  To save changes to a workbook or save a workbook as a new workbook on Tableau Server, a user must have a site role of **Publisher**. The **Save** and **Download/Save As** capabilities must be set to **Allowed** in the workbook permissions.

- **Save**—determines whether users can save changes to an existing workbook on the
server (overwrite a workbook).

**Note:** The **Save** permission determines whether a user can overwrite the content on the server. This permission does not determine whether a **Save** button is displayed for users who do not own the content. Only the workbook owner can save changes to an existing workbook on Tableau Server.

To save changes to a workbook, a user must have a site role of **Publisher**. The **Save** capability must be set to **Allowed** in the workbook permissions.

**Note:** Setting the **Save** capability to **Denied** for a project disables saving to the entire project, as well as disabling overwriting the existing workbook.

### To grant Web Edit permissions

1. Set the site role of the user to **Interactor** or **Publisher**. For more information, see [Change Site Roles](#) on page 820.

2. In the permission rules for a group or user at the workbook level, set the **Web Edit** capability to **Allowed**.

3. Save the rule.

### To grant Save and Download/Save As permissions

1. Set the site role of the user to **Publisher**. For more information, see [Change a Site Role](#).

   **Note:** **Interactors** are not allowed to save or download workbooks.

2. Create a permission rule for a group or user at the project and workbook level. Set the following capabilities:

   **To allow users (Publisher site role) to edit and save changes to existing and new workbooks**

<table>
<thead>
<tr>
<th>Permission</th>
<th>For the project</th>
<th>For specified workbooks in the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Save</td>
<td>Allowed</td>
<td>Allowed</td>
</tr>
</tbody>
</table>
Note: To apply the default permissions to all workbooks within the project, lock content permissions to the project. For more information, see Lock Content Permissions to the Project on page 869.

To allow users (Publisher site role) to edit and save changes to new workbooks, but not overwrite existing workbooks

<table>
<thead>
<tr>
<th>Permission</th>
<th>For the project</th>
<th>For specified workbooks in the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Save</td>
<td>Allowed</td>
<td>Denied</td>
</tr>
</tbody>
</table>

Important: In this scenario, permissions must be set manually on each workbook and the project permissions are not locked. If project permissions are locked, the permissions apply to all workbooks in the project.

3. Save the rule.

Note: When you deny Save permissions for a workbook, users can still click Save As when editing the workbook in Tableau Server, but a message appears that tells users they do not have permission to overwrite the workbook and the changes will not be saved.

About permissions for views in workbooks

Permissions for views in workbooks are inherited from the workbook permissions.

If a user selects Show sheets as tabs when publishing a workbook from Tableau Desktop or saving it on Tableau Server, the workbook permissions override the permissions on individual views. When the workbook is saved again without tabs, the default permissions are applied to the workbook and views, but view permissions can then be edited.

See also

Permissions Reference on page 874

Quick Start: Permissions

Quick Start: Lock Content Permissions to a Project on page 866

Permission Rules and User Permissions
Set Permissions for Workbooks and Views on page 846
Set Permissions for a Project on page 856
Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861
Create Project-Based Permissions below
Site Roles for Users on page 788

Create Project-Based Permissions

As an administrator, you can organize a collection of related workbooks and data sources in a project. You can then control access to that content by creating permission rules for groups of users who need similar access levels to publish or interact with that content.

Note: For this scenario, you set the permission rule for the All Users group for the project to None, which means that permissions are Unspecified for the All Users group.

Preparation

Before you begin creating projects and project-based permissions, document the projects and permission levels that you want users to have in each project.

This roadmap exercise helps you organize permissions to be most efficient to manage over time, and can help you identify any user or permission gaps in your solution. For a best-practice walkthrough on how to implement permissions, see Projects and Content Permissions.

Also read the following topics in the Tableau Server Help:

- Manage Permissions on page 834 and permissions-related topics
- Projects on page 756 and projects-related topics
- Grant Web Edit, Save, and Download Permissions on page 878

Step 1: Create projects and user groups

1. Sign in to Tableau Server with your administrator user name and password.
2. On the Projects page, click New Project.
3. Click Groups, and then click New Group.

Create groups that correspond to each project and access level. For example, for a project that allows users only to access the views, you might use a name similar to Project1_Viewer. For a project that allows interaction with the views, Project1_Interactor.

4. Click Users, and then click Add Users. Select one or more users in the list, select Actions > Group Membership, and then select a group for the users. Click Save to
Step 2: Assign permissions at the project level

After you set up your projects and user groups, you can start assigning permissions. Repeat these steps for each project. Also see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

1. On the Projects page, select a project, and then select Actions > Permissions.
2. For the All Users group permission rule, set the permission role template to None.
   Click the actions menu ( . . ) next to All Users, and then click Edit. Select None for Project, Workbooks, and Data Sources, and then click Delete. This means that all capabilities will be set to Unspecified.
3. Click Add a user or group rule, select Group, and then select the group name in the list.
   To edit an existing rule, click the actions menu ( . . ) next to the permission rule name, and then click Edit.
4. Select a permission role template for Project, Workbooks, and Data Sources to specify a predefined set of capabilities for the group or user.
5. To further change capabilities included in the rule, click a capability in the rule to set it to Allowed or Denied, or leave it Unspecified.
   Click Save when you are done.
   Repeat steps 3-5 for each group or user requiring project permissions.

Note: You can optionally lock content permissions to the project to enforce the default permissions for all content in the project. This overwrites any previous permissions assigned to workbooks and views in the project. For more information, see Lock Content Permissions to the Project on page 869.

Step 3: Check project permissions

- View the resulting user permissions.
  Click a group name or user name in the permission rules list to see the resulting permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

When you publish workbooks to the project, the permissions are updated accordingly.

For information on granting Save permissions to users, see Grant Web Edit, Save, and Download Permissions on page 878.
Enable Web Authoring

The ability for users to edit views in Tableau Server is a setting that administrators control. In addition to this setting being enabled, user must also have the **Web Edit** permission for a given content item.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.

2. In a site’s **Settings** page, make sure **Allow users to use web authoring** is selected.

3. In the permissions for a workbook or a view, make sure the permission rule for a user or group allows the **Web Edit** capability.

4. If your site is already in production, and you want the change to take effect immediately, restart the server.

To confirm which sites allow web authoring, on the site menu, click **Manage All Sites**, and then click the **Sites** menu.

For more information on web authoring and web editing in Tableau Server, also see these topics:

- **Disable web authoring**
- **The Web Authoring workspace**
Grant edit and save permissions

Disable Web Authoring

If you want users to be able to view published workbooks on Tableau Server but not access the web editing environment, you can use a site-level setting to disable authoring.

For example, you might have a select group of data analysts who use Tableau Desktop to create and publish workbooks, and a group of sales managers working in the field, who do not use Tableau Desktop but need to access the published dashboards from a web browser.

1. In a web browser, sign in to the server as an administrator and go to the site for which you want to disable authoring.
2. With Site selected, display the Settings page.

3. In the Site Settings page, clear the check box for Allow users to use web authoring, and then click Save.

If you disable web authoring while creating a new site, no cached sessions exist, and the setting takes effect immediately.

Otherwise, the change takes effect after server session caching expires or the next time a user signs in after signing out.

Until the change takes effect, users might have authoring access if they see an Edit link on a view, or if they enter the URL for the view’s edit mode. For example, they bookmarked the URL while they had the view open for editing.

4. If your site is already in production, and you want the change to take effect immediately, restart the server.

To confirm which sites allow web authoring, on the site menu, click Manage All Sites, and then click the Sites menu.
Refresh Data on a Schedule

As a server administrator, you can allow users to subscribe to views published to the server, or to schedule tasks for refreshing published data extracts or the subscription email deliveries.

You can also specify which other users are allowed to set schedules. Otherwise, non-administrator users can work with schedules in the following ways:

- Tableau Desktop publishers can set scheduled refresh tasks when they publish a data source or a workbook with a data extract.
- Tableau Server users can subscribe to views that are delivered by email on a schedule.

Changes to an existing schedule, as well as new schedules you create on the server, are reflected in the publishing steps in Tableau Desktop the next time the author publishes content. Similarly, changes to a subscription schedule are reflected in the choices a server user has when subsequently subscribing to a view.

Data Sources

A Tableau data source consists of metadata that describes the following:

- **The data connection information** that describes what data you want to bring in to Tableau for analysis.
- **Customization and cleanup** that helps you and others use the data source efficiently. For example, calculations, sets, groups, bins, and parameters, custom field formatting, and so on.
- **Information about how to access or refresh the data**, such as a path to an Excel file, credentials for accessing data on-premises or in the cloud, and so on.

Sharing data sources

After you create and customize a data source that you want others to use for their Tableau analysis, you publish it from Tableau Desktop. After it’s published, your team can connect to it when they create or edit workbooks.
If your data source contains an extract connection, set up a refresh schedule, so that when it is refreshed, workbooks that connect to it show the updates as well.

You can also publish a workbook that contains the data source if you want users to connect to the data source only from that workbook. This is also referred to as an *embedded* data source. Every published workbook has at least one embedded data source.

**Managing data sources**

As a best practice, administrators should manage data sources on the server. However, both administrators and data source owners can perform management tasks on published data sources.

To perform these tasks (described below), do the following:

1. Sign in to the site or server as an administrator or owner of the data source you want to work with.

   **Note:** Some tasks are available only to administrators, as described below.

2. Go to the **Data Sources** page, select the check box next to the data source, and in the upper-left of the Data Sources page, select **Actions**.

   - **Edit and view permissions:** Permissions can specify which users or groups can connect to, modify, or download data sources. For information, see [Set Permissions for a Data Source](page 851).
   - **Edit connection information:** Update embedded credentials or other metadata that
creates the connections to the original data.

- **Create a new workbook or download**: You can start a new workbook in the browser environment by connecting to a Tableau data source. Or download the data source to use locally.

- **Change the data source owner**

- **View the data source’s revision history**

- **Refresh extracts**: If a data source includes an extract, you can assign the extract to a refresh schedule. For information, see Refresh Data on a Schedule on page 885.

- **Add or remove keyword tags**: Tags can contain a single word or multiple words, delimited by a comma.

- **Delete**: Deleting a data source affects workbooks that connect to the data source. Before you delete a data source, ensure that there are no workbooks that connect to the data source or edit the workbooks to use another data source.

- **Move**: Only administrators can move a data source from one project to another.

In addition, for data sources that are proxy connections, administrators can stay aware of how users authenticate to the database, and whether the appropriate drivers are installed. For information, see Database Drivers on page 702 and Data Security on page 958.

**See also**

Best Practices for Published Data Sources in the Tableau Desktop Help

**About Tableau Data Sources**

The Tableau Server data server is a server component that lets you centrally manage and store Tableau Server data sources. A data source is a reusable connection to data. The data can be located either in Tableau’s data engine, as an extract, or in a live relational database. For relational database connections, the information stored in the data source is used for a pass-through connection to the database. The data source can also include customizations you’ve made at the field-level in Tableau Desktop, such as calculations, dimension aliases, groups, or sets.

For administrators, there are many advantages to using Tableau Server data sources. Because one data source can be used by many workbooks, a data source that includes an extract means you save on server space and processing time. Extract refreshes can be scheduled per-extract instead of per-workbook, and when a workbook using a Tableau Server data source is downloaded, the data extract stays on the server, resulting in less network traffic. Finally, if a database driver is required for a connection, you only have to install the driver once, on Tableau Server, instead of multiple times, on all your users’ desktops.

To use the data server, authors connect to data in Tableau Desktop, either by creating an extract or using a connection to a live relational database, and publish the data source to
Tableau Server. Once published, these data sources and the server contain everything workbook authors need to quickly connect to data and start authoring. To change a published data source, you download it to Tableau Desktop, make your changes, then republish, overwriting your original. Note that any new members you add to a parameter or any changes you make to the default sort order are not part of the data source (they are part of the workbook).

If you are running a distributed installation of Tableau Server and expect data sources to be heavily used, there are several ways you can optimize your server deployment. See Distributed Environments on page 694 for more information.

**Note:** To use published multidimensional (cube) data sources, you must download them to Tableau Desktop, so many of the above advantages do not apply. For more information, see Cube Data Sources on page 893.

View Data Sources or Connections

You can filter the view to data sources or connections.

**Data Source view**

**Connections view**

The Difference Between Published Data Sources and Embedded Data Sources

Published data sources contain connection information that is independent of any workbook and can be used by multiple workbooks. An embedded data source contains connection information and is associated with a workbook. Every workbook has one or more embedded
data sources. If a workbook uses a published data source, an embedded data source is listed for the workbook.

**Identifying Types of Data Sources**

The list of data sources gives you information about the data sources and what they are connected to:

![List of Data Sources](image)

Data sources are distinguished by a number of characteristics in the list:

- **Icon/Name**—The data source icon next to the Name lets you know whether the data source is published (️) or embedded in a workbook (️).
  - Published data source names are links. Clicking the name of a published data source opens the data source page, with tabs for viewing the data source connections (if any) and connected workbooks.
  - Embedded data source names link to their workbooks. Clicking the name of an embedded data source opens the workbook associated with the data source, with tabs for viewing its data sources.

- **Connection Type** —The connection type gives you information about the type of connection the data source is making. A connection type of Tableau Server indicates that the connection is to a published data source. A Tableau Data Extract connection type means that the data source has an extract which is stored in Tableau Server.

- **Connects To**—The Connects To list tells you what the data source is connecting to. This could be a database outside of Tableau Server, an extract, or a published data source.

- **Live or Last Extract**—This column tells you whether the connection to the data is live, or, if it is a connection to an extract, when the extract was last updated.

**Identifying Types of Connections**

The list of connections gives you information about connections, including the data source they are connected to, connection type, and authentication:
- Connects To—Indicates what the connection’s data source is connecting to. This could be a database outside of Tableau Server, an extract, or a published data source.

- Connection Type — The connection type gives you information about the type of connection the data source is making. A connection type of Tableau Server indicates that the connection is to a published data source. A Tableau Data Engine connection type means that the data source has an extract which is stored in Tableau Server.

Embedded Data Sources

Every workbook that is published to Tableau Server contains at least one embedded data source. These embedded data sources contain the connection information for the workbook and are listed on the Data Sources page:
By default the list of data sources is filtered to only display published data sources.

To view embedded data sources, under **Data Source**, click the drop-down menu and select **Embedded in workbook** to change the filter:
The Difference Between Published Data Sources and Embedded Data Sources

Embedded data sources are different from published data sources in that each embedded data source is associated with a single workbook and describes the attributes required for connecting to a data source (e.g., server name, database name, etc.). That means if you have three workbooks that connect to the same data source, you will still have three embedded data sources listed on the Data Sources page.

Searching for Embedded Data Sources

The Filter area on the left side of the Data Sources page helps you find embedded data sources by connection type, database server name, port, username, password status (whether or not the database password is embedded) and whether or not there is an extract:
Which Connections Can I Edit?

You can edit connection information for live database connections and for extracts that need to be refreshed by Tableau Server. For example, you may have a large number of workbooks that connect to a database on a specific database server. If the name of the server changes, you can update all of the workbooks at once so they reference the new server name. Another example is if a workbook connects to a database using a specific user name and password. You can quickly update all of the workbooks to use a different set of credentials.

For details on how to edit data connections, see Edit Connections on page 896.

Cube Data Sources

Cube (multidimensional) data sources have certain characteristics that make them unique in Tableau.

Cube data sources do not support pass-through connections. This means that when a cube data source is published, you cannot make a connection from Tableau Server using the data source. It also means you cannot create a workbook using the data source in Tableau Server.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally. To download a published data source you need:
• The **Download/Web Save As** permission for the data source. For more information, see **Manage Permissions** on page 834 and **Set Permissions for a Data Source** on page 851.

• Correct drivers installed and ports opened on computer running Tableau Desktop.

### Connect to Published Data Sources

You can publish data sources to Tableau Server or Tableau Online from Tableau Desktop. Publishing data sources to Tableau Server or Tableau Online enables sharing data among colleagues; including those who don’t use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

You can connect to these published data sources on Tableau Server or Tableau Online if you have permissions to create and edit views.

**Note:** Many of the topics in this section use the Sample-Superstore data source that comes with Tableau Desktop. To use this data source, you can publish it to Tableau Server or Tableau Online from Tableau Desktop. Follow the procedure in the **Publish a Data Source** topic in the Tableau Desktop Help to learn more.

If you do not have access to the Sample-Superstore data source, you can connect to your own published data and follow the procedures using similar measures and dimensions from it.

### Connect to a published data source on the web

At any time while you’re creating or editing a view on Tableau Server or Tableau Online, you can connect to one or more published data sources.

1. Sign in to Tableau Server or Tableau Online and select a view to edit.

2. In editing mode, click the New Data Source icon .

3. In the Connect to Data dialog box, select a published data source from the list, and then click **Add**.

**Note:** By default, only data sources that have been published to the server are displayed in the list.

You can also connect to a published data source when you’re creating a new workbook on Tableau Server or Tableau Online.
1. Sign in to Tableau Server or Tableau Online.

2. Navigate to the Content page and select Data Sources.

3. In the list of data sources, select the check box next to the one you want to use, and then click Actions and select New Workbook.

Connect to a published data source from Tableau Desktop

1. On the Connect to Data page in Tableau Desktop, click Tableau Server, and then provide the server name and your credentials.

2. Select a data source you want to use.

   The data source opens in the Data pane in the workbook. Tableau Server data sources
show a Tableau icon instead of a database icon.

See also

Edit Connections

Administrators and data source owners can manage the information that describes how a published data source connects to the original data. This information includes the database server, the server port, the database user name, and whether or not the password is embedded in the connection.

**Note:** From the permissions perspective, whether you can edit connections is evaluated separately from your permissions for editing data sources. Even if you have the Edit capability on a data source, to edit its connections, your site role must be Server Administrator, Site Administrator, or Publisher. If your role is Publisher, you also must be the data source owner.

1. Sign in to the site that has the data sources you want to modify, and open the **Data Sources** page.
2. Select the name of the data source with the connection you want to update.
   
   Use the search box or filters on the left to narrow the data source list. The values you type into the **Server** and **Database username** fields are treated as regular expressions.
3. In the **Connections** view, select the check box for the connection, and then click **Actions > Edit Connection**.
4. Update the connection information.

For **Server name**, if you want to use an IP address, make sure the database or its driver supports that type of connection. If it doesn’t, enter the database server name.

For connections to Google, Salesforce, and web data connector (WDC) data, see **Authentication types for Google, Salesforce, and WDC data** on the next page later in this topic.

5. Click **Save**.

6. Refresh the browser page for your changes to take effect.
Authentication types for Google, Salesforce, and WDC data

Google BigQuery, Google Analytics, Salesforce.com, and many web data connector (WDC) connections use the OAuth authentication standard, which uses secure access tokens instead of “raw” user name and password credentials. Database credentials do not need to be stored in Tableau, and all users connect through this access token, including Tableau Desktop users who want to create or edit workbooks that connect to this data source.

The following sections describe Google and Salesforce connection options. Web data connector options vary, but all involve signing in through the provider’s web-based sign-in form to establish the access token.

Google authentication options

When you edit Google BigQuery or Google Analytics connections, select either of the following options in the Edit Connection dialog box:

- Select Embed Google BigQuery (or Google Analytics) credentials in the connection to authenticate through a designated account, and then select an existing account from the list or select authenticate account now... to add a new one.

  When you add a new account, the Google sign-in page appears. After you provide your database credentials, Google prompts you to confirm Tableau access to the data. When you click Accept, Google returns an access token to use for connecting to the data.

  **Note:** If you create extracts of your Google data source, select this first option, so that you can schedule refresh tasks.

- Select Prompt user for Google BigQuery/Analytics credentials to require users to connect through their own individual access tokens or sign in each time they connect.

Salesforce.com authentication options

When you edit Salesforce.com connections, you can select any of the following options in the Edit Connection dialog box:

- Select Embed a Salesforce username and password to use a traditional authentication method.

- Select Embed Salesforce credentials in the connection to use an OAuth connection and schedule refresh tasks, and then select an existing account from the list or click Add a Salesforce Account to add a new one.

  When you add a new account, the Salesforce.com sign-in page appears. After you provide your database credentials, Salesforce.com prompts you to confirm Tableau access to the data. When you allow Tableau access, Salesforce.com creates an access token through which it connects to the data.
- Select **No Salesforce authentication** to require users to sign in to Salesforce.com each time they connect. (This option does not allow scheduled extract refreshes.)

**Monitor progress**

When you save your changes in the Edit Connection dialog box, the dialog displays the progress. If you close the dialog box, the modifications continue to run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped, but they will not impede other changes. For example, if you try to change the server name and add a password to several connections, the server names will be changed, and the passwords on workbooks will be changed. However, because you cannot add a password to a data source, the passwords for the data sources will not be changed.

For information about checking the progress of these tasks, see **Background Tasks for Extracts** on page 1103.

**Web Data Connectors in Tableau Server**

Web data connectors are web pages that provide a data connection that is accessible over HTTP for data sources that don't already have a connector in Tableau. Web data connectors allow users to connect to almost any data that is accessible over the web and to create extracts for their workbooks. Data sources for a web data connector can include internal web services, JSON data, REST APIs, and other sources that are available over HTTP or HTTPS. Users can create their own web data connectors or use connectors that were created by others.
For information about how to use a web data connector in Tableau Desktop, see Web Data Connector in the Tableau Desktop documentation.

For information about how to create a web data connector, see the Web Data Connector documentation on Github.

- Before you run connectors on Tableau Server below
- The safe list method vs. the import method below
- The safe list method on the next page
- The import method on the next page
- Refresh the extract for a connector on page 905
- Troubleshooting on page 906

Before you run connectors on Tableau Server

As a security measure, Tableau Server won't run web data connectors unless you approve the connector, as explained in this topic.

Note: You must be a server administrator to approve web data connectors for use on Tableau Server.

Web data connectors require your approval because they contain executable code and typically make requests to third-party websites. Before a user can use a web data connector via Tableau Server, you must either add the connectors to a safe list (to a whitelist) or import the connectors into Tableau Server. Before you do this, we recommend that you vet and test the connector so that you know what the connector does and what sites it connects to. For more information, see Testing and Vetting Web Data Connectors on page 906.

The safe list method vs. the import method

When you add a connector to the safe list (whitelist), you configure Tableau Server to allow connections to a particular URL where the connector is hosted. This is the recommended way of allowing Tableau Server to run web data connectors. The connectors can then be hosted on a server inside your organization's firewall or on an external domain.

Alternatively, you can import a web data connector. When you import a connector, you run a \texttt{tabadmin} command that imports (copies) the connector from a location on your network to all of the machines in your Tableau Server installation.

Note: In versions of Tableau Server before 10.0, importing was the only way to run web data connectors on Tableau Server.
Reasons to use a safe list

You might want to add web data connectors to a safe list if:

- Your organization wants to host the connector on a separate server in your network or on an external domain. (That is, on a computer that is not running Tableau Server.)
- Your organization makes updates to connectors frequently. By adding the connector to the safe list, you avoid the need to re-import the connector each time you change it.
- The connector references many files and you do not want to import each file to Tableau Server individually.

Reasons to import web data connectors

As noted, the recommended way to configure Tableau Server to be able to run web data connectors is to use a safe list. However, you might want to import web data connectors if:

- Your organization does not have an existing web server that you can use to host the connector.
- Your organization has imported many connectors to Tableau Server in previous versions and wants to manage the connectors in a central location.

By default, both ways of configuring Tableau Server to run connectors are allowed. However, you can restrict the ways that connectors can be added or imported with the `tabadmin set webdataconnector.whitelist.mode` option. For more information, see `tabadmin set options`.

The safe list method

To add a web data connector to the safe list, use the `tabadmin whitelist_webdataconnector` command. This command lets you perform the following tasks:

- Add a connector to the safe list.
- List connectors on the safe list.
- Remove a connector from the safe list.
- Configure an optional secondary safe list, that is, a list of domains that a particular connector can send requests to and receive requests from.

For more information, see `tabadmin whitelist_webdataconnector`.

The import method

Use the `import_webdataconnector` on page 1277, `list_webdataconnectors` on page 1279, and `delete_webdataconnector` on page 1270 commands to manage imported connectors.

To import a connector to Tableau Server, follow these steps:
1. Make sure you have the HTML file for the web data connector and any supporting files, such as .css files or .js files.

2. On the server, run the `import_webdataconnector` on page 1277 command, as in this example:

   ```
tabadmin import_webdataconnector connector1.html
   ```

   **Note:** The connector name (connector1.html in this example) can contain only these characters: a-zA-Z0-9()_.

   You can import a web data connector as a local file on the server or from a network share (for example, \myshare\connector1.html), as in these examples:

   ```
tabadmin import_webdataconnector
c:\webdataconnectors\connector1.html
   ```

   ```
tabadmin import_webdataconnector
\myshare\webdataconnectors\connector2.html
   ```

   If you want to re-import a web data connector that's already been imported (for example, you want to import an updated version of the connector), use the `import_webdataconnector` command with the `overwrite` option, as in this example:

   ```
tabadmin import_webdataconnector
\myshare\webdataconnectors\connector2.html --overwrite
   ```

   When the command finishes, it displays a URL, as in this example:

   ```
===== Importing web data connector to server...
   -- The web data connector with the following URL was imported to the server:
   http://myserver/webdataconnectors/connector1.html
   ```

3. Give the URL of the imported web data connector to any users who want to use that connector.

   **Note:** If you re-import a web data connector, the older version of the connector might still be available in the server's cache, and users who work with the connector might still see the older version. By default, the maximum lifetime for an item in the cache is eight hours. To force a cache reset, restart the server.
List imported connectors

As the server administrator, you can see a list of web data connectors by running the following command:

```
tabadmin list_webdataconnectors
```

In order to reference a web data connector in a workbook, users need to know the URL for the connector. To get a list of connector URLs, use this command:

```
tabadmin list_webdataconnectors --urls
```

Delete imported connectors

If you no longer need a web data connector, you should delete it from the server. Use the following command to remove an individual web data connector, where `connector_name` is the name of the connector file to delete:

```
tabadmin delete_webdataconnector connector_name
```

(To see a list of web data connectors on the server, use the `tabadmin list_webdataconnectors` command).

To remove all web data connectors from the server, use the following command:

```
tabadmin delete_webdataconnector --all
```

**Note:** When you delete a web data connector, a version of the connector might still be available in the server’s cache, and users might still be able to work with the connector. By default, the maximum lifetime for an item in the cache is eight hours. To force a cache reset, restart the server.

Reference external files from imported connectors

If a web data connector `.html` file references external files, you must make sure that those files are available on the server. For example, a web data connector might reference an external `.css` file in a `<link>` element or a `.js` file in a `<script>` element.

If the external files are referenced using a URL (`http://`), Tableau Server can access the external files as long as the files are on a server that is accessible to Tableau Server.

If the external files are referenced as local files, you can import them into Tableau Server using the `import_webdataconnector` command. For example, if a web data connector that you are importing references the `myconnectors.css` file, you import the connector and the `.css` file using this sequence of commands:

```
tabadmin import_webdataconnector connector1.html
```
tabadmin import_webdataconnector myconnectors.css

An important point is that all files imported using the import_webdataconnector command are stored in the same directory on the server—Tableau Server does not let you import external files into a subdirectory. Therefore, you must make sure that any local files referenced in <link> or <script> elements in the connector’s .html file do not include paths, only file names.

**Imported connectors in a distributed environment**

If your server is configured as a cluster, web data connectors are imported to each computer where a gateway process is running. This makes the web data connector available for distributed access across your cluster. Deleting a connector in a distributed environment removes the connector from all the computers where the gateway process is running.

In a distributed environment, the process of importing or deleting a web data connector might complete only partially. If you’re importing a connector, the connector might be copied to some of the computers where the gateway process is running, but not to all of them. In that case, the tabadmin import_webdataconnector command reports the error using text like this:

The web data connector with the following URL has been imported to some gateways on the server, but not all.

Similarly, if you’re deleting a web data connector, the connector might be removed from some computers but not all of them. The tabadmin delete_webdataconnector command reports the error using text like this:

The web data connector was deleted from some gateways on the server, but not all.

**Note:** If the delete process is partially successful, users can still access the connector.

If the import or delete process reports partial success, you can try either of the following solutions:

- Run the import or delete process again. If you’re importing, run the tabadmin import_webdataconnector command again, and use the --overwrite option to overwrite any instances of the connector that were successfully installed. If you’re deleting, run the tabadmin delete_webdataconnector command again. Tableau Server will remove any remaining instances of the connector.

- Stop the server, run tabadmin configure, and then restart the server. The configuration process makes sure that any web data connectors are correctly distributed (imported or deleted) in all nodes where the gateway process is running. Since this option requires you to stop the server, you would choose it if it’s practical to stop the server, or if you have some other reason to stop and restart the server.
Performing site import and site export with web data connectors

Web data connectors are imported as server-wide resources; they are not associated with a specific site on your server. Therefore, if you export a site using the `tabadmin exportsite` command, the resulting .zip file does not include web data connectors that might be referenced by workbooks on the site.

Managing imported connectors for failover in a cluster

If your server is configured as a cluster with a backup primary server, you must make sure that web data connectors that you have imported to the primary are available if you need to failover to your backup primary. If the web data connectors are not available on the new primary after a failover, running the configuration process on the primary server can end up removing the connectors from other computers where a gateway process is running.

To make sure that web data connectors are available after a failover, follow these steps:

1. Make sure that you keep an up-to-date backup of the web data connectors that have been imported to your server.

2. After the primary fails, and before you start the backup primary, copy the web data connectors from the backup location to the following folder on the backup primary:
   
   `C:\ProgramData\Tableau\Tableau Server\data\tabsvc\httpd\htdocs\webdataconnectors`

   If you have created a backup of the primary server using the `tabadmin backup` command, the .tsbak file created by the backup file contains the web data connectors. You can extract the contents of a .tsbak file and get the web data connectors.

   If you installed Tableau Server on a different drive, substitute that drive letter for `C:`.

3. Overwrite the tabsvc.yml file on the backup primary.

4. Run the `tabadmin failoverprimary` command. For more information, see Quick Start: Creating a Backup Primary on page 712

If necessary, you can also reimport the web data connectors, as described earlier in this topic.

Refresh the extract for a connector

When a user creates a workbook that uses a web data connector, Tableau creates an extract from the data returned by the connector. If the user then publishes the workbook, the publish process sends the workbook and the data extract to the server.

Tableau can refresh an extract that was created by a web data connector, the same as it can refresh any extract. Tableau Server cannot invoke a web data connector to refresh an extract if the connector requires credentials to sign in to the web-based data source. This is because the refresh can occur on a schedule or in some other background context, and the server cannot prompt for credentials.
If the background process that performs the refresh operation fails, it creates an alert and a log entry that indicates this issue. (Users will be able to see that the timestamp on the extract does not change.)

If you want, you can disable refresh for all web data connectors, even those that were previously imported. To disable refresh, use the `tabadmin set` command to change the `webdataconnector.refresh.enabled` setting to `false`, as in the following example:

```
tabadmin set webdataconnector.refresh.enabled false
```

**Troubleshooting**

If the server experiences problems with adding connectors to the safe list or importing connectors, you can examine the `tabadmin.log` files. Be sure to check the log files on both the primary server and on the other servers that are running the gateway process. For more information about log files, see [Server Log File Locations](#) on page 1190.

If the issue is that Tableau Server will not refresh an extract that was created by a web data connector, make sure that the `webdataconnector.refresh.enabled` configuration setting has been set to `true`.

If you have re-imported a changed web data connector on the server (overwriting an existing one), but users who work with the web data connector are not seeing the changes, the users might be getting a cached version of the older version. By default, the cache is reset after eight hours; after a cache reset, older versions of the web data connector will no longer be used. If you want to force the cache to reset, you can restart the server.

If you have deleted an imported connector from the server but users are still able to work with the connector, the connector is probably still in the server’s cache. A web data connector can stay available in the cache for up to eight hours. To clear the cache, restart the server. If you delete a web data connector from a server in a distributed environment, make sure that the connector has been successfully deleted from all computers where a gateway process is running.

**Testing and Vetting Web Data Connectors**

Web Data Connectors contain JavaScript that typically connects to data on another site. Because of this, you should test and vet web data connectors before users use them as data sources for a workbook, and before you import them into Tableau Server.

This topic includes some suggestions for testing and vetting web data connectors.

- Examine the source
- Test the web data connector in an isolated environment
- Monitor the traffic created by the connector
- Test the performance and resource usage of the connector
Examine the source

The code in a web data connector is in JavaScript, so you can open the file (and any external files that the connector uses) and examine the source code.

Many connectors reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe. If it is practical for you to change the source code of the connector, use HTTPS protocol (https://) to reference external libraries (if the source site supports HTTPS) to help verify the site’s authenticity.

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request.

**Note:** Experienced JavaScript programmers often compress (minify) their code to reduce the size of the code for download. Dense blocks of code that use cryptic function and variable names are not uncommon. While this can make it more difficult to examine the code, it is not a sign that the code was written to be deliberately difficult to understand.

Test the web data connector in an isolated environment

If possible, test the web data connector in an environment that is isolated from your production environment and from user computers. For example, import a web data connector onto a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor the traffic created by the web data connector

When you test a web data connector, use a tool like Fiddler, Charles HTTP proxy, or Wireshark to examine the requests and responses that the connector makes. Make sure that you understand what sites the connector makes requests to and what content the connector is requesting. Similarly, examine the responses and their content to be sure that the connector is not reading data or code that is not directly related to the connector's purpose.

Test the performance and resource usage of the web data connector

When you test a web data connector, use tools to monitor its CPU and memory usage. Remember that the web data connector will run on Tableau Server, which is an environment in which many processes are already running. You want to make sure that when the connector fetches data, the connector does not have an undue impact on server performance.
Check whether the connector writes to disk. If it does, check how much disk space it occupies, and examine the output to make sure you understand what it’s writing and why.

Troubleshoot Data Sources

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

- **Permissions for the data source:** Anyone connecting to a data source must have the Connect and View permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be licensed to Publish and also have the Write/Save As and Download/Web Save As permissions. See Manage Permissions on page 834 and Set Permissions for a Data Source on page 851 for more information.

  Multidimensional (cube) data sources have to be downloaded and used in Tableau Desktop, so they require Download/Web Save As permission. For more information about cubes in Tableau, see Cube Data Sources on page 893.

- **Ability to authenticate to the database:** There are several ways you can connect to data in Tableau and control who has access to what. Basically, whichever entity is connecting to the database must be able to authenticate. The entity could be Tableau Server performing an extract refresh. It could be a Tableau Desktop user connecting to a data source that then connects to a live database. It could also be a Tableau Server user who’s accessing a view that connects to a live database. Refer to Data Security on page 958 to learn more about your options.

- **Database drivers:** If the person who created and published the data source in Tableau Desktop needed to install additional database drivers, you may need to install them on Tableau Server as well. If you are running a distributed installation of Tableau Server where, for example, the data server process is running on a worker server, any required database drivers must be installed there as well as on the primary server. Other processes require drivers as well. See Database Drivers on page 702 for more information.

Data Source Error Messages

Here are some errors that workbook authors and other users may encounter as they work with data sources and views:

**Permission to access this Tableau Server data source denied:** Connecting to a data source requires the Connect permission. See Manage Permissions on page 834 and Set Permissions for a Data Source on page 851 for more information.

**Data source not found:** Someone working with a view may see this error if a data source is removed from Tableau Server or if their Connect to Data page needs to be updated. To update the Connect to Data page in Tableau Desktop, click the Refresh icon:
Unable to connect to this Tableau Server data source: This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct settings.

Unable to list Tableau Server data sources: This error may occur if a user is trying to access Tableau Server data sources and there are connectivity issues between Tableau Server and Tableau Desktop.

Can’t connect with a cube data source: To use a published multidimensional (cube) data source, you must download the data source and use it in Tableau Desktop. Verify that you have the Download/Web Save As permission for the data source. For more information about cubes in Tableau, see Cube Data Sources on page 893.

About Extracts and Schedules
Tableau Desktop authors can create data extracts, which are copies or subsets of data from the original data sources. Workbooks that use data extracts are generally faster than those that use live database connections because the extracted data is imported into the Tableau data engine. Extracts can also increase functionality. After an author publishes a workbook or a data source with an extract, the extract resides on Tableau Server.

Refreshing extracts on Tableau Server
You can use Tableau Server to refresh extracts on a schedule. Both server and site administrators can create, change, and reassign schedules. However, only a server administrator can enable scheduling. Any scheduling changes made in Tableau Server are reflected in the Schedule dialog box in Tableau Desktop when the workbook or data source is published again.

Schedules that you create have the following options:

Priority
The priority is a number which determines the order in which refresh tasks are run, where 0 is the highest priority and 100 is the lowest priority. The priority is set to 50 by default.

Execution mode
The execution mode determines how schedules are run by the Tableau Server backgrounder processes and can be set to parallel or serial. When you run a schedule in parallel, it runs on all available backgrounder processes, even if the schedule only contains one refresh task. When you run a schedule serially, it only runs on one backgrounder process. By default, the execution mode is set to parallel so that refresh tasks finish as quickly as possible. However, you may want to set the execution mode to serial if you have a very large schedule that is preventing other schedules from running because it is using all the available backgrounder processes.

**Frequency.**

The frequency determines how often a schedule is run. You can set the frequency to hourly, daily, weekly, or monthly. To run a schedule immediately, select the schedule and click **Actions > Run Now.**

**Note:** You can also refresh extracts from the command line using the `tabcmd refreshextracts` command. For more information, see `tabcmd Commands` on page 1319.

**Refreshing extracts from Tableau Desktop**

- **At publish time:** When an author publishes a workbook or data source that uses an extract, that author can assign it to a recurring refresh schedule on Tableau Server. The refresh can be a full refresh or an incremental refresh. Incremental refreshes reference a column in the extract that has a data type of date, date/time, or integer; such as a timestamp. Tableau uses this column to identify new rows that need to be added to your extract. See **Refreshing Extracts** and **Schedules** in the Tableau Desktop help for more information.

- **User interface:** You can use the **Refresh from Source, Add Data From File,** and **Add Data From Data Source** options in Tableau Desktop to upload an addition to or refresh an extract on Tableau Server. You may want to do this if Tableau Server doesn't have sufficient credentials to refresh data from the original data source. See **Updating Extracts on Tableau Server** in the Tableau Desktop online help for details on how to upload.

- **Data Extract command line utility:** The Data Extract command line utility installs with Tableau Desktop. You can use it to upload an addition to an extract on Tableau Server or refresh it. See **Tableau Data Extract Command Line Utility** in the Tableau Desktop online help for more information on how to upload.

See also

**Enable Extract Refresh Scheduling and Failure Notification** on the next page

**Automate Refresh Tasks** on page 922.
Manage Refresh Tasks

Administrators can change the priority or schedule of a scheduled refresh, a refresh manually, or delete schedules. You can do this on the Tasks page:

1. Sign in to the site that has the schedules you want to manage, and then click Tasks.
2. On the Tasks page, do any of the following:
   - Click Change Schedule, and select a new schedule from the list.
   - Select Run Now to initiate the refresh manually.
     
     **Note:** If an extract does not have a scheduled refresh, you can refresh it on demand from the Data Connections page.
   - Select Change Priority, and enter a new number between 1 and 100 to move the extract up or down in the priority list.
   - Select Delete to completely remove the schedule for the selected data sources.

See also

Enable Extract Refresh Scheduling and Failure Notification below

Enable Extract Refresh Scheduling and Failure Notification

Before your publishers can schedule extract refreshes, you must enable scheduling on the server.

While you’re enabling scheduling, you can decide whether also to enable sending email to data source or workbook owners when extract refreshes do not complete successfully. You can read more about these emails below. When you enable refresh failure notification, users can opt out individually by changing their account settings.

1. Sign in as a server or site administrator, and select Settings.
2. On the General page, do the following:
• To enable refresh failure notification, under Email Notification, select Send email to data source and workbook owners when scheduled refreshes fail.
• To enable scheduling, under Embedded Credentials, select both check boxes to allow publishers to embed credentials and schedule extract refreshing.

Automatic refresh schedules require direct access to the data, which you allow by embedding credentials in the connection.

Note: On a multi-site server, email notifications are a site setting and embedded credentials are a server setting.

Managing schedules from the server

It might be more appropriate in your organization to manage embedded credentials and refresh schedules centrally from the server. If you do that, you might clear the check boxes in the Embedded Credentials section described in the steps above, so that Tableau Desktop publishers do not see schedule options during publishing.

Managing schedules centrally allows you to distribute the tasks and run them when most people are offline. It also allows for more control over which credentials are embedded in which connections.

For more guidelines for managing schedules and refreshes on the server, see Provide access to data sources and Keep data fresh in Everybody’s Install Guide.
How refresh failure emails work

The email notification for a failed extract refresh lists the extract name and location on the server, gives the time of last successful refresh, the number of consecutive times the refresh has failed, and suggests the reason for the failure and possible solution.

After five consecutive failures, the refresh schedule is suspended until you or the data owner takes an action to address the cause of the failure, such as updating database credentials or a path to the original data file.

How the last successful refresh date is determined

The last successful refresh date and time are shown when that last refresh occurred within a number of days. By default it is 14 days, and this value is set in `wgserver.alerts.observed_days`. If the number of days since the last successful refresh exceeds the number specified in this setting, the message in the email shows "not in the last N days."

Create or Modify a Schedule

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

To create a new schedule

1. In a site, click **Schedules**.

2. Click **New Schedule**.
3. Specify a descriptive **Name** for the schedule. For example, End of week.

4. Select a **Task type** the schedule will handle—either refreshing extracts or delivering subscriptions.

5. Optionally you can define a **Default Priority** from 1 to 100, where 1 is the highest priority. This is the priority that will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the higher priority runs first. See **Manage Refresh Tasks** on page 911 to learn more about modifying a task’s priority.

6. Choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available backgrounder processes so that they can complete faster. For more information, see **About Extracts and Schedules** on page 909.

7. Finish defining the schedule. You can define an hourly, daily, weekly, or monthly schedule.
8. Click **Create**.

To modify an existing schedule

1. Navigate to the Schedules page.
2. Select an existing schedule, click the Actions drop-down arrow, and then select **Edit Settings**.

3. Finish editing the schedule, and click **Save**.
Add a Workbook or Data Source to a Schedule

You can set scheduled refresh tasks for published data source extracts and published workbooks that connect to data extracts.

1. When you’re signed in to Tableau Server, display Content > Data Sources or Content > Workbooks, depending on the type of content you want to refresh.

2. Select the check box for the data source or workbook you want to refresh, and then select Actions > Extract Refresh.

3. In the Refresh Extracts dialog, select Schedule a Refresh, and complete the following steps:
   - Select the schedule you want.
   - If available, specify whether you want a full or incremental refresh.
     By default, and if this option is not shown, a full refresh is run. Incremental refresh is available only if you configured it in Tableau Desktop before publishing the extract. For information, see Refreshing Extracts in the Tableau Desktop Help.
   - Click the Schedule Refresh button.

Note: If you want to add a new schedule, you can do so on the Schedules page.
Quick Start: Manage Incremental Extracts

When you publish a workbook that has an incremental extract, you can associate it with up to two refresh tasks that Tableau Server will handle for you: An incremental refresh of the extract and a full refresh. After you publish the workbook, you or a Tableau Server administrator can modify any tasks that are associated with the workbook. You can also delete tasks or add more.

1 Publish and Assign a Schedule

In Tableau Desktop, after you create a workbook that uses an extract, go to Server > Publish Workbook, and click Scheduling & Authentication. Next, choose schedules for your refreshes and click OK.

After you publish in Tableau Desktop and choose your refresh schedules, Tableau Server handles the refresh tasks for you.

2 Select the Workbook

To modify a workbook’s scheduled task, sign in to Tableau Server and on the Workbooks page, select the workbook:
Access the Refresh Schedule

Click **Refresh Schedule**.

Select the check box for the refresh task you want to modify:
4 Edit, Delete, or Add More Tasks

Select the action you want to take—for example, Change Schedule—and make your selection. You can also delete the task, change its priority, or add more refresh tasks.

Quick Start: Refresh Extracts on a Schedule

For published workbooks that connect to data extracts, you can set up the server to refresh the data on a recurring schedule, so all workbooks connected to them always show the most up-to-date data.

To schedule refreshes you need to have administrator or data owner permissions.

1 Set up a schedule on the server

Sign in to the server, go to the Schedules page, and click New Schedule.
Tableau provides a few refresh schedules. You create additional schedules you need.

2 Enable scheduled extract refreshes and failure emails

As a server or site administrator, you can enable schedules, as well as email notification when extract refreshes fail.

Select **Settings**, and then go to the **General** page.

- Under Email Notification, select **Send email to data source and workbook owners when scheduled refreshes fail**.
- Under **Embedded Credentials**, select both check boxes to allow publishers to embed credentials and schedule extract refreshes.
3 Publish a workbook with an extract

In Tableau Desktop, select Server > Publish Workbook. Sign in to the server if you’re not already. In the Publish Workbook to Tableau Server dialog box, click Schedules & Authentication. Under Extract Schedule, select the schedule from the list.

If the original data requires authentication, you will also need to select how you want people to access it.

4 Monitor refresh performance

You can monitor scheduled tasks by viewing Background Tasks for Extracts on the Status page.
Automate Refresh Tasks

You can associate extract refresh tasks with schedules in Tableau Server to automate refreshing data extracts. You can also automate extract refreshes using tabcmd, a command line utility that comes with Tableau Server and can be installed on a separate computer from Tableau Server. In particular, you can use the refreshextracts command in combination with other commands in your own script. For example:

```
tabcmd login - http://mytabserver -u jsmith -p P@ssw0rd!
refreshextracts --datasource salesq4
```

Handle Extract Refresh Alerts

When Tableau Server cannot complete a scheduled refresh, an alert appears to indicate that the refresh has failed. If a scheduled refresh fails five consecutive times, Tableau Server suspends the refresh. When a refresh is suspended, Tableau Server does not try to run it again until someone takes an action that attempts to correct the cause of the failure.

**Note:** The number of consecutive failures for a refresh is set to five by default, but can be changed by a Tableau Server administrator, using the `tabadmin set backgrounder.failure_threshold_for_run_prevention` command. For more information, see `tabadmin set options` on page 1294.

You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn’t be refreshed
- The author of a workbook that connects to a data source that couldn’t be refreshed

When you open the Alerts menu you can see more information about the refresh failure(s):
When a Data source is listed as Embedded it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the Data source (for example, Data source: sales_data), it means that the data source is a Tableau Server data source. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon:

<table>
<thead>
<tr>
<th>Data</th>
<th>Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>Sales by Region</td>
</tr>
</tbody>
</table>

Resolving Extract Refresh Problems

To resolve refresh issues, you can take any of these actions, based on the cause indicated in the alert:

- **Errors related to access token validation or user credentials**
  You can resolve some extract refresh problems by clicking the Connection Details in the alert. Select the check box next to the problematic data source, click Actions > Edit Connection, and then enter the missing information. Click Save when you're done. After you update the connection information, Tableau Server restarts the refresh schedule.
If you originally embedded the credentials or other data connection information when you published the workbook or data source from Tableau Desktop, you can also republish the workbook or data source. As part of the publishing process, you can choose to set a new refresh schedule. If you don't choose a new schedule, Tableau Server restarts the existing schedule.

- **Errors that indicate the database was unreachable**
  Confirm that the database is online and that you can sign in to access the data. You can use the **Try again** link in the alert to restart the refresh schedule.

If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

**Tip:** Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections

### Background Task Prioritization

**Note:** This topic only covers prioritization of background tasks for extract refreshes and schedules.

Scheduled extract refreshes and subscriptions are run in this order:

1. Any task that is already in process is completed first.
2. Any task that is manually **Run Now** will start when the next backgrounder process becomes available.
3. Tasks with the highest priority (the lowest number) start next, independent of how long they have been in the queue. For example, a task with a priority of 20 will run before a task with a priority of 50, even if the second task has been waiting longer.
4. Tasks with the same priority are executed in the order they were added to the queue. The first task added to the queue will be started first and the second task added will be started next.
5. When multiple tasks with the same priority are scheduled to run at the same time, they are started in the order they were created or enabled. There is no distinction between extract refreshes and email subscriptions.

The following limitations also impact when scheduled tasks run:

- Tableau Server can only run as many concurrent tasks as there are backgrounder processes configured.
- Separate extract refreshes for the same data cannot run at the same time.
- Tasks associated with a schedule that uses serial execution must run one at a time.
Manage Subscriptions

A subscription is a regularly scheduled email delivery of a Tableau Server view or workbook to subscribed users. When subscribers click the snapshot of the view or workbook in their email, it opens on Tableau Server.

Administrators, project leaders, and content owners have the option to subscribe other users to workbooks and views. For more information, see Subscribe others to a view.

To view information about each subscription, such as the subscriber’s email address and name, the name of the view, and the delivery schedule, click Tasks > Subscriptions.

Requirements

For Tableau Server users to receive subscriptions, the following things need to be in place:

- **Email settings configuration**: As the system administrator, you configure the basic SMTP server settings for subscriptions on the Alerts and Subscriptions tab in the Configuration dialog box, which is available during Setup. This is the “from account” Tableau Server uses to email subscriptions to server users. You can access this tab after Setup as well. See Reconfigure the Server on page 641 and To enable email subscriptions on page 619 for steps.

- **Credentials embedded or not required**: From Tableau Server’s perspective, a subscription includes a workbook, data, and a schedule. To deliver the data piece, Tableau Server needs to be able to access the data with no end-user involvement. This can be accomplished by using either a workbook with embedded database credentials, a Tableau Server data source, or by using data that doesn't require credentials, such as a file that’s included with the workbook at publish time. Workbooks that prompt for credentials for live database connections can’t be subscribed to.

- **User requirements**: If a user can see a view or workbook on Tableau Server and it has the subscription icon () in the upper right corner, he or she can subscribe to it. The ability to see a view or workbook is controlled by the View permission. A user must also have an email address. If Tableau Server doesn’t already have an email address for a subscribing user, it prompts for one at subscription sign-up time. Users can change their delivery options, unsubscribe, or update their email address on their account settings page.

- **Trusted authentication**: If you are using a restricted ticket (the default) to render an embedded view, subscriptions are disabled.

Additional subscription settings

As long as subscriptions are configured on the Alerts and Subscriptions tab in the and Tableau Server is using its default settings, server users can subscribe to the views and workbooks they see. To prevent users from subscribing or to customize their subscription experience, here’s where to go:
- Site Settings page (Site > Settings): By default, subscriptions are enabled for every site, but you can use the Site Settings page to disable subscriptions on a per-site basis or to customize it. For example, you can enter a custom Email From Address for subscriptions instead of the one you specified in the Configuration dialog box. You can also create your own footer for the subscription emails your users receive.

- Schedules page: Your users will need at least one subscription schedule to choose when they subscribe. Tableau provides two by default. As the server administrator, you can create additional schedules or remove the default ones. See Create or Modify a Schedule on page 913 for details.

- Subscriptions page (Tasks > Subscriptions): The Subscriptions page lists all the subscriptions on the server or, if you’re a site administrator, on the site. System administrators can use this page to change a server user’s subscription schedule or delete their subscription. See the topics below for details.

For steps on how to test whether you’ve configured subscriptions correctly, see Test your subscription configuration below. If you’re experiencing an issue with subscriptions, see Troubleshoot Subscriptions on page 1202.

Delete a subscription

1. In a site, click Tasks, and then click Subscriptions.
2. Select the subscription you want to remove, and then select Actions > Unsubscribe.

Edit a subscription schedule

1. In a site, click Tasks, and then click Subscriptions.
2. Select the subscription you want to update, and then select Actions > Change Schedule.

Test your subscription configuration

As the administrator, use the following steps to test if you’ve correctly configured subscriptions.

1. Subscribe to a view.
2. In a site, click Schedules. On the Schedules page, select the schedule that contains your subscription.
3. Select Actions > Run Now.
4. The view will be sent to your email address and should arrive within the next 10 minutes.

Quick Start: Set Up Subscriptions

When Tableau Server users subscribe to a workbook or view, they can see the latest updates without having to sign into Tableau Server—a snapshot of the view is delivered to their email on scheduled basis. Administrators, project leaders, and content owners have the option to
subscribe other users to workbooks and views. For more information, see Subscribe to Views on page 930.

As server administrator, you determine whether subscriptions are enabled for a site, and you create the schedules that are available to users.

1 Configure the mail server

Stop the server and then open the Tableau Server Configuration utility. To stop, start, or configure the server: click Start > All Programs > Tableau Server. Click the Alerts and Subscriptions tab and then select Enable users to receive emails for subscriptions to views.

Enter a user name and password only if your SMTP server requires it.

Next, click the SMTP Setup tab and then enter the name of your SMTP server and port number. Enter an email account for Send email from. For Tableau Server URL, enter http:// and the server computer name, such as http://myserver.myco.com. Click OK to save your changes, and then start the server.
2 Enable subscriptions in the site

As server administrator, you decide which sites allow subscriptions. In a site, click **Settings**, and then select **Allow users to subscribe to workbooks and views**. To let content owners subscribe other users to their content, select **Allow content owners to subscribe other users to workbooks and sheets**. When you're done, click **Save**.

3 Set up a test subscription

In a site, click **Content > Views** or **Workbooks**. Open a view, and then click **Subscribe** in the view toolbar. Enter the subject for the email message in the **Subject** field.
Select a subscription schedule. Click **Subscribe me** to subscribe yourself. Click **Add** to enter the user names to subscribe other people. When you are done, click **OK**, and then click **Subscribe**.

Administrators manage the subscription schedules that are available for subscriptions. For more information, see Create or Modify a Schedule.

**4 Test the schedule**

To test the subscription, click **Schedules**. Select the schedule that you used for the subscription, and then click **Actions > Run Now**.
An email with a snapshot of the view you subscribed to will be sent to you.

**Subscribe to Views**

When you open a view in Tableau Server and a subscription icon (●) is available in the upper-right corner, you can subscribe to that view or to all of the views in the workbook.

**Note:** Server administrators determine whether subscriptions are enabled for a site, and they create the subscription schedules that are available to users.

When you subscribe to a view or to all of the views in a workbook, a snapshot of that content is automatically delivered to you periodically via email. You don't need sign in to Tableau Server to see it.

If you are the owner of the workbook, you can also subscribe other users to your workbook and its views. If you're a project leader, you can subscribe other users to workbooks and views in your projects. If you're an administrator, you can subscribe users to any workbooks or view in a site.

You can also choose to unsubscribe from views you no longer want to receive.
Subscribe yourself to a view

1. In a site, click **Views** or **Workbooks**.

2. Open a view, or open a workbook and then open one of its views.

3. Click **Subscribe** in the toolbar.
If you don't own the workbook, you will not see the **Subscribe me** and **Subscribe Others** options.

4. If your Tableau Server account doesn't already have an email address, enter your email address.

You can change the email address that a view is sent to. For details, search for “Change Your Email Address” in the Tableau Server Help.

5. To subscribe to the current view, click **This View**. To subscribe to all views in the workbook, click **Entire Workbook**.

6. Pick a schedule, enter a subject line for your email, select **Subscribe me**, and then click the **Subscribe** button.

**Note**: If you don't own the workbook, you will not see the **Subscribe me** or **Subscribe Others** options. You will only need to click the **Subscribe** button.

When you receive the subscription by email, click the snapshot of the view to open it in Tableau Server.

**Note**: If a dashboard size is set to **Automatic**, the image included in the subscription email is fixed at 800 pixels by 600 pixels.

**Subscribe others to a view**

If you are the owner of the workbook, you can subscribe other users to your workbook and its views. If you're a project leader, you can subscribe other users to workbooks and views in your
projects. If you're an administrator, you can subscribe users to any workbooks or view in a site. If a user doesn't have permission to view the content, their subscription will not be saved.

1. In a site, click **Views** or **Workbooks**.
2. Open a view, or open a workbook and then open one of the views.
3. Click **Subscribe**.
4. Select the current view (**This View**), or to include all views in the workbook, select **Entire Workbook**. Pick a schedule and enter a subject line for the email that users will receive.
5. Under **Subscribers Others**, click **Add**.

![Subscribe screen capture](image)

Add Subscribers

- Jane Johnson
- agar
- Ashley Garcia

[Cancel] [OK]
For each user you want to subscribe, type the initial letters of each name, and then select the name from the results list. When you are done adding subscribers, click OK, and then click the Subscribe button.

**Note:** Subscriptions can't be added for subscribers who do not have email addresses in their Tableau Server account. Also, if a user doesn't have permission to view the content, their subscription will not be saved.

6. To change the subscription schedule or email subject line for a subscriber, or to unsubscribe a user, click Manage under Current Subscribers.
Unsubscribe yourself from a view

1. Open your account settings on Tableau Server in one of the following ways:
   - Click the Manage my subscriptions link at the bottom of a subscription email.

2. Click Subscriptions.

3. Select the check box next to the view you want to unsubscribe from, click Actions, and
then click **Unsubscribe**.

You can also change your subscriptions in your users settings page, such as selecting a different schedule or changing the email subject line. For more information, search for “Manage Your Subscription Settings” in the Tableau Server Help.

**Troubleshoot Subscriptions**

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials:** Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.

- **Database temporarily down:** If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.

- **Background process timeout:** By default, the background process that handles subscriptions times out after 30 minutes. In the majority of cases, this is plenty of time. However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the **Background Tasks for Non Extracts** on page 1105 admin view to see if that's the case. To increase the timeout threshold, use the tabadmin option `subscriptions.timeout`.

**Can't subscribe**

If you can see a view on Tableau Server and it has a subscription icon () in the upper right corner, you can subscribe to it.

Two things need to be in place for you to subscribe to a view: Tableau Server needs to be correctly configured (described in **Manage Subscriptions** on page 925) and the view you’re subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn’t being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the **Tableau Desktop help** for details).
No subscription icon

It's possible to see a view on Tableau Server but be unable to subscribe to it. This happens for views with live database connections, where you’re prompted for your database credentials when you first click the view. A subscription includes a view (or workbook), data, and a schedule. To deliver the data piece, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don’t require credentials or have them embedded.

You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is configured for trusted authentication. See Subscription Requirements for more information.

Receiving invalid or "broken" subscriptions

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don’t work, or receiving subscriptions even though they’ve unsubscribed from the view or workbook.

Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")

You may see the above error in Windows Event Viewer if subscriptions appear to be sent (according to the Background Tasks for Extracts on page 1103 admin view), yet subscriptions aren’t arriving, and your SMTP server is using encrypted (SSL) sessions. Subscriptions are only supported for unencrypted SMTP connections. The solution is to use an unencrypted SMTP server.

Custom scripts not working after upgrade to 8.1

To support better session management, starting with version 8.1, a hash tag (#) was added to the end of view URLs. If you had custom subscriptions scripting that generated views as PDFs or PNGs you may need to update your scripts to allow for the hash tag.

For example, prior to version 8.1, view URLs use this syntax: http://tableauserver/views/SuperStore/sheet1. To generate a view as a PNG, .png could be added to the end of the URL. For example, http://tableauserver/views/SuperStore/sheet1.png.

In versions 8.1, 8.2, or 8.3, view URLs use this syntax: http://tableauserver/views/SuperStore/sheet1#1. To generate a PNG, add .png before the hash tag. For example: http://tableauserver/views/SuperStore/sheet1.png#1
Custom scripts not working after upgrade to 9.0

In version 9.0, the session ID at the end of server URLs is indicated by an "iid" parameter, :iid=<n>. For example, http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1. This parameter replaces the hash tag "#<n>" used for the session ID in 8.x versions of Tableau Server.

If you use custom subscriptions scripts that generate views as PDFs or PNGs, you may need to update your scripts by removing the hash tag and number (#<n>), and inserting the ?:iid= session ID parameter before the number.

Starting in version 9.0, view URLs use this syntax:

To generate a PNG in version 9.0 and later, add .png before the session ID:
http://tableauserver/views/SuperStore/sheet1.png?:iid=2

Maintain a History of Revisions

Revision history allows you to keep copies of content (workbooks and data sources) that have been changed. When revision history is enabled, each time a user saves updates to the content, either from Tableau Desktop or on Tableau Server, Tableau Server creates a new version of that content and stores it with older versions. You and authorized users can view and restore older versions of the content. By enabling revision history, you give users (and yourself) the confidence to experiment with the content, knowing that older versions are always available.

When revision history is enabled on a site and users publish or save workbooks, or publish data sources, a revision of each workbook and data source is saved, and users with the necessary permissions can access revision history.

- **Workbooks.** Users can preview, delete, or download previous versions. Published workbooks can be restored online, in Tableau Server, or can optionally be downloaded and then republished. For more information, see Manage Workbook Revisions on page 945.

- **Data Sources.** Users can delete or download previous versions. To restore a previous version of a published data source, users view the revision history for the data source in Tableau Server, download a revision, and then republish the data source to the same location, using the same name. For more information, see Manage Data Source Revisions on page 950

Required permissions

To access revision history, a user must have a site role of Publisher, plus the following permissions:
• Project: **View** and **Save**
• Workbooks in the project: **View, Save, and Download Workbook/Save As**
• Data sources in the project: **View, Save, and Download Data Source**

**Administrator control of revision history**

Server administrators can enable revision history for all workbooks and data sources in a site, on a per-site basis. Revision history is enabled by default, and the default number of revisions saved in history for each resource is 25.

Server administrators can set a limit the number of versions stored in revision history, and can also clear all revisions for every workbook and data source on a site. The most recent revision of each published workbook and data source is always retained.

When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if you set the limit to 15, the 15 most recent versions of the workbook or data source are saved.

**More about revision history**

• If a different author publishes over a workbook or data source with the same name, the most recent author becomes the owner of the content and can see its entire revision history.

• Workbooks and data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook or data source.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

• When a workbook or data source is deleted from a site, all previous versions are also deleted.

• If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

**Security for previewing and restoring workbooks**

When users click **Restore** or **Preview** for workbook revisions, user passwords are exchanged between the Tableau Server browser client and the server back end. Tableau Server encrypts these passwords using public/private key encryption, but to ensure these public keys are provided by Tableau Server, you must configure Tableau Server to use SSL (HTTPS). For more information, see **SSL on page 969** in the Server Administrator Guide.
**About restoring workbooks that require credentials**

When you restore a workbook that uses a live connection and prompts for a username and password, you have the option to embed the credentials for the connection. If the workbook uses a data source with multiple connections, you may need to provide a user name and password for each connection that prompts for credentials.

**Extracts with embedded credentials and scheduled refreshes**

When you restore an extract that uses embedded credentials and scheduled extract refreshes, you will need to edit the data source connection and provide your credentials as part of the restore process. This ensures that the workbook has the credentials it needs when the refresh task runs.

1. Restore the workbook that uses embedded credentials and has scheduled extract refreshes.
2. Go to the Data Source page for data source used by the workbook.
3. Click the **Connections** tab.
4. For each connection in the data source, select the connection and click **Actions > Edit Connection**.
5. Enter (or keep) the connection information, and then click **Save**.

**Workbooks with OAuth connections**

When you restore a workbook that uses an OAuth connection, you cannot preview the workbook. If you are not able to restore the workbook in Tableau Server, download the workbook and then republish it to make it the current revision.

**Quick Start: Keep Content Revisions**

Every time you publish a workbook or data source to Tableau Server, Tableau can save a version of that content in its revision history. If you want to revert to a previous version, you can go to the workbook or data source in Tableau Server, view its revision history, and restore that version.

*Note:* A server administrator must enable **Revision History** in **Site Settings** to make this feature available.

1. **Publish your content**

In Tableau Desktop, click **Server > Publish Workbook** or **Server > Publish Data Source**. Make changes to the workbook or data source, and then publish it again to the same project, with the same name. Workbook revisions are also saved when a content owner edits and save a workbook in a project on Tableau Server.
Because your content has the same name, you need to confirm that you want to overwrite the workbook or data source when you publish the workbook or data source.

2 View revision history

Sign in to Tableau Server. Select the workbook or data source, and then in the actions menu (.), click Revision History.
Workbooks include the option to preview past versions.

3 Restore a previous version of a workbook

In the revision history for the workbook, select the revision, and then click Restore.
The restored version becomes the current version.

4 Restore a previous version of a data source

Select and then download the data source. Open the downloaded file in Tableau Desktop, and then republish it with the same name, to the same location in Tableau Server. This makes it the current revision of that data source.
More about revision history

- To access revision history, a user must have a site role of **Publisher**, plus the following permissions:
  - In the project: **View** and **Save**
  - Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**
  - Data sources in the project: **View**, **Save**, and **Download Data Source**

- Workbooks and data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you might need to make changes in the downloaded workbook or data source.

- You can delete specific revisions from revision history. Server administrators can clear all revisions for every workbook or data source in a site. The most recent revision of each published workbook and data source is always retained.

- When a workbook or data source is deleted from a site, all previous revisions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the data source are saved.

- If revision history has been turned on and then turned off, saved revisions are still retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

- Published workbooks can be previewed and restored online, in Tableau Server, or can optionally be downloaded, opened in Tableau Desktop, and then republished to the same location, using the same name. For more information, see **Manage Workbook Revisions** on the next page.

  Published data sources must be downloaded, opened in Tableau Desktop, and then republished to be restored. For more information, see **Manage Data Source Revisions** on page 950.

Enable and Manage Revision History

Server administrators can enable revision history for all workbooks and data sources in a site, on a per-site basis. The number of revisions saved in history can be unlimited, or history can be limited to a specific number of revisions.

Revision History is enabled by default, and the default number of revisions saved in history is 25.

**To enable revision history**

1. In a site, click **Settings**.
2. Under **Revision History**, select **Save a history of revisions**.
3. Click **Save**.

To set a limit on the number of revisions saved in history

1. In a site, click **Settings**.
2. Under **Revision History**, enter the maximum number of revisions to be saved.
3. Click **Save**.

When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if you set the limit to 15, the 15 most recent versions of the workbook or data source are saved.

**Note**: When you change the limit number, a resource’s revision history list may not update immediately to reflect the change. A background cleanup process must run first.

To clear all revisions

Server administrators can delete all previous revisions of workbooks from the site. The most recent version of each published workbook and data source is always retained.

1. In a site, click **Settings**.
2. Under **Revision History**, click **Clear Revision History**.
3. Click **Save**.

**Manage Workbook Revisions**

When revision history is enabled on your site and you publish a workbook from Tableau Desktop or save a workbook on Tableau Server, Tableau Server saves a version of that content in its revision history. Each time you publish or save that workbook again in the same location, with the same name, another revision is saved.

You can restore previous revisions of workbooks online, in Tableau Server. Optionally, you can download a revision, open it in Tableau Desktop, and then republish the workbook to the same location in Tableau Server, using the same name.

**Permissions for revision history**

To access revision history, you must have a site role of **Publisher**, plus the following permissions:

- In the project: **View** and **Save**
- Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**

**View revision history**
In Tableau Server, select a workbook, and then click Revision History in the actions menu ( . . ).

Preview a revision of a workbook

1. Select a workbook, and then click Revision History in the actions menu ( . . ).
2. In the revision history, click Preview in the actions menu ( . . ).
If a preview is available, it is displayed from Tableau Server in a new tab in the browser.

3. Click **Close Preview** in the preview page to return to the **Revision History** dialog box.

**Note:** If a workbook cannot be previewed online, you can download the workbook, and open it in Tableau Desktop to preview it.

**Restore a revision**

1. Select a workbook, and then click **Revision History** in the actions menu (・・・).
2. In the revision history, select a revision, and then click **Restore**.

The restored version becomes the current version.
About restoring workbooks that require credentials

When you restore a workbook that uses a live connection and prompts for a user name and password, you have the option to embed the credentials for the connection. If the workbook uses a data source with multiple connections, you might need to provide a user name and password for each connection that prompts for credentials.

Extracts with embedded credentials and scheduled refreshes

When you restore an extract that uses embedded credentials and scheduled extract refreshes, you will need to edit the data connection and provide your credentials as part of the restore process. This ensures that the workbook has the credentials it needs to complete the refresh successfully.

1. Restore the workbook that uses embedded credentials and has scheduled extract refreshes.
2. Go to the Data Source page for data source used by the workbook.
3. Click the Connections tab.
4. For each connection in the data source, select the connection and click Actions > Edit Connection.
5. Enter (or keep) the connection information, and then click Save.

Workbooks with OAuth connections

When you restore a workbook that uses an OAuth connection, you cannot preview the workbook. If you are not able to restore the workbook in Tableau Server, download the workbook and then republish it to make it the current revision.

Download a revision

1. Select a workbook, and then click Revision History in the actions menu (…).
2. In the revision history, click Download in the actions menu (…).
3. Open the workbook file in Tableau Desktop.

You can also republish the with the same name, in the same location in Tableau Server, to make it the most current version.

**Note:** When you publish the workbook from Tableau Desktop, because your content has the same name, you will have to confirm that you want to overwrite the workbook. Proceed by clicking **Yes**.

**Delete a revision from history**

You can delete any previous version of a workbook from its revision history. You cannot delete the most current revision except by deleting the entire workbook.

1. Select a workbook, and then click **Revision History** in the actions menu (...).
2. In the revision history, click **Delete** in the actions menu (...).
The revision history list updates to indicate a revision has been deleted.

![Revision History](image)

**Revision History**
Select a previous revision of workbook "Forecast".

<table>
<thead>
<tr>
<th>Revision Number</th>
<th>Owner</th>
<th>Publish Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision 4 (current)</td>
<td>Henry Wilson</td>
<td>Jan 25, 2016, 5:35 PM</td>
</tr>
<tr>
<td>Revision 3 (deleted)</td>
<td>Henry Wilson</td>
<td>Jan 14, 2016, 4:15 PM</td>
</tr>
<tr>
<td>Revision 2</td>
<td>Henry Wilson</td>
<td>Jan 11, 2016, 5:06 PM</td>
</tr>
<tr>
<td>Revision 1</td>
<td>Henry Wilson</td>
<td>Jan 11, 2016, 5:05 PM</td>
</tr>
</tbody>
</table>

More about revision history

- If a different author publishes over a workbook with the same name, the most recent author becomes the owner and can see its entire revision history.

- Workbooks are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook.

Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a workbook is deleted from a site, all previous versions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the workbook are saved.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

**Manage Data Source Revisions**

When revision history is enabled on your site and you publish a data source, Tableau Server saves a version of the data source in its revision history.

To restore a published data source, you download it, open it in Tableau Desktop, and then republish it to the same location in Tableau Server, with the same name.
Permissions for revision history

To access revision history, you must have a site role of Publisher, plus the following permissions:

- In the project: View and Save
- Data sources in the project: View, Save, and Download Data Source

View revision history

- Select a data source, and then click Revision History in the actions menu (\.\).

Download and restore a revision

1. Select a workbook or data source, and then click Revision History in the actions menu (\.\).
2. In the revision history, click Download in the actions menu (\.\).
3. Open the data source file in Tableau Desktop, and then republish it with the same name, to the same location in Tableau Server. The uploaded version becomes the most current version.

**Note:** When you publish from Tableau Desktop, because your content has the same name, you will have to confirm that you want to overwrite the data source. Proceed by clicking **Yes**.

**Delete a revision from history**

You can delete any previous version of a data source from its revision history. You cannot delete most current revision except by deleting the entire data source.

1. Select a data source, and then click **Revision History** in the actions menu (. . .).
2. In the revision history, click **Delete** in the actions menu (. . .).
The revision history list updates to indicate a revision has been deleted.

More about revision history

- If a different author publishes over a data source with the same name, the most recent author becomes the owner and can see its entire revision history.

- Data sources are downloaded with the latest permutation of their extract or data connection. If the data model or data connection has changed between revisions, you may need to make some changes in the downloaded workbook or data source.

  Versions of workbooks and data sources that use .xls, or .csv data are saved with an extract of that data in revision history.

  Versions of TDE files that are not refreshed extracts are saved in revision history.

- When a data source is deleted from a site, all previous versions are also deleted.

- When limits are set on revision history, the most recent set of revisions are the versions that are saved. For example, if the limit is 10, the 10 most recent versions of the data source are saved.

- If revision history has been turned on and then turned off, saved revisions are retained, but new versions will overwrite the latest version. If revision history is then turned on again, the version numbering starts from the last revision that was saved.

**Security**

This section provides information on helping to secure Tableau Server.

**Authentication**

Authentication verifies a user's identity.
Everyone who needs to access Tableau Server—whether to manage the server, or to publish, browse, or administer content—must be represented as a user in the Tableau Server identity store. The method of authentication may be performed by Tableau Server ("local authentication"), or authentication may be performed by an external process. In the latter case, you must configure Tableau Server for external authentication technologies such as Active Directory, SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Tableau Server identity store, which is managed by the repository.

Access and management permissions are implemented through site roles. Site roles define which users are administrators, and which users are content consumers and publishers on the server. For more information about administrators, site roles, groups, Guest User, and user-related administrative tasks, see Users and Site Roles for Users.

**Note:** In the context of authentication, it's important to understand that users are not authorized to access external data sources through Tableau Server by virtue of having an account on the server. In other words, in the default configuration, Tableau Server does not act as a proxy to external data sources. Such access requires additional configuration of the data source on Tableau Server or authentication at the data source when the user connects from Tableau Desktop.

**User identity in Tableau Server**

When you install Tableau Server, you must select the process that the server will use to manage user authentication: local authentication or Active Directory. Before you install Tableau Server, you should understand how these two options impact your overall authentication strategy. After you select and set the authentication process, Tableau Server will configure the various components for the authentication method that you have selected. After this configuration is complete, you cannot change the authentication method. In fact, to change this configuration, you must uninstall the server, delete the configuration on the computer, and then reinstall the server.

**Local authentication**

If the server is configured to use local authentication, then the Tableau Server identity store is used exclusively to authenticate users. When users sign-in and enter their credentials, either through Tableau Desktop, tabcmd, API, or web client, Tableau Server verifies the credentials.

To enable this scenario, you must first create an identity for each user. To create an identity, you specify a username and a password. To access or interact with content on the server, users must also be assigned a site role. User identities can be added to Tableau Server in the server UI, using tabcmd Commands, or using the REST API.

You can also create groups in Tableau Server to help manage and assign roles to large sets of related user groups (e.g., “Marketing”).

Use local authentication if any of the following are true:
Your organization does not manage users with Active Directory
You do not want to use Active Directory
You want to use OpenID for authentication and single sign-on

Active Directory

If Tableau Server is configured to use Active Directory authentication, then credentials are managed and verified by Active Directory. When a user logs onto Tableau Server from Tableau Desktop or a web client, the credentials are passed through to Active Directory, which then verifies them and sends an access token to Tableau Server. Tableau Server will then manage user access to Tableau resources based on the site roles stored in the local identity store.

In this scenario, Tableau Server will sync user and group metadata from Active Directory to the identity store. You do not have to manually add users. However, after the data is synchronized, you will need to assign site and server roles. You can assign these individually, or at the group level. Tableau Server does not synchronize any data back to Active Directory. Tableau Server manages content and server access according to the site role permission data is stored in the repository.

If you are already using Active Directory to manage users in your organization, then we recommend selecting Active Directory authentication during Tableau setup to make user provisioning and management easier. For example, by synchronizing Active Directory groups, you can set minimum site role Tableau permissions for users that are synchronized in the groups. You can synchronize specific Active Directory groups, or you can synchronize them all. For more information, see Synchronize All Active Directory Groups on the Server.

Be sure to review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

Single sign-on options for Tableau Server

Tableau Server supports several types of single sign-on (SSO) solutions. With SSO, users don’t have to explicitly sign in to Tableau Server. Instead, the credentials they’ve used to authenticate already (for example, by signing in to your corporate network) are used to authenticate them to Tableau Server, and they can skip the step of entering a username and password to access Tableau Server. With SSO, the user’s identity as established externally is mapped to a user identity defined in the Tableau Server identity store.

When you configure Tableau Server to use an SSO solution, all authentication is handled by the SSO solution. However, Tableau Server will manage user access to Tableau resources based on the site roles stored in the identity store.

Tableau Server supports these types of SSO:

- **SAML.** You can configure Tableau Server to use SAML (security assertion markup language) for SSO. With SAML, an external identity provider (IdP) authenticates the
user's credentials, and then sends a security assertion to Tableau Server that provides information about the user's identity.

You can use SAML to access Tableau Server if you have configured Active Directory or local authentication on Tableau Server. For more information, see SAML on page 1010.

- **Kerberos.** If Kerberos is enabled in your environment and if the server is configured to use Active Directory authentication, you can provide users with access to Tableau Server based on their Windows identities. You cannot use Kerberos if your Tableau Server is configured for local authentication. For more information, see Kerberos on page 987.

- **OpenID.** OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server. To use OpenID Connect on Tableau Server, the server must be configured to use local authentication. Active Directory authentication is not supported. For more information, see OpenID Connect on page 1050.

- **Trusted Authentication.** Trusted authentication lets you set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from a trusted web server, it assumes that the web server has already handled whatever authentication is necessary. Tableau Server receives the request with a redeemable token or ticket and presents the user with a personalized view which takes into consideration the user's role and permissions. For more information, see Trusted Authentication on page 1033.

- **Integrated Windows Authentication.** If you have configured Tableau Server with Active Directory authentication, you can enable automatic logon. Automatic logon uses Microsoft SSPI to sign in your users based on their Windows username and password. Users are not prompted for credentials, which creates an experience similar to single sign-on (SSO). To enable automatic logon see, Configure General Server Options on page 608.

**Related topics**

- Trusted Authentication on page 1033
- REST API: Signing In and Out (Authentication)

**Authorization**

Authorization refers to how and what users can access on Tableau Server after authentication has been verified. Authorization includes:

- What users are allowed to do with content hosted on Tableau Server, including projects, sites, workbooks, and views.
- What users are allowed to do with the data sources that are managed by Tableau Server.
• What tasks users are allowed to perform to administer Tableau Server, such as configuring server settings, running tabadmin, creating sites, and other tasks.

Authorization for these actions is managed by Tableau Server and determined by a combination of the user’s site role and permissions associated with specific entities such as workbooks and data sources.

Site Roles

Site roles are permission sets that are assigned to a user, such as System Administrator, Publisher, or Viewer. The site roles define collections of capabilities (delete, save, view, and others) that can be granted to users or groups on Tableau Server.

Site roles define who is an administrator. Administrators can be assigned at the site or server level. Site roles also determine whether non-admin users are allowed to publish to the server from Tableau Desktop. In general, site roles determine the maximum capabilities that can be granted for each non-admin user. For example, if a user’s site role is Interactor, the user cannot publish to the server, no matter what other permissions the user has, because the Interactor role denies permission to publish.

For more information about site roles, see Site Roles for Users on page 788.

Permissions

Permissions determine whether a given user is allowed or denied to perform a specific action on a specific resource.

As an administrator setting up Tableau Server, it’s important that you understand how permissions are evaluated. Understanding the Tableau permissions process will enable you to set up and configure permissions on sites, projects, and other resources so that you can control how content and data is shared, published, viewed, extracted, and imported.

Four important concepts to understand about permissions in Tableau are:

• Permissions are resource-based. Permissions are assigned to individual resources and are granted to users or groups. Permissions are evaluated for projects, workbooks, views, and data sources.

• Permissions are implicitly denied, and non-admin users must explicitly be allowed to access resources. The process by which Tableau Server determines the “allow” or “deny” permission is explained in detail in the topic, How Permissions are Evaluated on page 839.

• Permissions inheritance exists only in locked projects and in workbooks with tabbed views. When content permissions are locked to the project, its workbooks, views, and data sources will always use the default permissions in the project. In the case of workbooks saved with the option Show sheets as tabs, views will use the workbook permissions. For more information, see Manage Permissions on page 834.
In a project that is not locked, initial permissions are a one-time copy of the container item's permissions. A workbook, view, or data source will start with the default permissions, but authorized users can subsequently edit permissions on those resources. For more information on default permissions and projects, see Set Default Permissions for a Project, and its Workbooks and Data Sources on page 861.

Tableau Server provides a flexible permissions infrastructure that allows you to manage access to all content for countless scenarios. See Control Access to Published Content on page 833 for more detailed information.

Data Access and External Authorization

There are scenarios where Tableau Server and Desktop rely on external authorization to enable access to data. For example:

- Users connecting to external data sources may require authorization that is outside the scope of Tableau Server’s authority. If users publish an external data source, then Tableau Server will manage access and capabilities of data source. But if users embed an external data source in a workbook, then it’s up to the users who publishes the workbook to determine how other users who open the workbook will authenticate with the data source.
- Running Tableau Server in an organization with Active Directory where Tableau has been configured with a Run As user account results in a dependency on Active Directory and NTFS for authorization. For example, if you configure Tableau Server to use the Run As account to impersonate users connecting to SQL, then object-level authorization is reliant on NTFS and Active Directory.
- How users authenticate and are authorized by specific database solutions may differ. As noted, Tableau Server can be configured to provide access authorization when a data source is configured, but some databases will authorize access according to their own authentication scheme.

Server Administration: Authorization for Configuring Tableau Server

One or more users must have Windows local admin permissions to configure Tableau Server and to run tabadmin set options on page 1294 commands.

Data Security

Tableau provides several ways for you to control which users can see which data. For data sources that connect to live databases, you can also control whether users are prompted to provide database credentials when they click a published view. The following three options work together to achieve different results:
- Database login account: When you create a data source that connects to a live database, you choose between authenticating to the database through Windows NT or through the database’s built-in security mechanism.

- Authentication mode: When you publish a data source or a workbook with a live database connection, you can choose an Authentication mode. Which modes are available depends on what you choose above.

- User filters: You can set filters in a workbook or data source that control which data a person sees in a published view, based on their Tableau Server login account.

The table below outlines some dependencies with the above options:

<table>
<thead>
<tr>
<th>Database Connection Options</th>
<th>Data Security Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database login account uses...</td>
<td>Is database security possible per Tableau Server user?</td>
</tr>
<tr>
<td><strong>Window NT Integrated Security (Windows Authentication)</strong></td>
<td><strong>Server Run As account</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Impersonate via server Run As account</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Viewer Credentials</strong></td>
</tr>
<tr>
<td><strong>Username and Password</strong></td>
<td><strong>Prompt user:</strong> Viewers are prompted for their database credentials when they click a view. Credentials can be</td>
</tr>
<tr>
<td><strong>Database Connection Options</strong></td>
<td><strong>Data Security Questions</strong></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Database login account uses...</td>
<td>Authentication mode</td>
</tr>
<tr>
<td><strong>saved.</strong></td>
<td><strong>Embedded credentials</strong>: The workbook or data source publisher can embed their database credentials.</td>
</tr>
<tr>
<td><strong>Impersonate via embedded password</strong>: Database credentials with impersonate permission are embedded.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

* Because it can create unexpected results, Tableau recommends that you not use this authentication mode with user filters.

User filters, the embedded credentials option and the impersonation modes have similar effects—when users click a view, they are not prompted for database credentials and they see only the data that pertains to them. However, user filters are applied in the workbook by authors, and the impersonation authentication modes rely on security policies defined by administrators in the database itself.

Some of the options described above require configuration steps that must happen during Tableau Server Setup or before you publish a workbook or data source. See the following topics for more information:
Related Topics

Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)

When you install Tableau Server or upgrade from a previous version, the installation process generates a password for Tableau Server to use internally when it accesses the Repository PostgreSQL database. To help with security, the password generated during the installation process is unique to an installation. Because the password is used only by Tableau Server for access to the Repository, the password is not accessible to server administrators or other users.

Tableau Server can also generate an SSL certificate that can be used to protect internal communications to the Repository and other server components. Using SSL for internal communications between processes is optional. For more information, see Configure Internal SSL on page 976.

Note: If you need access to the Repository (for example, to monitor activity), you can use the administrative views that are built in to the server environment or create your own custom views. For more information, see Collect Data with the Tableau Server Repository on page 1117 for details.

Regenerating the password and certificate

If you need to generate a new password and certificate for internal use, you can use the `tabadmin regenerate_internal_tokens` command. For example, if you believe your installation of Tableau Server has been compromised, you should run the `regenerate_internal_tokens` command to generate a new password and SSL certificate.

Note: The SSL certificate is used for internal communication between server components and the PostgreSQL database and is independent of any SSL certificate that you might have on the server to use for HTTPS communication between the server and clients that connect to Tableau Server.

To manually regenerate a password and SSL certificate:
1. On the Tableau Server computer, open a command prompt as an administrator and navigate to <install directory>\Program Files\Tableau\Tableau Server\9.0\bin.

2. Enter the following:

   tabadmin stop
   tabadmin regenerate_internal_tokens
   tabadmin config
   tabadmin start

   See regenerate_internal_tokens on page 1282 for more information, including optional switches to specify regeneration of password or certificate.

Network Security

There are three main network interfaces in Tableau Server:

- **Client to Tableau Server**: The client can be a web browser, Tableau Mobile, Tableau Desktop, or the tabcmd on page 1315 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, a device running Tableau Mobile, Tableau Desktop, or tabcmd commands. Communications between Tableau Server and its clients use standard HTTP requests and responses. We recommend configuring Tableau Server for HTTPS for all communications. When Tableau Server is configured for SSL, all content and communications between clients are encrypted using SSL, and the HTTPS protocol is used for requests and responses.

By default, passwords are communicated from browsers and tabcmd to Tableau Server using 1024-bit public/private key encryption. This level of encryption is not considered robust enough for secure communications. Additionally, this method, where a public key is sent to the recipient in the clear and without network layer authentication is susceptible to man-in-the-middle attacks.

To adequately secure network traffic from clients to Tableau Server, you must configure SSL with a certificate from a trusted certificate authority. See **Configure External SSL on page 972**.

Client access from the Internet

We recommend a gateway proxy server to enable secure client access from the internet to your Tableau Server. We do not recommend running Tableau Server in a DMZ or otherwise outside
your protected, internal network.

Configure a reverse proxy server, with SSL enabled, to handle all inbound traffic from the internet. In this scenario, the reverse proxy is the only external IP address (or range of addresses if multiple reverse proxies are load-balancing inbound requests) that Tableau Server will communicate with. Reverse proxies are transparent to requesting clients, thereby obfuscating Tableau Server network information and simplifying client configuration. For configuration information, see Configuring Proxies for Tableau Server on page 579.

Clickjack Protection

By default, Tableau Server has clickjack protection enabled. This helps prevent certain types of attacks in which the attacker overlays a transparent version of a page on top of an innocuous-looking page in order to lure a user into clicking links or entering information. With clickjack protection enabled, Tableau Server imposes certain restrictions on embedding views. For more information, see Clickjack Protection on the next page.

Tableau Server to your database

Tableau Server makes dynamic connections to databases to process result sets and refresh extracts. It uses native drivers to connect to databases whenever possible and relies on a generic ODBC adapter when native drivers are unavailable. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Tableau.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server’s internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Tableau Server to the Internet

In some cases, where users connect to external data sources, such as the Tableau map servers, then Tableau Server will need to connect to the internet. We recommend that you run all components of Tableau inside your protected network. Therefore, connections to the internet may require that you configure Tableau Server to use a forward proxy. See the Knowledge Base article, Configuring Proxies for Tableau Server on page 579, for more information.

Communication with the repository

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic that is exchange with the Postgres repository to and from other server components. By default, SSL is disabled for communications between server components and the repository. For more information, see Configure Internal SSL on page 976.
Server component communication in a cluster

There are two aspects to communication between Tableau Server components in a distributed server installation: trust and transmission. Each server in a Tableau cluster uses a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. Computers in the cluster running a gateway process accept requests from third parties (clients), unless they are fronted by a load balancer, in which case the load balancer receives the requests. Servers not running a gateway process only accept requests from other trusted members of the cluster. Trust is established by a whitelist of IP address, port, and protocol. If any of these are invalid, the request is ignored. All members of the cluster can communicate with each other.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server’s internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Clickjack Protection

Tableau Server includes protection against clickjack attacks. Clickjacking is a type of attack against web pages in which the attacker tries to lure users into clicking or entering content by displaying the page to attack in a transparent layer over an unrelated page. In the context of Tableau Server, an attacker might try to use a clickjack attack to capture user credentials or to get an authenticated user to change settings on your server. For more information about clickjack attacks, see Clickjacking on the Open Web Application Security Project website.

Note: Clickjack protection was available in previous versions of Tableau Server, but was disabled by default. New installations of Tableau Server 9.1 and later will always have clickjack protection on unless you explicitly disable it.

Effects of clickjack protection

When clickjack protection is enabled on Tableau Server, the behavior of pages loaded from Tableau changes in the following ways:

- Tableau Server adds the X-Frame-Options: SAMEORIGIN header to certain responses from the server. In the current versions of most browsers, this header prevents the content from being loaded into an <iframe> element, which helps prevent clickjacking attacks.
- The top-level page from Tableau Server cannot be loaded in <iframe> elements. This includes the sign-in page. One consequence is that you cannot host Tableau Server pages in an application that you create.
- Only views can be embedded.
- If an embedded view requires data source credentials, a message is displayed in the
<iframe> element with a link to open the view in a secure window where the user can safely enter credentials. Users should always verify the address of the opened window before entering credentials.

- Views can be loaded only if they include the :embed=y parameter in the query string, as in this example:

```plaintext
http://<server>/views/Sales/CommissionModel?:embed=y
```

**Note:** View URLs that contain a hash mark (#) after the server name (for example, `http://<server>/#views/Sales/CommissionModel?:embed=y`) are blocked when clickjack protection is enabled.

Disabling clickjack protection

You should leave clickjack protection enabled unless it is affecting how your users work with Tableau Server. If you want to disable clickjack protection, use the following `tabadmin` commands:

1. `tabadmin stop`
2. `tabadmin set wgserver.clickjack_defense.enabled false`
3. `tabadmin config`
4. `tabadmin start`

Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

1. **Update to the current version**

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Security Bulletins page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the Customer Portal page. For more information see Upgrade Tableau Server on page 665.

2. **Configure SSL/TLS with a valid, trusted certificate**

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can
connect top the server over a secured connection. For more information, see SSL on page 969.

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS) is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect. We recommend that you only allow external clients to connect to Tableau Server with TLS v1.2.

Specially, we recommend that you disable TLS v1 and TLS v1.1 on Tableau Server. However, before you disable a specific version of TLS, verify that the browsers that your users connect to Tableau Server with support TLS v1.2. In some cases, you may need to preserve support for TLSv1.1.

The following tabadmin command enables TLS v1.2 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol).

```
tabadmin stop
stabadmin set sslprotocols "all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
stabadmin configure
stabadmin start
```}

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See Configure Internal SSL on page 976.

5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server in the same network with your internet gateway or in a DMZ. Tableau Server must be protected by external firewall. The platform firewall, such as the Windows firewall, should be enabled to protect Tableau sever in single and multi-node deployments.

In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. By default, the Tableau installation process configures ports in the Windows firewall so that server components can communicate with each other. If you're
configuring a different firewall, or if you need to configure the Windows firewall after you’ve installed Tableau Server, see Tableau Server Ports on page 1244 to understand which ports and services Tableau Server requires.

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

Important: Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network protected by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See Configure a reverse proxy server on page 584.

6. **Restrict access to the server computer and to important directories**

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the C:\ProgramData\Tableau directory. By default, the permissions on these directories are restrictive, therefore we do not recommend changing permissions at the directory level.

7. **Update the Tableau Server Run As User account**

By default, Tableau Server runs under the predefined Network Services (NT Authority\Network Service) Windows account. Using the default account is acceptable in scenarios where Tableau Server does not need to connect to external data sources that require Windows authentication. However, if your users require access to data sources that are authenticated by Active Directory, update the Run As User to a domain account. It’s important to minimize the rights of the account that you use for the Run As User. For more information, see Run As User on page 577.

8. **Generate fresh asset keys**

Tableau Server encrypts embedded database credentials before they are stored in the repository. The credentials are encrypted with asset keys. We recommend that after you install Tableau Server, you generate new encryption keys for your deployment. To do this, use the tabadmin assetkeys on page 1259 command.

9. **Refresh server token and encryption key**

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server Setup. In addition, the encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during Setup. If your organization follows a security policy to update shared secrets and encryption keys on a regular schedule, you should include the token and key in that
process. See the tabadmin regenerate_internal_tokens on page 1282 command for more information.

10. Disable services that you’re not using
To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

REST API
The REST API interface is enabled by default. If no applications will make REST API calls to your installation of Tableau Server 9.3 (or later), disable it by using the following sequence of tabadmin commands:

```bash
tabadmin stop
tabadmin set api.server.enabled false
tabadmin configure
tabadmin start
```

You can disable REST API only on versions of Tableau Server 9.3 and later.

JMX Service
JMX is disabled by default. If it's enabled but you’re not using it, you should disable it by using the following sequence of tabadmin commands:

```bash
tabadmin stop
tabadmin set service.jmx_enabled false
tabadmin configure
tabadmin start
```

11. Verify session lifetime configuration
By default, Tableau Server does not have an absolute session timeout. This means that client sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. (The default inactivity timeout is 240 minutes.)

If your security policy requires it, you can set an absolute session timeout. Before you do that, you must enable session lifetime timeout. Use the following sequence of tabadmin commands.

```bash
tabadmin stop

# Enable session lifetime timeout

# By default, Tableau Server uses 1440 minutes
# (24 hours) as the lifetime limit

tabadmin set wgserver.session.apply_lifetime_limit true

# Set the lifetime limit to a specific value (in minutes)
# For example, to use 240 minutes (4 hours)

# Not supported in Tableau Server 9.3 or earlier

tabadmin set wgserver.session.lifetime_limit "value", where value is the number of minutes. The default is 1440, which is 24 hours.

# Set the idle limit to a specific value (in minutes)
# For example, to use 240 minutes (4 hours)

# Not supported in Tableau Server 9.3 or earlier

tabadmin set wgserver.session.idle_limit "value", where value is the number of minutes. The default is 240.
```
tabadmin configure

12. Configure a server safelist for file-based data sources

By default, Tableau Server allows authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). In this scenario, files are accessed by the Run As User on page 577 account.

To prevent unwanted access to files, we recommend that you configure safelist (sometimes referred to as "whitelist") functionality. This lets you limit Run As User access to just the directory paths where you host data files.

1. On the computer running Tableau Server, identify the directories where you will host data source files.

   Important Make sure the file paths you specify in this procedure exist on the server. If the paths do not exist when the computer starts, Tableau Server will not start.

2. Run the following tabadmin commands:

   ```
   tabadmin stop
   tabadmin set native_api.allowed_paths "path", where path is the directory to add to the safelist. Note! All subdirectories of the specified path will be added to the safelist. If you want to specify multiple paths, separate them with a semicolon, as in this example:
   tabadmin set native_api.allowed_paths "c:\datasources;c:\HR\data"
   tabadmin configure
   tabadmin start
   ```

SSL

SSL (Secure Sockets Layer) is a standard security technology that establishes an encrypted link between a web server and clients. To use SSL, you need to install an SSL certificate on Tableau Server.

Tableau Server also supports mutual (two-way) SSL as an encryption and authentication method.

You can configure Tableau Server to use SSL in the following ways:

- Use SSL for external HTTP traffic.
- Use mutual (two-way) SSL between clients (Tableau Desktop, web browsers, and
Use SSL for all HTTP traffic between internal server components and the repository.

If you are using mutual SSL, each client also needs a certificate.

**Note:** Tableau Server uses SSL only for user authentication. Tableau Server does not use SSL to handle permissions and authorization for content hosted on Tableau Server, such as workbooks.

For more information, see the following topics:

**Quick Start: Mutual (Two-Way) SSL Authentication**

To provide a secure automatic sign-in experience with Tableau across all devices, use mutual SSL. With mutual SSL, when a client (Tableau Desktop on Windows, a web browser, or `tabcmd.exe`) with a valid certificate connects to Tableau Server, Tableau Server confirms the existence of a valid client certificate and automatically signs the user in, using the user name it finds in the certificate. If the client does not have a valid SSL certificate, Tableau Server refuses the connection. To configure Tableau Server for mutual SSL, you need the following:

- **Certificate file**: A PEM-encoded x509 certificate file with the extension `.crt`.
- **Certificate key file**: An RSA or DSA key file that is not password-protected and that has a `.key` file extension.
- **Certificate CA file**: A PEM-encoded x509 certificate file with the extension `.crt`.
- **Client certificate on client devices**: Tableau Server queries the client for an SSL certificate that it trusts, before it allows a connection to Tableau Server.

**Note:** Mutual SSL is not available for Tableau Desktop on the Mac.

The certificate files should be in the `C:\Program Files\Tableau\Tableau Server\SSL` folder.

1 **Use SSL for server communication**

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the Tableau Server Configuration utility after you have installed Tableau Server. Click the `SSL` tab, and then select **Use SSL for server communication**.
Specify values for **SSL certificate file** and **SSL certificate key file**.

2 Use mutual SSL

To add mutual authentication between the server and each client and allow for automatic sign-in experience, select **Use mutual SSL and automatic login with client certificates**.

Specify the **SSL CA certificate file**. The SSL CA certificate file identifies the certificate of the Certificate Authority (for example, Verisign). For information on how to configure multiple Certificate Authorities, see **Configure External SSL on the next page**.

Click **OK** to close the Tableau Server Configuration utility, and then start Tableau Server.
**Additional options for mutual SSL**

**Fallback authentication**

When Tableau Server is configured for mutual SSL, authentication is automatic and a client must have a valid certificate. If you need a fallback option, use the `tabadmin set ssl.client_certificate_login.fallback_to_password true` command to configure Tableau Server to allow user name / password authentication. Setting this option to `true` allows Tableau Server to fall back to using user name and password for authentication if SSL certificate authentication fails.

**Username mapping**

When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate so the client can be automatically signed in. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

- **Local Authentication**—Tableau Server uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the CN (Common Name) by using the `tabadmin set ssl.client_certificate_login.mapping_strategy` command.

**Certificate Revocation List (CRL)**

You may need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly. To specify a CRL, use the `tabadmin set ssl.revocation.file` command. For more information, see `tabadmin set Commands`.

**Configure External SSL**

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the web browser and the server or Tableau Desktop and the server is protected. Steps on how to configure the server for SSL are described in the topic below; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server. If you are running a Tableau Server cluster and you want to use SSL, see **Configure SSL for a Cluster** on page 974, below, for recommendations.

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

**Note:** Be sure to use a SHA-2 (256 or 512 bit) certificate. All major browsers will display warnings when connecting to SHA-1 certificates. By the end of 2017, it's likely that most browsers will no longer connect to servers that are presenting SHA-1 certificates.

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 10.0 folder. For example:

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

   This location gives the account that's running Tableau Server the necessary permissions for the files. You may need to create this folder.

3. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 10.0 > 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 has an embedded passphrase, and is not password protected with the file extension .key.
   - **SSL certificate chain file (Optional for Tableau Server, required for Tableau Mobile and Tableau Desktop on the Mac)**—Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate. All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

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

   - Select **Use mutual SSL and automatic login with client certificates**.

   **Note:** Tableau Server does not support mutual SSL and SAML together.
In **SSL CA certificate file**, browse to the location for the certificate file. The SSL CA certificate file must be a valid PEM-encoded x509 certificate with the extension .crt.

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

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

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

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

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

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

---

**Configure SSL for a Cluster**

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

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

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

- **Configure Tableau Server for SSL:** Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. See the procedure below for details.

Configure a Server Cluster for SSL

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

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

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

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

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

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

1. Open the Tableau Server Configuration Utility by selecting Start > All Programs > Tableau Server 10.0 > Configure Tableau Server.
2. In the Tableau Server Configuration dialog box, click the SSL tab.
3. Select one of the following options:
   - **Required for all connections**
     When this option is selected, Tableau Server uses SSL for communications between the repository database and other server components. In addition, direct connections to Tableau Server (connections using the "tableau" or "readonly" users) must use SSL.
   - **Optional for direct user connections**
     This option configures Tableau Server to use SSL between the repository and other server components and supports but does not require SSL for direct connections by "tableau" or "readonly" users.
   - **Off for all connections** (the default)
     This option disables SSL for internal communications and direct connections.
4. Click OK.

For more information on downloading the public certificate for direct connections, see Configure SSL for Direct Connections below.

Configure SSL for Direct Connections

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

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

1. Generate the SSL certificate file using the `regenerate_internal_tokens` on page 1282 command.
2. Locate the SSL cert file by looking in the workgroup.yml file on the primary Tableau Server node.
The workgroup.yml file is located on the primary Tableau Server node in the \ProgramData\Tableau\Tableau Server\data\tabsvc\config folder.

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

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

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

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

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

**How Mutual SSL Authentication Works**

Mutual (or two-way) SSL authentication provides a combination of an encrypted data stream, mutual authentication of both server and client, and automatic sign-in convenience. To use mutual SSL with Tableau Server, you need an SSL certificate for Tableau Server and a certificate on each client that will connect to Tableau Server. You also need to configure Tableau Server to use mutual SSL. Tableau Server and client verify that each other has a valid certificate, and Tableau Server signs in the user automatically, based on the user name that Tableau Server finds in the client certificate.

The following image shows the sequence of events that occurs with mutual SSL.

![Sequence of events](image)

1. The user navigates to Tableau Server.
Tableau Server sends its SSL certificate to the client computer.

The client computer verifies the Tableau Server certificate.

The client computer sends its certificate to Tableau Server.

Tableau Server verifies the client certificate.

Tableau Server signs the user in using the user name from the certificate.

**Mapping a Client Certificate to a User During Mutual Authentication**

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. Tableau Server can perform client mapping using different strategies, depending on the content of the client certificates in your organization.

This topic discusses the options for how a client certificate is mapped to a user identity and explains how to change how the server performs the mapping if necessary. Note that in order for you to understand how the mapping is performed and whether you need to change the default mapping for Tableau Server, you must understand how client certificates are structured in your organization.

- **Mapping options**
- **Changing the certificate mapping**
- **Ambiguous user names in multi-domain organizations**

**Mapping options**

Tableau Server can map a client certificate to a user identity by using one of the following approaches:

- **Use Active Directory.** If Tableau Server was configured during setup to use Active Directory for user authentication, when Tableau Server receives a client certificate, Tableau Server passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any explicit user name information in the certificate is ignored.
Using the user principal name (UPN). A client certificate can also be created so that the user name is in the user principal name (UPN) field of the certificate. Tableau Server can read the UPN value and use it to map to a user in Active Directory or to a local user.

Using the common name (CN). A client certificates can also be created so that the user name is in the common name (CN) field of the certificate. Tableau Server can read the CN value and use it to map to a user in Active Directory or to a local user.

If the server is configured for Active Directory authentication, and if you're using UPN or CN mapping, the user name should be in one of these formats: username, domain\username, or username@domain. For example, the name must be asmith, example.org\asmith, or asmith@example.org.

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Changing the certificate mapping

The approach that Tableau Server uses to map a client certificate to a user identity is specified using the ssl.client_certificate_login.mapping_strategy setting. Possible values for this setting are ldap for Active Directory mapping, upn for UPN mapping, or cn for CN mapping.

When you first install and configure Tableau Server, the server makes default settings for the mappings. By default, if Tableau Server is configured to use Active Directory, the server also uses Active Directory for mapping the certificate to the user identity (ssl.client_certificate_login.mapping_strategy is set to ldap). If the server is configured to use local authentication, by default the server gets the user name value from the UPN field in the certificate (ssl.client_certificate_login.mapping_strategy is set to upn).

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the tabadmin set command to change the value of ssl.client_certificate_login.mapping_strategy. As an example, the following sequence of commands shows how to set the mapping to use the CN value:

```
tabadmin stop
tabadmin set ssl.client_certificate_login.mapping_strategy cn
tabadmin configure
tabadmin start
```

Ambiguous user names in multi-domain organizations

Under some circumstances, the user name in a UPN or CN field in the certificate can be ambiguous. This can have unexpected results when the user name is mapped to a user identity.
on the server. This can occur when all of the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.
- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains (for example, asmith@example.org and asmith@example.com).
- The user name in the UPN or CN fields of the certificate does not include the domain as part of the user name—for example, the certificate simply includes asmith.

If Tableau Server gets a user name that has no domain, the server maps the user name to an identity using the default domain. This can result in incorrectly mapping the user name.

**Important:** Incorrect mapping of the user name can result in a user being granted an identity and permissions for a different user. To avoid this issue, you should make sure that the client certificates include full user names, with the domain.

To resolve this issue, the system administrator should make sure that the user name in the user’s certificate is fully qualified with a domain name using the format asmith@example.org or example.org\asmith.

**Troubleshooting Mutual SSL Authentication**

This topic describes possible mutual (two-way) SSL authentication issues and their causes, the messages that users might see, and possible mitigation for the issues.

- The client is missing a certificate
- The client doesn't support mutual SSL authentication
- Client certificates are not published to Active Directory
- Users unexpectedly see a sign-in dialog box that displays an error message
- The user name in the UPN or CN fields is missing or invalid
- The user is signed in using unexpected user name (LDAP mapping)
- The user is signed in as incorrect user (UPN or CN mapping)

For more information about mutual SSL authentication and LDAP, UPN, and CN user mapping, see the following topics:

- **Quick Start: Mutual (Two-Way) SSL Authentication** on page 970
- **Mapping a Client Certificate to a User During Mutual Authentication** on page 978
We couldn’t find a valid client certificate. Contact your Tableau Server administrator.

The client is missing a certificate.

If the client has no client certificate, the user sees this message during authentication:

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

To resolve the issue, the user should contact the system administrator to generate a certificate for the client computer.

Invalid user name or password

The client doesn’t support mutual SSL authentication.

Versions of Tableau Desktop older than version 9.1 do not support mutual SSL authentication. If an older version of Tableau Desktop is used to connect to Tableau Server that is configured for mutual SSL authentication, the following can occur:

- If Tableau Server is configured to use fallback authentication, the client displays a sign-in dialog box and the user can enter a user name and password.
- If the server is not configured to use fallback authentication, the user sees the following message and cannot connect to the server:

  Invalid user name or password

For more information about fallback authentication, see Quick Start: Mutual (Two-Way) SSL Authentication on page 970.

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

Client certificates are not published to Active Directory.

If Tableau Server is configured to use Active Directory for authentication, and if user mapping is set to LDAP, Tableau Server sends the client certificate to Active Directory for authentication. However, if client certificates have not been published to Active Directory, authentication fails and the user sees the following message:

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

To resolve this issue, the system administrator should make sure that client certificates are published to Active Directory. Alternatively, the server should be configured to use a different user mapping (UPN or CN), and the system administrator should be sure that client certificates contain user names in the UPN or CN fields.
Users unexpectedly see a sign-in dialog box that displays an error message.

If Tableau Server is configured to use mutual SSL authentication and certificates are available for use with users’ computers, a user should not see a sign-in dialog box, because Tableau Server uses the certificate to authenticate the user. However, if the server does not recognize the user name in the certificate, the user sees a sign-in dialog box with an error message that indicates why the certificate was not used. This can occur when all of the following conditions are true:

- Fallback authentication is enabled.
- If the server is using UPN or CN mapping, the user name in the certificate's UPN or CN field is not recognized. If the server is using LDAP mapping, the certificate is not mapped to the user in Active Directory.

To resolve this issue, the system administrator should do the following, depending on how user mapping is configured on Tableau Server:

- LDAP mapping: Make sure that the certificate is linked to the user, that the certificate is available for use with the user’s computer, and that the user is configured as a Tableau Server user.
- UPN or CN mapping: Make sure that the certificate is available for the user's computer, that the user name is in the certificate's UPN or CN field, and that the user name matches the user name on Tableau Server (including domain).

We couldn’t find your user name in the client certificate. Contact your Tableau Server administrator.

Certificate does not contain a valid Tableau Server user name.

The user name in the UPN or CN fields is missing or invalid.

When Tableau Server is configured to use UPN or CN mapping, the server reads the user’s name from the UPN or CN field of the certificate and then looks up the user name in Active Directory or in the local repository on Tableau Server. (The specific field that the server reads depends on which mapping—UPN or CN—the server is configured to use.) If the field that is supposed to contain the user name has nothing in it, the user sees the following message:

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.

If a client certificate contains a user name but Active Directory and Tableau Server don't recognize the user name, the user sees the following message:

Certificate does not contain a valid Tableau Server user name.

This can occur when all of the following conditions are true:

- Tableau Server is configured to use UPN or CN mapping.
- Fallback authentication is not enabled.
The client certificate has no user name in the UPN or CN field, or the user name in the UPN or CN field does not match a user name in Active Directory or on Tableau Server.

To resolve this issue, the system administrator should make sure that the user's certificate has the correct user name in the UPN or CN fields of the certificate.

The user is signed in using an unexpected user name (LDAP mapping)

When the server is configured to use Active Directory authentication and LDAP mapping, the certificate is linked to a user in Active Directory. If the certificate contains a user name in the UPN or CN field, that user name is ignored.

If the intention is that the user should be signed in with the user name in the UPN or CN fields, the server should be configured to use UPN or CN mapping.

The user is signed in as the incorrect user (UPN or CN mapping)

Under some circumstances, the user name in a UPN or CN field in the client certificate can be ambiguous. The result is that a user is signed in to the incorrect identity.

For more information about the conditions under which this issue can occur, see Ambiguous user names in multi-domain organizations in the topic Mapping a Client Certificate to a User During Mutual Authentication on page 978.

Example: SSL Certificate - Generate a Key and CSR

**Important:** This example is intended to provide general guidance to IT professionals who are experienced with SSL requirements and configuration. The procedure described in this article is just one of many available methods you can use to generate the required files. The process described here should be treated as an example and not as a recommendation.

When you configure Tableau Server to use Secure Sockets Layer (SSL) encryption, this helps ensure that access to the server is secure and that data sent between Tableau Server and Tableau Desktop is protected.

Tableau Server uses Apache, which includes OpenSSL. You can use the OpenSSL toolkit to generate a key file and Certificate Signing Request (CSR) which can then be used to obtain a signed SSL certificate.

**Steps to generate a key and CSR**

To configure Tableau Server to use SSL, you must have an SSL certificate. To obtain the SSL certificate, complete the steps:

1. Set the OpenSSL configuration environment variable (optional).
2. Generate a key file.
3. Create a Certificate Signing Request (CSR).
4. Send the CSR to a certificate authority (CA) to obtain an SSL certificate.
5. Use the key and certificate to configure Tableau Server to use SSL.
You can find additional information on the SSL FAQ page on the Apache Software Foundation website.

**Configure a certificate for multiple domain names**

Tableau Server allows SSL for multiple domains. To set up this environment, you need to modify the OpenSSL configuration file, openssl.conf, and configure a Subject Alternative Name (SAN) certificate on Tableau Server. See For SAN certificates: modify the OpenSSL configuration file below.

**Set the OpenSSL configuration environment variable (optional)**

To avoid using the `-config` argument with every use of openssl.exe, you can use the `OPENSSL_CONF` environment variable to ensure that the correct configuration file is used and all configuration changes made in subsequent procedures in this article produce expected results (for example, you must set the environment variable to add a SAN to your certificate).

Open the Command Prompt as an administrator, and run the following command:

```bash
set OPENSSL_CONF=c:\Program Files\Tableau\Tableau Server\<version>\apache\conf\openssl.cnf
```

**Notes:**

- When setting the Open SSL configuration environment variable, do not enclose the file path with quotation marks.
- If you are using a 32-bit version of Tableau Server on a 64-bit computer, run the `set OPENSSL_CONF=c:\Program Files (x86)\Tableau\Tableau Server\<version>\apache\conf\openssl.cnf` command instead.

**Generate a key**

Generate a key file that you will use to generate a certificate signing request.

1. Open the Command Prompt as an administrator, and navigate to the Apache "bin" directory for Tableau Server. For example, run the following command:

   ```bash
   cd C:\Program Files\Tableau\Tableau Server\<version>\apache\bin
   ```

2. Run the following command to create the key file:

   ```bash
   openssl.exe genrsa -out <yourcertname>.key 4096
   ```

   **Note:** This command uses a 4096-bit length for the key. You should choose a bit length that is at least 2048 bits because communication encrypted with a shorter bit length is less secure. If a value is not provided, 512 bits is used.
Create a certificate signing request to send to a certificate authority

Use the key file you created in the procedure above to generate the certificate signing request (CSR). You send the CSR to a certificate authority (CA) to obtain a signed certificate.

Note: If you want to configure a SAN certificate to use SSL for multiple domains, first complete the steps in For SAN certificates: modify the OpenSSL configuration file below, and then return to here to generate a CSR.

1. Run the following command to create a certificate signing request (CSR) file:

   openssl.exe req -new -key yourcertname.key -out yourcertname.csr

   If you did not set the OpenSSL configuration environment variable, OPENSSL_CONF, you might see either of the following messages:

   - An error message about the config information being unable to load. In this case, retype the command above with the following parameter: -config ..\conf\openssl.cnf.
   - A warning that the /usr/local/ssl directory cannot be found. This directory does not exist on Windows, and you can simply ignore this message. The file is created successfully.

   To set an OpenSSL configuration environment variable, see Set the OpenSSL configuration environment variable (optional) section in this article.

2. When prompted, enter the required information.

   Note: For Common Name, type the Tableau Server name. The Tableau Server name is the URL that will be used to reach the Tableau Server. For example, if you reach Tableau Server by typing tableau.example.com in the address bar of your browser, then tableau.example.com is the common name. If the common name does not resolve to the server name, errors will occur when a browser or Tableau Desktop tries to connect to Tableau Server.

Send the CSR to a certificate authority to obtain an SSL certificate

Send the CSR to a commercial certificate authority (CA) to request the digital certificate. For information, see the Wikipedia article Certificate authority and any related articles that help you decide which CA to use.

Use the key and certificate to configure Tableau Server

When you have both the key and the certificate, you can configure Tableau Server to use SSL by completing the steps in Configure External SSL on page 972.
For SAN certificates: modify the OpenSSL configuration file

In a standard installation of OpenSSL, some features are disabled by default. To use SSL with multiple domain names, before you generate the CSR, complete these steps to modify the openssl.cnf file.

1. Open Windows Explorer and browse to the Apache conf folder for Tableau Server.
   
   **For example:** `C:\Program Files\Tableau\Tableau Server\<version>\apache\conf`

2. Open openssl.cnf in a text editor, and find the following line: `req_extensions = v3_req`

   This line might be commented out with a hash sign (#) at the beginning of the line.

   ```
   [ req_distinguished_name ]
   countryName = Country Name (2 letter code)
   
   # req_extensions = v3_req # The extensions to add to a certificate request
   
   # so use this option with caution!
   string_mask = nonstd
   ```

   If the line is commented out, uncomment it by removing the # and space characters from the beginning of the line.

3. Move to the `[v3_req]` section of the file. The first few lines contain the following text:

   ```
   # Extensions to add to a certificate request
   basicConstraints = CA:FALSE
   keyUsage = nonRepudiation, digitalSignature, keyEncipherment
   ```

   After the `keyUsage` line, insert the following line:

   ```
   subjectAltName = @alt_names
   ```

   If creating a self-signed SAN certificate, do the following to give the certificate permission to sign the certificate:

   a. Add the `cRLSign` and `keyCertSign` to the `keyUsage` line so it looks like the following: `keyUsage = nonRepudiation, digitalSignature, keyEncipherment, cRLSign, keyCertSign`

   b. After the `keyUsage` line, add the following line: `subjectAltName = @alt_names`

4. In the `[alt_names]` section, provide the domain names you want to use with SSL.
DNS.1 = [domain1]
DNS.2 = [domain2]
DNS.3 = [etc]

The following image shows the results highlighted, with placeholder text that you would replace with your domain names.

```plaintext
[ v3_req ]

# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = domain1
DNS.2 = domain2
DNS.3 = etc
```

5. Save and close the file.

6. Complete the steps in Create a certificate signing request to send to a certificate authority section, above.

**Additional information**

If you prefer to use a different version of OpenSSL, you can download it from Open SSL for Windows.

**Authentication and Access**

You can configure Tableau Server to integrate with a number of different third-party user authentication solutions. To configure a solution listed below, you must first specify how Tableau Server will store user and manage identities (local authentication or Active Directory) during the install process. Most of the authentication solutions require that you set Tableau Server user identity store to a specific type. Therefore, before you install Tableau Server, be sure to understand the identity store requirements for the authentication solution that you want to deploy. See Authentication on page 953 for an overview of how the Tableau Server identity store interacts with authentication types.

**Kerberos**

Kerberos is a three-way authentication protocol that relies on the use of a trusted third-party network service called the Key Distribution Center (KDC) to verify the identity of computers and
provide for secure connections between the computers through the exchange of *tickets*. These tickets provide mutual authentication between computers or services, verifying that one has permission to access the other.

Tableau Server supports Kerberos authentication in an Active Directory Kerberos environment, with authentication to Tableau Server being handled by Kerberos.

**Note:** The Kerberos support in Tableau Server is for user authentication. It does not handle internal permissions and authorization related to Tableau Server content, such as workbooks.

**Quick Start: Single Sign-On with Kerberos**

Tableau Server now supports Kerberos-based single sign-on (SSO). Users with Active Directory (AD) accounts in a Kerberos-enabled environment can now use SSO to connect to Tableau Server from Tableau Desktop and web browsers. In addition, Tableau Server can use Kerberos for authentication to Kerberos-enabled Microsoft SQL, MSAS, PostgreSQL, Hive/Impala, and Terradata data sources. When Tableau Server is configured for Kerberos, you can make SSO connections to Cloudera Impala databases using server managed credentials for Impala LDAP authentication.

1 **Configure Tableau Server**

After you install Tableau Server, run the Tableau Server Configuration utility. On the **Kerberos** tab select **Enable Kerberos for single sign-on**.
2 Generate the Configuration Script

Click Export Kerberos Configuration Script to generate a batch file that will configure Kerberos in AD for Tableau Server.

Save the file and then send it to your AD domain administrator to run.

3 Run the Configuration Script

The domain administrator needs to run the script from a command prompt on any computer in the domain by typing the name of the script.
When your domain administrator runs the configuration script, the script registers Service Principal Names (SPNs) for Tableau Server using the Run As User account, and generates a .keytab file for your environment. (The .keytab file is created in a \keytabs folder in the folder where the script was run.

Have the domain administrator send you a copy of the .keytab file.

4 Copy the .keytab File

On the Kerberos tab of the Tableau Server Configuration utility, enter the path to the .keytab file in the text box in Step 3.

The utility will copy the file to each gateway node in the Tableau Server installation.

Click Test Configuration to verify that the configuration is correctly set up. If the SPNs are correctly set up, the test should display an OK. The number of services configured for delegation will be 0 (zero) unless you have completed the steps below in Configure Kerberos Delegation in AD.

Configure Kerberos Delegation in AD

To use Kerberos Authentication with SQL Server or MSAS data source, or to make SSO connections to Cloudera Impala, you need to configure Kerberos delegation in AD. You don't need to complete these steps if you will only be using Kerberos SSO to connect to Tableau Server.

To configure Kerberos delegation in AD:

- Enable the Run As User to act as the operating system. For more information, see Enable Run As User to Act as the Operating System on page 1004.

- Enable Kerberos delegation in AD. This step is specific to the supported connection type (s) that you will be using with Tableau:
  - **SQL Server** - See Enabling Kerberos Delegation for SQL Server in the Tableau Knowledge Base.
  - **MSAS** - See Enabling Kerberos Delegation for MSAS in the Tableau Knowledge Base.
  - **Hive/Impala** - See Enable Kerberos Delegation for Hive/Impala on page 997.
- Impala - See Enabling Delegation for Cloudera Impala in the Tableau Knowledge Base.

- PostgreSQL - See Enabling Delegation for PostgreSQL in the Tableau Knowledge Base.

- Teradata - See Enabling Delegation for Teradata in the Tableau Knowledge Base.

**Kerberos Authentication in Tableau Server**

When you configure Tableau Server for Kerberos in an Active Directory (AD) environment, the AD domain controller also serves as the Kerberos Key Distribution Center (KDC) and issues Ticket Granting Tickets to the other nodes in the domain. Users authenticated by the KDC do not have to authenticate further when connecting to Tableau Server.

The following is a diagram of the authentication workflow.

1. User logs into their Active Directory domain.
2. The Kerberos KDC authenticates the user and sends a Ticket Granting Ticket (TGT) to the user's computer.
3. The user connects to Tableau Server in Tableau Desktop or in a web browser.
4. Tableau Server authenticates the user.
Kerberos Requirements

To use Kerberos authentication with Tableau Server, you need the following:

- **Windows Server**: Tableau Server must be installed on a server version of Windows. Non-server versions (including Windows 7 and Windows 8) do not support the `ktpass` command required for generating a keytab file.
- **Active Directory**:
  - Tableau Server must use Active Directory (AD) for authentication.
  - The domain must be an AD 2003 or later domain for Kerberos connections to Tableau Server.
  - The domain must be an AD 2012 or later domain for delegated Kerberos connections to data sources. (2012 R2 is preferred because it has a dialog for configuring constrained delegation. 2012 non-R2 requires manual configuration.)
- **Run As User account**:
  - The Run As User account (the Tableau Server service account) must be an AD domain account. Local accounts, including `NTAUTHORITY\NetworkService` will not work.
  - The Run As User account must be in the same domain as the database services that will be delegated.
  - Constrained delegation: The Run As User account must be granted access to the target database Service Principal Names (SPNs).
  - Data Source authentication: If you plan to use Kerberos to authenticate to Microsoft SQL Server, MSAS, PostgreSQL or Teradata databases, enable the Run As User account to act as part of the operating system. For more information, see [Enable Run As User to Act as the Operating System](#) on page 1004.
- **Single-Sign On (SSO)**: Users must be granted a Kerberos Ticket Granting Ticket (TGT) from Active Directory when they sign into their computers. This is standard behavior for domain-joined Windows computers and standard for Mac computers that use AD as their network account server. For more information on using Mac computers and Active Directory, see [Join your Mac to a network account server](#) in the Apple Knowledge Base.
- **External Load Balancer/Proxy Server**: If you are going to use Tableau Server with Kerberos in an environment that has external load balancers (ELBs) or proxy server, you need to set these up before you configure Kerberos in the Tableau Server Configuration utility. See [Add a Load Balancer](#) on page 730 and [Configure a reverse proxy server](#) on page 584 for more information.
- **Smart Card Support**: Smart cards are supported when users sign into their workstations with a smartcard and this results in a Kerberos TGT being granted to the user from Active Directory.
- **iOS Browser Support**: An iOS user can use Kerberos authentication with mobile Safari if a Configuration Profile specifying the user’s Kerberos identity is installed. See [Configuring an iOS Device for Kerberos Support](#) in the Tableau Knowledge Base.
For more information about browser support for Kerberos SSO, see Browser Support for Kerberos SSO to Tableau Server in the Tableau Knowledge Base.

External load balancers:

- If you are using an external load balancer or a reverse proxy, complete the configuration for the external load balancer or reverse proxy before configuring Tableau Server for Kerberos.

  **Note:** If you configure these after configuring Tableau Server for Kerberos, the configuration script generated by the Tableau Server Configuration utility might use the wrong host names. See Add a Load Balancer on page 730 and Configure a reverse proxy server on page 584 for more information.

To use Kerberos authentication for delegated access with data sources:

- **Data Sources:**
  - The supported data sources (SQL Server, MSAS, PostgreSQL, Hive/Impala, and Teradata) must be configured for Kerberos authentication.
  - The data sources must be on the same domain as Tableau Server (users can be on different domains).

Kerberos connections to Tableau Server are supported in the following configurations:

- Tableau Server requires constrained delegation, where the Run As User account is specifically granted rights to the target database SPNs. Unconstrained delegation is not supported.

**Configure Kerberos**

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

1. Open a command prompt as an administrator and change directories to the location of Tableau Server's bin directory. The default location is `C:\Program Files\Tableau\Tableau Server\10.0\bin`.

2. Type the following command to stop Tableau Server:

   ```
   tabadmin stop
   ```

3. Open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server), and then click the Kerberos tab.

4. Select **Enable Kerberos for single sign-on**.
5. Click **Export Kerberos Configuration Script**. The generated script configures your Active Directory domain to use Kerberos with Tableau Server. For more information, see **Kerberos Configuration Script** on the next page.

![Kerberos Configuration Script](image)

**Note:** Verify the host names in the setspn lines of the script. If you are using an external load balancer or a reverse proxy, the host names should match the name you used when you configured Tableau Server for the load balancer or proxy. If you have not configured Tableau Server for your proxy or external load balancer, do that and then re-export the Kerberos configuration script to ensure it has the correct host names. See **Add a Load Balancer** on page 730 and **Configuring Proxies for Tableau Server** on page 579.

6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the .keytab file. The domain administrator should do the following:

   - Review the script to verify it contains correct values.
   - Run the script at a command prompt on any computer in the domain by typing the script name (not by double-clicking the script in Windows Explorer).
     
     The script creates a file, `kerberos.keytab`, in a `\keytabs` folder in the location that the script was run.

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

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

8. (optional) Click **Test Configuration** to confirm that your environment is configured correctly to use Kerberos with Tableau Server.
If you have not configured any data sources for Kerberos delegation, 0 is shown for the **Number of services configured for delegation**.

9. Click **OK** to save your Kerberos configuration.

10. Start Tableau Server.

**Confirm Your SSO Configuration**

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

![Web browser window with URL](image)

You should be automatically authenticated to Tableau Server.

**Kerberos Configuration Script**

When you click **Export Kerberos Configuration Script** in the Tableau Server Configuration utility, the **KerberosConfig.bat** script is generated. This script registers the Service Principal Names (SPNs) for Tableau Server in Active Directory (AD) and generates a Kerberos .keytab file.

**SPNs** - The script uses the setspn utility to register the SPNs for Tableau Server, using the Run As User account. These SPNs are used for generating the .keytab file, and for authenticating web browser connections to Tableau Server.

**.keytab** - The script uses the ktpass utility, to generate a **kerberos.keytab** file, located in the \keytabs folder in the folder where the script was run. The .keytab file contains the shared secret key for Tableau Server.
**Note:** The setspn and ktpass utilities may generate warning or errors. You can ignore these errors and warnings if the utilities run to completion.

**Enable Kerberos Delegation**

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer’s name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Tableau Server requires constrained delegation, with the Run As User account specifically granted delegation rights to the target database Service Principal Names (SPNs). Delegation is not enabled in Active Directory by default.

To configure Kerberos delegation:

1. On all nodes in Tableau Server, configure the Run As User to act as part of the operating system. For more information, see [Enable Run As User to Act as the Operating System](#) on page 1004.

2. In Active Directory:
   - Configure SPNs for the data sources you will be using.
   - Enable Kerberos delegation for the data sources’ SPNs
   - (Optional for multi-domain environments) Configure krb5_conf.html to map principal names to local user names for each Kerberos realm. See [Kerberos delegation multi-domain configuration](#) on page 1009.

3. Enable delegation for data connections:
   - **SQL Server**—See [Enabling Kerberos Delegation for SQL Server](#) in the Tableau Knowledge Base.
   - **MSAS**—See [Enabling Kerberos Delegation for MSAS](#) in the Tableau Knowledge Base.
   - **Hive/Impala**—See [Enable Kerberos Delegation for Hive/Impala](#) on the next page.
   - **PostgreSQL**—See [Enabling Kerberos Delegation for PostgreSQL](#) in the Tableau Knowledge Base.
Enable Kerberos Delegation for Hive/Impala

Disclaimer: This topic includes information about a third-party product. Please note that while we make every effort to keep references to third-party content accurate, the information we provide here might change without notice as Hive/Impala changes. For the most up-to-date information, please consult Hive/Impala documentation and support.

Starting with Tableau Server 10, Tableau Server supports Kerberos delegation to Hive/Impala data sources. You can use two different authentication approaches with Kerberos between Tableau Server and Hive/Impala:

- Constrained Delegation/Viewer Credentials
- Database Impersonation using Delegation UID

Before you can use constrained delegation, you need to configure Tableau Server for Kerberos. For more information see Configure Kerberos on page 993. Verify that the Hive/Impala driver that is installed on Tableau Server supports constrained delegation on Windows. With Tableau Server configured, you can use Kerberos for single sign-on (SSO) between Tableau Desktop or a web browser and Tableau Server.

Before you can use database impersonation using the Tableau Server Run As User, the Run As User must be a network account and configured to log in to Hive/Impala using Kerberos. You must also verify that the Hive/Impala driver that is installed on Tableau Server supports the delegationUID parameter.

The next step depends on which Kerberos authentication type you want to use:

- Constrained Delegation—To use constrained delegation, you need to enable Kerberos delegation in Active Directory (AD). See Enabling Constrained Delegation below.
- Database Impersonation—To use database impersonation, you need to configure delegation on your Hadoop distribution. This configuration is beyond the scope of Tableau Server documentation, but for more information, see Database Impersonation below.

Enabling Constrained Delegation

Enabling constrained delegation for Kerberos to Hive/Impala requires you to specify the Tableau Server Run As User for delegation, and add the Hive/Impala services account for delegation in Active Directory. You need to be a domain administrator for your AD domain to do these steps. After configuring Tableau Server for Kerberos, do the following:
Specify the Run As User for delegation

1. On the Active Directory domain controller, start the Active Directory Users and Computers (ADUC) tool.
2. In the left pane (Active Directory Domain Services), click Users.
3. In the Users pane, right-click the name of the Run As User who will be doing the delegation and then click Properties.
4. In the Properties dialog box, in the left pane, select Delegation.
5. In the Delegation section, select Trust this user for delegation to specified services only.
6. Select Use any authentication protocol.

Add Hive/Impala service accounts for delegation

1. To specify the services to be delegated, click Add.
2. In the Add Services dialog box, click Add Users or Computers.
3. In the text field, type the name of the Hive/Impala service account and then click Check Names. The account should be found.
4. Click OK.

The SPN (Service Principal Name) list is populated.
5. Select the SPNs registered for the Hive/Impala services you want to delegate to.

The SPNs should now appear in the SPN list in the delegation section of the properties window for the user.
6. Click OK.

When this configuration is complete and Tableau Server users publish workbooks or data sources to the server specifying Viewer Credentials, delegation is done to make the connection to the Hive/Impala data source.

Database Impersonation

Database impersonation for Kerberos to Hive/Impala requires you to configure your Hadoop distribution for delegation using the DelegationUID connection parameter.

Publish using Database Impersonation

You can publish using database impersonation with one of two methods:

- Embedded credentials—if you do not have a Kerberized cluster, or you have an LDAP frontend, you can embed the credentials for the impersonating user when publishing. The user you connect as when you publish must be configured with the ability to delegate
for other users.

- Impersonate with server run as—If you have a Kerberized cluster, you can connect with this option. In this case we will connect as the Tableau Server service user using Kerberos to the backend. The impersonating user, in this case Tableau Server, must be configured with the ability to delegate for other users.

You can validate that database impersonation is configured using the driver manager:
Enable Kerberos for Oracle

Starting with Tableau Server version 10.0, Tableau Server supports Kerberos authentication for Oracle data sources.

To use this feature, you must install and configure software on both Tableau Desktop and Tableau Server.

Disclaimer: This topic includes information about a third-party product. Please note that while we make every effort to keep references to third-party content accurate, the information we provide here might change without notice as Oracle changes. For the most up-to-date information, please consult Oracle documentation and support.

This topic contains the following sections:

- Configure Tableau Desktop
- Configure Tableau Server
- Use Kerberos authentication

Configure Tableau Desktop

This section describes how to configure Tableau Desktop for Windows computer to use Kerberos on a Oracle connector.

Prerequisites

Before you can configure Kerberos for Oracle on Tableau Desktop, you must perform the following tasks on each installation of Tableau Desktop:

- Install the Java SE Development Kit on the Tableau Desktop computer.
- Install either the 11g or 12c version of the Oracle Data Access (ODAC) driver.
- Install the Tableau Oracle driver. You can download the driver from the Tableau Drivers & Activation web page.

Step 1: Set system environmental variables

Follow the procedure in the Tableau Knowledge Base to set the required environment variables.

Note: All file path examples in this document use C: drive as system drive. If you have installed to a different drive, change paths accordingly. In all cases, verify the paths. Oracle client paths will include the latest version (for example, 11.2.0), which might not match the file versions exactly as shown here.

For the Oracle 11g client:
• Set the ORACLE_HOME variable to C:\app\user_name\product\11.2.0\client_1

• Set the TNS_ADMIN variable to C:\app\user_name\product\11.2.0\client_1\Network\Admin

For the Oracle 12c client:

• Set the ORACLE_HOME variable to C:\app\client\user_name\product\12.1.0\client_1

• Set the TNS_ADMIN variable to C:\app\user_name\product\12.1.0\client_1\Network\Admin

Step 2: Customize the sqlnet.ora file

1. In a text editor, open the %ORACLE_HOME%\Network\Admin\sqlnet.ora file.

2. Copy the following content into the file:
   - For the Oracle 11g client
     
     SQLNET.KERBEROS5_REALMS= C:\Windows\krb5.realms
     SQLNET.AUTHENTICATION_SERVICES = (BEQ, TCPS, KERBEROS5)
     SQLNET.KERBEROS5_CONF = C:\Windows\krb5.ini
     SQLNET.KERBEROS5_CONFMIT = TRUE
     SQLNET.KERBEROS5_CC_NAME = OSMSFT:
   
   - For the Oracle 12c client
     
     SQLNET.KERBEROS5_REALMS = C:\Windows\krb5.realms
     SQLNET.AUTHENTICATION_SERVICES = (BEQ, TCPS, KERBEROS5PRE, KERBEROS5)
     SQLNET.KERBEROS5_CONF = C:\Windows\krb5.ini
     SQLNET.KERBEROS5_CONFMIT = TRUE
     SQLNET.KERBEROS5_CC_NAME = OSMSFT:
   
3. Save and close the file. You will need this file later when you configure Tableau Server.

Step 3: (Optional) Create and customize the tnsnames.ora file

If your users will be using the Generic ODBC Connection, create the tnsnames.ora file.

1. Open a text editor and copy the following content into the editor:

   ORCL =
   (DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = \FQDN_of_Oracle_DB\)(PORT = 1521))
(CONNECT_DATA =
 (SERVER = DEDICATED)
 (SERVICE_NAME = orcl)
)

where \FQDN_of_Oracle_DB\ is the host name of the Oracle server your users will connect with, such as oracle1.dev.example.lan. You can add multiple host names to this parameter.

2. Save the file and name it \tnsnames.ora.\n3. Copy the file to the \%ORACLE_HOME%\Network\Admin\ folder.

Step 4: Create and customize the \krb5.ini\ file

1. Open a text editor and copy the following content into the editor:

```plaintext
[libdefaults]
forwardable = true
default_realm = \FQDN_user_domain\ndefault_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
[realms]
FQDN_user_domain = {
kdc = \FQDN_domain_controller\nadmin_server = \FQDN_domain_controller\n}
[domain_realm]
.FQDN_user_domain = FQDN_user_domainFQDN_user_domain = FQDN_user_domain
```

where:

- \FQDN_user_domain\ is the fully qualified domain name of the domain where users are authenticated, such as \users.dev.example.lan\n
- \FQDN_domain_controller\ is the fully qualified domain name of a domain controller in the domain where users are authenticated, such as \dc1.users.dev.example.lan\n
2. Save file and name it \krb5.ini.\n3. Copy the file to the \C:\Windows\ folder. When you configure Tableau Server, you will also copy this file to the computer running Tableau Server.
Configure Tableau Server

This section describes how to configure Tableau Server. You must follow these steps on each computer that is running Tableau Server.

Prerequisites

Before you can configure Kerberos for Oracle on Tableau Server, you must perform the following tasks:

- **Configure Kerberos** on page 993.
- ** Enable Run As User to Act as the Operating System** on the next page.
- **Install Java SE Development Kit** on the Tableau Server computer.

Step 1: Install Oracle Database Client on the Tableau Server computer

1. Download the Oracle Database Client 12c Release 1 ($winx64_12102_client.zip) from the Oracle website.
2. Extract the downloaded file and run Setup.exe.
3. Select the following options:
   - On the **Select Installation Type** page, select **Administrator**.
   - On the **Specify Oracle Home User** page, select **Use Windows Built-in Account**.

Step 2: Set system environmental variables

Follow the procedure in the Tableau Knowledge Base to set the following variables:

- **Set the** `ORACLE_HOME` **variable to** `C:\app\user_name\product\12.1.0\client_1`
- **Set the** `TNS_ADMIN` **variable to** `C:\app\user_name\product\12.1.0\client_1\Network\Admin`

Step 3: Install Oracle Database Patch on the Tableau Server computer

Download the Oracle Database Patch version 12.1.0.2.10 from the Oracle website. Follow the installation instructions in the Readme.html file that is included with the patch.

Step 4: Copy client files to Tableau Server

Find the following files that you created when configuring Tableau Desktop and copy them to Tableau Server:

- **Copy** `sqlnet.ora` **to the following path:**
  
  `C:\app\client\user_name\product\12.1.0\client_1\network\admin\sqlnet.ora`
Use Kerberos authentication

1. On a Tableau Desktop computer, open the Oracle connector.

   ![Oracle Connector](image)

   - **Server**: `oracle1.dev.example.lan`
   - **Service**: `Optional`
   - **Port**: `Optional`
   - **Enter information to sign in to the server**:
     - **Integrated Authentication**
     - **Use a specific username and password**:
       - **Username**: `test`
       - **Password**: 

2. In the **Server** field, enter the fully qualified host name of the Oracle server, such as `oracle1.dev.example.lan`.

3. Select **Integrated Authentication**.

4. Create a workbook with a view and publish it to Tableau Server. When you publish the workbook, configure authentication to use viewer credentials as described in Tableau Desktop help.

Enable Run As User to Act as the Operating System

To use Kerberos delegation with Tableau Server, you must configure the Run As User account to act as the operating system on each Tableau Server node.

1. On the computer that is running Tableau Server, select **Start > Control Panel > Administrative Tools > Local Security Policy**.
2. In the Local Security Settings window, expand **Local Policies**, click **User Rights**
Assignments, and then right-click Act as part of the operating system and select Properties.

3. In the Act as part of the operating system Properties window, click Add User or Group.

4. Type the `<domain>\<username>` for the Tableau Server Run As User account (for example: MYCOMPANY\tableau_server), and then click Check Names.

5. When the account resolves correctly, it is underlined. Click OK.

6. Click OK to close the Local Security Policy windows.

Troubleshoot Kerberos

The troubleshooting suggestions in this topic are divided into issues related to Single sign-on (SSO) on the server and issues with the delegated data sources.

Single Sign-on to Tableau Server

Kerberos Authentication Failed (unable to connect automatically to Tableau Server)

If you are using Kerberos for SSO and a user is prompted to sign in to Tableau Server when they connect with either a web browser or with Tableau Desktop, try these steps from the client computer:
Troubleshooting on the client computer

- **Account permissions**—Try to sign in to Tableau Server using the user's name and password. If they can't sign in to Tableau Server using their user name and password, they do not have permission to access Tableau Server and Kerberos authentication will fail.

- **Other accounts**—Try to connect with SSO to Tableau Server using other user accounts. If all users are affected, the problem may be in the Kerberos configuration.

- **Computer location**—Kerberos will not work when connecting from localhost. Clients must be connecting from a computer other than the Tableau Server computer.

- **URL address**—You cannot use Kerberos SSO when connecting using an IP address. In addition, the server name you use to access Tableau Server must match the name used in the Kerberos configuration (see Key table entry, below).

- **TGT (Ticket Granting Ticket)**—Confirm that the client computer has a TGT from the Active Directory domain. Kerberos requires a TGT to sign in. To confirm the client computer has a TGT, type:
- klist tgt at a command prompt on a Windows computer
  or
  klist at a terminal prompt on a Mac computer

The output should show a TGT for the user/domain trying to authenticate to Tableau Server.

The client computer may not have a TGT in the following circumstances:

- The client computer is using a VPN connection
- The client computer is not joined to the domain (for example, it is a non-work computer being used at work)
- The user signed into the computer with a local (non-domain) account
- The computer is a Mac that is not using Active Directory as a network account server

- Browser—Check which browser the user is using to access the server
  - Internet Explorer (IE) and Chrome work "out of the box" on Windows
  - Safari works "out of the box" on Mac
  - Firefox requires additional configuration

For more information about browser support for Kerberos Single Sign-On (SSO), see Browser Support for Kerberos SSO to Tableau Server in the Tableau Knowledge Base.

**Troubleshooting on the server**

If you cannot solve the problem from the client computer, your next steps are to troubleshoot on the computer running Tableau Server. The administrator can use the request ID to locate the sign-in attempt in the Apache logs on Tableau Server.

- Log files—Check the Apache error.log for an error with the exact time/date of the failed sign-in attempt.
  - In a ziplog archive, these logs are in the \httpd folder.
  - On Tableau Server, these logs are in the \data\tabsvc\logs\httpd\ folder.

- Key table entry—If the error.log entry says "No key table entry matching HTTP/<servername>.<domain>.<org>@", for example:

  [Fri Oct 24 10:58:46.087683 2014] [:error] [pid 2104:tid 4776] [client 10.10.1.62:56789] gss_acquire_cred() failed: Unspecified GSS failure. Minor code may provide more information (, No key table entry found matching HTTP/server-name.domain.com@)
This error is a result of a mismatch between any of the following:

- **Tableau Server URL** - The URL used by the client computer to access the server.
  This is the name that you type into Tableau Desktop or a browser address bar. It could be a shortname (http://servername) or a fully-qualified domain name (http://servername.domain.com)

![Tableau Server URL](image)

- **DNS reverse lookup** for the server IP address
  This looks up a DNS name using an IP address.
  At a command prompt type:

  ping servername

  with the IP address returned by pinging the server, do a reverse DNS lookup type:

  nslookup <ip address>

  The Tableau Server computer name needs to match in:
  - .keytab file
  - Service Principal Name (SPN) for the server

![DNS Reverse Lookup](image)

---

**Test Configuration and tabconfig.log**

Use the Test Configuration button in the Tableau Server Configuration utility:
If your SPNs are correctly set up for Kerberos, **SPNs are correctly configured** shows OK.

If any services are configured for delegation, the number of configured services will appear. A value of 0 (zero) does not indicate a problem unless you are using delegation and Kerberos authentication to SQL Server or MSAS.

Look in `tabconfig.log` for any problems or errors. For example:

```
2014-10-17 10:58:16.545 -0700 ERROR root: No SPN entries found
```

If the test does not show successful results, run the configuration script again.

**Data source SSO**

**Delegated data source access failures**

Check the vizqlserver log files for "workgroup-auth-mode":

- In a ziplog archive, these logs are in the `\vizqlserver\Logs` folder
- On the Tableau Server, these logs are in the `\data\tabsvc\vizqlserver\Logs` folder

Look for "workgroup-auth-mode" in the log files. It should say "kerberos-impersonate" not "as-is".

**Kerberos delegation multi-domain configuration**

Tableau Server has the ability to delegate users from other Active Directory domains. If your database uses MIT Kerberos, you may need to adjust your Kerberos principal to database user mapping. Specifically, you will need to update `krb5.conf` with rules for each Kerberos realm that users will connect from. Use the `auth_to_local tag` in the `[realms]` section to map principal names to local user names.

For example, consider a user, `EXAMPLE\jsmith`, whose Kerberos Principal is `jsmith@EXAMPLE.LAN`. In this case, Tableau Server will specify a delegated user, `jsmith@EXAMPLE`. Tableau Server will use the Active Directory legacy domain alias as the Kerberos Realm.

The target database may already have a rule such as the following to map the user, `jsmith@EXAMPLE.LAN` to the database user, `jsmith`.

```
EXAMPLE.LAN = {
    RULE:[1:$1@$0]{.*@EXAMPLE.LAN}s/@.*//
```
To support delegation, you must add another rule to map jsmith@EXAMPLE to a database user:

```plaintext
EXAMPLE.LAN = {
    RULE: [1:$1@$0]\.*@EXAMPLE.LAN}s/@.*//
    RULE: [1:$1@$0]\.*@EXAMPLE)s/@.*//
    DEFAULT
}
```

See the MIT Kerberos Documentation topic, krb5.conf, for more information.

**SAML**

SAML (Security Assertion Markup Language) is an XML standard that allows secure web domains to exchange user authentication and authorization data. You can configure Tableau Server to use an external identity provider (IdP) to authenticate Tableau Server users over SAML 2.0. This allows you to provide a single sign-on experience for your users across all the applications in your organization.

These are the options for configuring SAML with Tableau Server:

- **Server-wide SAML authentication.** All server users authenticate with the same SAML IdP.

- **Server-wide local authentication and site-specific SAML authentication.** The users from one or more sites on Tableau Server authenticate with one or more SAML IdPs. Each site can use a different IdP. In this option, users who are not configured to use SAML can sign in using local authentication.

- **Server-wide SAML authentication and site-specific SAML authentication.** All users authenticate with a SAML IdP. There is a default SAML IdP for users that belong to multiple sites. Each site can use a different IdP.

User authentication is performed by the SAML IdP, not by Tableau. SAML does not handle permissions and authorization having to do with Tableau Server content, such as workbooks.

**Note:** Tableau Server supports both service provider initiated and IdP initiated SAML. However, if you connect to Tableau Server from Tableau Desktop or Tableau Mobile, it is a service provider initiated connection.

See the following links for more information about SAML.
Quick Start: Single Sign-On with SAML

With Tableau’s SAML support, you use one or more external identity providers (IdP) to authenticate Tableau Server users. This quick start describes how to set up a server-wide SAML implementation that uses a single IdP.

To configure Tableau Server for SAML, you need the following:

- **Certificate file**: A PEM-encoded x509 certificate that has a .crt filename extension.
- **Certificate key file**: An RSA or DSA key file that is not password protected and that has a .key filename extension.
- **IdP account**: Examples are PingFederate, SiteMinder, and OpenAM.
- **Matching usernames**: Tableau Server usernames and the usernames stored in the IdP must match. Ensure that the username you plan to use for your Tableau Server administrator account exists with your IdP before you run setup.

1 Specify the Server and Certificates

Run Tableau Server setup. After you configure your general settings in the Configuration utility, click the SAML tab and select **SAML authentication for the server**:

In the **Tableau Server return URL** box, enter the customer-facing URL for your installation of Tableau Server. Enter the same value for **SAML entity ID**.

Create a folder named **SAML** under the following folder, and then copy the .crt and .key files to the new folder.

C:\Program Files\Tableau\Tableau Server

Use the new folder for the **SAML certificate file** and **SAML key file** boxes.
2 Export Metadata from Tableau

Leaving the SAML IdP metadata file box empty, click the Export Metadata File button.

Use the .xml file name of your choice.

In the next dialog box, save the XML file. You will need to provide this file to your IdP in the next step.

3 Export Metadata from the IdP

On the IdP’s website, add your installation of Tableau Server as a connection type for the IdP to authenticate. As part of this, you import the Tableau metadata .xml file you created in step 2, and confirm that your IdP’s settings use username as the attribute element to verify.

Next, export your IdP’s metadata .xml file and copy it to the following folder on the computer where Tableau Server is installed:

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

4 Test the SAML Sign-On

On the SAML tab in the Tableau Configuration utility, enter the location of the IdP’s file in the SAML IdP metadata box. Click OK. Finish the setup process.
To test your changes, start a new web browser session and go to the URL for your installation of Tableau Server. If SAML is properly configured, the Sign On prompt is from your IdP and not from Tableau:

![Sign On prompt](image)

**How SAML Authentication Works**

SAML (Security Assertion Markup Language) is an open standard for exchanging authentication information between a service provider and an identity provider (IdP). A third-party IdP is used to authenticate users and to pass identity information to the service provider in the form of a digitally signed XML document. Tableau Server is a service provider. Examples of IdPs include PingOne and OneLogin.

When you use a trusted IdP for the SAML connection, you can provide a single sign-on (SSO) experience, in which your users can access their web applications, including Tableau Server, through one set of credentials. In this environment, only the IdP has access to users' credentials.

Tableau supports authentication initiated by the service provider and the IdP. In service provider initiated authentication, your users navigate to Tableau and are redirected to the IdP for authentication. In IdP initiated authentication, your users authenticate with the IdP first and then navigate to Tableau.

The following image shows the steps to authenticate a user with single sign-on in a typical service provider initiated flow:
1. User navigates to the Tableau Server sign-in page or a published workbook, and enters the user name.

2. Tableau Server starts the authentication process and redirects the request to the registered IdP.

3. The IdP requests the user’s password and, after confirming that the user name submitted is identical to the user name stored in the IdP assertions, authenticates the user.

4. The IdP returns a SAML success response to Tableau Server.

5. Tableau Server displays the page the user requested in step 1.

**SAML Requirements**

Before you configure SAML for Tableau Server, ensure that you meet the following requirements:

**Certificate and identity provider (IdP) requirements**

To configure Tableau Server for SAML, you need the following:

- **Certificate file.** A PEM-encoded x509 certificate file with a .crt extension. This file is used by Tableau Server, not the IdP. If you have an SSL certificate, you can use the same certificate with SAML. See About the Certificate File later in this topic for details.

- **Certificate key file.** An RSA or DSA private key file that is not password protected, and
which has the .key extension. This file is used by Tableau Server, not the IdP. The certificate key file must have the passphrase embedded in it. If you have an SSL certificate key file, you can use it for SAML as well. See About the Certificate File later in this topic for details.

- **IdP account that supports SAML 2.0.** You need an account with an external identity provider. Some examples are PingFederate, SiteMinder, and Open AM. The IdP must support SAML 2.0.

- **IdP provider that supports import and export of XML metadata.** Although a manually created metadata file may work, Tableau Software Technical Support cannot assist with generating the file or troubleshooting it.

### About the certificate file and key files

If you are using a PEM-encoded x509 certificate file for SSL, you can use the same file for SAML. When it's used for SSL, the certificate file is used to encrypt traffic. When it's used for SAML, the certificate is used for authentication.

Tableau Server does not support certificate and certificate key files for SAML if the certificate/key require a chain file. If your SSL certificate and certificate key file require a chain file, you need to generate a new certificate and key file to use for SAML.

### User management requirements

When you configure SAML, the authentication is performed by the IdP outside of Tableau. However, the user management is performed either by Active Directory or by Tableau Server (which is called local authentication even though Tableau Server does not perform authentication when configured with SAML).

When you configure user authentication on the **General** tab of Tableau Server Configuration utility, you must choose an option that is correct for how you want to use SAML:

- **For site-specific SAML:** If you want different sites on Tableau Server to authenticate with different SAML IdPs, configure Tableau Server to use local authentication rather than Active Directory.

- **For server-wide SAML:** If you configure server-wide SAML with a single IdP, you can configure Tableau Server to use local authentication or Active Directory for user management. However, if you select Active Directory, you must disable the **Enable automatic logon** option.

### SAML compatibility requirements and notes

Note the following about using SAML with Tableau:

- **Service provider initiated:** Tableau Server only supports SAML authentication that begins at the service provider (SP).

- **No Kerberos:** Tableau Server does not support SAML and Kerberos together.
- No mutual SSL: Tableau Server does not support mutual SSL (two-way SSL) and SAML together, whether it is server-wide SAML or site-specific SAML. If you want to use mutual SSL, you can configure mutual SSL on the IdP side rather than on the Tableau Server side.

- No encrypted assertions for site-specific SAML: For site-specific SAML, Tableau Server does not support encrypted SAML assertions from the IdP. However, all SAML requests and responses are sent over HTTPS.

- User identity in Tableau Server for tabcmd users: To use tabcmd with the server, users must sign in to the server using the credentials of a user defined on the server; you cannot use SAML accounts with tabcmd. An initial system administrator user is created when the server is first installed and configured.

- IdP provider that uses forms-based authentication: Tableau Desktop supports signing in with SAML. However, your IdP must support forms-based authentication. Otherwise, you can disable SAML for Tableau Desktop with the wgserver.authentication.desktop_nosaml command. See tabadmin set options on page 1294 for more information.

- Distributed installations: Clusters configured for SAML must have the same SAML certificate, SAML key, and SAML IdP metadata files on each Tableau Server that’s running an Application Server process. See Configure a Server Cluster for SAML for details.

- Login URL: For users to be able to sign in, your IdP must be configured with SAML Login endpoint that sends a POST request to the following URL:

  http(s)://<IdP>/wg/saml/SSO/index.html.

- Logout URL: To enable users to sign out after signing in with SAML, your IdP must be configured with a SAML Logout endpoint that sends a POST request to the following URL:

  http(s)://<IdP>/wg/saml/SingleLogout/index.html.

- Post-logout redirect URL: By default, when a user signs out of Tableau Server, the sign-in page is displayed. To specify an alternate page to display after sign-out, use the tabadmin set wgserver.saml.logout.redirect_url command.

  - To specify an absolute URL, use a fully-qualified URL starting with http:// or https://, as in this example:

    tabadmin set wgserver.saml.logout.redirect_url
    http://example.com

  - To specify a URL relative to the Tableau Server host, use a page starting with a / (slash):
tabadmin set wgserver.saml.logout.redirect_url /ourlogoutpage.html

- **Active Directory Federation Service (AD FS):** You must configure AD FS to return additional attributes for Tableau authentication with SAML. The **Name ID** and **username** attributes can be mapped to the same AD attribute: **SAM-Account-Name**. For configuration information, see [Authenticating an External Tableau Server using SAML & AD FS in the Information Lab blog](#).

**Disclaimer:** Clicking this link will take you away from the Tableau website. Although we make every effort to ensure these links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

**Requirements for connecting from Tableau Desktop**

When you configure SAML for Tableau Server, users with SAML credentials can also sign into Tableau Server from Tableau Desktop. It is recommended that the version of Tableau Desktop match the version of Tableau Server for full compatibility. Note that if you connect to Tableau Server from Tableau Desktop or Tableau Mobile, it is a service provider initiated connection.

**Note:** To connect with site-specific SAML, users must run Tableau Desktop 10.0 or later.

**XML data requirements**

You configure SAML using XML metadata documents that are generated by Tableau Server and by your IdP. During the authentication process, the IdP and Tableau Server exchange authentication information using XML documents. To be sure that the XML that's used for SAML configuration and SAML-based authentication works correctly, review the following requirements. If the XML does not meet these requirements, errors can occur when you configure SAML or when users try to sign in.

- **HTTP POST:** Tableau Server only accepts HTTP POST requests for SAML communications. HTTP Redirect is not supported.

  The SAML metadata XML document that is exported by Tableau Server should contain the following elements, with the **Binding** attribute set to **HTTP-POST**.

  - Verify the following element which specifies the URL that the IdP redirects to after successful authentication:

    ```xml
    <md:AssertionConsumerService
    Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Location="http(s)://TABLEAU-SERVER/wg/saml/SSO/index.html index="0"
    ```

  - Verify the following element which specifies the URL that the IdP redirects to after successful authentication:
- Verify the following element which specifies the URL that the IdP will use for the logout endpoint:

```
<md:SingleLogoutService
  Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Location="http(s)://example-IdP/wg/saml/SingleLogout/index.html/>
```

- Verify the following element which specifies the URL for signin in:

```
<md:SingleSignOnService
  Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Location="http(s)://example-IdP/wg/saml/SSO/index.html/>
```

- **Attribute named username**: You must configure your identity provider to return an assertion that includes the `username` value in the `saml:AttributeStatement` element in a format like the following example. Make sure that the attribute is typed as `xs:string`. (It should *not* be typed as `xs:any`.)

```
<saml:Assertion assertion-element-attributes>
  <saml:Issuer>issuer-information</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
  <saml:Subject>
    ...
  </saml:Subject>
  <saml:Conditions condition-attributes >
    ...
  </saml:Conditions>
  <saml:AuthnStatement authn-statement-attributes >
    ...
  </saml:AuthnStatement>

  <saml:AttributeStatement>
    <saml:Attribute Name="username"
      NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
      <saml:AttributeValue
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">
```

To change the SAML attribute that the username value is passed in, use the `tabadmin set` command to set the `wgserver.saml.idpattribute.username` value to a different attribute name. You must change the attribute if you use a global ID. The attribute name is case sensitive.

- **Matching usernames**: Tableau Server usernames and the usernames stored in the IdP must match. For example, if the username for Jane Smith is stored in PingFederate as `jsmith`, it must also be stored in Tableau Server as `jsmith`. If you are configuring SAML as part of Tableau Server setup, part of setup is creating the Tableau Server administrator account. Before you run setup, make sure that the account you plan to use exists in your IdP.

If you are using Active Directory authentication with Tableau Server and have multiple Active Directory domains (that is, users belong to multiple domains, or your Tableau Server installation includes multiple domains), the IdP must send both the domain and username for a user, and these must match the user exactly in Tableau Server. While these can be sent either as `domain/username` or `username@domain.com`, we recommend using the `domain/username` format. See User Management in Active Directory Deployments on page 1251 for more information.

### Configure Server-Wide SAML

Configure server-wide SAML when you want users on Tableau Server to authenticate with a single SAML identity provider (IdP). For information about authenticating users with different IdPs for different sites on Tableau Server, see Configure Site-Specific SAML on page 1022.

Before you configure Tableau Server for SAML, make sure you meet the SAML Requirements on page 1014.

To configure Tableau Server to use server-wide SAML:

1. Place the certificate files in a folder named SAML, parallel to the Tableau Server 10.0 folder. For example:

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

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

2. If you are configuring SAML during Tableau Server setup, go to the SAML tab in the configuration utility.
If you are configuring SAML after you installing Tableau Server, open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server) and then click the SAML tab.

3. On the SAML tab, select **SAML authentication for the server** and provide the location for each of the following:

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

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

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

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

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

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

5. Save the XML file with the name of your choice.

6. On your IdP’s website or in its application:
   - Add Tableau Server as a Service Provider. Refer to your IdP's documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the file you saved in step 5.
   - Confirm that your IdP uses **username** as the attribute element to verify.

7. Still within your IdP, export your IdP’s metadata XML file.

   It's a good idea to verify that the metadata XML you get from the IdP includes a **SingleSignOnService** element in which the binding is set to HTTP-POST, as in the following example:

   ```xml
   ```

8. Copy your IdP’s metadata XML file to the following folder on the computer where Tableau Server is installed:
9. On the SAML tab in the Tableau Server Configuration dialog box, enter the location to the file in the **SAML IdP metadata file** text box:

![Tableau Server Configuration dialog box](image)

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

**Configure a Server Cluster for SAML**

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

1. Configure the primary Tableau Server as described in the procedure above.

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

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

Test Your Configuration

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

![Web browser window](image)

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

![Sign in prompt](image)

Configure Site-Specific SAML

Configure site-specific SAML when you want each site on Tableau Server to use a different SAML identity provider (IdP). Because each SAML IdP can be site-specific, you must also configure a server-wide authentication method for users that do not belong to a site, or that belong to multiple sites. The server-wide default authentication method is configured in the Tableau Server Configuration utility and can be set to either local authentication or to server-wide SAML authentication.
If there is only one site on Tableau Server, the Default site, then you must use server-wide SAML. For more information, see Configure Server-Wide SAML on page 1019.

To configure Tableau Server to use site-specific SAML, you must complete the following high-level steps. Details about each step appear later in this topic.

1. Use the Tableau Server Configuration utility to perform one of the following tasks:
   - Configure site-specific SAML with local authentication below
   - Configure site-specific SAML with server-wide SAML on the next page
2. Enable SAML for a site on page 1026
3. Configure SAML for a site on page 1027

**Important:** Before you configure Tableau Server for SAML, make sure you meet the SAML Requirements on page 1014. For example, you cannot use Active Directory with site-specific SAML, you must get certificate files, and you must ensure that your IdP provider meets the necessary requirements.

**Configure site-specific SAML with local authentication**

1. If you are configuring SAML during Tableau Server setup, go to the SAML tab in the configuration utility.
   If you are configuring SAML after Tableau Server has been installed, open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server) and then click the SAML tab.
2. On the SAML tab, select the Site-specific SAML authentication only option.
3. Place the certificate files that you want to use in a folder named SAML at the same level as the Tableau Server 10.0 folder. For example:
   C:\Program Files\Tableau\Tableau Server\SAML
   You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.
4. Provide the location for each of the following:
   - **Tableau Server return URL**—The URL that Tableau Server users enter in their browser to access the server, such as http://tableau_server. Using http://localhost will not work for an external-facing server. Using a URL with a trailing slash (for example, http://tableau_server/) is not supported.
   - **SAML entity ID**—Typically the same as the Tableau Server return URL. The
entity ID that you enter is used as a base for generating site-specific entity IDs. For example, if you enter http://tableau_server, a site configured for site-specific SAML might display the following entity ID:

http://tableau_server/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0

- **SAML certificate file**—A PEM-encoded x509 certificate (a file with the extension .crt). This file is used by Tableau Server, not the IdP.
- **SAML certificate key file**—An RSA or DSA private key file that is not password protected, and that has the file extension .key. This file is used by Tableau Server, not the IdP.

5. Click OK.
   Continue to Enable SAML for a site on page 1026.

Configure site-specific SAML with server-wide SAML

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

   If you are configuring SAML after Tableau Server has been installed, open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server) and then click the SAML tab.

2. On the SAML tab, select the SAML authentication for the server and for sites option.

3. Place the certificate files that you want to use in a folder named SAML that’s at the same level as the Tableau Server 10.0 folder. For example:
   C:\Program Files\Tableau\Tableau Server\SAML

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

4. Provide the location for each of the following:

   - **Tableau Server return URL**—The URL that Tableau Server users enter in their browser to access the server, such as http://tableau_server. Using http://localhost will not work for an external-facing server. Using a URL with a trailing slash (for example, http://tableau_server/) is not supported.

   - **SAML entity ID**—The entity ID that uniquely identifies your Tableau Server installation to the IdP. This is typically the same as the Tableau Server return URL. Additionally, the entity ID that you enter is used as a base for generating site-specific entity IDs. For example, if you enter http://tableau_server, a site
configured for site-specific SAML might display the following entity ID:

http://tableau_server/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0

- **SAML certificate file**—A PEM-encoded x509 certificate (a file with the extension .crt). This file is used by Tableau Server, not the IdP.
- **SAML certificate key file**—An RSA or DSA private key file that is not password protected, and that has the file extension .key. This file is used by Tableau Server, not the IdP.

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

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

6. Save the XML file. You can give the file any name you want.

7. On your IdP's website or in its application:
   - Add Tableau Server as a Service Provider. Refer to your IdP's documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will import the file that you just exported from Tableau Server.
   - Confirm that your IdP uses **username** as the attribute element to verify.

8. Still within your IdP, export your IdP's metadata XML file.

   It's a good idea to verify that the metadata you get from the IdP includes a **SingleSignOnService** element in which the binding is set to **HTTP-POST**, as in the following example:


9. Copy your IdP's metadata XML file to the following folder on the computer where Tableau Server is installed:

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

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

10. On the **SAML** tab in the Tableau Server Configuration dialog box, enter the location of
the file in the **SAML IdP metadata file** box:

![Tableau Server Configuration](image)

11. Click **OK**. Tableau Server is now configured for SAML authentication.

Continue to the next step, **Enable SAML for a site** below.

**Enable SAML for a site**

1. Sign in to Tableau Server as a site administrator.

2. If you have more than one site for Tableau Server, select the site for which you want to enable SAML in the sites drop-down.

3. Click **Settings**.

![Tableau Server](image)

4. Click the **Authentication** tab.
5. On the **Authentication** tab, select **Use site-specific SAML**.

Continue to **Configure SAML for a site** below.

**Configure SAML for a site**

This section corresponds to and supplements the steps shown on the Authentication page in Tableau Server.

**Note:** To complete this process, you will also need the documentation your IdP provides. Look for topics that refer to configuring or defining a service provider for a SAML connection.

To display all collapsed content, click the (Expand all) button at the top of the page.

**Step 1: Export metadata from Tableau Server**

To create the SAML connection between Tableau Server and your IdP, you need to exchange required metadata between the two services. To get metadata from Tableau Server, do either of the following:

- Select **Export metadata** to download an XML file that contains the Tableau Server SAML entity ID, Assertion Consumer Service (ACS) URL, and X.509 certificate. Note that the entity ID is site-specific and generated from the entity ID that you entered into the Tableau Server Configuration utility. For example, if you entered `http://tableau_server` into the Tableau Server Configuration utility, you might see the following entity ID for the site:

  `http://tableau_server/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0`

- Select **Download signing and encryption certificate** if your IdP expects the required information in a different way. For example, if it wants you to enter the Tableau Server entity ID, ACS URL, and X.509 certificate in separate locations.
See the IdP’s SAML configuration steps to confirm the correct option.

**Steps 2 and 3: External steps**

For Step 2, to import the metadata you exported in step 1, sign in to your IdP account, and use the instructions provided by the IdP’s documentation to submit the Tableau Server metadata.

For Step 3, the IdP’s documentation will guide you also in how to provide metadata to a service provider. It will instruct you to download a metadata file, or it will display XML code. If it displays XML code, copy and paste the code into a new text file, and save the file with a .xml extension.

**Step 4: Import metadata to Tableau Server**

On the Authentication page in Tableau Server, import the metadata file that you downloaded from the IdP or configured manually from XML it provided.

**Step 5: Match assertions**

Assertions contain authentication, authorization, and other attributes about a user. In the **Identity Provider (IdP) Assertion Name** column, provide the names of the assertions that contain the information Tableau Server requires.

- **Username or Email**: (Required) Enter the name of the assertion that stores user names or email addresses.
- **Display name**: (Optional but recommended) Some IdPs use separate assertions for first and last names, and others store the full name in one assertion.

Select the button that corresponds to the way your IdP stores the names. For example, if the IdP combines first and last name in one assertion, select **Display name**, and then enter the assertion name.

**Step 6: Manage users**

Select existing Tableau Server users, or add new users you want to approve for single sign-on.

When you add or import users, you also specify their authentication type. On the Users page, you can change users’ authentication type any time after adding them.
Important: Users that authenticate with site-specific SAML can only belong to one site. If a user needs to belong to multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.

Step 7: Troubleshooting

Start with the troubleshooting steps suggested on the Authentication page. If those steps do not resolve the issues, see Troubleshoot SAML on page 1204.

Configure SAP HANA SSO

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

Note: You do not need to use SAML sign on with Tableau Server in order to use HANA SSO. You can sign in to Tableau Server using whatever method you choose.

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

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

Configure SSO for SAP HANA

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 10.0 folder. For example:

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

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

2. After you install Tableau Server, run the Configuration utility (Start > All Programs > Tableau Server 10.0 > Configure Tableau Server), and then click the SAP HANA tab.

3. Select Use SAML to enable single sign-on for SAP HANA and provide the location
for each of the following:

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

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

4. Select the format of the user name.

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

**Troubleshoot SAML**

Use the following topics to troubleshoot SAML issues.

**SAML and Enable Automatic Logon**

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select **Enable automatic logon**. **Enable automatic logon** and SAML cannot both be used on the same server installation.

**HTTP Status 500 error when configuring SAML**

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:
org.opensaml.saml2.metadata.provider.MetadataProviderException: User specified binding is not supported by the Identity Provider using profile urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.
- The IdP is configured to use SP-initiated authentication. (IdP-initiated authentication is not supported.)
- The IdP is configured to use HTTP-POST requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in SAML Requirements on page 1014.

Signing In from the Command Line

SAML is not used for authentication when you sign in to Tableau Server using the command linetools `tabcmd` on page 1315 or the Tableau Data Extract command line utility (provided with Tableau Desktop), even if Tableau Server is configured to use SAML. These tools require the authentication configured when Tableau Server was originally installed (either local authentication or AD).

Login Failed

Login can fail with the following message:

Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.

This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as `jsmith` it must be stored in Tableau Server as `jsmith`. 
SAML Error Log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see Archive Log Files on page 1184.

**Note:** To log SAML-related events, `vizportal.log.level` must be set to debug. For more information, see Change Logging Levels on page 1197.

Check for SAML errors in the following files in the unzipped log file snapshot:

`\vizportal\vizportal-<n>.log`

In Tableau Server 9.0 and later, the application process (vizportal.exe) handles authentication, so SAML responses are logged by that process.

**Trailing Slash**

On the SAML tab, confirm that the Tableau Server return URL does not end with a trailing slash (correct: http://tableau_server; incorrect: http://tableau_server/):

![Tableau Server Configuration](image)

**Confirm Connectivity**

Confirm that the Tableau Server you are configuring has either a routeable IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running `telnet` on Tableau Server and attempting to connect with the SAML IdP. For example: `C:\telnet 12.360.325.10 80`

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.
**Trusted Authentication**

When you embed Tableau Server views into webpages, everyone who visits the page must be a licensed user on Tableau Server. When users visit the page they are prompted to sign in to Tableau Server before they can see the view. If you already have a way of authenticating users on the webpage or within your web application, you can avoid this prompt and save your users from having to sign in twice by setting up trusted authentication.

Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from these trusted web servers it assumes that your web server has handled whatever authentication is necessary.

If your web server uses SSPI (Security Support Provider Interface), you do not need to set up trusted authentication. You can embed views and your users will have secure access to them as long as they are licensed Tableau Server users and members of your Active Directory.

**Note:** Client browsers must be configured to allow third-party cookies if you want to use trusted authentication with embedded views.

**How Trusted Authentication Works**

The diagram below describes how trusted authentication works between the client's web browser, your web server(s) and Tableau Server.
1. **User visits the webpage:** When a user visits the webpage with the embedded Tableau Server view, it sends a GET request to your web server for the HTML for that page.

2. **Web server POSTS to Tableau Server:** The web server sends a POST request to the trusted Tableau Server (for example, http://tabaserver/trusted, not http://tabserver). That POST request must have a `username` parameter. The `username` value must be the username for a licensed Tableau Server user. If the server is running multiple sites and the view is on a site other than the Default site, the POST request must also include a `target_site` parameter.

3. **Tableau Server creates a ticket:** Tableau Server checks the IP address or host name of the web server (192.168.1.XXX in the above diagram) that sent the POST request. If it is set up as a trusted host then Tableau Server creates a ticket in the form of a unique 24-...

4. **Web server passes the URL to the browser:** The web server constructs the URL for the view using either the view’s URL or its object tag (if the view’s embedded), and inserts it into the HTML for the page. The ticket is included (e.g., http://tabserver/trusted/<ticket>/views/requestedviewname). The web server passes all the HTML for the page back to the client’s web browser.

5. **Browser requests view from Tableau Server:** The client web browser sends a request to Tableau Server using a GET request that includes the URL with the ticket.

6. **Tableau Server redeems the ticket:** Tableau Server sees that the web browser requested a URL with a ticket in it and redeems the ticket. Tickets must be redeemed within three minutes after they are issued. Once the ticket is redeemed, Tableau Server logs the user in, removes the ticket from the URL, and sends back the final URL for the embedded view.
Tableau Server responds to the POST request with that ticket. If there is an error and the ticket cannot be created Tableau Server responds with a value of -1.

**Add Trusted IP Addresses or Host Names to Tableau Server**

The first step in setting up trusted authentication is to configure Tableau Server to recognize and trust requests from one or more web servers:

1. Open a command prompt as an administrator and navigate to your Tableau Server bin directory (for example, C:\Program Files\Tableau\Tableau Server\10.0\bin).
2. Type the following command to stop Tableau Server:

```
tabadmin stop
```

3. Next, type the following command:

```
tabadmin set wgserver.trusted_hosts "<trusted IP addresses or host names>"
```

In the command above, `<trusted IP addresses>` should be a comma-separated list of the IPv4 addresses or host names of your web server(s).

**Note:** The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts in the `set` command. (You cannot amend the list of hosts by running the `set` command repeatedly.)

For example:

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

or

```
tabadmin set wgserver.trusted_hosts "weberv1, webserv2, web-serv3"
```

**Notes:**
The comma separated list should be in quotes, with one space after each comma.
The web servers you specify must use static IP addresses, even if you use host names (learn more).

4. If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in step 2, above) and Tableau Server, you also need to add them as trusted gateways. See Configure a reverse proxy server on page 584 for steps.

5. Type the following command to save the changes to all the server configuration files:

   `tabadmin config`

6. Finally, type the following command to start the server again:

   `tabadmin start`

Next, you need to configure your web server to receive tickets from Tableau Server.

**Get a Ticket from Tableau Server**

After you’ve added trusted IP addresses to Tableau Server, you’re ready to configure your web server to get tickets from Tableau Server via POST requests (step 3 in the diagram). The POST request must be sent to `http://<server name>/trusted`, not `http://tabserv`. For example `http://tabserv/trusted`.

**Note:** If SSL is enabled you must use https instead of http. For example: `https://tabserver/trusted`.

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

C:\Program Files\Tableau\Tableau Server\10.0\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 ID that was provided when the site was created. This value is case sensitive. If the Site ID is `SAles`, then the `target_site=SAles`. 
• **client_ip=<IP address>** (optional): Used to specify the IP address of the computer whose web browser is accessing the view (for example, `client_ip=123.45.67.891`). It is not the IP address of the web server making the POST request of Tableau Server. If you decide to use this parameter, see Optional: Configure Client IP Matching on page 1039 for more information.

Tableau Server’s response to the POST request will be a unique 24-character string (the ticket). If Tableau Server isn’t able to process the request, the return will be -1. See Ticket Value of -1 Returned from Tableau Server on page 1039 for tips on how to correct this. Also, in order for users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

Next, you need to add code that allows the web server to construct an URL for the view that includes the view’s location and the ticket.

**Display the View with the Ticket**

After you create the POST request, you need to write code that provides the web server with the view’s location and the ticket from Tableau Server. It will use this information to display the view. How you specify it depends on whether the view is embedded, and if Tableau Server is running multiple sites.

**Tableau Server View Examples**

Here’s an example of how to specify a view that users only access via Tableau Server (the view is not embedded):

http://tabserver/trusted/<ticket>/views/<workbook>/<view>

If Tableau Server is running multiple sites and the view is on a site other than the Default site, you need to add t/<site ID> to the path. For example:

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

Use the same capitalization that you see in the Tableau Server URL.

**Embedded View Examples**

Here are some examples of how to specify embedded views. Because there are two approaches you can take with embed code, both ways are provided below. Regardless of which you use, there is some information unique to trusted authentication that you must provide. For more information, search for "Writing Embed Code" in the Tableau Server Help.

**Script Tag Examples**

This example uses the ticket object parameter:

```html
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
```
Here’s what the above example looks like for a multi-site Tableau Server, where the view is published on the Sales site:

```
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value="Etdpsm_Ew6rJY-9kRrALjauU" />
</object>
```

Instead of using `ticket`, you can use the `path` parameter to state the full path of the view explicitly. When `path` is used, you do not also need the `name` parameter, which is usually a required parameter in Tableau JavaScript embed code:

```
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="site_root" value="/t/Sales" />
  <param name="path" value="trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/MyCoSales/SalesScoreCard" />
</object>
```

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

```
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="path" value="trusted/Etdpsm_Ew6rJY-9kRrALjauU/t/Sales/views/MyCoSales/SalesScoreCard" />
</object>
```

**Iframe Tag Example**
Optional: Configure Client IP Matching

By default, Tableau Server does not consider the client web browser IP address when it creates or redeems tickets. To change this, you need to do two things: specify an IP address using the client_ip parameter in the POST request that obtains the ticket, and follow the steps below to configure Tableau Server to enforce client IP address matching.

1. Open a command window and change directories to the location of Tableau Server's bin directory. The default location is `C:\Program Files\Tableau\Tableau Server\10.0\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
   ```

Troubleshoot Trusted Authentication

Below are some common issues and errors you might encounter when you're configuring trusted authentication. Trusted authentication information is written to `ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\vizql*.log`. To increase the logging level from info to debug, use the `tabadmin` setting `vizqlserver.trustedticket.log_level`.

For tips on testing trusted authentication, see the Tableau Knowledge Base.

Ticket Value of -1 Returned from Tableau Server

Tableau Server returns -1 for the ticket value if it cannot issue the ticket as part of the trusted authentication process. The exact reason for this message is written to the `vizql*.log` files in the following folder:

`ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver`

Here are some things to confirm:
- All web server host names or IP addresses are added to trusted hosts

The IP address or host name for the computer sending the POST request must be in the list of trusted hosts on Tableau Server. See Add Trusted IP Addresses or Host Names to Tableau Server on page 1035 to learn how to add IP addresses or host names to this list.

- Value of wgserver.trusted_hosts is properly formatted

The list of trusted hosts you provided using the wgserver.trusted_hosts setting must be a comma-separated list with a space after each comma. For example, "the list should be similar to the following: 192.168.1.101, 192.168.1.102, 192.168.1.103, or bigbox1.example.lan, bixbox2.example.lan, bigbox3.example.lan.

- IP addresses are IPv4

If you are using IP addresses to specify trusted hosts, they must be in Internet Protocol version 4 (IPv4) format. An IPv4 address looks like this: 123.456.7.890. IPv6 addresses (for example, fe12::3c4a:5eab:6789:01c%34) are not supported as a way of inputting trusted hosts.

- Username in POST request is a valid Tableau Server user

The username you send in the POST request must be a licensed Tableau Server user with a Viewer or Interactor license level. You can see a list of users and their license levels by signing in to Tableau Server as an administrator and clicking the Licensing link on the left side of the page.

- Username in POST request includes domain

If Tableau Server is configured to use Local Authentication, the username that you send in the POST can be a simple string. However, if the server is configured for Active Directory you must include the domain name with the user name (domain\username). For example, the username parameter might be: username=dev\jsmith

- Content-Type is specified

If you are designing an ASP.NET or C# application, you need to declare the content type in your HTTP request. For example, http.setRequestHeader("Content-Type","application/x-www-form-urlencoded; charset=UTF-8"). If you do not specify content type and Tableau Server returns a -1, the log files contain the error: "missing username and/or client_ip".

HTTP 401 - Not Authorized

If you receive a 401- Not Authorized error, you may have configured Tableau Server to use Active Directory with SSPI (see Enable automatic login). If your web server uses SSPI, you do not need to set up trusted authentication. You can embed views and your users will have access to them as long as they are licensed Tableau server users and members of your Active Directory.
HTTP 404 - File Not Found

You may receive this error if your program code references a Tableau Server URL that does not exist. For example, your web server may construct an invalid URL that cannot be found when the webpage tries to retrieve it.

Invalid User (SharePoint or C#)

You may encounter this error if you’ve configured Tableau Server for trusted authentication. The example code for the SharePoint .dll references the following GET request:

```
```

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:

```
```

After you make the change, recompile the SharePoint .dll.

Attempting to Retrieve the Ticket from the Wrong IP Address

You may encounter this error if you’ve configured Tableau Server for trusted authentication. The client web browser IP address is not considered by default when redeeming the ticket. If Tableau Server is configured to enforce client IP address matching, make sure that the client's web browser IP address that is sent in the POST to Tableau Server is the same as when the browser tries to retrieve the embedded view. For example, in the Trusted Authentication diagram, if the POST request in step 3 sends the parameter client_ip=74.125.19.147, then the GET request in step 5 must come from that same IP address.

See Optional: Configure Client IP Matching on page 1039 to learn how to configure Tableau Server to enforce client IP address matching.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as the browser's address bar, it is considered a first-party cookie. If a user's browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.
An error occurred communicating with the server (403)

If Tableau Server is configured for trusted authentication, you may receive this error after opening a new view in a browser and attempting to navigate back to views you'd opened earlier. Tableau Server provides protection against unauthorized reuse of VizQL sessions through the tabadmin set option `vizqlserver.protect_sessions`, which is set to `true` by default. Because Tableau Server is configured for trusted authentication, you may not also need to enable `vizqlserver.protect_sessions`. To disable it, use `set` on page 1286 to change it to `false`.

**SQL Server Impersonation**

Impersonation in the context of Tableau Server means allowing one user account to act on behalf of another user account. You can configure Tableau and Microsoft SQL Server to perform database user impersonation, so that the SQL Server database account used by Tableau Server queries on behalf of SQL Server database users, who are also Tableau users.

The main benefit of this feature is it allows administrators to implement and control their data security policy in one place: their databases. When Tableau users access a view with a live connection to a SQL Server database, the view only displays what the users’ database permissions authorize them to see. An additional benefit is that the users don't have to respond to a database sign-in prompt when they open the view. Also, workbook publishers don't have to rely on user-specific filters to restrict what's seen in views.

Use the topics below for more information on what you need to use this feature.

**Impersonation Requirements**

Here’s what you need to use feature:

- **Live connections to SQL Server only**: Impersonation can only be used for views that have a live connection to a SQL Server database, version 2005 or newer.

- **Individual database accounts**: Each person who'll be accessing the view must have an explicit, individual account in the SQL Server database to which the view connects. Members of an Active Directory (AD) group cannot be impersonated. For example, if Jane Smith is a member of the AD group Sales, and her database administrator adds the Sales AD group to the SQL Server database, Jane cannot be impersonated.

- **Matching credentials and authentication type**: The credentials of each Tableau user's account and their Tableau user authentication type must match their credentials and authentication type in the SQL Server database. In other words, if Jane Smith’s Tableau Server user account has a username of MyCo\jsmith and Tableau Server is using Active Directory for user authentication, her username on the SQL Server database must also be MyCo\jsmith and SQL Server must be using Windows Integrated Authentication.

- **SQL Server prerequisites**: In SQL Server you should have a data security table, a
view that enforces data security, and you should require that your database users use the view.

- **SQL IMPERSONATE account**: You need a SQL Server database account that has IMPERSONATE permission for the above database users. This is either an account with the sysadmin role or one that has been granted IMPERSONATE permission for each individual user account (see the MSDN article on EXECUTE AS). This SQL Server account must also be one of two accounts on the Tableau side of things:
  - The Tableau Server Run As User account (see Impersonate with a Run As User Account on the next page).
  - The workbook publisher's account (see Impersonate with Embedded SQL Credentials on page 1046).

**How Impersonation Works**

Here’s an illustration of how database user impersonation works:

In the above illustration, Jane Smith (MyCo\jsmith) is a West Coast sales representative and Henry Wilson (MyCo\hwilson) covers the East. In the SQL Server database, the account permissions for Jane’s account, MyCo\jsmith, only give her access to West Coast data. Henry’s account, MyCo\hwilson, can only access data for the East Coast.

A view has been created that displays data for the entire country. It has a live connection to a SQL Server database. Both users sign in to Tableau Server and click the view. Tableau Server connects to SQL Server using a database account with IMPERSONATE permission for each user’s database account. This account acts on behalf of each user's database account.

When the view displays, it is restricted by each user's individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.
Impersonate with a Run As User Account

Impersonating via a Run As User account is the recommended way to perform impersonation. The Run As User account is an Active Directory (AD) account the Tableau Server service can run under on the machine hosting Tableau Server (see Run As User on page 577). This same account must have IMPERSONATE permission for the database user accounts in SQL Server. From a data security standpoint, using the Tableau Server Run As account for impersonation gives the administrator the most control.

To set up impersonation with a Run As User account:

1. When you configure Tableau Server as part of Setup, under Server Run As User, enter the Run As User AD account that has IMPERSONATE permission for the user accounts. Under User Authentication, select Use Active Directory:

2. Click OK to finish configuration.

3. Create a workbook in Tableau Desktop. When you create the data connection, select Use Windows NT Integrated security for the workbook's live connection to a SQL Server database:
4. In Tableau Desktop, publish the workbook to Tableau Server (Server > Publish Workbook).

5. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via server Run As account** from the drop-down list:

6. Click **OK**.

7. Test the connection by signing into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.
**Impersonate with Embedded SQL Credentials**

You can also perform impersonation by having the person who publishes a view embed their SQL Server account credentials in the view. Tableau Server can be running under any type of account, but it will use these credentials, supplied by the publisher, to connect to the database.

This may be the right choice for your site if the account that handles the impersonation cannot be an Active Directory (AD) account and if you're comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

**Note:**

To use this approach, *Embedded Credentials* must be enabled on the server Settings page in Tableau Server:

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:

   ![SQL Server Connection](image)

   - Step 1: Select or enter a server name:
     - MSSQL2008_R2
   - Step 2: Enter information to log on to the server:
     - Use a specific username and password:
       - Username: sqluser
       - Password: ********

2. Publish the workbook to Tableau Server (*Server > Publish Workbook*).
3. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select `Impersonate via embedded password` from the drop-down list:
4. Click OK.

5. Test the connection by signing in to Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

**Troubleshoot 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.

This article describes some common issues you may encounter after enabling impersonation and how to troubleshoot them.

**Tableau Server view fails to load**

There are several potential causes for a Tableau Server view failing to load:

- Account performing impersonation doesn't have IMPERSONATE permission for the database user account of the person who's trying to access the view. Depending on how you've configured impersonation, the account doing the impersonation is either the server Run As User account or the account whose credentials are being embedded in the view. See "Granting IMPERSONATE Permission for a User" section, below.
User credentials don't match. The credentials of each Tableau Server user's account must match their credentials in the SQL Server database. In other words, if Jane Smith's Tableau Server user account has a username of MYCO\jsmith, her username on the SQL Server database must also be MYCO\jsmith.

User authentication type doesn't match. If you've configured Tableau Server to use Active Directory to authenticate users, the SQL Server database must also be using Active Directory (in SQL Server 2008, it's called Windows Authentication). Alternatively, if Tableau Server is using Local Authentication to authenticate its users, SQL Server must also be using "local" authentication for its users. In SQL Server this is called SQL Server Authentication.

Tableau Server view shows too much or incorrect data

If a published view shows too much or incorrect data, it could be for one of the following reasons:

- Impersonation is not enabled. The workbook author did not enable impersonation when he or she published the view. See Impersonate with a Run As User Account on page 1044.
- Live database connection/impersonation is not being used. The workbook author created a data extract instead of creating a live connection to a SQL Server data source and enabling impersonation. See Impersonate with Embedded SQL Credentials on page 1046.
- The SQL Server database view is incorrect. If you have configured impersonation correctly but still have a view that is showing too much data or the wrong data, it could be because your SQL Server database view is not correctly configured. See "SQL Server Prerequisites" section, below.
- The SQL Server data security lookup table has incorrect mappings. This could also be the cause of a view displaying too much or incorrect data. See "SQL Server Prerequisites" section, below.

Tableau Server view prompts for credentials

If the person attempting to access a view is prompted for credentials, the workbook author did not enable Impersonate via embedded credentials or Impersonate via server Run As User account when they published the workbook.

Publish preview shows different data than seen on desktop

When a workbook author publishes a view, they are prompted to log into Tableau Server. After successfully publishing a view, Tableau will show a preview of that view as it appears on Tableau Server. If that author's current Windows login is different from their Tableau Server user credentials, the view they see in Desktop while they're authoring may look different than the preview of the view they see after they publish. The preview reflects the permissions of the account they used to log into Tableau Server.
Workbook author doesn't see the "Impersonate via embedded password" option

Workbook authors who use impersonation via embedded credentials select the following option when they publish:

If an author does not see the above option in Tableau Desktop's Authentication dialog box, the Tableau Server administrator needs to enable Embedded Credentials (All Sites > Settings):

**SQL server prerequisites**

The power of Tableau's impersonation feature is that it leverages the data security model you've already created in SQL Server. This topic won't attempt to describe how to set that up, but on a very high level, the minimum you need to use Tableau's impersonation feature is a data security table in SQL Server and a view for enforcing data security. The following example
will get you started. For specific guidance on how to use and configure SQL Server to secure your data, see your Microsoft SQL Server documentation.

First, assume you have the following data security table (for example, `[UserAccess]`) in your BigSales database:

<table>
<thead>
<tr>
<th>uaID</th>
<th>uaMarket</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYCO\jsmith</td>
<td>West</td>
</tr>
<tr>
<td>MYCO\hwilson</td>
<td>East</td>
</tr>
</tbody>
</table>

The following SQL Server command would create a view that enforces data security so that jsmith only sees sales data from states in the West and hwilson only sees data from states in the eastern sales territory:

```
CREATE VIEW dbo.BigSales AS
SELECT *
FROM dbo.Sales
JOIN dbo.UserAccess ua
ON Market = ua.uaMarket
WHERE ua.uaID = SUSER_SNAME()
```

**Granting IMPERSONATE permission for a user**

The following example illustrates how to create an account in SQL Server then grant it IMPERSONATE permission for another account. In the example, Tableau Server is running under an Active Directory account named TableauServer. The domain is MYCO. The following command creates a "matching" account in SQL Server:

```
CREATE USER [MYCO\TableauServer] FOR LOGIN [MYCO\TableauServer]
WITH DEFAULT_SCHEMA=[dbo];
```

The next command grants `[MYCO\TableauServer]` IMPERSONATE permission for Jane Smith ([MYCO\jsmith]). Jane Smith is a Tableau Server user and has an individual account in the SQL Server database.

```
GRANT IMPERSONATE ON USER::[MYCO\jsmith] to [TSI\TableauServer];
```

The GRANT must be performed for each database user account to be impersonated.

**OpenID Connect**

You can configure Tableau Server to support OpenID Connect for single sign-in (SSO). OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server.
Configuring OpenID Connect involves several steps. The topics in this section provide general information about using Tableau Server with OpenID Connect, and provide a sequence for configuring the IdP and Tableau Server.

**Requirements for Using OpenID Connect**

To use OpenID Connect with Tableau Server, you must have the following.

**IdP account**

You must have access to an IdP that supports the protocol, such as Google. You must also have an account with the IdP.

**Local authentication**

To use OpenID Connect on Tableau Server, the server must be configured to use local authentication—that is, the server must be configured so that you explicitly create users on the Tableau Server, rather than importing them from Active Directory. Active Directory authentication is not supported.

**User names with email addresses**

In Tableau Server, each user who can sign in must have an existing identity in Tableau Server—that is, you must previously have created a user for each person who will sign in. By default, the user’s user name in Tableau Server must match the user name in the IdP, and it must be an email address—for example, if you use Google as the IdP, the user name in Tableau Server must be the user’s Gmail address (alice@gmail.com). Using a complete email address in this way helps to guarantee the uniqueness of the user name in Tableau Server, even when two users have the same email but are on different email hosts.

**Note:** When you create a user identity in Tableau Server, you specify a user name, password, and optionally an email address. For using OpenID Connect, the user name is the value that must match the user’s name in the IdP. (The optional email address in the Tableau Server user identity is not used for OpenID authentication.)

**Ignoring the domain name**

You can configure Tableau to ignore the domain portion of an email address when matching the IdP user name in Tableau Server. In this scenario, the user name in the IdP might be alice@example.com, but this will match a user named alice in Tableau Server. Ignoring the domain name might be useful if you already have users defined in Tableau Server whose names match IdP user names except for the domain.

**Important:** We do not recommend ignoring the user domain name without taking precautions. Specifically, verify that user names are unique across the configured domains that you’ve created in your IdP.
Setting Tableau Server to ignore the user domain name has the potential to result in unintended user log on. Consider the case where your IdP has been configured for multiple domains (example.com and tableau.com). If two users with the same first name, but different user accounts (alice@tableau.com and alice@example.com) are in your organization, then the first one to complete the OpenID provisioning sequence will claim the sub mapping in the IdP. If the wrong user is mapped, then the other user will be unable to log on until the associated sub value is reset.

To configure Tableau Server to ignore domain names in user names from the IdP, use the following sequence of tabadmin commands:

```bash
tabadmin stop
tabadmin set vizportal.openid.ignore_domain true
tabadmin configure
tabadmin start
```

When you change the `vizportal.openid.ignore_domain` setting to ignore the domain in user names, all user names in Tableau Server must have a domain name.

**How Tableau Server Works with OpenID Connect**

OpenID Connect is a flexible protocol that supports many options for the information that’s exchanged between a service provider (here, Tableau Server) and an IdP. The following list provides details about the Tableau Server implementation of OpenID Connect. These details can help you understand what types of information Tableau Server sends and expects, and how to configure an IdP.

- Tableau Server supports only the OpenID Implicit Flow as described in the OpenID Connect final specification.
- Tableau Server relies on using discovery or a provider URL to retrieve the OpenID Provider metadata.
- Tableau Server supports only the `client_secret_jwt` Client Authentication method specified in the OpenID Connect specification. In addition, Tableau Server supports only RSA Asymmetric Encryption for handling the JWT.
- Tableau Server expects a `kid` value in the `id_token` attribute's JOSE Header. This value is matched with one of the keys found in the JWK Set document, whose URI is specified by the `jwks_uri` value in the OpenID discovery document. A `kid` value must be present even if there is only one key in the JWK Set document.
- Tableau Server does include OpenID support for the JWK `x5c` parameter or for using X.509 certificates.
Mobile users cannot sign in to Tableau Server using the Tableau Mobile app because the app does not support OpenID Connect, but they can sign in with a web browser.

For more information about OpenID Connect, see the following:

- OpenID Connect Core 1.0 incorporating errata set 1
- OpenID Connect Discovery 1.0 incorporating errata set 1

**Configure the Identity Provider (IdP) for OpenID Connect**

This topic provides information about configuring an identity provider (IdP) to use OpenID Connect with Tableau Server. This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- [OpenID Connect](#) on page 1050
- Configure the Identity Provider (IdP) for OpenID Connect (you are here)
- [Configure Tableau Server for OpenID Connect](#) on the next page
- [Signing In to Tableau Server Using OpenID Connect](#) on page 1056
- [Changing IdPs in Tableau Server for OpenID Connect](#) on page 1058

**Configure the IdP**

Before you can use OpenID Connect with Tableau Server, you must have an account with an IdP and a project or application with the IdP. When you configure Tableau Server, you will need to be able to provide the following information:

- Provider client ID. This is the identifier that the IdP assigned to your application.
- Provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.
- Provider configuration URL. This is the URL at the provider's site that Tableau Server should send authentication requests to.

The following procedure provides an outline of the steps that you follow with the provider. As an example, the procedure discusses using Google as a provider. However, each provider has a somewhat different flow, so the specifics of the steps (and their order) might vary depending on your provider.

1. Register at the provider's developer site and sign in. For example, for Google, you can go to the Developers Console at this URL: [https://console.developers.google.com](https://console.developers.google.com)
2. Create a new project, application, or relying party account.
3. In the developer dashboard, follow the steps for getting an OAuth 2.0 client ID and client secret. Record these values for later.
4. On the developer site, find the URL of the endpoint that the IdP uses for OpenID Connect discovery. For example, Google uses the URL https://accounts.google.com/.well-known/openid-configuration. Record this URL for later.

The IdP configuration requires an additional step that you cannot finish until after you've configured Tableau Server, as described in Configure Tableau Server for OpenID Connect below.

Configure Tableau Server for OpenID Connect

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

- OpenID Connect on page 1050
- Configure the Identity Provider (IdP) for OpenID Connect on the previous page
- Configure Tableau Server for OpenID Connect (you are here)
- Signing In to Tableau Server Using OpenID Connect on page 1056
- Changing IdPs in Tableau Server for OpenID Connect on page 1058

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

Important notes

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

- You can use OpenID Connect with Tableau Server only if the server is configured to use local authentication. OpenID Connect is not available if the server is configured to use Active Directory authentication. For more information, see Configure General Server Options on page 608.

- We recommend that you configure Tableau Server to use SSL for external communications. This helps to maintain secure communications between Tableau Server and the IdP during the exchange of authentication information. For details, see Configure External SSL on page 972.

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

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

**Configure the server**

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

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

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

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

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

   **Note:** Make sure that you update the external URL if the default value is not the URL for how your server can be reached from an external source.
9. Copy the URL in the box labeled **Configure the OpenID provider using the following redirect URL for Tableau Server**. You'll use this value in the next procedure to finish configuring the IdP.

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

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

### Add the redirect URL to the IdP configuration

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

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

### Signing In to Tableau Server Using OpenID Connect

This topic provides information about signing in to Tableau Server using OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- **OpenID Connect** on page 1050
- **Configure the Identity Provider (IdP) for OpenID Connect** on page 1053
- **Configure Tableau Server for OpenID Connect** on page 1054
Signing In to Tableau Server Using OpenID Connect (you are here)

Changing IdPs in Tableau Server for OpenID Connect on the next page

Signing in using OpenID Connect

Once Tableau Server has been configured to use OpenID Connect, users who access the server and aren't already signed in are redirected to the IdP site, where they are prompted to sign in. Users enter the credentials that they have with the IdP. In many cases, the user is also asked to authorize the IdP to share information with Tableau Server, as in the following example:

When a user signs in using OpenID Connect, the IdP sends a unique user identifier (known in OpenID as the sub value) as part of the information that's redirected to Tableau Server. This sub value is associated with the user's Tableau user identity.

Note: Tableau Server does not support using OpenID Connect to sign in from mobile devices.

Restricting sign-in to server administrators for command-line tools

Command-line tools for working with Tableau Server (tabcmd, tabadmin, and tableau.com) do not support sign-in using OpenID Connect. When OpenID Connect is
enabled for the server, these tools still require sign-in using a Tableau Server username and password.

Even if users normally authenticate using OpenID Connect, each user has a Tableau Server username and password. This means that users could use command-line tools like `tabcmd`. As a security measure, you can make sure that only server administrators can use command-line tools. To do this, use `tabadmin set` to set

`wgserver.authentication.restricted` to `true`. When this setting is `true`, only server administrators can sign in to Tableau Server using a username and password; all other users must sign in to the server using a single sign-on (SSO) option like OpenID Connect. The effect is that users who are not administrators also cannot then use command-line tools. To make this change, do the following:

1. Stop the server.
2. Run the following sequence of `tabadmin` commands:
   ```bash
   tabadmin set wgserver.authentication.restricted true
   tabadmin configure
   ```
3. Start the server.

### Changing IdPs in Tableau Server for OpenID Connect

This topic provides information about changing an identity provider (IdP) if you have configured Tableau Server to use OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect on page 1050
- Configure the Identity Provider (IdP) for OpenID Connect on page 1053
- Configure Tableau Server for OpenID Connect on page 1054
- Signing In to Tableau Server Using OpenID Connect on page 1056
- Changing IdPs in Tableau Server for OpenID Connect (you are here)

### Changing providers

You might decide to change the IdP that Tableau Server is configured to use. To do so, you follow the procedure that you used to configure the first IdP: establish an account, get a customer ID and secret, configure Tableau Server with that information, and provide the IdP with the redirect URL for Tableau Server. For more information, see Configure Tableau Server for OpenID Connect on page 1054.

However, you also need to perform an additional step: you must clear any user identifiers (sub values) that have already been associated with Tableau Server users. The new IdP will have different sub values for each user, and you must clear the existing ones so that Tableau Server can store a new sub value when the user signs in using the new IdP.
To clear sub values for users, use the tabadmin reset_openid_sub command. You can reset (that is, clear) sub values for an individual user, as in the following example:

```
tabadmin reset_openid_sub Alice
```

You can also clear the sub value for all users using this command:

```
tabadmin reset_openid_sub all
```

**OpenID Connect Authentication Request Parameters**

The OpenID authentication request sent from Tableau Server passes information using a limited set of parameters, as listed in this topic. If your OpenID IdP requires parameters that are not on the list above, it is not compatible for use with Tableau Server.

- **scope.** This value specifies a profile that tells the IdP what user information claims to return. This value can be configured by a Tableau Server administrator. The default value is "openid email profile". For more information, see Configure the scope value later in this document.

- **response_type.** OpenID Connect supports multiple flows. This value tells the IdP which flow Tableau Server expects. Tableau supports only the authorization code flow, and the value is always set to "code".

- **client_id.** This value specifies the server's ID (Provider client ID in the Tableau Server Configuration dialog box), which lets the IdP knows where the request came from. It is provided by the IdP when the service is registered. The value is configurable by a Tableau Server administrator.

- **redirect_uri.** This value specifies the URL that the IdP redirects to after the user has authenticated using OpenID Connect. The URL must include the host and protocol (for example, http://example.tableau.com), but Tableau provides the URL endpoint.

- **nonce.** Tableau Server generates a nonce value to verify that the client that it redirected to matches the entity that comes back from the IdP.

**Configure the scope value**

The **scope** value indicates to the IdP the information that Tableau Server requests about the user. By default, Tableau Server sends the value "openid profile email". This indicates that Tableau uses OpenID to authenticate (this part of the scope attribute value must always be included) and that Tableau Server is requesting the user profile and email information during the exchange of the user authorization code.

If this default scope is not appropriate for your scenario, you can have Tableau Server request custom information about the user. To do so, you configure the IdP with a custom profile (for example, something like "tableau-scope"). You can then configure Tableau Server to send the scope request using the name of the custom profile.
To change the scope value that Tableau Server requests, use the following tabadmin command:

```
tabadmin set vizportal.openid.custom_scope custom-scope-name
```

**Note:** Tableau Server always includes "openid" as part of the scope value (even if you don't include it in the custom_scope setting).

**Troubleshoot OpenID Connect**

Use the following topics to troubleshoot OpenID Connect issues in Tableau Server.

**Signing In from the Command Line**

Even if Tableau Server is configured to use OpenID, it is not used if you sign in to Tableau Server using tabcmd on page 1315, the REST API, or the Tableau Data Extract command line utility (provided with Tableau Desktop).

**Login Failed**

Login can fail with the following message:

```
Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.
```

This error typically means that there is a mismatch between a username stored in Tableau Server and the username provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as jsmith it must be stored in Tableau Server as jsmith as well.

**OpenID Error Log**

OpenID authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, sign-in attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see Archive Log Files on page 1184.

**Note:** To log OpenID-related events, vizportal.log.level must be set to debug. For more information, see Change Logging Levels on page 1197.

Check for OpenID errors in the following files in the unzipped log file snapshot:

```
\vizportal\vizportal-<n>.log
```
OAuth Connections

For Google BigQuery, Google Analytics, Salesforce.com, and some web data connector data sources, an alternative to storing your sensitive database credentials with Tableau Server is to create connections using the OAuth 2.0 standard.

From Tableau, when you sign in to data with a provider that uses OAuth, you are redirected to the provider’s sign-in page. After you provide your credentials and authorize Tableau to access your data, the data provider sends Tableau an **access token** that uniquely identifies requests from Tableau. For more information, see **Overview of the OAuth process** below below.

Using OAuth connections provides the following benefits:

- **Security**: Your database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau.
- **Convenience**: Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider.

  In addition, for live connections to Google BigQuery data, each workbook viewer can have a unique access token that identifies the user, rather than sharing a single user name and password credential.

**Overview of the OAuth process**

The following steps describe a workflow in the Tableau environment that calls the OAuth process.

1. You take an action that requires access to a cloud data source.
   
   For example, you open a workbook that’s published to Tableau Server.

2. Tableau directs you to the cloud data provider’s sign-in page. The information that is sent to the data provider identifies Tableau as the requesting site.

3. When you sign in to the data, the provider prompts you to confirm your authorization for Tableau Server to access the data.

4. Upon your confirmation, the data provider sends an access token back to Tableau Server.
5. Tableau Server presents your workbook and data to you.

The following workflows can use the OAuth process:

- Creating a workbook and connecting to the data source from Tableau Desktop or from Tableau Server.
- Publishing a data source from Tableau Desktop.
- Signing in to Tableau Server from an approved client, such as Tableau Mobile or Tableau Desktop.

**Access tokens for data connections**

You can embed credentials based on access tokens with data connections, to enable direct access after the initial authentication process. An access token is valid until a Tableau Server user deletes it, or the data provider revokes it.

It is possible to exceed the number of access tokens your data source provider allows. If that's the case, when a user creates a new token, the data provider uses length of time since last access to decide which token to invalidate to make room for the new one.

**Access tokens for authentication from approved clients**

By default, Tableau Server sites allow users to access their sites directly from approved Tableau clients, after users provide their credentials the first time they sign in. This type of authentication also uses OAuth access tokens to store the users’ credentials securely.

For more information, see **Disable Automatic Client Authentication** on page 1175
Configure the Server for OAuth Support

Instead of individual usernames and passwords, OAuth works through limited-purpose access tokens. Before you can obtain access tokens needed to create an OAuth connection in Tableau, you need to configure your server so that the data provider sending the token can recognize Tableau Server as a trusted destination. The following section describes how to prepare for setting up OAuth regardless of your data provider. The topics listed below it contain the steps for configuring your server for specific data providers.

Preparing for Configuring OAuth Support

Before you begin the configuration steps specific to your data provider, complete the following prerequisites:

- Obtain the fully qualified domain name of each Tableau Server node that will host views that connect to this data source. For example:
  https://sales.your_domain.com
  If you use Salesforce.com, you will need to provide an https address.
- Make sure at least one of your data-provider accounts is enabled for API access.
  For Google data types, you need access to the developers console on the Google Cloud Platform.
  For Salesforce.com, you need access to the Force.com platform.
  For QuickBooks Online, you need access to the Intuit platform.

Configure Settings for Your Data Provider

When you complete the OAuth-preparation steps, you can configure the appropriate settings with your data provider.

- Set up OAuth for Google below
- Set up OAuth for Salesforce.com on page 1066
- Set up OAuth for QuickBooks Online on page 1069

Set up OAuth for Google

This topic describes how to set up your Google BigQuery and Google Analytics data sources for OAuth. Complete these steps for each Tableau Server instance.

**Note** Before you complete these steps, make sure you have completed the prerequisites described in Preparing for Configuring OAuth Support above.

Set up OAuth by following these two procedures:
• Get required information from Google and enable API access.
• Use the information you obtained to configure your server.

Obtain a Client ID and Enable Google APIs

**Note** These steps reflect the settings in the Google Cloud Platform console at the time of this writing. For more information, see Using OAuth 2.0 for Web Server Applications in the Google Developers Console Help.

1. Sign in to Google Cloud Platform, and then click Go to my console.

![Google Cloud Platform Console](image)

2. Select Projects, and on the Project page, click Create Project.

![Create Project](image)

3. In the new project form that appears, complete the following:
   - Give the project a meaningful name that reflects the Tableau Server instance for which you’ll use this project.
   - Determine whether you want to change the project ID.

**Note** After you create the project, you will not be able to change the project ID.
4. Open the new project, and navigate to APIs & auth > Credentials.

5. Click Create a New Client ID, and in the Create Client ID page, complete the following:
   - Select Web Application.
   - For Authorized JavaScript Origins, type the local computer name of your Tableau Server.
   - For Authorized Redirect URI, replace the existing text with the Internet address for your server, and add the following text to the end of it: auth/add_oauth_token. For example:
     
     https://your_server_url.com/auth/add_oauth_token

6. Copy the Authorized Redirect URI, and paste it in a location that you can access from your Tableau Server computer.

7. Click Create Client ID.

8. Copy the following values that Google returns, and paste them in a location that you can access from your Tableau Server computer:
   - Client ID
   - Client secret

9. In the Google Developer Console, with your new project open, select APIs & auth > APIs, and then set the status to On for BigQuery API or Analytics API.
Configure Tableau Server for Google OAuth

Using the information you obtained by completing the steps in Obtain a Client ID and Enable Google APIs on page 1064, configure your Tableau Server:

1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory.
   
   ```
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:
   
   ```
   tabadmin stop
   ```

3. Type the following commands to configure the server with the client ID and client secret you obtained from Google, as well as your server URI. Press Enter after each command.
   
   ```
   tabadmin set oauth.google.client_id <your_client_ID>
   tabadmin set oauth.google.client_secret <your_client_secret>
   tabadmin set oauth.google.redirect_uri <your_authorized_redirect_URI>
   ```

4. Type the following commands to complete the configuration and restart the server:
   
   ```
   tabadmin config
   tabadmin start
   ```

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens on page 1070.

Set up OAuth for Salesforce.com

This topic describes how to set up your Salesforce.com data sources for OAuth. Complete these steps for each Tableau Server instance.

**Note:** Before you complete these steps, make sure you have completed the prerequisites described in Preparing for Configuring OAuth Support on page 1063.
Set up OAuth by following these two procedures:

- Create a Connected App in Salesforce
- Use the information you obtained to configure your server.

**Create a Connected Salesforce App**

1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.

2. In the left navigation column, under App Setup, select **Create > Apps**.

3. In the Connected Apps section, click **New**.

4. Complete the **Basic Information**, and in the API section, select **Enable OAuth Settings**.

5. In the new OAuth settings that appear, for **Callback URL**, type the fully qualified domain name of your server, using the https protocol, and append the following text to the URL: `auth/add_oauth_token`.

   For example:
https://www.your_server.com/auth/add_oauth_token

6. Move the following items from Available OAuth Scopes to Selected OAuth Scopes:
   - Access and manage your data (api)
   - Access your basic information (id)
   - Perform requests on your behalf at any time (refresh_token)

![Available OAuth Scopes](image)

7. Click **Save**.

After you save the app, Salesforce populates the API section with the following IDs that you will use to configure Tableau Server:

- Consumer Key
- Consumer Secret
- Callback URL

**Configure Tableau Server for Salesforce.com OAuth**

1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory:
   
   ```
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:
   
   ```
   tabadmin stop
   ```

3. Type the following commands to configure the server with the consumer ID and secret you obtained from Salesforce and the callback URL. Press **Enter** after each command:
   
   ```
   tabadmin set oauth.salesforce.client_id <your_consumer_ID>
   tabadmin set oauth.salesforce.client_secret <your_consumer_secret>
   tabadmin set oauth.salesforce.redirect_uri <your_callback_URL>
   ```

4. (Optional) To change the default login server, type the following command:
tabadmin set oAuth.salesforce.server_base_url <URL>

By default, this is set to https://login.salesforce.com.

5. Type the following commands to complete the configuration and restart the server:
   
   tabadmin config
   
   tabadmin start

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens on the next page.

Set up OAuth for QuickBooks Online

This topic describes how to set up your QuickBooks Online data sources for OAuth authentication. Complete these steps for each Tableau Server instance.

Note: Before you complete these steps, make sure you have completed the prerequisites described in Preparing for Configuring OAuth Support on page 1063.

Setting up OAuth for QuickBooks Online consists of two tasks:

- Create a Connected App on the Intuit developer platform.
- Use the information you get as part of the Connected App to configure your server.

Create a Connected Intuit App

1. Sign in to your Intuit developer account, and then click My Apps.
2. In the Just start coding section, click Select APIs.
3. Select Accounting and click Create App.
4. In the Get your app ready for submission section, click the link to get your production keys.

   Important: You must use production keys rather than development keys.

5. Copy the app token, OAuth consumer key, and OAuth consumer secret.

Configure Tableau Server for QuickBooks Online

1. On the Tableau Server computer, open a command prompt as an administrator and change to the Tableau Server bin directory using the following command:

   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
2. Type the following command to stop the server:
   
   `tabadmin stop`

3. Type the following commands to configure the server with the app token, consumer key, and consumer secret that you copied from the Intuit site:
   
   `tabadmin set oauth.quickbooks.oauth_callback_uri http://YOUR-SERVER/auth/add_oauth_token`
   
   `tabadmin set oauth.quickbooks.consumer_key <your_consumer_key>`
   
   `tabadmin set oauth.quickbooks.consumer_secret <your_consumer_secret>`

4. Type the following commands to complete the configuration and restart the server:
   
   `tabadmin config`
   
   `tabadmin start`

Managing access tokens

If you run an extract refresh job for your QuickBooks Online data source, Tableau Server attempts to renew access tokens for you. To help ensure that your access tokens do not expire, run your extract refresh jobs more than once a month. Otherwise, the access tokens from QuickBooks Online expire and your extract refresh jobs fail. If your access tokens do expire, you can edit your saved credentials from the Settings page.

The saved credentials can be managed centrally or by your users. For more information, see Allow Saved Access Tokens below.

Allow Saved Access Tokens

After you configure Tableau Server for OAuth, you can decide to allow users to manage their own OAuth credentials, or you want to manage them centrally. If you want users to manage their own, you need to enable user profile settings from the server.

Note: If you have not yet configured your server to enable OAuth data connections, see the related topics listed below.

1. Sign in to Tableau Server as a server administrator.

2. **Single-site:** Click Settings > General.
   
   **Multisite:** In the site menu, click Manage All Sites and then click Settings > General.

3. In the Saved Credentials section, select the following:
- Allow users to save passwords for data sources (allows users to save their individual credentials with data sources).
- Allow users to save OAuth access tokens for data sources

4. Click **Save**.

After you select these check boxes, users will see a **Manage Credentials** section in their profile settings, where they can add access tokens for OAuth data connections.

Managing credentials centrally

Server administrators alternatively can manage OAuth credentials centrally. This can work well, for example, if multiple users work from the same data, and you have a dedicated user account for your data provider.

To manage credentials centrally, you do the following:

- Clear the check boxes described in the preceding procedure.
- Edit connection information as data sources are published.

  When you edit the connection, you embed credentials that use an OAuth access token instead of an individual’s user name and password.

When the settings for saving passwords and access tokens are not enabled, the Manage Credentials section is excluded from users’ profile settings.
Enterprise Tableau Desktop Deployment

The following topics describe how to manage Tableau Desktop installation in your enterprise or organization.

Enterprise Desktop Licensing Overview

If you are responsible for deploying or managing Tableau Desktop installations in your organization, then managing and tracking licensing entitlement, and understanding desktop usage are essential tasks.

Tableau provides two main tools that will help you track Tableau Desktop licensing and usage in your organization:

- Tableau Software customer portal: the online portal is where you purchase, manage, and view registration information for licenses assigned to your users. The portal is also where you manage your Tableau account on behalf of your organization.
- Desktop license reporting: you can configure Tableau Desktop to report usage information to an instance of Tableau Server running in your organization.

This topic describes how you can use these two tools to manage Desktop licenses and track Desktop usage in your organization.

Customer Portal: asset and account management

The Tableau Software customer portal is where you manage all elements of licensing entitlement. The portal provides you with access to your purchased license keys along with a platform upon which you can track license key assignments.

Use the portal for the following tasks:

- View your purchased license entitlements
- Track the assignment of license entitlements to specific departments and assigned users
- Monitor and compare user registration received by Tableau to your purchased and assigned licenses
- Open support cases and manage current and prior case interactions with Tableau Software
- Download Tableau installation packages
- Manage your organizational account and invoicing
The portal is not intended to provide detailed usage data, however, you can determine desktop installations in your organization according to the user registration records that are housed in the portal.

**Activation, Deactivation, and Registration**

It's important to understand that the Tableau Customer portal only logs registration events from installations in your organization. This means that as the asset manager for your organization, you must manage deactivating and reactivating licenses, as these events are not logged in the customer portal. Desktop licenses are perpetual, which means that as long as they are registered and active for an authorized user, the license key will not expire.

The reason it's important to understand these details is for scenarios where a computer with a desktop license is no longer accessible. For example, if a computer is lost, stolen, or formatted before the license has been deactivated, the license key may not always be able to be reassigned.

Therefore, it's important for users to deactivate their licenses before decommissioning a computer where Tableau Desktop is installed.

Users can deactivate licenses in Tableau Desktop by opening Manage Product Keys (Help > Manage product keys...), selecting the Product Key and then clicking Deactivate. Other command line tools are available for bulk deactivate and silent deactivate. Contact your account representative for more information.

Be sure that the license key-registration pairs that are shown in the portal match the activated Desktop-user pairs in your organization. For larger organizations, use Desktop Reporting and Tableau Server to identify activated Desktop-user pairs.

**Desktop Reporting: Monitoring usage in your organization**

Deploying Tableau Desktop in your organization shows a commitment to data analysis as a core business requirement. For many organization, quantifying the return on software investments is an important business need. Understanding how often and to what extent your users are utilizing Tableau Desktop can be important as you plan asset allocation.

After you configure desktop reporting in your organization, you can view usage reports on Tableau Server to answer questions like the following:
- What types of licenses are installed in your organization.
- Which users have Tableau licenses.
- Which licenses are used most and least often.
- Whether trial licenses need to be converted to paid licenses.
- Which licenses are expired or might soon expire.
- When maintenance renewals are due in your organization.

Desktop reporting is enabled by configuring each Tableau Desktop installation with a pointer to at least one Tableau Server in your organization. You can configure each Desktop during the install process with a command line option, or you can deploy a registry update to existing Desktop clients. For more information see Configure Tableau Desktop License Reporting on page 1081.

**Automate Tableau Desktop Installation**

This topic describes how to automate the installation of Tableau Reader, Tableau Public, or Tableau Desktop 9.3 and 10.0.

**Note:** These instructions apply to Tableau Reader, Tableau Public version of Tableau Desktop, and to Tableau Desktop starting with version 9.3. For information about how to install earlier versions of these products, see Performing a Quiet Installation of Tableau Desktop, Tableau Reader, or Tableau Public in the Tableau Knowledge Base.

This topic also describes how to automate licensing as part of your installation process.

**Tableau Installer**

In this topic, when we use the term "Tableau installer," we are referring to the .exe files that you use to install Tableau Desktop, Tableau Reader, or Tableau Public version of Tableau Desktop.

Before you begin, click the link below to open and read the End User License Agreement (EULA) for the product that you are installing.

- Tableau Desktop: www.tableau.com/eula
- Tableau Reader: www.tableau.com/eula-reader
- Tableau Public: www.tableau.com/eula-public

**Note:** By installing or using all or any portion of the software, you are accepting all of the terms and conditions of EULA as published on Tableau’s website at www.tableau.com

**Getting the Tableau Installer**

All product installers for each version are available on the Tableau customer portal.

**Running the Installer**

The Tableau installer can only be automated on computers running Windows.
Before you run the installer to install Tableau products, download the installer to a local directory on the computer where you are installing.

The installer has been optimized to run on the computer where Tableau will be installed. Do not run the installer from a shared directory on your network.

**Syntax**

The syntax for running the Tableau installer from command line is:

```
Tableau_Product_installer_name.exe /option1 /option2 PROPERTY1 PROPERTY2
```

The `Tableau_Product_installer_name.exe` file is the Tableau installer for the product and version you are installing.

For example, the following command performs the following:

- Installs Tableau Desktop version 9.3.3 in quiet mode
- Sets installation to finish without restarting
- Configures a non-default product update server
- Configures the product to send license reporting to an internal Tableau Server

```
TableauDesktop-64bit-9-3-3.exe /quiet /norestart
AUTOUPDATESERVER="assets.intranet.lan"
REPORTINGSERVER="http://mytableau"
```

You must run the command from the directory where the installer file is located.

**Installer Options**

You can specify one or more options as part of the command:

- Each option is delimited with a slash (/).
- Options must come before properties.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/quiet</td>
<td>Run the installer without messages (status or installation progress) and without requiring user interaction. The product does not launch after installation is complete.</td>
</tr>
<tr>
<td>/passive</td>
<td>Run the installer and display dialog boxes, and installation status. Does not prompt user for input. The product launches after installation is complete.</td>
</tr>
<tr>
<td>/norestart</td>
<td>Suppress any attempts to restart. By default, the Setup will prompt before restart.</td>
</tr>
<tr>
<td>/log &quot;logfile.txt&quot;</td>
<td>Log setup information to specified path and file. Specify path and file name, for example, /log &quot;c:\logs\logfile.txt&quot;. Default log file is the system %TEMP% directory.</td>
</tr>
</tbody>
</table>
/repair  Runs the installer to repair existing installation of Tableau product.

/h  Lists available options and properties that you can set on the installer.

**Installer Properties**

You can specify one or more property:

- Property names are case sensitive
- Each property value is enclosed in double quotes
- Each property set is delimited by a space

<table>
<thead>
<tr>
<th>Property (case sensitive)</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
</table>
| DESKTOPSHORTCUT | Create a desktop shortcut. | "1" = yes (default)  
"0" = no |
| STARTMENUSHORTCUT | Create a Tableau entry on the Windows Start menu. | "1" = yes (default)  
"0" = no |
| INSTALLDIR | Specify a different installation directory. | Takes a path, for example, "D:\Software\Tableau" 
Default is "%SYSTEMDRIVE%:\Program Files\Tableau" |
| DATABASEDRIVERS | Install core database drivers: MySQL, Microsoft SQL Server, PostgreSQL, and Amazon Redshift. 
The Tableau installer will install only these drivers. To download other drivers, see the Drivers & Activation page on the Tableau website. You must distribute other drivers to desktops using whatever asset management tools you use for software deployment in your organization. | "1" = yes (default)  
"0" = no |
| AUTOUPDATE | Configure Tableau product to check for updates. | "1" = yes (default)  
"0" = no |
### AUTOUPDATESERVER

Specify a server to check for Tableau product updates. You can control updates for your users by customizing and hosting the TableauAutoUpdate.xml file along with the appropriate Tableau installer.

A host name, such as `assets.internal.lan`.

Default: "downloads.tableau.com"

See [Tableau Knowledge Base](#) for information.

### REPORTINGSERVER

Specify the instance of Tableau Server where license reporting is stored. Requires Tableau Server in your organization. For full implementation information, see [Configure Tableau Desktop License Reporting](#) on page 1081.

Takes a server URL, for example, "http://mytableau"

---

**Extracting and running the Windows Installer**

You can also deploy Tableau using the Windows Installer and the associated command line options.

To use the Windows Installer, you must first extract the .msi file from the Tableau installer exe.

**Disclaimer:** This solution includes information about a third-party product. While we make every effort to keep references to third-party content accurate, WiX Toolset options might change without notice. For the most up to date information, please consult [WiX documentation](#). For assistance with the WiX utilities, contact the [WiX users mailing list](#).

Use the WiX Toolset to extract the .msi file:

1. On the computer to which the Tableau Desktop or Tableau Reader installation file has been downloaded, download and install the WiX Toolset version 3.9 from the WiX website [http://wixtoolset.org/](http://wixtoolset.org/).
2. Open Command Prompt as an administrator and navigate to the root of the folder where WiX was installed.
3. Run the following command:

   ```
   Dark.exe <path to the Tableau .EXE installer> -x <output folder>
   ```
For example,

```bash
Dark.exe C:\tableau-setup-std-tableau-9-3.16.0614.1319-x64.exe -x c:\output
```

4. The output folder specified above will contain a folder named AttachedContainer which includes the necessary MSI files.

**Licensing Tableau Desktop**

After Tableau Desktop is installed, you can automate licensing.

Use the `-activate` option when you start Tableau Desktop in order to activate the license key for Tableau Desktop. The `-activate` option takes one parameter, which is the license key.

**Windows**

Run the `-activate` option with the `tableau.exe` file as in the following:

```bash
tableau.exe -activate <license-key>
```

By default, the `tableau.exe` file is located in `C:\Program Files\Tableau\Tableau <version>\bin`.

For information about exit codes that you can capture and evaluate if licensing fails, see `Automated Licensing Task Exit Codes` in the Tableau Knowledge Base.

**Mac**

Run the `-activate` option on the Tableau object in the Applications path:

```
./Applications/Tableau.app/Contents/MacOS/Tableau -activate <license_key>
```

**Registering Tableau Desktop**

You can ensure consistent registration across your organization by automating the task of registration and by and prepopulating registration information. For details, see `Command Line Registration` on page 1085.

**Quick Start: Tableau Desktop License Reporting**

Use Tableau Desktop License Reporting to help you manage the Desktop licenses in your organization.

License Reporting is useful for organizations with large numbers of Tableau Desktop. But it can be useful to anyone who has Tableau Desktop and an installation of Tableau Server. When License Reporting is configured, Tableau Desktop sends usage information every eight hours to Tableau Server (while Tableau Desktop is running). It also sends information when a
desktop license is activated or returned. All these actions can be viewed in administrative views in Tableau Server when License Reporting is enabled in Tableau Server.

Enable Desktop License Reporting

By default, license reporting is disabled on Tableau Server. To view license information, you need to enable license reporting. Once Desktop License Reporting has been enabled on Tableau Server, Administrative Views are visible that allow you to see gathered license information. No links to these views appear until you configure Tableau Desktop to report to the server. For details on enabling license reporting on Tableau Server, see Enable Desktop License Reporting.

Configure Tableau Desktop

To gather license information in Tableau Server, each copy of Tableau Desktop needs to be configured with the name of the server you want it to report to. The easiest way to do this is with an automated install process for Tableau Desktop, but you can also configure Tableau Desktop when it's already installed. In most cases, this is best done by your IT department. For details on configuring Tableau Desktop, see Configure Tableau Desktop License Reporting on page 1081.

Tip: You can use command-line options during Tableau installation to set this key automatically. For more information, see Configure Tableau Desktop License Reporting on page 1081

View Desktop License Usage

On Tableau Server, server administrators can use the Desktop License Usage administrative view to answer questions about what types of licenses are installed in your organization, which users have Tableau licenses, which licenses are used most and least often, and if there are trial licenses that need to be converted to paid licenses. You can identify heavy usage as well as users who have not been using Tableau and might need additional training.
For more information about the License Usage view, see Desktop License Usage on page 1111.

View Desktop License Expiration

Server administrators can use the Desktop License Expiration administrative view to answer questions about license expiration and when maintenance renewals are due in your organization.
For more information about the License Expiration view, see Desktop License Expiration on page 1113.

### Filtering the Views

Both the Usage and Expiration views have filters at the top to help you control what is displayed in the views. By default, the views display information for the last 183 days. You can also click individual segments of the bar graphs (the Trial section of the upper graph on the Usage view, for example) to filter the rest of the view to only trial licenses. Click around in the views and experiment to see how different selections change what you see.

### Configure Tableau Desktop License Reporting

License reporting gathers usage information from individual instances of Tableau Desktop and stores the information centrally in Tableau Server. You can use license reporting to help you manage the Tableau Desktop licenses in your organization. License reporting is especially useful for organizations with large numbers of Tableau Desktop, but it can also be useful to anyone who has Tableau Desktop and an installation of Tableau Server.

#### How Desktop License Reporting Works

When an instance of Tableau Desktop is configured for license reporting, the instance sends information (if it's running) to Tableau Server every eight hours. It also sends information whenever a Tableau Desktop license is activated or returned. This information is stored in the Tableau Server repository and, if license reporting is enabled on the server, is viewable using built-in administrative views.

Usage information is sent in the background, using SSL if the server is configured for external SSL. Tableau Desktop users don't need to have an account on Tableau Server for the license information to be sent and saved.

### Configure Desktop License Reporting

Desktop License Reporting is disabled by default. To use the feature you need to:

1. Enable Desktop License Reporting on Tableau Server
2. Configure Tableau Desktop installations to report to at least one Tableau Server (maximum of six)

**Step 1: Enable Desktop License Reporting on Tableau Server**

Use the following commands on each server that Tableau Desktop will be reporting to:

```
tabadmin set features.DesktopReporting true
```
tabadmin config

Step 2: Configure Tableau Desktop for license reporting

You configure Tableau Desktop for license reporting by adding a Windows registry key (ReportingServer) or a Mac .plist file value (com.tableau.ReportingServer.plist) with the address of one or more Tableau Servers that the information should be sent to. You can configure the Tableau Desktop instance to send license reporting information to up to six different servers.

By default Tableau Desktop sends license reporting information to the configured server or servers every eight hours. You can change the interval frequency if needed, for example, for troubleshooting purposes. For more information, see Changing the default reporting interval on page 1226.

Update existing Tableau Desktop installations

If your organization has already deployed Tableau Desktop, you must update the registry key or .plist file value on each computer where Tableau Desktop is installed. You can do this manually or by using a third-party desktop management solutions. This section describes how to make these settings manually.

Note: The instructions in this section are intended for an IT professional who is comfortable editing the registry and adding .plist files. Make a backup of the registry or .plist file before you make any changes to it.

Windows

Edit the registry to add a ReportingServer key with these values:

- **Path:** HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ReportingServer
- **Name:** add a string value named Server.
- **Data:** Add the URL or URLs of the Tableau Server instances that the Desktop instance will send reporting data to. Include the protocol (http or https) and the port number if needed. Separate multiple addresses with a comma (,). You can include up to six addresses. Any addresses after the sixth one will be ignored.

For example, the following **Server** value configures Tableau Desktop to report to two Tableau Server instances. The first, https://mytableau02, is configured for SSL and listens on port 8010. The second, http://mytableau does not use SSL and listens on the default port, 80:


The illustration below shows what the **Server** key looks like in the registry editor.
Macintosh

To begin, make the /Library/Preferences location visible by running the following command:

defaults write com.apple.finder AppleShowAllFiles YES

Create a com.tableau.ReportingServer.plist file in /Library/Preferences with a Server key. Set this to the address of the server you want the Tableau Desktop instance to report to. The following example shows the contents of a .plist file that's configured to send information to two servers, https://mytableau02:8010 and http://mytableau.

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PropertyList 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Server</key>
  </dict>
</plist>

Configure license reporting during Tableau Desktop setup (Windows only)

If you haven't installed Tableau Desktop for Windows yet, you can use a command-line option during installation to update the registry and configure an instance of Tableau Desktop for license reporting.

Run the setup process from the command line and add the REPORTINGSERVER option to specify the server to report to:

tableau-setup-std-x64.exe REPORTINGSERVER="<server URL>"

For example, the following command installs Tableau Desktop and configures it to report to http://mytableau:

tableau-setup-std-x64.exe REPORTINGSERVER="http://mytableau"

Separate multiple server URLs with a comma. The following command installs Tableau Desktop and configures it to report to two Tableau Server instances:
Note: If you do not use a comma as a separator, only the first server will be recognized and sent license usage data.

Viewing desktop usage and expiration information on Tableau Server

After you've configured instances of Tableau Desktop for license reporting, they send usage information to Tableau Server. If you have enabled license reporting on Tableau Server, you can use the Tableau Desktop License Usage administrative view to learn about license usage.

The Tableau Desktop License Usage administrative view can answer questions like the following:

- What types of licenses are installed in your organization.
- Which users have Tableau licenses.
- Which licenses are used most and least often.
- Whether trial licenses need to be converted to paid licenses.
- Which licenses are expired or might soon expire.
- When maintenance renewals are due in your organization.

You can also use the administrative view to identify heavy usage, and you can determine whether specific users have not been using Tableau and might need additional training.
For more information, see Desktop License Usage on page 1111 and Desktop License Expiration on page 1113.

**Troubleshooting Desktop License Reporting**

If you cannot access the administrative views, or do not see data you expect in the views, see Troubleshoot Desktop License Reporting on page 1224.

**Command Line Registration**

The command line registration feature gives Tableau administrators a way to automate Tableau Desktop installation and registration, and provides control over the accuracy of registration data by allowing administrators to pre-fill and automatically send registration information.

This means that end-users don't have to register Tableau, and enterprises can be confident that the registration information is accurate. Accurate registration data helps enterprises track licenses and license use.

**Registering Tableau Desktop from the Command Line**

To register Tableau from the command line, you need to pre-fill the registration registry values on each computer that will run Tableau Desktop and then use a command line option when running Tableau. The easiest way to do this is to manually install and register Tableau on one computer, and use the registry entries that are populated by that process as a template for updating the registry on other computers.

**Note:** These instructions assume you are registering Tableau Desktop on Windows. For Tableau Desktop on the Mac, you need to follow similar steps but will be updating the Property List file com.tableau.Registration.plist. One way to do this is using Xcode or the defaults command. You can find more information on editing OS X Property List files at Apple Support.

Follow these steps to use the command line registration feature of Tableau Desktop.

Make a backup of the registry or .plist file before you make any changes to it.

1. Install and register Tableau Desktop on one Windows computer.
   
   The registration information is saved in the Windows registry under:
   
   HKEY_CURRENT_USER\Software\Tableau\Registration\Data.
   
2. Export the data (Key) to a registry text file.

3. Create an automated script to update unique user fields and update the registry for each computer that will run Tableau Desktop. The fields you update will depend on the
information you want sent with the registration. For information on how to automate updates to the registry, see the Microsoft MSDN.

Fields include location fields that may be the same from computer to computer (city, country, and zip for example), and user fields that are likely unique to each computer (email, first_name, last_name, title).

The registry key will look like this:

![Registry Editor screenshot]

4. Install Tableau Desktop.

5. Run Tableau.exe with a register option:

   ```
   tableau.exe -register
   ```

   If registration is successful, the install log file will have an entry:

   ```
   Current registration state: complete / not needed
   ```

   If registration is not successful, Tableau will close with the error code: error 117, as well as with information about invalid or missing fields. If this happens, the automated script created in step 3 needs to be updated.

**Tableau Server Performance**

When you take the time to understand the performance of Tableau Server, you make it easier to serve your users by improving the efficiency of Tableau Server. Although every server environment is unique, and there are many variables that can impact performance, the general
steps that you take to understand and act on performance data in Tableau Server are the same.

- **Alerts.** Configure email notifications for important server events. For example, you can receive notifications when server processes become unavailable and when the server is running out of disk space.

- **Monitoring.** Collect and analyze data about Tableau Server to understand how well the server is performing.

- **Tuning.** Make adjustments to tasks, process configurations, and more to improve the performance of Tableau Server.

- **Troubleshooting.** Identify bottlenecks in resources, workbooks, and more to improve the performance of Tableau Server.

**General Performance Guidelines**

**Hardware and Software**

*Add more cores and memory:* Regardless of whether you’re running Tableau Server on one computer or several, the general rule is that more CPU cores and more RAM will give you better performance. Make sure you meet the Tableau Server recommended hardware and software requirements and see the topic *When to Add Workers and Reconfigure* on page 1134 to assess whether you should add additional machines.

If you are running Tableau Server in a virtual environment, use your VM host’s best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host.

**Configuration**

*Schedule refreshes for off-peak hours:* Backup tasks tend to stall other background tasks until the backup is completed. Use the *Background Tasks for Extracts* on page 1103 administrative view to see your refresh and backup task schedules. Your refresh tasks should be scheduled for off-peak hours that don’t overlap with your backup window.

*Look at caching:* Caching helps Tableau Server respond to client requests quickly, especially for views that connect to live databases. Confirm that *Refresh Less Often* on the Data Connections tab of the Configuration dialog box is selected.

**Consider changing two session memory settings:**

- **VizQL session timeout limit:** The default VizQL session timeout limit is 30 minutes. Even if a VizQL session is idle, it is still consuming memory and CPU cycles. If you can make do with a lower limit, use *tabadmin* on page 1255 to change the `vizqlserver.session.expiry.timeout` setting.

- **VizQL clear session:** By default, VizQL sessions are kept in memory even when a user navigates away from a view. This consumes a good deal of session memory. Instead,
you can end sessions when users move away from a view by changing the value of the `vizqlserver.clear_session_on_unload` setting to `true` (the default is `false`).

Assess your process configuration: Tableau Server is divided into six different components called server processes. While their default configuration is designed to work for a broad range of scenarios, you can also reconfigure them to achieve different performance goals. Specifically, you can control on which computers the processes run and how many are run. See Performance Tuning Examples on page 1135 for general guidelines for one-, two-, and three-node deployments.

Tableau Server Alerts

Alerts are email notifications that you receive when something happens on Tableau Server. You can set up alerts for when the server is running out of disk space and for when server processes stop or start. These conditions often mean that there is an immediate problem.

Note: We discuss alerts in this section as a tool for getting information about server health. But as an entirely separate benefit, users can also make use of alerts. After you set up alerts, your users can subscribe to views to periodically receive a snapshot of views they are interested in on a recurring basis.

To send alerts, Tableau Server must connect to a mail server, also known as a Simple Mail Transfer Protocol (SMTP) server. An SMTP server is a service that you can send outbound email messages to. It then relays the messages to whoever they’re addressed to. (It doesn’t handle incoming email.) To set up alerts, you must configure Tableau Server to communicate with your SMTP mail server.

SMTP information you’ll need

Many organizations already have an SMTP server in-house. Before you continue, ask your IT department if there is an SMTP server that you can use.

Here’s the SMTP server information that you need from your IT department:

- The server address. This is often something like `smtp.example.com` or `mail.example.com`, but other addresses are also possible.
- The port. This is 25 for most servers.
- A user name.
- A password.

Some servers don’t require a user name or password because they are only meant for internal use.

You’ll also need to decide on a `from` address for the alerts that the server sends. When people receive an alert email from Tableau Server, this is the name that’s on the `from` line of the
message. Because alerts are simply informational, you generally don’t need to worry about who’s on the from line, so people use addresses like no-reply@example.com or tableau-admin@example.com.

After you get the SMTP server information from your IT department, you can use the Tableau Server Configuration utility to set up alerts.

Step 1: Stop the server

1. Stop Tableau Server. (In the Windows Start menu, search for Stop Tableau Server.)
2. In the Windows Start menu, search for Configure Tableau Server.

Step 2: Configure SMTP information for Tableau Server

1. In the Tableau Server Configuration utility, click the SMTP Setup tab.
2. Enter the information that you received from your IT department.
3. In the Send email from box, enter the email address that you want all server emails to be sent from. For example, you might enter tableau_admin@example.com or no-reply@example.com.
4. In the Send email to box, enter the email address or addresses that you want server-health emails to be sent to. For example, you might enter your own email address and the email address of your IT person.
5. Click OK.

6. Start Tableau Server (in the Windows Start menu, search for **Start Tableau Server**).

**Step 3: Set up alerts**

In the Tableau Server Configuration utility, click the **Alerts and Subscriptions** tab. We recommend that you select all the checkboxes on this tab to enable all alerts. You’ll know that alerts are working when you restart Tableau Server and receive an email.

![Tableau Server Configuration](image)

If you do select all the check boxes, here are the alerts that get activated.

**Subscriptions to views**

Users can periodically receive a snapshot of views that they’re interested in. This can be useful if your users want to see information about views on a recurring basis. For example, users can get a view in their inboxes every week.

**Server component events**

You can receive a notification when Tableau Server processes stop or start. If you install Tableau Server on multiple nodes, you can see a notification for each process that stops or starts. If you install Tableau Server on a single node, you can see a notification for each process that starts, but not for each process that stops. Because part of the server must be running in order to send an alert that processes have stopped, you see only notifications when the data engine, repository, and gateway processes stop.
Anytime that server processes stop or that the server restarts unexpectedly, you should investigate the cause of the restart. For example, you may discover that the Windows Server computer is configured to restart automatically after Windows updates—in which case you may want to schedule updates for off-peak hours.

**Low disk space**

You can receive a notification when the disk space on the server computer falls below a threshold that you specify. As a general rule, we recommend that the server computer maintain at least 20% free disk space. The farther that the disk space falls below this threshold, the more likely that the server’s performance will be affected. Eventually, the server may even stop responding.

**Step 4: Restart the server**

On the Windows Start menu, click **All Programs > Tableau Server 10.0 > Start Tableau Server**. If alerts are configured correctly, Tableau Server sends an email titled "Multiple services on your-server are UP."

**Quick Start: Disk Space Alerts**

You can configure Tableau Server to monitor free disk space on computers running Tableau Server, and to send alerts when free space drops below thresholds that you define. If you choose to have Tableau Server save historical usage information, this is available to Tableau Server administrators through one of the Administrative Views.

**1 Configure SMTP**

Before you can configure alerts for disk space usage, you need to configure Tableau Server for SMTP on the **SMTP Setup** tab in the Tableau Server Configuration utility.
For more information about how to configure SMTP in Tableau Server, see Configure SMTP Setup on page 617.

2 (Optional) Change the historical disk usage data option

By default Tableau Server is configured to save data about disk space usage. If you do not want to save this data, clear the Record disk space usage information, including threshold violations box.
Note: You do not need to save disk space usage information to receive alerts about low disk space, but if Tableau Server is not saving disk space usage data, you cannot view historical disk space usage in Administrative Views on page 1097.

3 Configure alerts

Tableau Server can send alerts to let you know when space on one of the Tableau Server nodes drops below the warning and critical thresholds of the entire disk. Tableau Server continues to send alerts at the frequency specified in Send email alert every as long as disk space remains below the warning threshold.

To receive email alerts when free disk space falls below either of the two thresholds, select Send alerts when unused drive space drops below thresholds:
4 Change alert thresholds and frequency

By default, the warning threshold is set to 20% and the critical threshold is set to 10%. As long as the free disk space remains below a threshold, Tableau Server will continue to send alerts at the frequency you specify in Send email alert every. You can change these values.

Performance Monitoring

When you monitor a server, you collect and analyze data that signals whether the server is performing badly or running into problems. For example, if you notice that your server is using 100% of its processing capacity for long periods of time, you know that there’s a problem.

The data that you need to collect and analyze can be broken down into the following broad categories:

- Resource usage data—how Tableau Server uses hardware resources like disk space, memory, and processors.
- Session and load time data—how users interact with Tableau Server, including how long it takes for views to load and how many concurrent users there are.
- Background task data—how Tableau Server runs tasks that are not directly tied to a user action. For example, background tasks include extract refresh tasks, subscription tasks, and more.

Some of this data, including load time data and extract refresh data, is already accessible from the administrative views that are built into Tableau Server. However, to collect resource usage data you need to use an external performance monitoring tool. (For the purposes of this section, we’ll use Windows Performance Monitor as an example, because it’s included with Windows Server.) To collect additional load time data and background task data, you can connect to the Tableau Server repository.
After you’ve collected the performance data that you want to analyze, you can use the sample workbook included in this section as a starting point for analyzing your performance data. To make it easier to analyze your performance data in one place, you can then publish the views that you create to Tableau Server as custom admin views.

**Note:** To use the sample workbook and to publish views to Tableau Server, you must have Tableau Desktop.

**Built-In Monitoring Tools**

Use the Tableau Server web interface to monitor server health. You can view the status of server processes on each computer where Tableau Server is installed, and you can use administrative views to understand activity on Tableau Server, whether the activity comes from users or from server tasks like extracts.

Here are the most important administrative views for monitoring Tableau Server:

- Traffic to views
- Background tasks for extracts
- Stats for load times

**View Server Process Status**

You can use the Process Status table on the Server Status page to view the state of Tableau processes on each Tableau server:
Possible status indicators are listed at the bottom of the table:

When Tableau Server is functioning properly, most processes will show as Active, Busy or Passive (Repository):

- **Active**—The process is functioning as intended. See File Store in Troubleshoot Server Processes on page 1212 for details on possible active states.
- **Busy**—The process is completing some task. See File Store and Repository in Troubleshoot Server Processes on page 1212 for more information.
- **Passive**—The repository is in passive mode
- **Unlicensed**—The process is unlicensed.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

If there is additional information, a message appears below the status icon:
For more information about troubleshooting process status, see Troubleshoot Server Processes on page 1212.

Administrative Views

The Status page contains an embedded Tableau workbook with various administrative views. These views help you to monitor different types of server or site activity.

- Shows server and site activity for Tableau Server.
- Shows site activity for Tableau Online.

Navigating to administrative views

To see administrative views, click Status. Site administrators can see administrative views for their site. Administrators of multiple sites can see views for the current site.

On a multi-site server, server administrators can see views for the entire server. Click the site menu, and then click Manage All Sites to access the server menus.

To see views for individual sites on a multi-site server, click the site menu, select the site name, and then click Status.

Create Custom Administrative Views

In addition to the pre-built administrative views available on the Maintenance page on the Server, you can use Tableau Desktop to query and build your own analyses of server activity. To do this, you can connect to and query views in the Tableau Server repository using one of two built-in users: the "tableau" or "readonly" user.

To connect to the Tableau Server repository, see Collect Data with the Tableau Server Repository on page 1117.
The **tableau** user—The tableau user has access to special views and a subset of tables in repository database. These views and tables are provided so that administrators can create custom administrative views. Tableau makes an effort to limit changes to these tables and views so that custom views built with them do not break.

The **readonly** user—The readonly user has access to a large number of the repository tables, providing more data about server usage. Administrators can use these to create custom administrative views too, but many of the tables are intended primarily to support the functioning of Tableau Server and may be changed or removed without warning. This means that views created from these tables can break when the database structure is changed.

**Note:** The readonly user is available in Tableau Server 8.2.5 and later.

For examples of using the readonly user to connect to the workgroup database, see the following articles in the Tableau Knowledge Base: Group Membership, Server Access, Server Access (2), and Workgroup Usage.

Before you can connect using one of the built-in users, you must enable access to the Tableau Server database. After doing this you can use Tableau Desktop to connect to and query the database as the tableau user or the readonly user.

The `tabadmin` set option `auditing.enabled` controls whether Tableau Server collects historical user activity and other information in the repository. It is enabled by default. Be aware that collecting historical events impacts the size of Tableau Server’s backup file (.tsbak).

- All hist_tables are controlled by the `tabadmin` set option `wgserver.audit_historyExpiration_days`, which controls how many days of event history are kept in the repository and has default value of 183 days.
- The `_http_requests` table is cleaned of all data older than 7 days every time `tabadmin cleanup` on page 1263 or `tabadmin backup` on page 1262 is used. For more information, see Remove Unneeded Files on page 1152.
- The `_background_tasks` table is cleaned automatically and keeps data for the last 30 days.
- All other tables with names that begin with a "_" prefix contain current data.

**Traffic to Views**

The Traffic to Views view gives you the ability to see how much of your user traffic goes to views.

You can filter what information is displayed and the time frame it comes from by selecting the view, the workbook, and the time range. Server administrators can specify the site.
Two time lines at the top of the view show you how views are being used over a time range you specify (the default is the last 7 days):

- **What is the Total View Count by Day**—This shows total view count by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count of views. Select the point to update the other sections of the view based on your selection.
- **What is the Total View Count by Time**—This shows the view count by time of day. The filters and any selection impact this graph.

Two bar graphs at the bottom of the view show results that are filtered by the **Min View Count** filter at the top of the view. These show you the views that are most often accessed, and the users who most frequently access views Only those views and users with counts greater than or equal to the minimum view count value are displayed:

- **What Views are Seen the Most**—This is a list of the most visited views. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Accesses Views Most Often**—This shows the users who most often access the views and is limited by filters and any selection you make.

**Traffic to Data Sources**

The Traffic to Data Sources view gives you the ability to see usage of data sources on your Tableau Server installation. This can help you determine which data sources are most heavily used and those that are less often used. You can filter the information you see by selecting the data source, the action taken on that data source, and the time range. Server administrators can specify the site.
A time line at the top of the view shows you how data sources are being used over a time range you specify (the default is the last 7 days):

- **What is the Total Data Source Usage by Day**—This shows total data source usage by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count. Select the point to update the other sections of the view based on your selection.

Two bar graphs at the bottom of the view show results that are filtered by the **Min Interactions** filter at the top of the view. These show you which data sources are most used, and who uses data sources most often. Only those data sources and users with interaction counts greater than or equal to the minimum interactions value are displayed:

- **What Data Sources are Used Most**—This is a list of the most used data sources. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Uses Data Sources Most Often**—This shows the users who most often use the data sources. This is impacted by filters and any selection you make.

**Actions by All Users**

The Actions by All Users view gives you insight into how your Tableau Server installation is being used. You can filter the view by actions and by time range. Server administrators can filter by site. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of active users who have performed one of the selected actions.
Up to three separate groups of time lines show you how users are using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

**Actions by Specific User**

The Actions by Specific User view gives you insight into how individual users are working in your Tableau Server installation. You can filter the view by user name, actions, and time range. Server administrators on multi-site installations can filter by site.
Up to three separate groups of time lines show you how a selected user is using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, or if no actions were taken, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

A bar graph at the bottom of the view shows which items the selected user is using.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

### Actions by Recent Users

The Actions by Recent Users view shows you which signed-in users have been active on Tableau Server recently. This can be useful if you need to perform some maintenance activity and want to know how many and which users this will affect, and what they are doing on Tableau Server.

The view **Active, Recently Active**, and **Idle** users that are currently signed in to Tableau Server. For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last
took an action more than 30 minutes ago. The actions are displayed in the lower section of the view.

Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.

Background Tasks for Extracts

The Background Tasks for Extracts view displays extract-specific tasks that run on the server.
A table lists the extracts that ran in the time period specified in Timeline. Click **Success** or **Error** to filter the table based on status. Click a specific task to update the **How Much Time did Extracts Take** graph for the selected task. The **How Many Extracts Succeeded or Failed** table updates for the status (success or failure) of the task, but the count of extracts that succeeded or failed does not change.

Tasks can have a status of successful or error:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
<td>Success—Server completed the task.</td>
</tr>
<tr>
<td>🚫</td>
<td>Error—Server was unable to complete the task.</td>
</tr>
</tbody>
</table>

For details on a task, hover over its icon:
Background Tasks for Non Extracts

The Background Tasks for Non Extracts view displays tasks that the server runs that are not related to refreshing extracts. For example, edited OAuth connections, subscription notifications, and so on.

A table lists the tasks that ran in the time range specified. Click Success or Error to filter the table based on status. Select a specific task in the How Many Tasks Succeeded or Failed on this Site table to update the What Background Tasks Ran on this Site graph for the selected task.

Tasks can have a status of successful or error.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Error—Server was unable to complete the task.</td>
</tr>
<tr>
<td></td>
<td>Success—Server completed the task.</td>
</tr>
</tbody>
</table>

For details on a task, hover over its icon.

Stats for Load Times

The Stats for Load Times view shows you which views are the most expensive in terms of server performance. You can filter by view and time range. Server administrators can filter by site. You can also limit the view based on load time in seconds, using the sliding Load Time
filter. Load times are for the server. Depending on your client browser and networking, actual load time may vary slightly.

The Average Load Times graph shows average load times for views based on the filters you set. Hover over a point to see details. Select a point on the line to update the rest of the view for the selection:

The Exact Load Times view shows exact time to load the listed views. A vertical line shows the average load time for each view. Select a mark to see details of a specific instance of the view loading:

Stats for Space Usage

The Stats for Space Usage view can help you identify which workbooks and data sources are taking up the most disk space on the server. Disk space usage is displayed by user, project, and by the size of the workbook or data source and is rounded down to the nearest number:
Use the **Min Size** filter to control which data sources and workbooks are displayed, based on the amount of space they take up.

Three bar graphs give you information about space usage on your Tableau Server:

- **What Users Use the Most Space**—This shows the users who own data sources and workbooks that are taking up the most space. Click a user name to filter the next two graphs for that user. Click the data source bar or the workbook bar for a user to filter the next two graphs for that type of object for that user. Click the selected user or bar to clear the selection.

- **What Projects Use the Most Space**—This shows the projects with the data sources and workbooks that are using the most space. If a user or object type is selected in the What Users Use the Most Space graph, this displays information specific to the selection.

- **What Workbooks and Data Sources Use the Most Space**—This shows the workbooks and data sources that are taking the most space. The bars are color-coded based on the length of time since the last refresh.

Move your cursor over any bar to display usage details:
Click on a bar to select it and update the other areas of the view based on that selection.

**Background Task Delay**

The Background Task Delay view displays the delay for extract refresh tasks and for subscription tasks—that is, the amount of time between when they are scheduled to run and when they actually run. You can use the view to help you identify places you can improve server performance by distributing your task schedules and by optimizing tasks.

Here are possible reasons for the delays, and ways that you might reduce the delays:

- Many tasks are scheduled for the same time. In the example view, tasks that show long delays are clustered at the same time every day, which creates spikes in the wait time.
Note that you can set the Timeline filter to a single day to view task delays by hour and identify the hours of the day which have many tasks scheduled at the same time. A solution to this issue can be to distribute the tasks to off-peak hours to reduce load on the server.

- Specific tasks take a long time to run and are preventing other tasks from running. For example, there might be an extract refresh job that is connecting to a slow data source or that is processing a large amount of data. Use the Background Tasks for Extracts administrative view to identify which extract refresh tasks are running slowly. You can then optimize the extract refresh task by filtering the data, aggregating the data, or creating multiple data sources for individual tables in a data source.

- Other server processes are running at the same time and are consuming server resources and slowing down performance. Monitor the CPU and memory usage of server processes to see which processes are consuming the most resources and then adjust the configuration of processes on your server. For more information on monitoring processes, see Collect Data with Windows Performance Monitor on page 1114. For more information on tuning the performance of server processes, see Performance Tuning on page 1123.

### Performance of Views

The Performance of Views administrative view displays how long it takes for views to load and how many sessions are running at a time on the server.
You can compare spikes in the number of sessions with spikes in slow load times to identify the times of day when high user traffic is slowing down the server. You can also look at the individual views by load time to understand which views take the longest to load. For information on how to optimize the server, see Optimize for User Traffic on page 1124.

Some views might take a long time to load regardless of when they are viewed. You can identify which workbooks need to be optimized with the Stats for Load Times administrative view. Some simple ways to optimize workbooks includes the following:

- Display less information in each view.
- Break up views.
- Reduce the number of filters.
- Use data extracts.

**Server Disk Space**

Use the Server Disk Space view to see how much disk space is in use on the computer or computers that run Tableau Server, where disk space refers only to the partition where Tableau Server is installed. You can also use this view to identify sudden changes in disk space usage.

For a distributed installation, the view displays information about each computer in the cluster.

The Server Disk Space view includes two graphs:

- **What is the most recent disk space usage?**—This graph shows disk space usage for the last 30 days both in gigabytes and as a percentage. Disk space refers only to the
partition where Tableau Server is installed.

- **How has free disk space trended in the last month?**—This graph shows changes to disk space usage over the last month. Rest your pointer on a line to view the exact amount of free disk space for a point in time.

When Tableau Server is low on disk space, you can remove files to free space.

For more information, see [Troubleshoot Disk Space Usage on Tableau Server Nodes](#) on page 1231

**Tip:** You can have Tableau Server alert you when free disk space falls below a threshold that you specify. For more information, see [Quick Start: Disk Space Alerts](#) on page 1091.

### Desktop License Usage

The Desktop License Usage view lets server administrators see usage data for Tableau Desktop licenses in your organization. This can help you manage licenses efficiently and determine if you need more or fewer licenses. This view can help you answer the following questions:

- Who is using a Tableau Desktop license in my enterprise?
- Have any licenses been shared or transferred?
- Is any license being used on a computer where it should not be?
- Does a specific user use their license?
- What types of licenses are being used in my enterprise?
- Do I need to convert any trial licenses?

**Note:** To get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau, or after installation, by modifying the registry or property list file. For more information, see [Configure Tableau Desktop License Reporting](#) on page 1081.

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See [Enable Desktop License Reporting](#)

### Filters

- **Product Keys.** Type a string to filter the dashboard to only those licenses that include the string anywhere in the license key. For example, to see only licenses that begin with TDTD, type TDTD and press Return to filter the view. Click the X after the string to reset the filter.

- **Action.** Use this filter to control what the dashboard displays, based on the action taken.
Actions are **Activate**, **Use**, and **Return**. If the **Use** action is not selected, nothing is displayed in the top bar graph.

- **Department.** Use this filter to control what departments the dashboard displays licenses for. The filter is populated based on the **Department** values specified when Tableau Desktop is registered.
- **Select time duration in days.** Use this slider to specify the time length in days that the dashboard displays information for. The default value is 183 days.

When you hover over the filter card in the first three filters, a drop-down icon appears. Click the icon to specify whether the view should include data that matches the filter (the default) or exclude data that matches the filter:

![Include values](image)

- **Include values**
- **Exclude values**

**Who has used Tableau in the last <nn> days?**

This area of the dashboard shows a bar graph of three types of Tableau Desktop licenses (Perpetual, Trial, and Term) and the number of users who have used each license type during the specified time period. Hover over a license type segment to see an explanation of the license type. Click a segment to filter the rest of the dashboard for only that license type. This action filters both the tables that show licenses that have been used and those that have not been. For example, to see a list of term licenses that have been used during the time period, click the Term bar. The "used" and "not been used" lists are filtered to just show term licenses.

A table of detailed information shows under the bar graph. For each row in the table, action icons display on the right, above a timeline that shows you when the action last took place.

To see a list of the underlying data in a format that allows you to select and copy values like email or product key, click a row in the list of licenses and click the View Data icon:

![Keep only](image)

- **Keep only**
- **Exclude**

The data displays in summary form. Click **Full data** to see all the data. From this view you can select and copy individual values, or download the data as a text file.
What licenses have not been used in the last <nn> days

This area of the dashboard shows a list of licenses that have not been used during the specified time period. A timeline shows the last use date. Hovering over a last use mark gives you information including the registered user of the copy of Tableau.

Desktop License Expiration

The Desktop License Expiration view gives server administrators information about which Tableau Desktop licenses in your organization have expired or need maintenance renewal. This can help you manage licenses efficiently. This view can help you answer the following questions:

- What trial or term licenses have expired?
- What perpetual licenses have expired maintenance?
- What perpetual licenses have maintenance renewals coming up?

Note: In order to get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau. For more information, see Configure Tableau Desktop License Reporting on page 1081.

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See Enable Desktop License Reporting.

Filters:

- **Product Keys**—Type a string to filter the dashboard to only those licenses that include the string. For example, to only see licenses that begin with TDTD, type TDTD and press return to filter the view. Click the "x" after the string to reset the filter.

- **Department**—Use this filter to control what department(s) the dashboard displays licenses for. The filter is populated based on the Department values used when registering copies of Tableau Desktop.

- **Time Duration**—Use this filter to control the length of time for which the dashboard displays information.

The view includes the following tables, which are affected by the filters you set at the top of the view:

- **What keys have expired maintenance**—This table shows the product keys for which maintenance has expired, with a vertical line indicating the point at which the six month
window for renewing maintenance closes. If maintenance for a key is expired for more than six months you need to purchase a new key in order to qualify for support or upgrades.

- **What trial and term licenses have expired**—This shows the trial or term product keys that have expired.
- **What is the maintenance schedule for my keys**—This shows the keys and their maintenance status.

**Collect Data with Windows Performance Monitor**

To monitor resource usage and server processes, you can use Windows Performance Monitor (PerfMon), which is included with Windows Server. Use PerfMon to gather detailed performance information, including how often the CPU is being used, how much memory is being used, information about each Tableau Server process, and more.

For more information about what each Tableau Server process does, see **Tableau Server Processes** on page 1240.

**Disclaimer:** This information refers to a third-party product. This example is not an endorsement of this product over any other competing products.

Before you can use PerfMon, you set up a data collector set, which is how PerfMon stores the data that it collects. To collect information about Tableau Server processes with PerfMon, Tableau Server must be running when you create the data collector set. The data that you collect in PerfMon are often referred to as performance counters.

**Step 1: Create a new data collector set**

1. Click the Windows Start menu and search for "performance".
2. Right-click **Performance Monitor** and then click **Run as administrator**.
3. In the left pane, click **Data Collector Sets**.
4. In the right pane, right-click **User Defined**, click **New**, and then click **Data Collector Set**.
5. In the **Create new Data Collector Set** wizard, enter a name for the data collector set. For example, you might enter **Tableau Server Performance**.
6. Select **Create manually (Advanced)** and then click **Next**.
7. Under **Create data logs**, select **Performance counter**, and click **Next**.

**Step 2: Select performance counters**

1. Set the sample interval to 30 seconds.
2. Click **Add**.
3. Select the computer that you want to record performance data for.
If you run Tableau Server on a single computer, select <Local Computer>. If you run Tableau Server on multiple computers, you must repeat some of these steps to gather data about each computer.

4. Select performance counters from the list.

The following table lists some performance counters that we recommend for tracking Tableau Server performance.

<table>
<thead>
<tr>
<th>Category</th>
<th>Performance Counters</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Disk</td>
<td>Current Disk Queue Length</td>
<td>The number of outstanding write requests and the amount of bytes read and written to the server’s hard disk. Select these counters for the disk on which you installed Tableau Server (referred to as the instance in PerfMon).</td>
</tr>
<tr>
<td></td>
<td>Disk Read Bytes/sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disk Write Bytes/sec</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>% Committed Bytes in Use</td>
<td>The percentage of virtual memory in use, and the amount of memory available in megabytes.</td>
</tr>
<tr>
<td></td>
<td>Available MBytes</td>
<td></td>
</tr>
<tr>
<td>Processor Information</td>
<td>% Processor Time</td>
<td>The percentage of time that the processor spends active, and the percent of processing capacity being used by the processor.</td>
</tr>
<tr>
<td></td>
<td>% Processor Utility</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>% Processor Time</td>
<td>The percentage of processing capacity being used by a particular process, and the amount of memory reserved for the process. Select these counters for the following processes (referred to as instances in PerfMon):</td>
</tr>
<tr>
<td></td>
<td>Private Bytes</td>
<td>• backgrounder (Backgrounder)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• dataserver (Data server)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• redis-server (Cache server)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• tdeserver (Data engine)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• vizqlserver (VizQL Server)</td>
</tr>
</tbody>
</table>

To select each performance counter:
1. Double-click to select a category in the drop-down list.
2. Select the performance counter or counters that you want to use.
3. Under **Instances of selected object**, if appropriate, select the process (or instance) that you want to collect information about.

4. Click **Add**.
5. If you run Tableau Server on multiple computers, return to step 3, select another computer and then repeat the above steps.

**Important:** Depending on how you configured server processes to run across computers, you might have to monitor only a subset of the processes listed for each computer. For example, it’s a best practice to isolate the backgrounder processes on a separate computer. As a result, for that computer, you would only monitor the backgrounder processes.

6. Click **OK** and then click **Next**.
Step 3: Save the data collector set

1. Browse to the directory where you want to store the data, and then click **Next**.

   **Important:** You must store the data in a place that's accessible by Tableau. For example, you might want to store the data on a network drive. If you don't have a network drive mapped, right-click **This PC** and select **Add a Network Location**.

2. Click **Finish**.

3. In the left pane of the main **Performance Monitor** window, select the data collector set that you created.

4. In the right pane, right-click the performance counter **DataCollector01** and then click **Properties**.

5. Select **Comma separated** as the log format and then click **OK**.

Step 4: Run the data collector set

In the left pane, right-click the name of the data collector set that you created and click **Start**. The Windows Performance Monitor tool starts monitoring your server and storing information in the location that you specified.

Step 5: Allow Remote Access for Multiple Computers

For PerfMon to collect data about other computers, you need to make sure that the other computers can be reached—that is, that they are on the same network and do not have firewall rules that prevent access. The firewall rules that you need to set differ across versions of Windows, so you might need to contact your network administrator for information. In addition, you must make sure that the Run As user account has permission to collect data on the remote computers. By default, PerfMon runs the data collector set as the SYSTEM user. To change the Run As user, complete the following steps:

1. In the left pane of PerfMon, open the **Data Collector Sets** node and then the **User Defined** node.

2. Right-click the name of the data collector set and then click **Properties**.

3. In the **Run As** section of the **General** tab, click the **Change** button and specify a different account.

Collect Data with the Tableau Server Repository

Before you start analyzing the resource usage data that Windows Performance Monitor (PerMon) collects, connect to the Tableau Server repository to get additional data about load times and background tasks. The Tableau Server repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more.
After you enable access to the Tableau Server repository, you can create views with data from the repository. The views that you create with this data are sometimes called custom administrative views. In addition to being used for performance monitoring, custom admin views can be used for tracking user activity, workbook activity, and more. For more information on the type of data that you can use for these views, see Create Custom Administrative Views on page 1097. Alternatively, if you are only interested in performance data, you can use the preselected database tables in the sample performance workbook. For more information about the sample performance workbook, see Analyze Data with the Sample Performance Workbook on page 1120.

Get access to the Tableau Server repository

You can use Tableau Desktop to connect to and query the Tableau Server repository using two built-in users. The user named tableau has access to several database views you can use as part of building your own analyses of Tableau Server activity. The user named readonly has access to additional database tables that you can use to create views for even more in-depth analysis.

Use the readonly user to monitor Tableau Server.

To access the Tableau Server repository, you need to enable access to the database by using tabadmin commands.

1. Open a command prompt as an administrator and type the following:
   
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"

2. Enter the following command to enable external access to the database for the tableau user or the readonly user:

   tabadmin dbpass --username [tableau|readonly] password

   For example, to enable access for the readonly user with a password of p@ssword, use this command:

   tabadmin dbpass --username readonly p@ssword

   **Note:** If no user is specified, the dbpass command enables access for the tableau user.

3. Restart Tableau Server by using this command:

   tabadmin restart

   **Note:** If you later decide that you want to disable access to the Tableau Server repository, use the tabadmin dbpass --disable command. For more information, see tabadmin Commands on page 1256.
Connect to the Tableau Server repository

This section describes how to connect to a custom set of tables from Tableau Server repository. If you want to use the sample performance workbook, see Analyze Data with the Sample Performance Workbook on the next page and follow the steps to edit the connection information.

1. In Tableau Desktop select **Data > Connect to Data**, and then select **PostgreSQL** as the database to connect to.

   **Note:** You might need to install the PostgreSQL database drivers. You can download drivers from www.tableau.com/support/drivers.

2. In the PostgreSQL connection dialog box, enter the name or URL for Tableau Server in the **Server** box. If you have a distributed server installation, enter the name or IP address of the node where the repository is hosted.

   Connect using the port you have set up for the pgsql.port, which is 8060 by default. For more information about ports, see Tableau Server Ports on page 1244.

   **Note:** The tabadmin dbpass command does not open any ports in the firewall. You may need to manually open the port in any firewall between your external client and the Tableau Server database.

3. Specify **workgroup** as the database to connect to.

4. Connect using one of the following users and the password you specified:

   **Username:** tableau or readonly.
   **Password:** The password you specified when you enabled access to the Tableau Server database for the specified user.

5. Click **Connect**.
6. Select one or more tables to connect to.

The `tableau` user has access to all of the tables that start with an underscore or with `hist_`. For example, you can connect to `background_tasks` and `datasources`. The `hist_` tables include information about server users that isn’t currently presented in the Actions by Specific User on page 1101 view. The readonly user has access to additional tables that can be used to query other information about server usage.

7. Click **Go to Worksheet**.

**Analyze Data with the Sample Performance Workbook**

To get started analyzing the data that you collect with Windows Performance Monitor (PerfMon) and with the Tableau Server repository, you can download and use the sample performance workbook that Tableau provides. The sample workbook contains worksheets for some of the most important performance indicators, including CPU and memory utilization by
process, disk activity, view load times, and more. After you download the workbook, use it as a starting point for your data exploration and extend it to meet your needs.

Download the sample performance workbook

1. Click the **Download** button in the lower-right corner of the workbook.
2. Navigate to the directory where you downloaded the workbook and double-click on the workbook to open it in Tableau Desktop.

Edit the connections to the Tableau Server repository

After you open the workbook, you must edit the data connections to point to the PostgreSQL repository for your installation of Tableau Server.

The sample workbook includes the following data source connections:

- **Background Tasks**—a connection to the Tableau Server repository that joins the `_background_tasks` table and the `_sites` table.
- **Historical Events**—a connection to the Tableau Server repository that joins the `historical_events`, `hist_users`, and `historical_event_types` tables.
- **Resource Usage**—a connection to the data you collect from PerfMon.
- **Sessions and Load Times**—a connection to the Tableau Server repository that joins the `_users`, `_http_requests`, `_sessions`, and `_sites` tables.

1. From any of the sheets in the workbook, right-click the **Background Tasks** data source in the **Data** pane, and then click **Edit Data Source**.
2. Replace the server address with the hostname or IP address of your server.
3. Enter the password that you set for the readonly user.
4. Click OK.
5. Repeat the steps above for the Sessions and Load Times data source.

**Edit the connection to the PerfMon data**

1. Click the Data Source tab.
2. In the data source list, select Resource Usage.
3. In the menu, click Data > Resource Usage > Edit Connection.
4. Navigate to the directory where you stored the data from PerfMon.
5. Select the file and click Open.

**Replace references to PerfMon fields**

Because the data that you collect from PerfMon includes references to specific host names, you must also replace the references in the workbook for specific field names. For example, when you click on the CPU worksheet, you might notice that most of the fields have a red exclamation mark next to them. This indicates that the field is missing from the new data source.
To map the fields in your data to the fields in the workbook, follow these steps:

1. Navigate to the **CPU** worksheet or to any of the worksheets that use the Resource Usage data source.
2. Right-click a field in the **Measures** list that has a red exclamation mark next to it.
3. Click **Replace references**.
4. Select the corresponding field in the dialog.

   For example, you might replace references to `\YOUR-SERVER\LogicalDisk (C:)\Current Disk Queue Length`.

**Update calculations**

The sample workbook includes calculations that aggregate processor utilization for multiple processes. To view data for all of the processes on your server, you must edit the calculations to include additional instances of server processes.

For example, the **VizQL Server CPU %** calculation includes a reference to one process:

```
[\YOUR-SERVER\Process(vizqlserver)\% Processor Time]
```

However, if your server runs more than one VizQL Server process, then you must aggregate the additional process in the calculation. For example, you might enter the following:

```
[\YOUR-SERVER\Process(vizqlserver)\% Processor Time] + [\YOUR-SERVER\Process(vizqlserver#1)\% Processor Time]
```

**Publish to Tableau Server**

Optionally, when you finish updating your performance workbook, you can publish it to Tableau Server so that the views in the workbook are accessible from the Tableau Server interface.

**Performance Tuning**

This section describes how to use the performance data that you collect to identify ways to improve the performance of Tableau Server. Because no two server environments are identical, we can't provide hard and fast rules for tuning server performance. However, you can draw conclusions about performance from patterns in the data that you collected.

For example, are there recurring spikes? Do any of the patterns that you notice in the administrative views correspond to similar patterns in Windows Performance Monitor? Observing patterns like this can guide you in testing and incremental tuning.

Most performance tuning for Tableau Server boils down to these general approaches:

- Optimize for user traffic. This tunes the server to respond to user requests and to display views quickly.
- Optimize for extracts. This tunes the server to refresh extracts for published data
sources. You might want to optimize for extract refreshes if your organization has a lot of data and the data needs to be as up to date as possible.

Rendering views and refreshing extracts generate the most load on the server, so you should optimize for the task that your organization is most interested in.

**Optimize for User Traffic**

You can optimize for traffic if you have many active Tableau Server users and few published data sources that need extract refreshes.

*Note:* This topic uses the sample performance workbook from the monitoring section. For more information, see Analyze Data with the Sample Performance Workbook on page 1120.

- When to optimize for user traffic
- Ways to optimize for user traffic

**When to optimize for user traffic**

**Slow load times for views**

Use the *Requests and Sessions* dashboard of the sample performance workbook to analyze how long views take to load.

If multiple views take longer than 10 seconds to load, and if the slow load times correspond to a large number of sessions, that can indicate that user traffic is slowing down the server.

However, if a particular view takes a long time to load regardless of when it is viewed, it's a sign that the workbook for the view needs to be optimized. You can identify which workbooks need to be optimized with the *Stats for Load Times* administrative view. Some simple ways of
optimizing workbooks includes displaying less information in each view or breaking up views, reducing the number of filters, and using data extracts.

**High resource usage corresponding to user traffic**

If your server displays high CPU and memory usage during peak traffic hours, you should optimize for user traffic. To determine peak traffic hours and analyze how many concurrent users are on your server, use the **Users and Actions** dashboard. In addition, you can use the **Traffic to Views** administrative view to see how much user traffic involves accessing views (as opposed to performing administrative functions, publishing, or other tasks).

![Users and Actions Dashboard](image)

If you click a point in the **Number of Users** view, the dashboard displays the users that were active at the time and the number of user actions that those users performed. By default, the only user actions displayed are user views, but you can use the **Action Types** filter to display additional user actions.

Make a note of the times of day when there are many concurrent users and views so that you can compare this to resource usage. As a rule of thumb, the number of users should correspond to a high number of user actions. However, the view in this example displays an artificially high number of actions for a single user as part of a load generation test. As an example, you can compare the high number of views at 12 AM on June 28th with the resource usage in the dashboard illustrated later.

Use the **CPU Usage** dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. In the following example, note the large spike in total CPU usage and in the VizQL server process at 12 AM on June 28th. Because the VizQL server process loads and render views, the VizQL server process is often the first process to show strain under high user traffic.
**Note:** The percent of CPU usage for individual processes may add up to more than 100 percent. This is because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.

Use the **Memory Usage** dashboard to display the percent of total memory usage and the average memory usage in gigabytes. As a general rule, memory usage increases steadily with user traffic. Here again the VizQL server process is the first to show strain under high traffic.
Ways to optimize for user traffic

When high user traffic corresponds to high resource usage as it does in the example shown previously, you should optimize for user traffic.

Adjust the number of VizQL server processes

The most effective way of optimizing for user traffic is to adjust the number of VizQL server processes. Add one VizQL server process at a time and measure the effect with more performance monitoring. Because VizQL server processes can consume a lot of CPU and memory, adding too many processes can slow down the server instead. If you see consistently high memory usage, try to reduce the number of VizQL server processes to reduce the amount of memory reserved.

1. Stop Tableau Server and open the Tableau Server Configuration utility.
2. Click the Servers tab.
3. Click Edit.
4. Increase the number of VizQL server processes by one.
5. Restart Tableau Server.

Adjust the number of other processes

Although the most effective way of improving performance for user traffic is to adjust the number of VizQL server processes, you can also tune other processes that support the VizQL server process or that prevent the VizQL server process from accessing resources. For example, the VizQL server process makes frequent requests to the cache server process, so you might also want to increase the number of cache server processes. On the other hand, the backgrounder and data engine processes might contend for CPU resources with the VizQL server process. As a result, if you do not need to run frequent extract refreshes, you might reduce the number of processes for the backgrounder or the data engine. If you do need additional instances of these processes, and if you’re running Tableau Server on a cluster, you can move these processes to a dedicated node.

Adjust the VizQL session timeout limit

In the example shown previously, the amount of memory used by the VizQL server process increases with user traffic, and it remains reserved by Tableau Server for some time after the traffic finished. This is because the VizQL server process reserves memory for each session for a specified amount of time. If the VizQL server process uses a high percentage of the available memory, try reducing the timeout for each session to make memory available more quickly. To do this, use tabadmin on page 1255 to reduce the vizqlserver.session.expiry.timeout setting.

Refresh the cache less often

If your users do not always need the most up-to-date data, you can optimize for user traffic by configuring Tableau Server to cache and reuse data as much as possible.
1. Stop Tableau Server and open the Tableau Server Configuration utility.
2. Click the **Data Connections** tab.
3. Select **Refresh less often** as the caching option.
4. Click **OK**.
5. Restart Tableau Server.

**Assess view responsiveness**

When a user opens a view, the components of the view are first retrieved and interpreted, then displayed in the user's web browser. For most views, the display rendering phase occurs in the user's web browser and in most cases, this yields the fastest results and highest level of interactive responsiveness. Handling most interactions in the client web browser reduces bandwidth and eliminates round-trip request latencies. If a view is very complex, Tableau Server handles the rendering phase on the server instead of in the client web browser, because that generally results in the best performance. If you find that views aren't as responsive as you'd like, you can test and change the threshold that causes views to be rendered by the server instead of in the client web browser. For more information, see **About Client-Side Rendering** below.

**About Client-Side Rendering**

When you navigate to a view in Tableau Server, the processing required to display the view (the rendering) can either be performed by your client web browser or by Tableau Server depending on the complexity of the view. The complexity of the view is determined by the number of marks, rows, columns, and more. If a view is less complex, then it is faster for you web browser to render the view than it is to send a request to Tableau Server. If a view is more complex, then it is faster to send a request to Tableau Server and take advantage of the server's computing power.

As a server administrator, you can configure when client-side rendering happens both for web browsers on your computer and web browsers on mobile devices by adjusting the complexity threshold. Alternatively, you can disable client-side rendering with tabadmin.

**Requirements**

- **Supported browsers**: Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which is used by client-side rendering.

- **Polygons and the page history feature**: If a view uses the polygon mark type or the page history feature, server-side rendering is performed, even if client-side rendering is otherwise enabled.

**Configure the complexity threshold for computers and mobile devices**

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering in your computer's web browser than in your mobile
device's web browser. You can adjust how much client-side rendering happens for computers and mobile devices with the complexity thresholds. You might want to adjust the complexity thresholds if you notice that views display slowly on mobile devices. Alternatively, you might want to increase the thresholds to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computer web browsers, use the following tabadmin command:

```
tagadmin set vizqlserver.browser.render [new value]
```

By default, the complexity threshold for web browsers on mobile devices is 60. To adjust the complexity threshold for web browsers on mobile devices, use the following tabadmin command:

```
tagadmin set vizqlserver.browser.render_threshold_mobile [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tagadmin set vizqlserver.browser.render_threshold_mobile 40
```

For more information on how to use tabadmin, see How to Use tabadmin on page 1255.

**Disable client-side rendering**

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following tabadmin command to disable client-side rendering:

```
tagadmin set vizqlserver.browser.render false
```

For more information on how to use tabadmin, see How to Use tabadmin on page 1255.

**Testing with the URL Parameter**

To test server-side rendering on a session basis, type `?:render=false` at the end of the view's URL. For example:

```
http://localhost/views/Supplies/MyView?:render=false
```

If client-side rendering is disabled on Tableau Server, enter `?:render=true` to enable it for the session:

```
http://localhost/views/Supplies/MyView?:render=true
```

You can also test particular complexity thresholds on individual views to see if it's appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server's
default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view's URL. For example:

http://localhost/views/Supplies/MyView?:render=80

**Optimize for Extracts**

Try to optimize for extracts if the extract schedules correspond to high resource usage or if extracts take a long time to finish.

**Note:** This topic uses the sample performance workbook from the monitoring section.
For more information, see [Analyze Data with the Sample Performance Workbook on page 1120](#).

- When to optimize for extracts
- Ways to optimize for extracts

**When to optimize for extracts**

**High CPU usage corresponds to extract schedules**

Use the **Background Jobs** dashboard of the sample performance workbook to view the number of background jobs run by Tableau Server, including extract refresh jobs. The dashboard also displays how long background jobs are delayed—that is, the amount of time between when a background job is scheduled and when it actually runs. If you see long delays at particular times of the day or if many jobs are running at the same time, try distributing the job schedules across different times of the day to reduce the load on the server.
Also compare the times when there are many background jobs or long delays with the CPU usage of the server. Use the CPU Usage dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. Because the backgrounder process runs background jobs, it is the first process to show strain when there are many extract refresh jobs or when there are slow extract refresh jobs. Note that the CPU usage of the backgrounder process periodically but briefly reaches 100 percent. This indicates that there are intensive refresh jobs on a recurring schedule.

Note: The percent of CPU usage for individual processes may add up to more than 100 percent because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.
Extracts fail or run slowly

Use the **Background Tasks for Extracts** administrative view to determine how many extracts fail and how long extracts take to complete. Frequent failures can indicate a problem with a particular data source.

Ways to optimize for extracts

When high CPU usage corresponds to extract refresh schedules like it does in the example shown previously, you should optimize for extracts.
Adjust the extract refresh schedule

Use the **Background Jobs** dashboard of the sample performance workbook to identify optimal times for running extracts. In addition to running extracts in off-peak hours, you can distribute extract refreshes to minimize concurrent server load. If extract refreshes continue to cause problems, reduce the frequency of extract refreshes as much as possible in these ways:

- Schedule extracts for times when the server isn’t busy.
- Reduce the frequency of refreshes.

Speed up specific extracts

Use the **Background Tasks for Extracts** administrative view to identify failing extracts and long-running extracts.

- Reduce the size of extracts. You can help improve server performance by keeping the extract’s data set short, through filtering or aggregating, and narrow, by hiding unused fields. To make these changes, use the Tableau Desktop options **Hide All Unused Fields** and **Aggregate data for visible dimensions**. For more information, see [Creating an Extract](https://www.tableau.com) in the Tableau Desktop Help.

  For general tips on building well-performing workbooks, search for “performance” in the Tableau Desktop Help. To see how workbooks perform after they’ve been published to Tableau Server, you can create a performance recording. For more information, see [Create a Performance Recording](https://www.tableau.com) on page 1139.

- Use incremental refresh jobs. Incremental refresh jobs append new rows to an existing extract instead of creating the extract from scratch. This type of extract refresh runs quickly because it processes only the data that has been added since the last time the extract refresh job ran. However, it does not account for data that has been updated rather than appended to a data source. As a result, if you run incremental refresh jobs, you should still occasionally run full refresh jobs. For example, you might run a full refresh job once or twice a week for a data source instead of every day.

Configure the execution mode for extract refreshes

When you create extract refresh schedules, ensure that they run in parallel execution mode. When you run a schedule in parallel, it runs on all available backgrounder processes, even if the schedule contains only one refresh task. When you run a schedule serially, it only runs on one backgrounder process. By default, the execution mode is set to parallel so that refresh tasks finish as quickly as possible.

However, in some circumstances, it can make sense to set the execution mode to serial. For example, you might set the execution mode to serial if a very large job is preventing other schedules from running because it uses all available backgrounder processes.
Increase the number of backgrounder processes

A single backgrounder process can consume 100 percent of a single CPU core for certain tasks. As a result, the total number of instances you should run depends on the computer’s available cores. If you have Tableau Server installed in a cluster and you run backgrounder processes on a separate node, a good rule of thumb is to set the number of backgrounder process to between half the number of cores and the full number of cores of the computer running the backgrounder processes.

To increase the number of backgrounder processes, complete the following steps.

1. Stop Tableau Server and open the Tableau Server Configuration utility.
2. Click the Servers tab.
3. Click Edit.
4. Increase the number of Backgrounder processes by one.
5. Restart Tableau Server.

Isolate processes

If you have Tableau Server installed in a cluster, you see the largest benefit from moving the backgrounder processes to a separate node to avoid resource contention. This is because the backgrounder process is very CPU-intensive and running it on the same node where other CPU-intensive processes are running can slow down the server. For example, both the VizQL server process and the data engine process can be CPU-intensive.

When to Add Workers and Reconfigure

Tableau Server can scale up and out as your needs and requirements evolve. Here are some guidelines to help you figure out whether it’s time to add more worker nodes to your system, reconfigure the server, or both:

- **More than 100 concurrent users**: If your deployment is user-intensive (>100 simultaneous viewers), it’s important to have enough VizQL processes—but not so many that they exceed your hardware’s capacity to handle them. Also, enabling the Tableau Server Guest User account can increase the number of potential simultaneous viewers beyond the user list you may think you have. The administrative view can help you gauge this. For more information, see [Actions by Specific User](#) on page 1101.

- **Heavy use of extracts**: Extracts can consume a lot of memory and CPU resources. There’s no one measurement that qualifies a site as extract-intensive. Having just a few, extremely large extracts could put your site in this category, as would having very many small extracts. Extract heavy sites benefit from isolating the data engine process on its own machine.

- **Frequent extract refreshes**: Refreshing an extract is a CPU-intensive task. Sites where extracts are frequently refreshed (for example, several times a day) are often...
helped by more emphasis on the background process, which handles refresh tasks. Use the Background Tasks for Extracts on page 1103 administrative view to see your current refresh rate.

- **Downtime potential**: If your server system is considered mission critical and requires a high level of availability, you can configure it so there’s redundancy for the server processes that handle extracts, the repository, and the gateway. For more information, see High Availability on page 709.

### Performance Tuning Examples

This topic lists example process configurations for Tableau Server installations with one, two, and three nodes. Use these process configurations as a starting point when tuning the number of server processes in your installation of Tableau Server.

**One-node example: Balanced**

This example shows a 64-bit, 8+ core, 16+ GB system configured for heavy extract usage.

For this configuration, the **Process Status** table on the Server Status page would look like this:

![Server Status Table](image)

**Configuration notes**

- The primary server runs two VizQL Server processes, two Cache Server processes, and two Data Server processes. These are the recommended values and are the defaults from installation.
- As a general rule, run a Cache Server process for every VizQL Server process on the node.
- Calculate the minimum number of Backgrounder processes to run by dividing the
computer’s total number of cores by 4. To calculate the maximum number, divide the computer’s total cores by 2.

- Both the Backgrounder and Data Engine processes are CPU-intensive.
- Schedule extract refreshes for off-peak times to help the VizQL Server, Application Server, Data Engine, and Backgrounder processes to not compete for system resources.

**Two-node example: Optimized for heavy extract usage**

This example shows a possible configuration for a two-node Tableau Server deployment that handles heavy extract usage. Both nodes are 64-bit, 8+ core, 16+ GB systems.

Note that the VizQL Server, Application Server, Data Server, and Data Engine processes on the primary node are isolated from the background processes, which are running on the worker node.

The **Process Status** table for this configuration would look like this:

![Server Status Table](image)

**Configuration notes**

- The primary node runs two VizQL Server processes, two Cache Server processes, and two Data Server processes.
- As a general rule, run a Cache Server process for every VizQL Server process on the node.
- Isolate the Backgrounder processes by configuring them to run on the worker node. To calculate the minimum number of Backgrounder processes to run, divide the computer’s total number of cores by 4. To calculate the maximum number, divide the computer’s total cores by 2.

- Isolate the Backgrounder processes from the VizQL Server, Application Server, Data Server, and Data Engine processes.

**Two-node example: Optimized for user traffic**

This example shows the configuration for a two-node deployment with light extract usage and heavier viewing. Both nodes are 64-bit, 8+ core, 16+ GB systems.

The **Process Status** table for this configuration would look like this:

![Server Status Table]

**Configuration notes**

- The primary node runs two VizQL Server processes, two Cache Server processes, and two Data Server processes.

- As a general rule, run a Cache Server process for every VizQL Server process on the node.

- Isolate the Backgrounder processes by configuring them to run on the worker node. To calculate the minimum number of Backgrounder processes to run, divide the computer’s total number of cores by 4. To calculate the maximum number, divide the computer’s total cores by 2.
- Run Data Engine processes on both nodes to split view requests between the two nodes. In a deployment where extracts are refreshed infrequently, the Data Engine and Backgrounder processes can be on the same node.

- If extract refresh jobs will be run only during off hours, you can add Backgrounder processes on each node to maximize the number of parallel jobs that can run at one time.

Three-node example: Optimized for a balance between extracts and user traffic

A configuration of three nodes or more is recommended to achieve the best performance when you have a high amount of extract refreshing and usage, and a high number of concurrent users. In this example, all computers are assumed to be 64-bit, 16 core, 16+ GB systems.

The Process Status table for this configuration would look like this:

![Server Status Table]

Configuration Notes

- For this configuration, 16 cores are recommended for each node.
  
  Run two VizQL Server processes, two Cache Server processes, and two Data Server processes on the nodes that are not running Backgrounder processes.

- As a general rule, run a Cache Server process for every VizQL Server process.

- The Backgrounder processes are on their own node so that they do not compete for resources with the other processes. Because this node is dedicated to Backgrounder processes and they might consume 100% of the CPU resources, the recommended
number of Backgrounder processes is the number of cores divided by 2. However, for a more extract heavy environment, you might want to increase the number of Backgrounder processes to equal the number of cores for the node.

- Run Data Engine processes on the primary node and on the worker node that is not running Backgrounder processes. This allows Tableau Server to split view requests between the two nodes.
- The user loads for the Application Server and Data Server processes can typically be handled by a single instance of each process. However, you can configure two of each process to provide redundancy.
- Under most conditions, the primary Data Server processes and the Data Engine processes will not be a bottleneck for the system’s overall throughput as long as sufficient CPU cycles exist for them. To increase viewing capacity, add worker nodes and run dedicated VizQL Server processes on them. To increase capacity for refreshing extracts, add worker nodes and run dedicated Backgrounder processes on them.

**Performance Troubleshooting**

This section describes how to identify bottlenecks in resources, workbooks, and more to improve the performance of Tableau Server.

**Create a Performance Recording**

The Performance Recording feature in Tableau records performance information about key events as you interact with a workbook. You can then view performance metrics in a workbook that Tableau creates to analyze and troubleshoot different events that are known to affect performance:

- Query execution
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation
- Blending data
- Server blending (Tableau Server only)

Tableau support may ask that you create a performance workbook as they work with you to diagnose performance issues.

: Enable Performance Recording for a Site

By default, performance recording is not enabled for a site. A server administrator can enable performance recording site by site.
1. Navigate to the site for which you want to enable performance recording.

2. Click **Settings**: 

   ![Tableau Server Settings](image)

3. Under Workbook Performance Metrics, select **Record workbook performance metrics**.

4. Click **Save**.

---

**Start a Performance Recording for a View**

1. Open the view for which you want to record performance.

   When you open a view, Tableau Server appends ":iid=<n>" after the URL. This is a session ID. For example:

   ```
   http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1
   ```

2. Type **:record_performance=yes&** at the end of the view URL, immediately before the session ID. For example:

   ```
   http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:record_performance=yes&:iid=1
   ```

3. Load the view.

   A visual confirmation that performance recording has started is the **Performance** option in the view toolbar:

   ![Tableau View Toolbar](image)
: View a Performance Recording

1. Click Performance to open a performance workbook. This is an up-to-the-minute snapshot of performance data. You can continue taking additional snapshots as you continue working with the view; the performance data is cumulative.

2. Move to a different page or remove :record_performance=yes from the URL to stop recording.

Interpret a Performance Recording

A performance recording workbook is a Tableau dashboard that contains three views: Timeline, Events, and Query.

For information on how to create a performance recording in Tableau Server, see Create a Performance Recording on page 1139.

Timeline

The uppermost view in a performance recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.

In the Timeline view, the Workbook, Dashboard, and Worksheet columns identify the context for events. The Event column identifies the nature of the event, and the final column shows each event’s duration and how it compares chronologically to other recorded events:

![Timeline View](image)

Events

The middle view in a performance recording workbook shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.
Different colors indicate different types of events. The range of events that can be recorded is:

- **Computing layouts**
  
  If layouts are taking too long, consider simplifying your workbook.

- **Connecting to data source**
  
  Slow connections could be due to network issues or issues with the database server.

- **Executing query**
  
  - For live connections, if queries are taking too long, it could be because the underlying data structure isn’t optimized for Tableau. Consult your database server’s documentation. As an alternative, consider using an extract to speed performance.
  
  - For extracts, if queries are taking too long, review your use of filters. If you have a lot of filters, would a context filter make more sense? If you have a dashboard that uses filters, consider using action filters, which can help with performance.

- **Generating extract**
  
  To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.

- **Geocoding**
  
  To speed up geocoding performance, try using less data or filtering out data.

- **Blending data**
  
  To speed up data blending, try using less data or filtering out data.

- **Server rendering**
  
  You can speed up server rendering by running additional VizQL Server processes on additional machines.

**Query**

If you click on an *Executing Query* event in either the **Timeline** or **Events** section of a performance recording dashboard, the text for that query is displayed in the Query section. For example:
If it makes sense, you can use the query text to work with your database team on optimizing at the database level. Sometimes the query is truncated and you’ll need to look in the Tableau log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

**Performance Resources**

This topic describes external resources that you can use to monitor and tune performance.

---

**Disclaimer:** This topic includes information about third-party products. Please note that while we make every effort to keep references to third-party content accurate, the information we provide here might change without notice. For the most up-to-date information, please consult the documentation for products referenced here.

- **TabJolt.** A load generation tool that you can use to understand how Tableau Server responds to user interactions over time. Use TabJolt to establish a baseline for server performance and test deployments before pushing them to production environments.
- **TabMon.** A monitoring tool that uses Windows Performance Monitor and Java Management Extensions to record performance data about Tableau Server to a PostGreSQL database.
- **Microsoft System Center.** A set of server management products for monitoring, configuration, automation, and more.
- **HP Sitescope.** An agentless application monitoring tool.
- **Zabbix.** An open-source, real-time monitoring tool.
- **Splunk.** A tool for monitoring and analyzing machine data, including logs.
- **Graylog.** An open-source log management tool.
Maintenance

Database Maintenance

A Tableau Server administrator should perform regular database maintenance, monitor disk usage on the server, and clean up unnecessary files to free up space on the server. Taking these steps can help ensure that Tableau Server runs with maximum efficiency.

You can use the tabadmin command line tool to back up and restore your Tableau data, and to clean up (remove) unnecessary log and temporary files. Tableau data includes Tableau Server's own PostgreSQL database, which stores workbook and user metadata, data extract (.tde) files, and server configuration data. Tableau Server log files capture activity and can help you diagnose problems. Logs are written to folders on the server and you can archive and remove them to save disk space. Use the commands described in the topics below, along with the built-in Windows task scheduler to automate backing up data and cleaning up unnecessary files.

**Note:** You can only use backups made with the `tabadmin backup` command when restoring Tableau Server data. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

Back Up Tableau Server Data

Backing up Tableau Server using the `tabadmin backup` command is an important part of proper administration and maintenance of your server.

**Important:** Only backups created with the `tabadmin` command can be used to restore Tableau Server data.

The frequency of your backups depends on your environment, including how much use the server gets and how much and frequently the content and users change. Any changes or updates that happen after your backup will be lost if there is a system failure and you need to use the backup to restore Tableau Server. Keep this in mind when you determine how often you should be backing up your system.

In addition to your regular backups, you should always create a current backup of Tableau Server before upgrading to a new version.

Starting with Tableau Server version 9.3, an option to verify the integrity of the backup was included. Use this option to make sure there is no issue with the database that would result in your backup not being usable. For more information about the verify option, see Verify the Tableau Postgres Database on page 1148.
Creating a Regular Backup

When you back up your Tableau data regularly, you can quickly restore published workbooks, data sources, and other information if there is a system failure. How often you create a backup depends on how heavily your Tableau Server is used. The more activity there is, the more often you need to back the server up.

Tableau Server data consists of Tableau's own PostgreSQL database, which contains workbook and user metadata, data extract (.tde) files, and configuration data. When you use tabadmin to create a backup, all these things are saved in a single file with a .tsbak extension. If you are running a distributed installation of Tableau Server you create the backup on the primary, and data from all the nodes is backed up.

For safety, after you create the backup, store the .tsbak file on a computer that is not a part of your Tableau Server installation.

**Note**: Running the `backup` command also removes Tableau Server log files older than seven days as well as some of the information displayed in certain Tableau Server Administrative Views on page 1097.

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

   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   ```

2. Create a backup file by typing `tabadmin backup <filename>`, where `<filename>` is the name or location and name of your backup file. Beginning with version 9.3, include the `-v` option to verify the integrity of the backup.

   Starting with version 8.1, there is no need to stop the server before you create the backup.

   For example:

   ```
   tabadmin backup tabserver -v
   ```

   or

   ```
   tabadmin backup C:\backups\tableau\tabserver -v
   ```

   You can also optionally use `-d` to append the current date to the file name.

   Add `-t` followed by a path, to specify a location for temporary files that are created during the backup process. The path for the temporary files is not the location where the backup file will be written. For example:

   ```
   tabadmin backup tabserver -t C:\mytemp\tableau
   ```
In the above example, the backup file `tabserver.tsbak` will be created in the Tableau Server bin directory (`C:\Program Files\Tableau\Tableau Server\10.0\bin`) not in `C:\mytemp\tableau`.

**Note:** The `-v` option is available beginning with version 9.3 and verifies the integrity of the backup. After creating the backup, `tabadmin` verifies that the file can be used to restore the database. For more information, see [Verify the Tableau Postgres Database on page 1148](#).

### Creating a Pre-Upgrade Backup

You should always create a backup before upgrading Tableau Server. Starting with version 10.0, Tableau Server Setup offers to create a backup before upgrading to a new version. If you have created a backup yourself, before the upgrade, you can choose to skip this and save time during the upgrade. You can create a backup while Tableau Server is running and minimize the amount of time the server is unavailable during upgrade. For more on the backup option during upgrade, see [Tableau Server Upgrade Backup Options on page 688](#).

The process for creating a pre-upgrade backup is the same as for creating regular backups, with one additional consideration for distributed installations.

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

- Plan on using the `--no-config` option when you restore the backup file to your new installation. With this option, no configuration information is restored, including configuration information for the primary Tableau Server node.

- Remove the workers from the Tableau Server configuration before creating the backup.

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

### Scripting the Backup Process

The `tabadmin backup` command should be run by the Run As account that you designated for Tableau Server, or an account with administrative rights that include "modify" permissions. For more information, see [Run As Account Settings to Confirm](#) in the Tableau Server Help.
**Example backup commands**

Create the backup:

```
tabadmin backup <backupfilename> -d -v
```

Copy the backup to a location separate from Tableau Server:

```
copy <original_backup_path_and_filename> <network_drive_or_other_location_path_and_name>
```

**Backup command tips**

- Include the `-v` command option to verify the integrity of the backup (version 9.3 and later).
- Add `-d` to the command to include the date in the file name.
- For `<backupfilename>`, you can specify a path and filename if appropriate. If you specify only a filename, the backup file is saved in the current directory, the Tableau Server bin directory.
- The backup file extension is `.tsbak`.
- As a best practice, copy the backup file to another location that is separate from Tableau Server.

**Scripting Maintenance Commands**

**Saving logs before cleanup**

A backup will clean logs older than seven days. If you want to preserve the logs before cleaning them up, run the following command:

```
tabadmin ziplogs -l -n -f
```

**Cleanup**

To clean all log files older than few days, run the following command:

```
tabadmin cleanup
```

**Full cleanup (optional)**

To remove all Tableau Server log files and clean temp folders, run the following commands:

```
tabadmin stop
tabadmin cleanup
tabadmin start
```

**Additional options**

You can verify that the Tableau Server is running when the script is complete, and trigger an email if it is not. If you are checking the status shortly after starting Tableau Server, give 90 seconds for the processes to warm up.
[sleep 90]
if tabadmin status != 'RUNNING' then <code_to_email_an_alert>

Full example script

The following is an example script combining logs, backup, cleanup and alerting capabilities.

Note: Remember that if you use a version of Tableau Server earlier than 8.1, you need to include the tabadmin stop command to stop Tableau Server before running the backup and maintenance commands, and then tabadmin start to restart Tableau Server when finished.

tabadmin ziplogs -l -n -f
copy logs.zip <path_and_filename>
tabadmin backup <backupfilename> -d -v
copy <original_backup_path_and_name> <other_location_path_and_name>
tabadmin cleanup
[sleep 90]
if tabadmin status != 'RUNNING' then <code_to_email_an_alert>

Scheduling

If you follow best practice and back up Tableau Server regularly, you may want to schedule the task, depending on how often content on your server is updated, and based on your business needs. When you create a backup and maintenance script, you can use the Windows Task Scheduler to schedule when to run it.

- Select Start > Control Panel > Task Scheduler

Follow the Windows Task Scheduler wizard to complete the setup. For more information, see Task Scheduler How To in the Microsoft TechNet library.

Note: When creating the scheduled task, use the Tableau Server Run As account. To confirm what account that is, select Start > All Programs > Tableau Server > Configure Tableau Server.

Verify the Tableau Postgres Database

Under rare circumstances, the PostgreSQL database that Tableau Server users for its repository can become corrupted. (If corruption occurs, it's often a result of a hardware problem on the computer hosting the repository.) The corruption may not be immediately obvious and may not cause the database to stop functioning, but it can impact your ability to restore a backup of the data.

To help you avoid problems due to database corruption, as a best practice you should regularly perform these tasks:
Verify the integrity of the PostgreSQL database.
Back up your Tableau data.

Note: The verify option is available beginning with version 9.3 of Tableau Server.

Verifying the database
You can verify database integrity while you perform a backup, or you can verify the database, or a backup of the database, as a separate step.

To verify the database during a backup, add the -v option to the backup command:

tabadmin backup tabserver -v

To verify the current database or a backup of the database, use the verify_database command:

tabadmin verify_database

Note: You do not need to stop Tableau Server to verify the database.

This command verifies that a backup of the PostgreSQL database can be restored successfully. If you cannot restore the database, your backups aren’t useful, and Tableau Server upgrades can fail.

Verify the integrity of the Tableau PostgreSQL database using the procedure below.

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

   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"

   Note: If you are running a distributed installation of Tableau Server, perform this step on the primary computer.

2. Verify the current database or a backup of the database. You do not need to stop Tableau Server to verify the database.

   - Verify the integrity of the Postgres database by typing the following:

     tabadmin verify_database

   - Verify the integrity of a backup file by typing the following:

     tabadmin verify_database -f <filename>, where <filename> is the name of your backup file.

     For example:
tabadmin verify_database -f
c:\backups\tableau\tabserver\tsbak

- Verify the integrity of the database while creating a backup of the Tableau Server data by adding -v to the backup command:

tabadmin backup tabserver -v

You can optionally specify a location for temporary files that are created during the verification process.

tabadmin verify_database -t C:\mytemp\tableau

If you don't specify a location, the default Tableau temp folder is used.

**Restore from a Backup**

Use the `tabadmin restore` command to restore your Tableau Server data. You might do this if you had a system failure and need to restore your data, if you need to switch back to a previous version of Tableau Server (for example, if there is a problem with an upgrade), or if you are moving Tableau Server to new hardware.

Only backups created using `tabadmin backup`, or created by the Tableau Server uninstall process can be used to restore Tableau Server data.

When you use `tabadmin` to restore your Tableau data, the contents of the PostgreSQL database, data extracts, and configuration files are overwritten with the content in the backup file (.tsbak). If you are running a distributed installation of Tableau Server, perform the restore on the primary node.

**Note:** Beginning with version 9.3, a `verify_database` command allows you to verify that your backup file does not have a hidden problem that will cause the restore to fail. If you have version 9.3 or later, verify the integrity of the backup using the `tabadmin verify_database` command before you restore the database. For more information, see **Verify the Tableau Postgres Database** on page 1148.

**Restore Tableau Server from a backup file**

1. Stop the server:

   `tabadmin stop`

2. Restore from a backup file:

   `tabadmin restore <filename>`

   In the above line, replace `<filename>` with the name of the backup file you want to restore from.
Note: When restoring from a backup, Tableau Server prompts for the password of the Run As user account. You can type the password when prompted, or use a .txt file that contains only the password (if you are scripting the restore, for example).

To restore only the data but no configuration settings (for example, if you are moving Tableau Server to a new computer), include the --no-config option:

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

3. Restart the server:

```
tabadmin start
```

4. If you ran the `tabadmin assetkeys` command at any time before you created the backup file that you're now restoring, run the following command:

```
tabadmin assetkeys --validate
```

You'll be prompted to enter the passphrase needed to re-create the custom encryption keys in use in the backup file.

When you restore a .tsbak file, Tableau Server automatically creates a copy of its current data folder, names it `tabsvc.bak-*`, and places it in `ProgramData\Tableau\Tableau Server\data`. This folder is an emergency backup of Tableau Server data which Tableau Support may be able to use in case something goes wrong during backup restoration.

When the restore is complete and you have verified that Tableau Server is running correctly with all the expected data, it's safe to remove any `tabsvc.bak-*` folders from `ProgramData\Tableau\Tableau Server\data` to free additional disk space. In Tableau Server clusters, `tabsvc.bak-*` folders are created on each machine running Tableau Server.

Important: Only remove the `tabsvc.bak-*` folders. Do not remove the `tabsvc` folder, which is also located under `ProgramData\Tableau\Tableau Server\data`. It contains necessary Tableau Server data.

Providing the Run As User Password in a File

Note: If you choose to store private information like passwords in a file, keep security considerations in mind. As a best practice we recommend you include a step in your process to remove the file after it is used to prevent unauthorized access.

When restoring from a backup, Tableau Server prompts you for the password of the Run As user account. If you are scripting the restore, you can provide the password in a .txt file that contains only the password.
For example, to restore to a new computer, use these commands:

```bash
tabadmin restore --no-config <backupfilename.tsbak> --password-file <passwordfile.txt>
tabadmin start
```

By default, Tableau Server looks in its bin folder for the password file. If you save the file in a different location, include the path to the file. For example:

```bash
tabadmin restore --no-config <backupfilename.tsbak> --password-file <c:\<location>\passwordfile.txt>
tabadmin start
```

**Recover Extracts from a Backup**

The file `uninstall-<version>.tsbak` (for example, `uninstall-9.3.tsbak`) is created as part of the uninstall process. After you upgrade to version 10.0, 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 10.0 Tableau Server bin directory, type the following:**

   - **Windows Server 2012, Windows Server 2008, Windows 7, Windows 8:** `tabadmin restore \ProgramData\Tableau\Tableau Server\uninstall-9.3.tsbak`
   - **32-bit Tableau Server installed on 64-bit Windows Server:** `tabadmin restore \Program Files (x86)\Tableau\Tableau Server\uninstall-9.3.tsbak`
   - **32-bit Tableau Server installed on 32-bit Windows Server:** `tabadmin restore \Program Files\Tableau\Tableau Server\uninstall-9.3.tsbak`

**Remove Unneeded Files**

As a best practice, you should monitor space usage on your server. If you need to make more space available, you can use the `cleanup` command to remove Tableau Server log files, temporary files, and unneeded entries in the PostgreSQL database. If you might need older logs for troubleshooting, you should create a log file archive before doing the cleanup. For more information, see [Archive Logs on Command Line (tabadmin)](page) on page 1189.

To perform a cleanup, use this command:

```bash
tabadmin cleanup
```

You can add the `restart` option, which is the equivalent of running `tabadmin stop`, `tabadmin cleanup`, and then `tabadmin start:`
The files and database entries that are removed by `tabadmin cleanup` command depend on whether Tableau Server is running or stopped. Therefore, to clean up all possible files and database entries, you should run `tabadmin cleanup` twice: once when Tableau Server is running, and once when it is stopped. Here’s a summary of what’s removed when you run `tabadmin cleanup` with the server running and stopped.

**When you run `tabadmin cleanup` with Tableau Server stopped:**

- All log files are removed from `ProgramData\Tableau\Tableau Server\data\tabsvc\logs`. *(Log files from `ProgramData\Tableau\Tableau Server\logs` are not removed.)*
- Temporary files are removed from `ProgramData\Tableau\Tableau Server\temp` and `ProgramData\Tableau\Tableau Server\data\tabsvc\temp`.
- No rows for HTTP requests are removed from the `http_requests` table of the Tableau Server PostgreSQL database, because the database is not accessible when the server is stopped.

**When you run `tabadmin cleanup` with Tableau Server running:**

- Log files older than the log file rotation interval are removed from `ProgramData\Tableau\Tableau Server\data\tabsvc\logs`. *(By default, the rotation interval is one day.) Active logs and log files from `ProgramData\Tableau\Tableau Server\logs` are not removed.*
- Temporary files are not removed.
- Files that are in use (that is, locked by the operating system) are not removed.
- Rows for HTTP requests that are older than seven days are removed from the `http_requests` table of the Tableau Server PostgreSQL database.

**Note:** Rows for HTTP requests older than seven days are also removed when you back up Tableau data. For more information, see *Back Up Tableau Server Data* on page 1144.

**More Information**

For more information about the Tableau Server PostgreSQL repository, see *Collect Data with the Tableau Server Repository* on page 1117.

For tips on how to automate running the cleanup and backup commands, refer to the following Knowledge Base article: *Server Backup and Maintenance Automation*
If you have created a log file archive but you no longer need it, you can remove it from the server by using the **Delete Snapshot** option on the Status page. For more information, see *Archive Logs on Status Page (Snapshot)* on page 1187.

**Server Maintenance**

As an administrator, you will want to check the status of the server, analyze and monitor the activity on the server, manage scheduled tasks, or perform certain maintenance activities such as clearing saved data connection passwords. In addition, there are several settings that you may want to specify to customize the user experience for people using the server. You can do some of these tasks from the General page of the Status page and others from the Settings page.

**View Server Process Status**

You can use the Process Status table on the Server Status page to view the state of Tableau processes on each Tableau server:

Possible status indicators are listed at the bottom of the table:
When Tableau Server is functioning properly, most processes will show as Active, Busy or Passive (Repository):

- **Active**—The process is functioning as intended. See File Store in Troubleshoot Server Processes on page 1212 for details on possible active states.
- **Busy**—The process is completing some task. See File Store and Repository in Troubleshoot Server Processes on page 1212 for more information.
- **Passive**—The repository is in passive mode
- **Unlicensed**—The process is unlicensed.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

If there is additional information, a message appears below the status icon:

![Ready for removal](image)

For more information about troubleshooting process status, see Troubleshoot Server Processes on page 1212.

**Access Status Remotely**

As the Tableau administrator, only you can see the Status table, but you can grant remote access to make the machine-readable version of the Status table available to non-admin users and to computers other than the one that’s hosting Tableau Server. You might do this as part of a remote monitoring process.

To grant remote access to Tableau Server status:

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

   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
   ``

2. Enable remote access by typing the following:

   ```
   tabadmin set wgserver.systeminfo.allow_referrer_ips <ip address>
   ``

   In the above command, `<ip address>` is the IPv4 address of the computer for which you want to enable remote access to the Tableau Server status XML.
   
   For example:

   ```
   tabadmin set wgserver.systeminfo.allow_referrer_ips 10.32.139.31
   ```
If you are enabling remote access for more than one computer, use commas to separate each IP address.

```
tabadmin set wgserver.systeminfo.allow_referrer_ips 10.32.139.31,10.32.139.35
```

3. Commit the configuration change:

```
tabadmin config
```

4. Restart Tableau Server:

```
tabadmin restart
```

Now, users of computers with the IP addresses that have been added can view Tableau process status by entering the URL `http://<server>/admin/systeminfo.xml` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo.xml`).

This functionality can also be used as part of an automated remote monitoring process.

**Get Process Status as XML**

To get a machine-readable version of the server process status, that is, a version of the status formatted in XML, use the following URL:

```
http://my_tableau_server/admin/systeminfo.xml
```

You must be signed in to Tableau Server to view the machine-readable process status, or have enabled remote access.

The server returns a status report similar to the following:

```
<systeminfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <machines>
    <machine name="my_tableau_server">
      <repository worker="my_tableau_server:8060" status="Active" preferred="false"/>
      <dataengine worker="my_tableau_server:27042" status="Active"/>
      <applicationserver worker="my_tableau_server:8600" status="Active"/>
      <apiserver worker="my_tableau_server:8000" status="Active"/>
      <vizqlserver worker="my_tableau_server:9100" status="Active"/>
      <dataserver worker="my_tableau_server:9700" status="Active"/>
    </machine>
  </machines>
</systeminfo>
```
<backgrounder worker="my_tableau_server:8250" status="Active"/>
<gateway worker="my_tableau_server:80" status="Active"/>
<searchandbrowse worker="my_tableau_server:11000" status="Active"/>
<cacheserver worker="my_tableau_server:6379" status="Active"/>
<filestore worker="my_tableau_server:9345" status="Active" pendingTransfers="0" failedTransfers="0" syncTimestamp="2015-02-27T20:30:48.564Z"/>
<clustercontroller worker="my_tableau_server:12012" status="Active"/>
<coordination worker="my_tableau_server:12000" status="Active"/>
</machine>
</machines>
</service>
</systeminfo>

Status values in the XML

- **<process> worker** - The name of the node running the process and the port the process is using.
- **status** - The status of the process on the node. Possible values are: Active, Passive, Unlicensed, Busy, Down, ReadOnly, ActiveSyncing, StatusNotAvailable, StatusNotAvailableSyncing, NotAvailable, DecommissionedReadOnly, DecomisioningReadOnly, and DecommissionFailedReadOnly
- **pendingTransfers** - A count of the workbook or data source extracts the node needs to get to be fully synced. These represent items that were published to this file store node, and items that were published to other file store nodes and need to be copied to this node.
- **failedTransfers** - A count of the workbooks or data sources that did not transfer successfully to this file store node during the last automated job. The automated job normally runs about every 15 to 30 minutes, but may take longer when transferring a large number of extracts or large extracts.

Failed transfers do not necessarily indicate a problem with Tableau Server. The recurring automated job will normally transfer files that failed during the previous sync. Reasons for failed file transfers are listed in the logs.
- **syncTimestamp** - The time in UTC of the last automated job that ran and synchronized files.
**Troubleshoot Server Processes**

When Tableau Server is functioning properly, processes will show as Active, Busy or Passive (Repository). If there is additional information, a message appears below the status icon:

<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td></td>
<td>&quot;Node degraded&quot;</td>
<td>• Repository on the node is stopped.</td>
<td>No action is necessary unless the cluster controller is regularly down or is down for an extended period of time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Node cannot respond to fail-over elsewhere in the cluster.</td>
<td>If that occurs, take the following actions, in order, until the problem is resolved:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• If Tableau Server is configured for high availability and this is the active repository, fail-over to the second repository occurs.</td>
<td>1. Check disk space. If disk space is limited, save the log files (use <code>tabadmin ziplogs</code>) in case you need them for Support, then remove unnecessary files (<code>tabadmin cleanup</code>).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• No status available for repository or file store on this</td>
<td>2. In Windows Task</td>
</tr>
</tbody>
</table>

Possible status indicators are:

- Ready for removal

Use this table to help troubleshoot issues with your Tableau Server installation.
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>node.</td>
<td></td>
<td>Manager, stop the cluster-controller.exe process tree and let it restart automatically.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3. Restart Tableau Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (tabadmin stop), clean up files (tabadmin cleanup --reset-coordination), and then start the cluster (tabadmin start).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5. If Cluster Controller continues to show as down, save the log files (tabadmin zip logs) and contact Support.</td>
</tr>
</tbody>
</table>

**File Store**

File Store status only reflects the state of the file store when the page was loaded.

<table>
<thead>
<tr>
<th></th>
<th>none</th>
<th></th>
<th>No extracts were being synchronized when the page was loaded. (It is possible that the recurring &quot;catch-all&quot; job is running and syn-</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>None.</td>
</tr>
<tr>
<td>Process</td>
<td>Status (Icon)</td>
<td>Message</td>
<td>Implications</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>---------</td>
<td>--------------</td>
</tr>
</tbody>
</table>
|         | ![Green Circle] | "Synchronizing" | - Extracts were being synchronized across file store nodes when the page was loaded.  
- Initial status following installation (both single-node and multi-node). Should disappear within 15 or 20 minutes. | None. |
|         | ![Red X] | "Data Extracts unavailable" | - Single-node installation: existing extracts may be available but publish/refresh will fail.  
- Multi-node installation: extract synchronization will fail for this node. | No action is necessary unless the file store is regularly down or is down for an extended period of time.  
If that occurs, take the following actions, in order, until the problem is resolved:  
1. Check disk space.  
   If disk space is limited, save the log files (`tabadmin ziplogs`) in case you need them for Support, and then |
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;Decommissioning&quot;</td>
<td>- File store is in read-only mode. - Any unique files on this node are</td>
<td>Wait until the status message changes to &quot;Ready for removal&quot;.</td>
<td>remove unnecessary files (tabadmin cleanup). 2. Stop the filestore.exe process using Windows Task Manager and let it restart automatically. 3. Restart Tableau Server. 4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (tabadmin stop), clean up files (tabadmin cleanup --reset-coordination), and then start the cluster (tabadmin start). 5. If the file store continues to be down, save the log files (tabadmin zip-logs) and contact Support.</td>
</tr>
<tr>
<td>Process</td>
<td>Status (Icon)</td>
<td>Message</td>
<td>Implications</td>
<td>Actions</td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td>---------</td>
<td>--------------</td>
<td>---------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|         | ![Checkmark]  | "Ready for removal" | • File store is in read-only mode.  
• Ready for user to stop cluster and remove data engine/file store or remove entire node. | Stop Tableau Server (tabadmin stop) and then run the Configuration utility to remove Data Engine and File Store or the entire node. |
|         | ![Checkmark]  | "Decommission failed" | • File store is in read-only mode.  
• At least one unique file failed to replicate to another file store node. | Take the following actions in order until the problem is resolved:  
1. Run the `tabadmin decommission` command again.  
2. Check disk space on other file store nodes. Decommissioning will fail if another file store node does not have enough space to store all the extracts.  
3. Check the `tabadmin.log` file on the primary node and workers for errors.  
4. Stop Tableau |
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td>Server (<code>tabadmin stop</code>) and then try running the <code>tabadmin decommission</code> command again.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5. Put the file store node back into read/write mode (<code>tabadmin recommission</code>), collect logs, and then contact Support.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6. With Support: copy and merge extracts directory from this file store node to the same directory on another file store node.</td>
</tr>
</tbody>
</table>
| Repository | ![Setting up](https://example.com) | "Setting up" | - Passive repository is being synchronized with active repository.  
- Repository is not ready to handle failover.  
- Repository may have gotten more than two minutes behind active repository and is being setup | Wait until the repository status message changes to "Passive".  
If this message does not appear, or if it is taking a long time:  
1. Check disk space and free space if possible.  
2. Check cluster controller logs for errors.  
3. Restart node. |
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>again (this is faster than waiting for a sync).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Failover occurred and this former active repository is rejoining the cluster.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Synchronize" /></td>
<td>&quot;Synchronizing&quot;</td>
<td>• Repository is synchronizing, for example after a failover.</td>
<td>None.</td>
</tr>
<tr>
<td></td>
<td><img src="image" alt="Stop" /></td>
<td>none</td>
<td>• If the installation is configured for high availability, failover of the repository occurred.</td>
<td>Take these actions in order until the problem is resolved:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Processes are restarting with updated database connection configurations after failover.</td>
<td>1. Wait several minutes for cluster controller to attempt to restart.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• If another active repository is not available, Tableau Server is down.</td>
<td>2. Restart Tableau Server (<a href="#">tabadmin restart</a>).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3. Check disk space to make sure there is free space. Collect logs (<a href="#">tabadmin ziplogs</a>) in case you need them for Support, and then cleanup files (<a href="#">tabadmin cleanup</a>).</td>
</tr>
<tr>
<td>Process</td>
<td>Status (Icon)</td>
<td>Message</td>
<td>Implications</td>
<td>Actions</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------</td>
<td>------------</td>
<td>--------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>none</td>
<td>• Working as intended.</td>
<td>4. Restart Tableau Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Node is ready if needed for failover.</td>
<td>5. Stop Tableau Server, collect logs and cleanup coordination service files (tabadmin cleanup --reset-coordination)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7. Collect logs (tabadmin zip logs) and contact Support.</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>×</td>
<td>none</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>!</td>
<td>none</td>
<td></td>
<td>For information about unlicensed status for a VizQL Server process, see Handle an Unlicensed VizQL Server Process on page 1199.</td>
</tr>
</tbody>
</table>

**Archive Logs on Status Page (Snapshot)**

You can generate and download a snapshot (archive) of the Tableau Server log files from a web browser, without opening a command prompt. This zipped snapshot contains a copy of up to seven days of log file data from Tableau Server and any worker servers (if you have a
distributed environment). The snapshot process does not change or remove either the Tableau Server log files or the log archives created with tabadmin.

**Note** To specify the amount of data you want to collect or the name of the zip file you are creating, use tabadmin to create an archive of server logs. For more information, see *Archive Logs on Command Line (tabadmin)* on page 1189.

To generate a snapshot of server log files:

1. Open the Status page:
   - Multi-site: Select **Server > Status**.
   - Single-site: Select **Status**.

2. Click **Generate Snapshot** to create a snapshot of the Tableau Server logs. The Generate Snapshot button is available only if there is no existing snapshot.

   **Note**: This option is available whether or not you have created log archives with tabadmin.

3. Select the number of days of logs you want to include. The default is **Last 7 days**, but you might want to select fewer if you want to reduce the size of the zip file. For example, if you just reproduced an issue and are collecting logs related to the issue, you may want to select **Today** to create the smallest zip file necessary.

4. Click **Download Snapshot** to download the log snapshot to your web browser’s default download location. This option is available after you create a snapshot.

   Google Chrome shows you the download in the bottom of the window:
5. Click the arrow and then click **Open** to unzip the snapshot or **Show in folder** to see where it was downloaded:

6. (Optional) Click **Delete Snapshot** to delete a log snapshot. This option is available after you create a snapshot. You need to delete the existing snapshot before you can create a new one.
For example, you might want to delete the snapshot that you created before an event that you want to investigate.

**Uploading log archives for Tableau Support**

If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.

**Rebuild the Search Index**

If search is returning incomplete or incorrect results, or if the Search & Browse process is down for an extended period of time, you may need to rebuild the search index.

**Important:** The recommended way to reindex search is to use the `tabadmin` `reindex` command while Tableau Server is stopped. Reindexing while the server is running can result in content, including sites and projects, temporarily disappearing from server pages.

The search index is built or rebuilt at key points during installation or upgrade of Tableau Server, when you restore a backup, and when you add the Search & Browse process to a new or existing node. The index is kept updated by a background task when content changes. If necessary you can force a rebuild of the index using the `tabadmin` `reindex` command.

1. To rebuild the search index, click **Status**.

   ![Server Status](image)

   In a multi-site environment, select **Server > Status**.

2. At the bottom of the page, click **Rebuild Search Index**.

   ![Rebuild Search Index](image)

**Note:** You might not see all available server content while the index is rebuilding, and larger search indexes can take longer times to finish rebuilding. Reindexing first removes
all content from the index, and then re-adds the content to the index. If you do this while Tableau Server is running, users who are logged into the server will see content disappear, and then slowly start to reappear in server pages. Reindexing while Tableau Server is stopped provides a better user experience.

Clear Saved Data Connection Passwords

As the administrator, if you enable the Allow users to save data source passwords setting, server users can save data source passwords across multiple visits and browsers so they are no prompted for their credentials each time they connect to a data source.

You can reset the data source passwords for all Tableau Server users. Doing this forces them to sign in to the data sources the next time they visit a view that requires database authentication. Server users can also clear their saved data connection passwords on an individual basis using their User Preferences page.

To clear saved data connection passwords for all server users:

1. In a site, click Settings > General.

2. Under Saved Credentials, click Clear All Saved Credentials.

View Server Licenses

Server administrators can view the license and product key information for Tableau Server.

Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.
To view server licenses

- In a site, click **Settings > Licenses**.

If you have a user-based Tableau Server license, you can review how these levels have been distributed.

If you have a core-based Tableau Server license, the Licenses page shows how many cores are allowed, how many have been licensed, and how many are in use (and on what server computers).

Also see:

- **Overview of Tableau Server Licenses** on page 640
- **Handle an Unlicensed Server** on page 1198.

**Add Capacity to Tableau Server**

You may need to add capacity to your Tableau Server installation to allow you to increase the number of users (if you have a user-based license) or the number of cores (if you have a core-based license).

Tableau Software will provide you with a new product key that adds capacity to your existing Tableau Server installation. You need to activate this key and use it together with your existing product key(s) to get the combined capacity you are licensed for.

Follow the steps below to add a product key to Tableau Server.

**Note:** This process requires a restart of Tableau Server.

1. Start the Product Key Manager:
   - In Windows, select **Start > All Programs > Tableau Server <version> > Manage Product Keys**.

2. Click **Activate** in the Manage Product Key dialog box:
3. Enter or paste your new product key and click **Activate**:

4. Restart Tableau Server after registration is complete.

**Synchronize All Active Directory Groups on the Server**

As a server administrator, you can synchronize all Active Directory groups on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.
The **Last synchronized** time indicates the time that synchronization most recently began.

**Synchronize Active Directory groups on a schedule**

1. **Single-site**: Click **Settings > General**.
   - **Multisite**: In the site menu, click **Manage All Sites** and then click **Settings > General**.

2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.
3. Select the frequency and time of synchronization.

4. Click Save.

Synchronize all Active Directory groups on demand

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. **Single-site**: Click Settings > General.

   **Multisite**: In the site menu, click Manage All Sites, and then click Settings > General.

2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

View synchronization activity

You can view the results of synchronization jobs in the Background Tasks for Non Extracts administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site**: Click **Status**.

   **Multisite**: In the site menu, click Manage All Sites and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.

3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.
Set the minimum site role for users in an Active Directory group

In the Groups - Details page, you can set the minimum site role for group users to be applied during Active Directory synchronization.

This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as Unlicensed by default.

**Note:** A user's site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization on page 791.

1. In a site, click Groups.
2. On the Groups page, select a group.
   - Click Actions > Minimum Site Role.
3. Select the minimum site role, and then click Change Site Role.

What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user's content.
before removing the user’s account completely. For more information, see Sync behavior when removing users from Active Directory on page 1253.

**Set the Default Start Page for All Users**

By default, Tableau Server installs with the Views page as the default start page for all users. As the administrator, you can change this to another page that all users have access to, such as the Workbooks page. Individual users will be able to override your setting (search for "Access Your Profile and Account Settings" in the Tableau Server Help for details).

**To set the default start page for all users**

1. Navigate to the page you want to be the default page.
2. Click your name on the upper right corner of the page.
3. Select **Make This the Start Page for All Users**.

![Admin Menu](image)

**Disable Automatic Client Authentication**

By default, after Tableau Desktop clients first successfully sign in to Tableau Server, they are automatically authenticated in the future. As a Tableau Server administrator, however, you can disable automatic authentication.

To immediately disconnect all clients from Tableau Server and require users to sign in every time they connect:

1. In the site menu, click **Manage All Sites**, and then click **Settings > General**.
2. Under **Connected Clients**, clear the option **Allow clients to automatically connect to Tableau Server**.
3. Click **Save**.
Tableau Server Monitor

Tableau Server Monitor is installed as part of Tableau Server and can be accessed in the Windows system tray.

Using this tool you can start and stop the server, open Tableau Server, and display server status.

**Open the Server**
This command launches Tableau Server in your web browser. This is an easy way to access the web application and the associated maintenance tools.

**Start/Stop the Server**
You can start and stop the server using these commands. When you stop the server you make it unavailable to all of your users and terminate any sessions that are currently in progress. If someone is publishing a workbook when the server is stopped, the process is abandoned. As a result, only some of the worksheets in the workbook may be published to the server. Because stopping the server can be very disruptive to your users, be sure to warn them prior to this operation or plan maintenance during non-business hours.

**Restart the Server**
This command restarts the server. While the server is restarting it will be unavailable to all users. Be sure to warn your users of the outage prior to this operation. You will need to restart the server if you make changes to the Tableau Server configuration.

**Display Status**
This command opens a screen tip containing the status of each process. For more detailed status, use the Maintenance page.

**Manage Product Keys**
This command opens the product key manager where you can add and remove product keys.
Exit

This command closes Tableau Server Monitor. It does not stop Tableau Server. You can re-open the application by selecting All Programs > Tableau Server 10.0 > Tableau Server Monitor on the Windows Start menu.

Server Settings (General)

The following settings are available on the General page in Server - Settings.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Embedded Credentials - Allow publishers to embed data source credentials in a workbook</strong></td>
<td>Allows publishers to attach passwords to published workbooks that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks and are only accessible on server. That is, when the workbook is opened in Tableau Desktop, users will still need to enter a user name and password to connect to the data source. When this setting is turned off, all existing embedded passwords are saved but are not used for authentication. If you turn the setting back on, users don't have to re-embed the passwords.</td>
</tr>
<tr>
<td><strong>Embedded Credentials - Allow publishers to schedule data extract refreshes</strong></td>
<td>Allows publishers to assign workbooks to schedules. This option is only available if Allow publishers to embed data source credentials in a workbook is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.</td>
</tr>
<tr>
<td><strong>Saved Credentials - Allow users to save data source passwords</strong></td>
<td>Allows users to save data source passwords across multiple visits and browsers. By default users can choose to &quot;Remember my password until I sign out,&quot; which lets them save their password during a single browser session. When the Saved Passwords setting is selected a user can instead choose 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.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Saved Credentials - Allow users to save data source access tokens</strong></td>
<td>Allows users to store access tokens with their user preferences. Access tokens are provided by cloud data sources that support OAuth connections, and they are used instead of user names and passwords to grant access to the data. For more information, see <a href="#">OAuth Connections</a> on page 1061.</td>
</tr>
<tr>
<td><strong>Connected Devices - Allow devices to automatically connect to Tableau Server</strong></td>
<td>Controls whether mobile users must sign in and provide their credentials every time they connect to Tableau Server, or if users can connect with their devices to Tableau Server without providing credentials after they authenticate their device successfully the first time. For more information, see <a href="#">Disable Automatic Client Authentication</a> on page 1175.</td>
</tr>
<tr>
<td><strong>Guest Access - Enable Guest account</strong></td>
<td>Allows users to view and interact with embedded views without having to sign in to a Tableau Server account. Permission can be assigned to the Guest User account to control the interactivity allowed for each view. This option is only available if you have a core-based server license. This option can be used with <a href="#">Enable automatic logon</a>, an option you can select during Setup.</td>
</tr>
<tr>
<td><strong>Default Start Page</strong></td>
<td>Takes you to the server's current default start page for all users. For more information on how to change the default start page, see <a href="#">Set the Default Start Page for All Users</a> on page 1175. Individual users will be able to override this setting (search for &quot;Access Your Profile and Account Settings&quot; in the Tableau Server Help for details).</td>
</tr>
<tr>
<td><strong>Language and Locale</strong></td>
<td>Controls the language used for the server user interface and the locale used for views. Individual users can override this setting on their Account Settings page. Also, web browser settings are evaluated first to determine which language and locale should be used. For more information, see <a href="#">Language and Locale</a> on page 661.</td>
</tr>
<tr>
<td><strong>Active Directory Synchronization - Synchronize Active Directory groups on a regular schedule</strong></td>
<td>Controls the synchronization of all Active Directory groups in Tableau Server based on a schedule</td>
</tr>
</tbody>
</table>

- 1178 -
that you specify after you select the option **Synchronize Active Directory groups on a regular schedule**. For more information, see **Synchronize All Active Directory Groups on the Server** on page 1171.

| Reset to Default Settings | Any server settings that have been changed since setup are returned to their original state. |

**Troubleshooting**

Use the following topics to troubleshoot issues you may be having with Tableau Server. For tips on troubleshooting trusted authentication, see **Troubleshoot Trusted Authentication** on page 1039:

**Work with Log Files**

Tableau Server creates log files as a normal part of its activities. You may need to use the server log files when you are troubleshooting issues with Tableau Sever or if Tableau Support requests logs to help you resolve an issue.

You can create a zipped log file archive (snapshot) from the command line on the server, or using the Generate Snapshot option on the Maintenance page. The zipped archive contains copies of the logs you can copy or download using a web browser, and send to Tableau Support. Once you have a copy of the archive, you can delete the archive from your server. For more information on creating, downloading and deleting log file archives, see **Archive Logs on Status Page (Snapshot)** on page 1187.

This collection of topics provides information about how to create log file archives, the contents of specific log files, and details about when and how you might want to look at a log.

**Investigating Tableau Server Issues**

The range and complexity of possible issues with Tableau Server means that there is no simple process you can use to investigate all problems, but a general approach would include these steps:

1. **Clean up** existing log files to reduce their size. For more information, see **Remove Unneeded Files** on page 1152.
2. **Set the appropriate logging level**. This is something that Tableau Support will instruct you on. For more information, see **Change Logging Levels** on page 1197.
3. **Reproduce the issue** you are troubleshooting so the logs capture the events related to the problem.
4. **Create an archive** of the logs. For more information see **Archive Log Files** on page 1184.
Important: Use this archive when looking at the log files. You should not edit, move or delete any files directly on the server.

5. **Review the server configuration file** (`\config\tabsvc.yml`) to get a basic understanding of the server environment.

6. **Review the admin log** (`\logs\tabadmin.log`) to understand any maintenance that has been done on the server.
   
   Search for run as: `<script>` to find entries specific to tabadmin activity.

7. **Review the Apache logs** (`\httpd\access.###_##_##_##.log` and `\httpd\error.log`) for requests that may be related to the issue you are investigating.

   The Apache logs will contain a fair amount of "noise" that does not apply to issues you are experiencing.
   
   - If you find a request that seems to be related to your issue, search `\vizqlserver` for entries that include the unique request ID from the Apache logs.
   - Look for the response code and message associated with the request ID.
   - Search for the name of the workbook, view, dashboard, or data source that is related to your issue. Make sure to look for a relevant timestamp.
   - If you find a request that seems to be related to your issue, look at the response code associated with the request. (200s are good, 500s indicate problems.)
   - Locate the unique request ID associated with the request you’ve identified (the unique request ID is a 24 character alphanumeric string at the very end of the request).

8. **Review the log archive** further to search for other messages and possible errors.
   
   - Use the request ID from the Apache logs to search the `\vizqlserver` folder of the log archive for files containing related log entries. Look for indications of a problem (for example, error messages or long-running queries).

9. **Contact support**

   If you are not able to solve the issue yourself, or if requested by Tableau Support, send the zipped archive to Tableau.

See the following topics for more information:

**Tableau Server Processes**

There are Tableau Server processes whose default configuration you can change to achieve different results. The topics **Performance Tuning Examples** on page 1135 and **High Availability** on page 709 describe some of the approaches you can take. High-level status for each process is displayed on the server’s Status page and more detailed information related to
some of the processes—such as the background process—is in the Administrative Views on page 1097 topic.

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

For information on log files generated by these processes, see Server Log File Locations on page 1190.

<table>
<thead>
<tr>
<th>Process</th>
<th>File Name</th>
<th>Purpose</th>
<th>Multi-Threaded?</th>
<th>Performance Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>vizportal.exe</td>
<td>Handles the web application, REST API calls, supports browsing and searching</td>
<td>Yes</td>
<td>Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>backgrounder.exe</td>
<td>Executes server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
</tr>
<tr>
<td>Cache Server</td>
<td>redis-server.exe</td>
<td>Query cache</td>
<td>No</td>
<td>A query cache distributed and shared</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>acros across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and API server and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>clustercontroller.exe</td>
<td>Responsible for monitoring various components, detecting failures, and executing failover when needed</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Coordination Service</td>
<td>zookeeper.exe</td>
<td>In distributed installations, responsible for ensuring there is a quorum for making decisions during failover</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server installations of more than five nodes, also installed on the first four worker nodes.</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>------------</td>
<td>---------------</td>
<td>--------------------------------------------------</td>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data Engine</td>
<td>tdeserver64.exe</td>
<td>Stores data extracts and answers queries</td>
<td>Yes</td>
<td>The data engine’s workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory consumption is primarily based on the size of the data extracts being loaded. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.</td>
</tr>
<tr>
<td>Data Server</td>
<td>dataserver.exe</td>
<td>Manages connections to Tableau Server data sources</td>
<td>Yes</td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.</td>
</tr>
<tr>
<td>File Store</td>
<td>filestore.exe</td>
<td>Automatically replicates extracts across data engine nodes</td>
<td>n/a</td>
<td>Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.</td>
</tr>
<tr>
<td>Repository</td>
<td>postgres.exe</td>
<td>Tableau Server database, stores</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>workbook and user metadata</td>
<td></td>
<td>such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository on page 653.</td>
</tr>
<tr>
<td>Search &amp;</td>
<td>searchserver.e</td>
<td>Handles fast search, filter, retrieval, and display of content metadata</td>
<td>Yes</td>
<td>The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.</td>
</tr>
<tr>
<td>Browse</td>
<td>exe</td>
<td>on the server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VizQL</td>
<td>vizqlserver.exe</td>
<td>Loads and renders views, computes and executes queries</td>
<td>Yes</td>
<td>Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.</td>
</tr>
<tr>
<td>Server</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Archive Log Files**

You can create archives (snapshots) of log files in two different ways: from the Status page using a browser, or from a command prompt using `tabadmin` on Tableau Server. Creating a log file archive gives you a zipped snapshot of logs that you can use for troubleshooting or to send to Tableau Support for help with an issue.

**Quick Start: Generate a Snapshot of Server Logs**

Server administrators can quickly generate and download a zipped snapshot of Tableau Server logs from the Server Status page. The snapshot contains a copy of up to seven days of log information and does not affect the actual logs on the server. You can create the snapshot from any browser, and there’s no need to stop the server first.
1 **Navigate to the Snapshot Feature**

On the **Server > Status** page, scroll to the bottom of the page:

2 **Generate a Snapshot**

Click **Generate Snapshot**. If you’re running a distributed installation of Tableau Server this will collect logs from all servers in the cluster.
You do not have to stop the server before generating a snapshot.

The Generate Snapshot button is available only if no snapshot exists. If an earlier snapshot exists, you need to delete it before generating a new snapshot. Download the existing snapshot first, if you think it contains information you might need.

3 Download a Snapshot

Click Download Snapshot to copy the zipped log files to your local computer:

The Download Snapshot button is available after the snapshot is generated. The downloaded snapshot is saved to the default download location for your web browser.

4 Delete a Snapshot

Click Delete Snapshot to remove an existing snapshot from Tableau Server:
Deleting the snapshot does not delete Tableau Server log files. You are just deleting the snapshot created from those files.

**Archive Logs on Status Page (Snapshot)**

You can generate and download a snapshot (archive) of the Tableau Server log files from a web browser, without opening a command prompt. This zipped snapshot contains a copy of up to seven days of log file data from Tableau Server and any worker servers (if you have a distributed environment). The snapshot process does not change or remove either the Tableau Server log files or the log archives created with tabadmin.

**Note** To specify the amount of data you want to collect or the name of the zip file you are creating, use tabadmin to create an archive of server logs. For more information, see [Archive Logs on Command Line (tabadmin)](on page 1189).

To generate a snapshot of server log files:

1. Open the Status page:
   - Multi-site: Select **Server > Status**.
   - Single-site: Select **Status**.
2. Click **Generate Snapshot** to create a snapshot of the Tableau Server logs. The Generate Snapshot button is available only if there is no existing snapshot.

   **Note:** This option is available whether or not you have created log archives with tabadmin.
3. Select the number of days of logs you want to include. The default is **Last 7 days**, but you might want to select fewer if you want to reduce the size of the zip file. For example, if you just reproduced an issue and are collecting logs related to the issue, you may want to select **Today** to create the smallest zip file necessary.

4. Click **Download Snapshot** to download the log snapshot to your web browser's default download location. This option is available after you create a snapshot.

   Google Chrome shows you the download in the bottom of the window:

   ![Log Files](image)

5. Click the arrow and then click **Open** to unzip the snapshot or **Show in folder** to see where it was downloaded:
6. (Optional) Click Delete Snapshot to delete a log snapshot. This option is available after you create a snapshot. You need to delete the existing snapshot before you can create a new one.

For example, you might want to delete the snapshot that you created before an event that you want to investigate.

Uploading log archives for Tableau Support

If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.

Archive Logs on Command Line (tabadmin)

If you have command line access on the primary Tableau Server computer, you can archive Tableau Server log files using the tabadmin ziplogs command.

This command creates a zip file containing all of the log files and is useful when you're working with Tableau Support. If you are running a distributed installation of Tableau Server, perform this step from the primary server. Any worker logs will be included in the zip file.

You may also want to create a log file archive before you run the tabadmin cleanup command, because that command removes logs. The ziplogs command does not remove the log files, rather it creates an archive by copying them into a zip file. For more information about cleaning up Tableau Server files, see Remove Unneeded Files on page 1152.
To create a log file archive:

1. Open a command prompt as administrator and navigate to the Tableau Server bin directory. For example:
   
   cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"

2. Create the zip file by typing `tabadmin ziplogs -l -n <filename>` where `<filename>` is the name of the zipped file you want to create. Choose a unique name with no spaces. Tableau will not overwrite an existing file. For example:
   
   tabadmin ziplogs -l -n my_logs

   If you don't specify a file name, the file is named `logs.zip`.

   You can also use `-d mm/dd/yyyy` to only include logs generated since a certain date. For example:
   
   tabadmin ziplogs -l -n -d 12/14/2015

   The above command creates a zipped file named `logs.zip` that includes logs dated December 14, 2015 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`.

   You can find the zipped log file archive in the Tableau Server bin directory.

   **Uploading log archives for Tableau Support**

   If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.

   **Server Log File Locations**

   By default, Tableau Server log file archives are gathered in a zip file called `logs.zip` (you can specify a different name if you create the archive using `tabadmin`). You can copy the archive from the server to a local computer and open it there, or send it to Tableau Support. When you unzip the archive, a series of folders are created with related log files. This table explains the possible contents of each folder, along with the original location the files came from on the Tableau Server, the process that created the log files, and details about the files.

   The Tableau Server log directory is `C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs` if you installed Tableau Server on drive C, unless otherwise noted in the table below.
## Log Archive File Locations

<table>
<thead>
<tr>
<th>Files/folders in logs.zip</th>
<th>Details</th>
<th>Files</th>
<th>Generated by</th>
<th>Location on Tableau Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>build-\version.txt</td>
<td>The build version of Tableau Server.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tabsvc.yml</td>
<td>Logs related to repository encryption.</td>
<td></td>
<td></td>
<td>\config</td>
</tr>
<tr>
<td>assetkey-\encryption</td>
<td>assetkeyencryptio.log</td>
<td>tabadmin assetkeys</td>
<td>\log-sslassetkeyencryption</td>
<td></td>
</tr>
<tr>
<td>back-\grounder</td>
<td>backgrounder-.log</td>
<td>backgrounder.exe</td>
<td>\logs\backgrounder</td>
<td></td>
</tr>
<tr>
<td>backgrounder.exe</td>
<td>spawn.####.log</td>
<td>backgrounder.exe</td>
<td>\logs\backgrounder</td>
<td></td>
</tr>
<tr>
<td>backgrounder-.log</td>
<td>tomcat-#.#####-.####-.log</td>
<td>backgrounder.exe</td>
<td>\logs\backgrounder</td>
<td></td>
</tr>
<tr>
<td>backgrounder.exe</td>
<td>redis-server.exe</td>
<td>\cacheserver</td>
<td>\cacheserver</td>
<td></td>
</tr>
<tr>
<td>cacheserver</td>
<td>Logs related to the Cache Server process.</td>
<td></td>
<td>\cacheserver</td>
<td></td>
</tr>
<tr>
<td>clustercontroller.exe</td>
<td>Logs related to the Cluster Controller process.</td>
<td></td>
<td>\clustercontroller</td>
<td></td>
</tr>
<tr>
<td>clustercontroller.log</td>
<td>clustercontroller.log</td>
<td>clustercontroller.exe</td>
<td>\clustercontroller</td>
<td></td>
</tr>
<tr>
<td>clustercontroller.log</td>
<td>clustercontroller.log</td>
<td>clustercontroller.exe</td>
<td>\clustercontroller</td>
<td></td>
</tr>
<tr>
<td>config</td>
<td>Configutation files.</td>
<td>connections.yml workgroup.yml</td>
<td>Tableau Server Configuration</td>
<td>\config</td>
</tr>
</tbody>
</table>
This is a good place to start gathering information when troubleshooting. Confirm that the configuration settings are what you expect.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
<th>Logs/Files</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>data-collector</td>
<td>There will be a tdeserver log file for each day with information about data extracts and queries, and responses to VizQL server requests.</td>
<td>tdeserver_###.log, tdeserver.exe, tdeserver64.exe</td>
<td>\logs\datacollector</td>
</tr>
<tr>
<td>dataengine</td>
<td>Information about connections to Tableau Server data sources.</td>
<td>dataserver-.log, dataserver.exe, dataserver64.exe</td>
<td>\logs\dataengine</td>
</tr>
<tr>
<td>httpd</td>
<td>Apache logs. Look here for authentication.</td>
<td>access.###.log, error.log, startup.log, Apache daemon</td>
<td>\logs\httpd</td>
</tr>
</tbody>
</table>
Each request in the Apache log will have a request ID associated with it. This request ID is used throughout the server logs and you can use it to associate log entries with a request.

<table>
<thead>
<tr>
<th>Licensing</th>
<th>\logs\licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logs</td>
<td>\logs</td>
</tr>
</tbody>
</table>
| This is the location of the logs of most interest and usefulness. Look here after reviewing the configuration files. | tabadmin.log  
tabconfig.log  
tablecsvrv.log  
tabsrvlic.log |
<table>
<thead>
<tr>
<th>Process</th>
<th>Description</th>
<th>File</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>notify-tabadmin.log</td>
<td>Contains errors from tabadmin.log (the errors are also included in tabadmin.log).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tabicsrv.log and tabsrvlic.log</td>
<td>are related to licensing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>notify-tabsvc.log</td>
<td>Related to server processes.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>postgres.exe</td>
<td>PostgreSQL database logs, including files related to launching server processes.</td>
<td></td>
<td>\logs\pgsql</td>
</tr>
<tr>
<td>repository</td>
<td></td>
<td></td>
<td>\logs\repository</td>
</tr>
<tr>
<td>service</td>
<td>notify-tabsvc.log tabsvc.log</td>
<td></td>
<td>\logs\service</td>
</tr>
<tr>
<td>solr</td>
<td>Related to search indexing.</td>
<td></td>
<td>\logs\solr</td>
</tr>
<tr>
<td>svcmonitor</td>
<td></td>
<td></td>
<td>\logs\svcmonitor</td>
</tr>
<tr>
<td>tabad-minservice</td>
<td>Related to log archives created using the tabad-minservice.</td>
<td></td>
<td>\log-s\tabadminservice</td>
</tr>
<tr>
<td><strong>Generate a Snapshot of Server Log Files</strong> option.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>tabadmwrk</strong></td>
<td>Server Worker Manager process that is used for auto-discovery of worker servers in a distributed environment.</td>
<td><code>tabadmwrk.exe</code> \logs\tabadmwrk</td>
<td></td>
</tr>
<tr>
<td><strong>vizportal</strong></td>
<td>Related to showing and interacting with views. When running multiple instances of VizQL Server, the instances are distinguished by port number. notify production logs contain exceptional events.</td>
<td>\logs\vizportal</td>
<td></td>
</tr>
<tr>
<td><strong>vizqlserver</strong></td>
<td></td>
<td><code>vizqlserver.exe</code> \logs\vizqlserver</td>
<td></td>
</tr>
</tbody>
</table>

```

```
Tableau Server log files can be found in the following folders on the server:

**Tableau Service Logs**

The following log files track activities related to the web application, database, and index:

C:\ProgramData\Tableau\Tableau Server\data\tabsvc

**VizQL Logs**

These log files track activities related to displaying views, such as querying the database and generating images:

C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Logs

**Temporary Files**

Any file that starts with exe_ in the folder below is a Tableau Server file and can be deleted.

C:\ProgramData\Tableau\Tableau Server\temp
**Change Logging Levels**

By default, Tableau Server logs events at the **Info** level. You can change this if you need to gather more information (if you are working with Tableau Support, for example). As a best practice you should not increase logging levels except when troubleshooting an issue.

**Logging Levels**

The following logging levels are listed in order of increasing amount of information logged:

- off
- fatal
- error
- warn
- info (the default)
- debug
- trace

**Note:** Increasing the log level to debug or trace increases the amount of information being logged and can have a significant impact to performance. You should only set a logging level to debug when investigating a specific issue. Reproduce the issue and then reset the logging level back to info.

**Change Logging Levels**

Set logging levels for Tableau Server using one of several `tabadmin set` commands. The command you use depends on which component of Tableau Server you want to change the logging level for.

<table>
<thead>
<tr>
<th>Command</th>
<th>Location of affected logs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(path begins with <code>\ProgramData\Tableau\Tableau Server\data\tabsvc</code>)</td>
</tr>
<tr>
<td>server.log.level</td>
<td><code>\vizqlserver\Logs\*.*.txt</code></td>
</tr>
<tr>
<td>vizportal.log.level</td>
<td><code>\vizportal\*.log</code></td>
</tr>
<tr>
<td>vizqlserver.log.level</td>
<td><code>\vizqlserver\*.log</code></td>
</tr>
</tbody>
</table>

For more information, see `tabadmin set options` on page 1294.

You need to stop Tableau Server before changing the logging levels, and restart it afterward. If you are running a distributed installation of Tableau Server, set logging levels from the primary server.

To change the logging level:
1. Open a command prompt as administrator and navigate to the Tableau Server bin directory.
   If Tableau Server is installed on the C drive:
   C:\Program Files\Tableau\Tableau Server\10.0\bin
   or
   C:\Program Files (x86)\Tableau\Tableau Server\10.0\bin
2. Stop Tableau Server by typing:
   `tabadmin stop`
3. Set the logging level to by typing `tabadmin set [command][option]`
   where [command] is `server.log.level` or `vizqlserver.log.level`
   and [option] is a valid logging level.
   Examples:
   - `tabadmin set server.log.level debug`
   - `tabadmin set vizqlserver.log.level warn`
   - `tabadmin set vizportal.log.level debug`
4. Restart Tableau Server by typing:
   `tabadmin restart`

Reset Logging Levels

After you gather the information related to the issue you are investigating, reset the logging levels so there is no lingering performance impact.

Reset the logging level back to its default (info) using the appropriate command with a `-d` option.

Examples:
- `tabadmin set server.log.level -d`
- `tabadmin set vizportal.log.level -d`
- `tabadmin set vizqlserver.log.level -d`

Handle an Unlicensed Server

Tableau offers two licensing models: user-based and core-based. User-based licensing requires each active user account to be covered by a license. User-based licenses have a
defined capacity, or number of users that it allows. Each user is assigned a unique user name on the server and is required to identify himself when connecting to the server.

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

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 node has more cores than the license allows. When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tabadmin command line tool. Follow the steps below to see a list of your licenses and number of cores by machine.

1. Open a command prompt and type the following: cd C:\Program Files\Tableau\Tableau Server\10.0\bin

2. Type the following: tabadmin licenses.

Handle an Unlicensed VizQL Server Process

There are several status indicators on the Tableau Server Status page that help you understand the state of Tableau Server processes. An orange-color status box, "Unlicensed", indicates that one of the VizQL server processes is unable to retrieve the Tableau Server license information.
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.

**Cookie Restriction Error**

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or sub-domain as the browser's address bar, it is considered a first-party cookie. If a user’s browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that's different from the one shown in the browser's address bar. If a user's web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

**Troubleshoot Data Sources**

For users to work with Tableau Server data sources, up to three things need to be in place:
• **Permissions for the data source:** Anyone connecting to a data source must have the **Connect** and **View** permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be licensed to Publish and also have the **Write/Save As** and **Download/Web Save As** permissions. See [Manage Permissions on page 834](#) and [Set Permissions for a Data Source on page 851](#) for more information.

Multidimensional (cube) data sources have to be downloaded and used in Tableau Desktop, so they require **Download/Web Save As** permission. For more information about cubes in Tableau, see [Cube Data Sources on page 893](#).

• **Ability to authenticate to the database:** There are several ways you can connect to data in Tableau and control who has access to what. Basically, whichever entity is connecting to the database must be able to authenticate. The entity could be Tableau Server performing an extract refresh. It could be a Tableau Desktop user connecting to a data source that then connects to a live database. It could also be a Tableau Server user who’s accessing a view that connects to a live database. Refer to [Data Security on page 958](#) to learn more about your options.

• **Database drivers:** If the person who created and published the data source in Tableau Desktop needed to install additional database drivers, you may need to install them on Tableau Server as well. If you are running a distributed installation of Tableau Server where, for example, the data server process is running on a worker server, any required database drivers must be installed there as well as on the primary server. Other processes require drivers as well. See [Database Drivers on page 702](#) for more information.

**Data Source Error Messages**

Here are some errors that workbook authors and other users may encounter as they work with data sources and views:

**Permission to access this Tableau Server data source denied:** Connecting to a data source requires the Connect permission. See [Manage Permissions on page 834](#) and [Set Permissions for a Data Source on page 851](#) for more information.

**Data source not found:** Someone working with a view may see this error if a data source is removed from Tableau Server or if their Connect to Data page needs to be updated. To update the Connect to Data page in Tableau Desktop, click the Refresh icon:
Unable to connect to this Tableau Server data source: This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct settings.

Unable to list Tableau Server data sources: This error may occur if a user is trying to access Tableau Server data sources and there are connectivity issues between Tableau Server and Tableau Desktop.

Can’t connect with a cube data source: To use a published multidimensional (cube) data source, you must download the data source and use it in Tableau Desktop. Verify that you have the Download/Web Save As permission for the data source. For more information about cubes in Tableau, see Cube Data Sources on page 893.

Troubleshoot Subscriptions

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials**: Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.

- **Database temporarily down**: If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.

- **Background process timeout**: By default, the background process that handles subscriptions times out after 30 minutes. In the majority of cases, this is plenty of time. However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the Background Tasks for Non Extracts on page 1105 admin view to see if that's the case. To increase the timeout threshold, use the tabadmin option `subscriptions.timeout`. 
**Can't subscribe**

If you can see a view on Tableau Server and it has a subscription icon (_GRAPH) in the upper right corner, you can subscribe to it.

Two things need to be in place for you to subscribe to a view: Tableau Server needs to be correctly configured (described in *Manage Subscriptions* on page 925) and the view you're subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn't being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the *Tableau Desktop help* for details).

**No subscription icon**

It's possible to see a view on Tableau Server but be unable to subscribe to it. This happens for views with live database connections, where you're prompted for your database credentials when you first click the view. A subscription includes a view (or workbook), data, and a schedule. To deliver the data piece, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don't require credentials or have them embedded.

You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is configured for trusted authentication. See *Subscription Requirements* for more information.

**Receiving invalid or "broken" subscriptions**

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don't work, or receiving subscriptions even though they've unsubscribed from the view or workbook.

**Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")**

You may see the above error in Windows Event Viewer if subscriptions appear to be sent (according to the *Background Tasks for Extracts* on page 1103 admin view), yet subscriptions aren't arriving, and your SMTP server is using encrypted (SSL) sessions. Subscriptions are only supported for unencrypted SMTP connections. The solution is to use an unencrypted SMTP server.
Custom scripts not working after upgrade to 8.1

To support better session management, starting with version 8.1, a hash tag (#) was added to the end of view URLs. If you had custom subscriptions scripting that generated views as PDFs or PNGs you may need to update your scripts to allow for the hash tag.

For example, prior to version 8.1, view URLs use this syntax:

In versions 8.1, 8.2, or 8.3, view URLs use this syntax:
http://tableauserver/views/SuperStore/sheet1#1. To generate a PNG, add .png before the hash tag. For example:
http://tableauserver/views/SuperStore/sheet1.png#1

Custom scripts not working after upgrade to 9.0

In version 9.0, the session ID at the end of server URLs is indicated by an "iid" parameter, :iid=<n>. For example, http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1. This parameter replaces the hash tag "#<n>" used for the session ID in 8.x versions of Tableau Server.

If you use custom subscriptions scripts that generate views as PDFs or PNGs, you may need to update your scripts by removing the hash tag and number (#<n>), and inserting the ?:iid= session ID parameter before the number.

Starting in version 9.0, view URLs use this syntax:

To generate a PNG in version 9.0 and later, add .png before the session ID:
http://tableauserver/views/SuperStore/sheet1.png?:iid=2

Troubleshoot SAML

Use the following topics to troubleshoot SAML issues.

SAML and Enable Automatic Logon

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select Enable automatic logon. Enable automatic logon and SAML cannot both be used on the same server installation.

HTTP Status 500 error when configuring SAML

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:

org.opensaml.saml2.metadata.provider.MetadataProviderException:
User specified binding is not supported by the Identity Provider using profile urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.
- The IdP is configured to use SP-initiated authentication. (IdP-initiated authentication is not supported.)
- The IdP is configured to use HTTP-POST requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in SAML Requirements on page 1014.

Signing In from the Command Line

SAML is not used for authentication when you sign in to Tableau Server using the command linetools tabcmd on page 1315 or the Tableau Data Extract command line utility (provided with Tableau Desktop), even if Tableau Server is configured to use SAML. These tools require the authentication configured when Tableau Server was originally installed (either local authentication or AD).

Login Failed

Login can fail with the following message:

Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.

This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith’s username is stored in the IdP as jsmith it must be stored in Tableau Server as jsmith.

SAML Error Log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a
snapshot of log files and use them to troubleshoot problems. For more information, see Archive Log Files on page 1184.

**Note:** To log SAML-related events, `vizportal.log.level` must be set to `debug`. For more information, see Change Logging Levels on page 1197.

Check for SAML errors in the following files in the unzipped log file snapshot:

`\vizportal\vizportal-<n>.log`

In Tableau Server 9.0 and later, the application process (vizportal.exe) handles authentication, so SAML responses are logged by that process.

**Trailing Slash**

On the SAML tab, confirm that the Tableau Server return URL does not end with a trailing slash (correct: `http://tableau_server`; incorrect: `http://tableau_server/`):

![Tableau Server Configuration](image)

**Confirm Connectivity**

Confirm that the Tableau Server you are configuring has either a routeable IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running `telnet` on Tableau Server and attempting to connect with the SAML IdP. For example: `C:\telnet 12.360.325.10 80`

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.
Troubleshooting Mutual SSL Authentication

This topic describes possible mutual (two-way) SSL authentication issues and their causes, the messages that users might see, and possible mitigation for the issues.

- The client is missing a certificate
- The client doesn't support mutual SSL authentication
- Client certificates are not published to Active Directory
- Users unexpectedly see a sign-in dialog box that displays an error message
- The user name in the UPN or CN fields is missing or invalid
- The user is signed in using unexpected user name (LDAP mapping)
- The user is signed in as incorrect user (UPN or CN mapping)

For more information about mutual SSL authentication and LDAP, UPN, and CN user mapping, see the following topics:

- Quick Start: Mutual (Two-Way) SSL Authentication on page 970
- Mapping a Client Certificate to a User During Mutual Authentication on page 978

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

The client is missing a certificate.

If the client has no client certificate, the user sees this message during authentication:

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

To resolve the issue, the user should contact the system administrator to generate a certificate for the client computer.

Invalid user name or password

The client doesn't support mutual SSL authentication.

Versions of Tableau Desktop older than version 9.1 do not support mutual SSL authentication. If an older version of Tableau Desktop is used to connect to Tableau Server that is configured for mutual SSL authentication, the following can occur:

- If Tableau Server is configured to use fallback authentication, the client displays a sign-in dialog box and the user can enter a user name and password.
- If the server is not configured to use fallback authentication, the user sees the following
message and cannot connect to the server:

Invalid user name or password

For more information about fallback authentication, see Quick Start: Mutual (Two-Way) SSL Authentication on page 970.

**We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.**

Client certificates are not published to Active Directory.

If Tableau Server is configured to use Active Directory for authentication, and if user mapping is set to LDAP, Tableau Server sends the client certificate to Active Directory for authentication. However, if client certificates have not been published to Active Directory, authentication fails and the user sees the following message:

**We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.**

To resolve this issue, the system administrator should make sure that client certificates are published to Active Directory. Alternatively, the server should be configured to use a different user mapping (UPN or CN), and the system administrator should be sure that client certificates contain user names in the UPN or CN fields.

**Users unexpectedly see a sign-in dialog box that displays an error message**

If Tableau Server is configured to use mutual SSL authentication and certificates are available for use with users' computers, a user should not see a sign-in dialog box, because Tableau Server uses the certificate to authenticate the user. However, if the server does not recognize the user name in the certificate, the user sees a sign-in dialog box with an error message that indicates why the certificate was not used. This can occur when all of the following conditions are true:

- Fallback authentication is enabled.
- If the server is using UPN or CN mapping, the user name in the certificate's UPN or CN field is not recognized. If the server is using LDAP mapping, the certificate is not mapped to the user in Active Directory.

To resolve this issue, the system administrator should do the following, depending on how user mapping is configured on Tableau Server:

- LDAP mapping: Make sure that the certificate is linked to the user, that the certificate is available for use with the user's computer, and that the user is configured as a Tableau Server user.
- UPN or CN mapping: Make sure that the certificate is available for the user's computer,
that the user name is in the certificate's UPN or CN field, and that the user name matches
the user name on Tableau Server (including domain).

**We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.**

**Certificate does not contain a valid Tableau Server user name.**

The user name in the UPN or CN fields is missing or invalid

When Tableau Server is configured to use UPN or CN mapping, the server reads the user’s
name from the UPN or CN field of the certificate and then looks up the user name in Active
Directory or in the local repository on Tableau Server. (The specific field that the server reads
depends on which mapping—UPN or CN—the server is configured to use.) If the field that is
supposed to contain the user name has nothing in it, the user sees the following message:

**We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.**

If a client certificate contains a user name but Active Directory and Tableau Server don’t
recognize the user name, the user sees the following message:

**Certificate does not contain a valid Tableau Server user name.**

This can occur when all of the following conditions are true:

- Tableau Server is configured to use UPN or CN mapping.
- Fallback authentication is not enabled.
- The client certificate has no user name in the UPN or CN field, or the user name in the
  UPN or CN field does not match a user name in Active Directory or on Tableau Server.

To resolve this issue, the system administrator should make sure that the user’s certificate has
the correct user name in the UPN or CN fields of the certificate.

**The user is signed in using an unexpected user name (LDAP mapping)**

When the server is configured to use Active Directory authentication and LDAP mapping, the
certificate is linked to a user in Active Directory. If the certificate contains a user name in the
UPN or CN field, that user name is ignored.

If the intention is that the user should be signed in with the user name in the UPN or CN fields,
the server should be configured to use UPN or CN mapping.

**The user is signed in as the incorrect user (UPN or CN mapping)**

Under some circumstances, the user name in a UPN or CN field in the client certificate can be
ambiguous. The result is that a user is signed in to the incorrect identity.
For more information about the conditions under which this issue can occur, see Ambiguous user names in multi-domain organizations in the topic Mapping a Client Certificate to a User During Mutual Authentication on page 978.

Handle Extract Refresh Alerts

When Tableau Server cannot complete a scheduled refresh, an alert appears to indicate that the refresh has failed. If a scheduled refresh fails five consecutive times, Tableau Server suspends the refresh. When a refresh is suspended, Tableau Server does not try to run it again until someone takes an action that attempts to correct the cause of the failure.

Note: The number of consecutive failures for a refresh is set to five by default, but can be changed by a Tableau Server administrator, using the `tabadmin set backgrounder.failure_threshold_for_run_prevention` command. For more information, see `tabadmin set options` on page 1294.

You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn’t be refreshed
- The author of a workbook that connects to a data source that couldn’t be refreshed

When you open the Alerts menu you can see more information about the refresh failure(s):
When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the **Data source** (for example, **Data source: sales_data**), it means that the data source is a **Tableau Server data source**. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon:

<table>
<thead>
<tr>
<th>Data</th>
<th>Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>🇺🇸 Population</td>
<td>🇺🇸 Sales by Region</td>
</tr>
</tbody>
</table>

**Resolving Extract Refresh Problems**

To resolve refresh issues, you can take any of these actions, based on the cause indicated in the alert:

- **Errors related to access token validation or user credentials**
  
  You can resolve some extract refresh problems by clicking the **Connection Details** in the alert. Select the check box next to the problematic data source, click **Actions > Edit Connection**, and then enter the missing information. Click **Save** when you're done. After you update the connection information, Tableau Server restarts the refresh schedule.

  If you originally embedded the credentials or other data connection information when you published the workbook or data source from Tableau Desktop, you can also republish the workbook or data source. As part of the publishing process, you can choose to set a new refresh schedule. If you don't choose a new schedule, Tableau Server restarts the existing schedule.

- **Errors that indicate the database was unreachable**
  
  Confirm that the database is online and that you can sign in to access the data. You can use the **Try again** link in the alert to restart the refresh schedule.

If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.

**Tip:** Administrators can edit data connections at any time on the **Data Connections** page, accessible from each site by clicking the **Content** tab and Data Connections.
Troubleshoot Server Processes

When Tableau Server is functioning properly, processes will show as Active, Busy or Passive (Repository). If there is additional information, a message appears below the status icon:

![Ready for removal](image)

Possible status indicators are:

Use this table to help troubleshoot issues with your Tableau Server installation.

<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td><img src="image" alt="X" /></td>
<td>&quot;Node degraded&quot;</td>
<td>• Repository on the node is stopped.</td>
<td>No action is necessary unless the cluster controller is regularly down or is down for an extended period of time.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Node cannot respond to fail-over elsewhere in the cluster.</td>
<td>If that occurs, take the following actions, in order, until the problem is resolved:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• If Tableau Server is configured for high availability and this is the active repository, fail-over to the second repository occurs.</td>
<td>1. Check disk space. If disk space is limited, save the log files (use <code>tabadmin ziplogs</code>) in case you need them for Support, then remove unnecessary files (<code>tabadmin cleanup</code>).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• No status available for repository or file store on this node.</td>
<td>2. In Windows Task Manager, stop the</td>
</tr>
<tr>
<td>Process</td>
<td>Status (Icon)</td>
<td>Message</td>
<td>Implications</td>
<td>Actions</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>---------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cluster-controller.exe process tree and let it restart automatically.</td>
<td>none</td>
<td>- No extracts were being synchronized when the page was loaded. (It is possible that the recurring &quot;catch-all&quot; job is running and synchronizing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3. Restart Tableau Server.

4. Clean up the coordination service (ZooKeeper) files: Stop the cluster *(tabadmin stop)*, clean up files *(tabadmin cleanup --reset-coordination)*, and then start the cluster *(tabadmin start)*.

5. If Cluster Controller continues to show as down, save the log files *(tabadmin zip logs)* and contact Support.
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
</table>
|         |              | "Synchronizing" | - Extracts were being synchronized across file store nodes when the page was loaded.  
- Initial status following installation (both single-node and multi-node). Should disappear within 15 or 20 minutes. | None. |
|         |              | "Data Extracts unavailable" | - Single-node installation: existing extracts may be available but publish/refresh will fail.  
- Multi-node installation: extract synchronization will fail for this node. | No action is necessary unless the file store is regularly down or is down for an extended period of time.  
If that occurs, take the following actions, in order, until the problem is resolved:  
1. Check disk space.  
   If disk space is limited, save the log files (tabadmin ziplogs) in case you need them for Support, and then remove unne- |
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;Decommissioning&quot;</td>
<td></td>
<td>• File store is in read-only mode.</td>
<td>Wait until the status message changes to &quot;Ready for removal&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Any unique files on this node are being rep-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>cessary files (<code>tabadmin cleanup</code>).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2. Stop the filestore.exe process using Windows Task Manager and let it restart automatically.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3. Restart Tableau Server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (<code>tabadmin stop</code>), clean up files (<code>tabadmin cleanup --reset-coordination</code>), and then start the cluster (<code>tabadmin start</code>).</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5. If the file store continues to be down, save the log files (<code>tabadmin zip logs</code>) and contact Support.</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Status (Icon)</td>
<td>Message</td>
<td>Implications</td>
<td>Actions</td>
</tr>
<tr>
<td>---------</td>
<td>---------------</td>
<td>---------</td>
<td>--------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
|         |               | "Ready for removal" | · File store is in read-only mode.  
       |               |         | · Ready for user to stop cluster and remove data engine/file store or remove entire node. | Stop Tableau Server (tabadmin stop) and then run the Configuration utility to remove Data Engine and File Store or the entire node. |
|         |               | "Decommission failed" | · File store is in read-only mode.  
       |               |         | · At least one unique file failed to replicate to another file store node. | Take the following actions in order until the problem is resolved:  
1. Run the tabadmin decommission command again.  
2. Check disk space on other file store nodes. Decommissioning will fail if another file store node does not have enough space to store all the extracts.  
3. Check the tabadmin.log file on the primary node and workers for errors.  
4. Stop Tableau Server (tabad- |
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&quot;Setting up&quot;</td>
<td>Passive repository is being synchronized with active repository.</td>
<td>Wait until the repository status message changes to &quot;Passive&quot;. If this message does not appear, or if it is taking a long time:</td>
</tr>
<tr>
<td>Repository</td>
<td></td>
<td></td>
<td>Repository is not ready to handle fail-over.</td>
<td>1. Check disk space and free space if possible.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Repository may have gotten more than two minutes behind active repository and is being setup again (this is min stop) and then try running the <code>tabadmin decommission</code> command again.</td>
<td>2. Check cluster controller logs for errors.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3. Restart node.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4. Put the file store node back into read/write mode (tabadmin recommission), collect logs, and then contact Support.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5. With Support: copy and merge extracts directory from this file store node to the same directory on another file store node.</td>
</tr>
<tr>
<td>Process Status (Icon)</td>
<td>Message</td>
<td>Implications</td>
<td>Actions</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
<td>--------------</td>
<td>---------</td>
<td></td>
</tr>
</tbody>
</table>
|                       | faster than waiting for a sync).  
- Failover occurred and this former active repository is rejoining the cluster. | | |
| "Synchronizing" | Repository is synchronizing, for example after a failover. | None. | |
| none | If the installation is configured for high availability, failover of the repository occurred.  
- Processes are restarting with updated database connection configurations after failover.  
- If another active repository is not available, Tableau Server is down. | Take these actions in order until the problem is resolved:  
1. Wait several minutes for cluster controller to attempt to restart.  
2. Restart Tableau Server (tabadmin restart).  
3. Check disk space to make sure there is free space. Collect logs (tabadmin ziplogs) in case you need them for Support, and then cleanup files (tabadmin cleanup).  
4. Restart Tableau | |
<table>
<thead>
<tr>
<th>Process</th>
<th>Status (Icon)</th>
<th>Message</th>
<th>Implications</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>✓</td>
<td>none</td>
<td>• Working as intended.</td>
<td>None.</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>✗</td>
<td>none</td>
<td></td>
<td>For information about unlicensed status for a VizQL Server process, see Handle an Unlicensed VizQL Server Process on page 1199.</td>
</tr>
<tr>
<td></td>
<td>!</td>
<td>none</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Troubleshoot Inconsistent Process Status**

**Disclaimer:** This topic includes information about a third-party product. Inclusion of this information is not an endorsement of the product, but is provided as a convenience for our customers. Please note that while we make every effort to keep references to third-party content accurate and up to date, the information we provide here might change without notice as the third-party product changes.
Follow the suggestions to resolve issues with Tableau Server process status. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes on page 1212.

**Issue**

When Tableau Server is configured with multiple network cards, the Status page may report inconsistent or inaccurate process status. These potential inconsistencies and inaccuracies can result in other functionality such as alerting or notifications unreliability.

**Environment**

- Tableau Server 9.0 and higher
- Tableau Server computers with multiple network cards (NICs)

**Resolution**

To address this problem, you can disable the second NIC, or use the following procedure to assign metrics to each NIC on the computer.

**Note:** Updating DNS or using local routing in the etc\hosts file to refer to the preferred IP address will not resolve this issue.

Assign metrics for each network interface

A metric is a way to indicate the “cost” of using a network interface. The higher the metric, the more expensive it is to use. By default in Windows, Automatic Metric is enabled, but you can manually assign metrics to each network interface to indicate which network interface is preferred. The lower a metric value the more preferred the interface is.

To manually configure metrics for a network interface:

1. In Control Panel, click Network and Internet.
2. Click Network and Sharing Center.
3. Click Change adapter settings.
4. Right-click on a network interface and click Properties.
5. Select Internet Protocol Version 4(TCP/IPv4) and click Properties.
7. On the IP Settings tab, clear Automatic metric and enter the metric that you want in the Interface metric box.

The metric indicates the cost of using the interface, so give your preferred interface a lower value than the other interface(s) on the computer.
Repeat the process for any other interfaces, giving them metrics based on their preference. The interface that Tableau Server uses should be the preferred interface and have the lowest value metric. For example, give the preferred network interface a metric of 5 and the secondary interface a value of 10.

Disclaimer: Although we make every effort to ensure links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

For more information about Windows and using the metric feature for IP routes, see the following Microsoft documentation:

- https://support.microsoft.com/en-us/kb/299540

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes on page 1212.

General Troubleshooting Steps

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

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

2. Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the tabadmin restart command. This will stop all the processes associated with Tableau Server and then restart them.

3. Clean up files associated with the Coordination Service (ZooKeeper). To clean up Coordination Service files, use the tabadmin cleanup --reset-coordination command.

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."
If you see this message after starting, verify that Tableau Server is running as expected by using a `tabadmin status -v` command.

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

**Tableau Server doesn’t start**

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

**Installing Tableau Server**

**Install fails due to hardware requirements**

Starting with version 9.0, Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to both primary server computers and worker computers. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server on page 674.

**Install or upgrade generates an error when PostgreSQL ODBC driver does not install correctly**

In certain circumstances (when a system reboot is pending, or another program is being installed or updated, the Tableau Server PostgreSQL ODBC driver does not install correctly. When this happens, this message displays:

PostgreSQL ODBC driver (64-bit) version 09.03.0400 did not install properly.

**Note:** The version may be different, depending on what version of Tableau Server you are installing.

If this occurs, follow these steps to correct the issue:

1. Check to see if the driver shows as installed in Control Panel.
2. If the driver is not installed, download it from the Tableau Drivers page and install it.
3. If the driver is installed, uninstall it from Control Panel, restart the computer, download the driver, and install it again.
### Upgrading Tableau Server

#### Extract migration is slow

Tableau Server 9.0 introduced a more reliable storage mechanism for data extracts called the File Store. Upgrading from a previous version requires migration of the extracts. This can take a long time (up to several hours) if you have a large number of extracts or extracts that have a lot of data. During migration a message displays:

_Migrating extracts to File Store
This process may take up to several hours._

If the migration progress appears to be stalled or stuck, you can verify that migration is continuing by watching the `tabadmin.log`. An entry is written to this log for each extract that is migrated. You can periodically copy the log and open your copy in a text editor like Notepad to verify that entries are being written to it.

#### Upgrading fails due to lack of disk space

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

To free up disk space:

1. Zip and save logs using the `tabadmin ziplogs` command.
   
   After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tabadmin cleanup` command. For more information, see Remove Unneeded Files on page 1152

### Reindexing Tableau Server Search & Browse

#### Problems that can be solved by reindexing Search & Browse

Symptoms of an index that needs to be rebuilt include:

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

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

When Tableau Server and Tableau Desktop instances are properly configured, Tableau Desktop license usage information is available in two administrative views, Desktop License Usage on page 1111 and Desktop License Expiration on page 1113. If you can't see these views, or if there is no data in them, you can use this topic to help troubleshoot.

For detailed information on configuring Desktop License Reporting, see Configure Tableau Desktop License Reporting on page 1081.

Administrative views aren't available

The Desktop License Reporting administrative views are available only to Tableau Server administrators. If you do not see links to the Desktop License Usage and Desktop License Expiration views on the Server Status page (select Manage All Sites from the sites menu), verify the following:

- You are signed in as a Tableau Server administrator.
- You are running a version of Tableau Server 10.0 or later, and users are running Tableau Desktop version 10.0 or later. Desktop License Reporting is available beginning with version 10.0 of Tableau Server and Tableau Desktop.
- Tableau Server has Desktop License Reporting enabled. (The feature is disabled by default.) For more information, see Step 1: Enable Desktop License Reporting on Tableau Server on page 1081.

Administrative views don't include expected content

If you aren't seeing the data you expect, it could be for one of the following reasons.

Tableau Desktop was configured less than eight hours ago

You might not see usage data in the administrative views if it has been less than eight hours since instances of Tableau Desktop were configured for reporting. After an initial report to server, when running Tableau Desktop reports every eight hours by default. You can change this default to a more frequent interval for troubleshooting. For more information, see Changing the default reporting interval on page 1226.

The following events force a report from a properly configured Tableau Desktop instance to Tableau Server:

- Activating Tableau Desktop version 10.0 or later.
- Deactivating Tableau Desktop version 10.0 or later.
Tableau Desktop has not been restarted since Desktop License Reporting was configured
If Tableau Desktop was already running when Desktop License Reporting was configured with a server address to report to, Tableau Desktop must be restarted.

Tableau Desktop has not been opened since being configured for license reporting
If Tableau Desktop has not been opened since being configured for reporting, you might not see usage data in the administrative views.

Tableau Desktop is incorrectly configured for license reporting
Verify that instances of Tableau Desktop are configured correctly with the address of the Tableau Server to report to.

Use log files on the Tableau Desktop computer to help determine if the instance is configured correctly. Find the log file `%My Documents%\My Tableau Repository\Logs\log.txt`. Search for "licUsageReport" to find entries related to Desktop License Reporting.

The following table lists log messages that can help identify issues with license reporting.

<table>
<thead>
<tr>
<th>Log message</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;licUsageReport: Response code from server: &lt;server&gt; is: 200&quot;</td>
<td>Desktop was successful reporting to the configured server.</td>
</tr>
<tr>
<td>&quot;Internet communication error: Couldn't connect to server (server_name).&quot;</td>
<td>• Tableau Desktop is configured for the wrong protocol. For example, server is configured for HTTPS and Tableau Desktop is configured to report using HTTP.</td>
</tr>
<tr>
<td></td>
<td>• Tableau Desktop is reporting to a non-existent server.</td>
</tr>
<tr>
<td></td>
<td>• Tableau Desktop is reporting to an instance of Tableau Server that is not running.</td>
</tr>
<tr>
<td>&quot;licUsageReport: License reporting server config does not exist&quot;</td>
<td>• The registry key is not set or is in the wrong place in the registry (Windows) or .plist file (Mac).</td>
</tr>
<tr>
<td></td>
<td>• On Mac computers, this message can be logged when</td>
</tr>
<tr>
<td>Log message</td>
<td>Details</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>a .plist file has been created in the wrong location. For example, the .plist file was created in or copied to ~\Library\Preferences instead of \Library\Preferences.</td>
<td></td>
</tr>
<tr>
<td>On Mac computers, this message can be logged when a .plist file has been created but the file is then updated with a new or changed server string and the Mac computer is not restarted.</td>
<td></td>
</tr>
<tr>
<td>&quot;licUsageReport: License reporting server config does not exist.&quot;&quot;</td>
<td>The registry key is not set or is in the wrong place in the registry (Windows) or .plist file (Mac). On Mac computers, this message can be logged when a .plist file has been created but the file is then updated with a new or changed server string and the Mac computer is not restarted.</td>
</tr>
</tbody>
</table>

**Changing the default reporting interval**

By default, when configured for Desktop License Reporting, Tableau Desktop reports to the configured server or servers every eight hours. You can modify the registry or .plist file on Tableau Desktop computers to change this interval. This is especially useful for troubleshooting.

**Windows**

To change the frequency that Tableau Desktop reports to the server from a Windows computer, edit the registry to add a string value to the ReportingServer key:

- **Path:** HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ReportingServer
- **Name:** add a string value named scheduleReportInterval.
- **Data:** The amount of time, in seconds, between reports sent by Tableau Desktop to the
server. By default this is 8 hours and if there is no entry the default is used. Increase this for troubleshooting if necessary, but keep in mind that the more frequent the interval, the more network traffic generated.

For example, the following image shows a registry configured so Tableau Desktop reports to the configured Tableau Servers every hour (3600 seconds):

[Image of registry editor showing scheduleReportInterval set to 3600 seconds]

**Macintosh**

To change the frequency that Tableau Desktop reports to the server from a Macintosh computer, edit the `com.tableau.ReportingServer.plist` file in `/Library/Preferences` and add a `scheduleReportInterval` key. Set this to the length of time, in seconds, between reports from Tableau Desktop to the configured Tableau Server. The following example shows the contents of a `.plist` file that’s configured to send information every hour (3600 seconds) to two servers, `https://mytableau02:8010` and `http://mytableau`.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Server</key>
    <key>scheduleReportInterval</key>
    <string>3600</string>
  </dict>
</plist>
```

For details on how to configure Tableau Desktop, see [Configure Tableau Desktop License Reporting](#) on page 1081.

**Troubleshoot 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.

This article describes some common issues you may encounter after enabling impersonation
and how to troubleshoot them.

**Tableau Server view fails to load**

There are several potential causes for a Tableau Server view failing to load:

- Account performing impersonation doesn't have IMPERSONATE permission for the data-
  base user account of the person who's trying to access the view. Depending on how
  you've configured impersonation, the account doing the impersonation is either the
  server Run As User account or the account whose credentials are being embedded in
  the view. See "Granting IMPERSONATE Permission for a User" section, below.
- User credentials don't match. The credentials of each Tableau Server user's account
  must match their credentials in the SQL Server database. In other words, if Jane Smith's
  Tableau Server user account has a username of MYCO\jsmith, her username on the
  SQL Server database must also be MYCO\jsmith.
- User authentication type doesn't match. If you've configured Tableau Server to use Active
  Directory to authenticate users, the SQL Server database must also be using Active
  Directory (in SQL Server 2008, it's called **Windows Authentication**). Alternatively, if
  Tableau Server is using Local Authentication to authenticate its users, SQL Server must
  also be using "local" authentication for its users. In SQL Server this is called **SQL Server
  Authentication**.

**Tableau Server view shows too much or incorrect data**

If a published view shows too much or incorrect data, it could be for one of the following
reasons:

- Impersonation is not enabled. The workbook author did not enable impersonation when
  he or she published the view. See **Impersonate with a Run As User Account** on
  page 1044.
- Live database connection/impersonation is not being used. The workbook author cre-
  ated a data extract instead of creating a live connection to a SQL Server data source and
  enabling impersonation. See **Impersonate with Embedded SQL Credentials** on
  page 1046.
- The SQL Server database view is incorrect. If you have configured impersonation cor-
  rectly but still have a view that is showing too much data or the wrong data, it could be
  because your SQL Server database view is not correctly configured. See "SQL Server
  Prerequisites" section, below.
- The SQL Server data security lookup table has incorrect mappings. This could also be
  the cause of a view displaying too much or incorrect data. See "SQL Server Pre-
  requisites" section, below.
Tableau Server view prompts for credentials

If the person attempting to access a view is prompted for credentials, the workbook author did not enable Impersonate via embedded credentials or Impersonate via server Run As User account when they published the workbook.

Publish preview shows different data than seen on desktop

When a workbook author publishes a view, they are prompted to log into Tableau Server. After successfully publishing a view, Tableau will show a preview of that view as it appears on Tableau Server. If that author's current Windows login is different from their Tableau Server user credentials, the view they see in Desktop while they're authoring may look different than the preview of the view they see after they publish. The preview reflects the permissions of the account they used to log into Tableau Server.

Workbook author doesn't see the "Impersonate via embedded password" option

Workbook authors who use impersonation via embedded credentials select the following option when they publish:

![Authentication dialog box](image)

If an author does not see the above option in Tableau Desktop's Authentication dialog box, the Tableau Server administrator needs to enable Embedded Credentials (All Sites > Settings):
SQL server prerequisites

The power of Tableau's impersonation feature is that it leverages the data security model you've already created in SQL Server. This topic won't attempt to describe how to set that up, but on a very high level, the minimum you need to use Tableau's impersonation feature is a data security table in SQL Server and a view for enforcing data security. The following example will get you started. For specific guidance on how to use and configure SQL Server to secure your data, see your Microsoft SQL Server documentation.

First, assume you have the following data security table (for example, [UserAccess]) in your BigSales database:

<table>
<thead>
<tr>
<th>uaID</th>
<th>uaMarket</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYCO\jsmith</td>
<td>West</td>
</tr>
<tr>
<td>MYCO\hwilson</td>
<td>East</td>
</tr>
</tbody>
</table>

The following SQL Server command would create a view that enforces data security so that jsmith only sees sales data from states in the West and hwilson only sees data from states in the eastern sales territory:

CREATE VIEW dbo.BigSales AS
SELECT * FROM dbo.Sales
JOIN dbo.UserAccess ua
ON Market = ua.uaMarket
WHERE ua.uaID = SUSER_SNAME();

Granting IMPERSONATE permission for a user

The following example illustrates how to create an account in SQL Server then grant it IMPERSONATE permission for another account. In the example, Tableau Server is running under an Active Directory account named TableauServer. The domain is MYCO. The following command creates a"matching" account in SQL Server:

CREATE USER [MYCO\TableauServer] FOR LOGIN [MYCO\TableauServer]
WITH DEFAULT_SCHEMA=[dbo];
The next command grants MYCO\Tableau Server IMPERSONATE permission for Jane Smith (MYCO\jsmith). Jane Smith is a Tableau Server user and has an individual account in the SQL Server database.

GRANT IMPERSONATE ON USER::[MYCO\jsmith] to [TSI\TableauServer];

The GRANT must be performed for each database user account to be impersonated.

Troubleshoot Disk Space Usage on Tableau Server Nodes

When available disk space on a Tableau Server primary or worker node is low, performance can be degraded. If free space falls too low, Tableau Server may begin to perform erratically. To monitor free disk space, configure Tableau Server to save disk usage information (this is on by default) and, if desired, enable alerts about low disk space. For more information, see Quick Start: Disk Space Alerts on page 1091.

**Note:** Disk space monitoring measures free disk space on each server node. Available space may be impacted by programs or processes that are not a part of Tableau Server.

If you find that your Tableau Server installation is running into free disk space limitations, you should take steps to make more space available. This topic suggests some ways you can do that.

Viewing Disk Usage on Tableau Server Nodes

When disk space usage monitoring is enabled (this is the default), server administrators can use the Server Disk Space on page 1110 administrative view to see current disk space usage, and one month of usage data on your Tableau Server nodes. Use this view to help you determine whether one of your server nodes is experiencing a jump in space usage, or if space usage has increased over time.

Cleaning Up Tableau Server-Related Files

To minimize server space used by Tableau Server, you can clean up unnecessary files.

Use the `tabadmin cleanup` command to remove log files, temporary files, and unneeded entries in the PostgreSQL database. If you want to save the logs before you clean them up, you can make an archive. For more information, see Remove Unneeded Files on page 1152.

Once you have cleaned up log files and temporary files, you may want to use the administrative views to determine which workbooks and data sources are taking up the most space on your server, and whether any of these is not being used. For more information, see Administrative Views on page 1097.

Identifying and Cleaning Up Other Files

There are a number of tools, like WinDirStat, you can use for viewing disk usage and doing cleanup.
Troubleshoot Run As User

As discussed in the topic, Run As User on page 577, Tableau Server requires administrative-like access to the machine on which it is installed. Therefore, when you update the Run As User in Tableau Server Configuration, a background process will configure permissions on the Tableau computer for that account. However, in some complex deployment scenarios you may need to verify or manually configure the Run As User permissions on the local Tableau Server computer. Use this section to verify how permissions are configured on the machines running Tableau Server in your deployment. This section also includes procedures that describe how to set permissions and configure security policies for the Run As User.

Required Run As User Account Settings

The Run As User account needs permissions that allows it to modify files and registry settings. In addition, because the Run As User is used as the security context for the Tableau Server Application Manager service (tabsv), the account must also be given rights to log on as a service.

These permissions are set automatically when you update the Run As User account in Tableau Server Configuration as described in the topic, Create and Update the Run As User Account on page 578.

If you have recently changed Run As User or are getting permission errors, use this section to confirm that Tableau Server meets the permission requirements that are detailed here. If you’re running a distributed installation, all Run As User permission configurations must be the same across the primary server and all worker nodes.

Note: Do not hide the files created by the Tableau Server installer.

Verify Folder Permissions

The account the Tableau Server service runs under needs permission to modify files in the path where Tableau Server is installed.

For example, if Tableau Server is installed in the default location on the system drive (typically, the C:\drive), the account needs modify permissions for C:\Program Files\Tableau\Tableau Server and C:\ProgramData\Tableau\Tableau Server, including all folders and files in all subfolders. If you have installed Tableau Server on a drive other than the system drive or if you have installed it to a non-default location, then all Tableau files and folders are created in the location you specify. The \ProgramData\Tableau\Tableau Server\ folder is not created. Instead, a data folder is created at\Tableau\Tableau Server\data\.
**Important:** The Modify permission in Windows requires the following permissions for full functionality: Read & execute; List folder contents; Read; Write. When editing permissions on a folder’s Security tab, Windows will automatically select the additional permissions to enable full Modify functionality. This topic refers to the full Modify functionality where all of the sub-permissions are included.

When you update the Run As User in Tableau Server Configuration, a background process (tabadmin) will configure the folder permissions on the Tableau computer for the account you specify. In this case, where you are installing on the system drive into the default folder (C:\Program Files\Tableau), the configuration of folder permissions will be handled by the tabadmin process when you update the Run As User account in Tableau Server Configuration. You do not need to verify or change any folder permissions for this scenario.

**Installing on non-system drive or in a different folder**

If you have installed Tableau Server on a drive other than the system drive, then you will need to configure the system drive to allow the Run As User additional permissions. The system drive is the drive where Windows is installed. For example, if Windows is installed on the C:/ drive, then C:/ is your system drive. If you install Tableau Server on any other drive (D:/, E:/, etc), then you will need to configure permissions to allow the Run As User to modify the system drive. See the procedure below for information about how to set Modify permission on the C:/ drive.

If you have installed Tableau into a folder other than the default path (\Program Files\Tableau) and you've updated the Run As User, then you should verify that the root Tableau folder and all subfolders have been configured with Modify permissions for the Run As User account. If they haven't, then use the procedure below to set the Modify permission on the \Tableau install folder and all subfolders.

This procedure describes how to set Modify permissions for the Run As User on a given folder in Windows. Use this procedure to verify that permissions on the Tableau folder and subfolders are set to Modify. If you have installed Tableau onto a drive other than the system drive (typically C:\), then you must set Modify permissions for the Run As User on the root of the system drive.

1. On the computer hosting Tableau Server (and on Tableau Worker nodes, if distributed), use Windows Explorer to right-click the drive, for example **Local Disk (C:)**, and select **Properties**.
2. In the Local Disk Properties Window, select the **Security** tab.
3. Click **Edit**, then **Add**.
4. In the Select Users, Computers, Service Accounts, or Groups dialog box, type the `<domain><username>` for the Tableau Server Run As User account.
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 **Modify** permissions. Selecting **Modify** automatically selects **Read & execute**, **List folder contents**, **Read** and **Write**.

7. Click **Advanced**:

![Tableau Properties](image)

8. In the Advanced Security Settings for Tableau window, click **Change Permissions**.

9. 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:
10. Click **OK** to apply changes to all subfolders and files - this may take a few minutes. It's typical to receive several error messages from Windows when you apply these changes. There's no need to cancel the process; instead, click **Continue**.

11. Click **OK** to confirm changes, then click **OK** in the Tableau Properties dialog box.

12. Click **OK** to exit.

**Verify Registry Permissions**

The account the Tableau Server service runs under needs permission to modify the registry on the local machine.

When you update the Run As User in Tableau Server Configuration, a background process (tabadmin) will configure the registry permissions on the Tableau computer for the account you specify. It's unlikely that you will need to apply these permissions manually.

Verify that the Run As User has been granted permissions to the following registry branches. If account that you have specified as the Run As User is a member of the local administrative group or a member of the Domain Admins security group, then the account will not be displayed on the Permissions page.

- HKEY_CURRENT_USER\Software\Tableau
- HKEY_LOCAL_MACHINE\Software\Tableau

**Permissions**

Tabadmin will grant Read permission and the following Special permissions to these branches:
To view or edit permissions on registry directories:

1. Open the Registry Editor by entering `regedit` in Windows Run, and then clicking **OK**.
2. In Registry Editor, navigate to the directory where you want to view or edit permissions. Right-click the directory, and then click **Permissions...**
3. In Permissions, on the Security tab, select the Run As User account, and then click **Advanced**.
   If you are adding your Run As User account, then click **Add** and follow the Windows process for adding a user account to the Security tab. After you have added the account, then select the Run As User account, and then click **Advanced**
4. In Advanced Security Settings, on the Permissions tab, select the Run As User account, and then click **Edit**.
5. On the Permission Entry, under Basic permissions, verify that **Read** and **Special permissions** are selected. Verify that **Only apply these permissions to objects and/or containers within this container** is not selected.
6. To view or edit Special permissions, click **Show advanced permissions**.
7. Under Advanced permissions, verify that the permissions enumerated at the beginning of this topic are selected. Verify that **Only apply these permissions to objects and/or containers within this container** is not selected.
8. If you have set new permissions, then click **OK** through the multiple windows to finish. If you have viewed permissions and not edited anything, then click **Cancel** to close all windows.

**Verify the Local Security Policy**

After you specify a Run As User account in Tableau Server Configuration (as described in the topic, **Create and Update the Run As User Account** on page 578), a background process (tabadmin) will update the local security policy on the computer running Tableau Server. Tabadmin will update the local security policy to give "log on as a service" permissions to the Run As User account. This elevated policy is required because the Run As User is used as the security context for the Tableau Server Application Manager service (tabsv).

**Note:** If the Run As User account that you specify in Tableau Server Configuration is a member of the local administrators or a domain administrator, then tabadmin may not update the local security policy. Updating the Run As User with an account that is a
member of local administrators or domain administrators is not a good security practice. We recommend using a domain User account for the Run As User.

In some cases, you may need to manually set security policy for your Run As User. For example, some organizations run Windows Group Policy that remove "Log on as service" rights that have been set on user accounts. Or an organization may run a policy that creates a permission conflict by specifying "Deny log on as a service." If your organization does this, then you will need to disable or edit such Group Policies so that your Run As User account is not affected.

The following procedure describes how to configure security policy, Log on as service, manually. You can also use the procedure below to verify that your Run As User is appropriately configured with local security policy rights. For example, you should verify that the Run As User account is not specified on the Deny log on as service policy.

If you are running a distributed installation, then configuration must be the same across the primary and all worker nodes.

To verify or update the local security policy:

2. In Local Security Policy, open Local Policies, select User Rights Assignments.
   - To verify or set Log on as a service policy:
a. Right-click **Log on as a service** policy and then click **Properties**.

![Local Security Policy]

b. In **Log on as a service Properties**, click **Add User or Group**.

c. Type the `<domain>\<username>` for the Tableau Server Run As User account (for example: MYCO\tableau_server), and click **Check Names**.

d. When the account resolves correctly, it is underlined. Click **OK**.

**To verify Run As User account is not specified in the Deny log on as a service policy:**

a. Right-click **Deny log on as a service** policy, and then click **Properties**.

b. In **Deny log on as a service Properties**, verify that the Run As User account is not listed. If it is, remove it. When you are finished, click **OK**.

3. Click **OK** to close the Local Security Settings windows.

**Verify Tableau Service Settings**

Confirm that Tableau services are assigned the correct Log On and Startup values. If you are running a **distributed installation** of Tableau Server, perform these steps on the workers as well as on the primary.
1. Log on as administrator to the computer running Tableau Server.

2. On the Tableau Server computer, select **Start > Control Panel > Administrative Tools > Computer Management > Services and Applications > Services.**

3. Open Services and Applications, then click **Services.** Confirm that the following services have the correct settings:

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Logon Value</th>
<th>Startup Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLEXnet Licensing Service</td>
<td>Local System</td>
<td>Automatic</td>
</tr>
<tr>
<td>Secondary Logon</td>
<td>Local System</td>
<td>Automatic</td>
</tr>
<tr>
<td>Tableau Server Application Manager (tabsvc)</td>
<td>&lt;domain&gt;&lt;username&gt; This is the Run As User account. See below. If you have not specified a Run As User account, then Network Service account is used.</td>
<td>Automatic</td>
</tr>
<tr>
<td>Tableau Server License Manager (tablicsrv)</td>
<td>Local Service</td>
<td>Automatic</td>
</tr>
<tr>
<td></td>
<td>The License Manager relies on default Windows folder permissions that are applied to the Local Service. If you are seeing licensing errors in the tabadmin log files, then you may need to modify permissions on the Tableau installation directory. See <strong>Verify Folder Permissions</strong> on page 1232 for more information.</td>
<td>(Delayed Start)</td>
</tr>
</tbody>
</table>

**Note:** Do not change the default settings on the **Recovery** tab of the **Tableau Server Application Manager Properties** dialog box; leave the settings for failure recovery as **Take No Action.** If you change these settings, Tableau Server will restart after being stopped via the **tabadmin** command or **Stop Tableau Server** command.

**Changing the Log On Value**

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

1. Select **Start > All Programs > Tableau Server > Stop Tableau Server.**
2. Select **Start > All Programs > Tableau Server > Configure Tableau Server.**
3. On the General tab, enter the domain, user name, and password for Tableau Server's Run As User account.
4. Click **OK**, and then select **Start > All Programs > Tableau Server > Start Tableau Server**.

**Server Administrator Reference**

This section provides reference material for server administrators.

- Tableau Server Processes below
- Tableau Server Ports on page 1244

**Tableau Server Processes**

There are Tableau Server processes whose default configuration you can change to achieve different results. The topics Performance Tuning Examples on page 1135 and High Availability on page 709 describe some of the approaches you can take. High-level status for each process is displayed on the server’s Status page and more detailed information related to some of the processes—such as the background process—is in the Administrative Views on page 1097 topic.

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

For information on log files generated by these processes, see Server Log File Locations on page 1190.

<table>
<thead>
<tr>
<th>Process</th>
<th>File Name</th>
<th>Purpose</th>
<th>Multi-Threaded?</th>
<th>Performance Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>vizportal.exe</td>
<td>Handles the web application, REST API calls, supports browsing and searching</td>
<td>Yes</td>
<td>Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>backgrounder.exe</td>
<td>Executes server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
</tr>
<tr>
<td>Cache Server</td>
<td>redis-server.exe</td>
<td>Query cache</td>
<td>No</td>
<td>A query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and API server and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.</td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>clustercontroller.exe</td>
<td>Responsible for monitoring various components, detecting failures, and executing failover when</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Coordination Service</td>
<td>zookeeper.exe</td>
<td>In distributed installations, responsible for ensuring there is a quorum for making decisions during failover</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server installations of more than five nodes, also installed on the first four worker nodes.</td>
</tr>
<tr>
<td>Data Engine</td>
<td>tdeserver64.exe</td>
<td>Stores data extracts and answers queries</td>
<td>Yes</td>
<td>The data engine's workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory consumption is primarily based on the size of the data extracts being loaded. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.</td>
</tr>
<tr>
<td>Data Server</td>
<td>dataserver.exe</td>
<td>Manages connections to Tableau</td>
<td>Yes</td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by</td>
</tr>
<tr>
<td>Process</td>
<td>File Name</td>
<td>Purpose</td>
<td>Multi-Threaded?</td>
<td>Performance Characteristics</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>File Store</td>
<td>filestore.exe</td>
<td>Server data sources&lt;br&gt;Automatically replicates extracts across data engine nodes</td>
<td>n/a</td>
<td>Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.</td>
</tr>
<tr>
<td>Repository</td>
<td>postgres.exe</td>
<td>Tableau Server database, stores workbook and user metadata</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see <a href="#">Tableau Server Repository</a> on page 653.</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>searchserver.exe</td>
<td>Handles fast search, filter, retrieval, and display of content metadata on the server</td>
<td>Yes</td>
<td>The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>vizqlserver.exe</td>
<td>Loads and renders views, computes and executes queries</td>
<td>Yes</td>
<td>Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.</td>
</tr>
</tbody>
</table>
Tableau Server Ports

The following table lists the ports that Tableau Server uses by default, and which must be available for binding. If you install multiple instances of a process (Cache Server for example) on a node, consecutive ports are used, starting at the base port. If Windows Firewall is enabled, Tableau Server will open the ports it needs for internal communication between processes. (There are circumstances when you may need to take action in addition. If you are making an external connection to the Tableau Server database you may need to open ports manually. If you have a distributed installation with a worker running Windows 7, see the Tableau Knowledge Base.)

Dynamic port remapping

When dynamic port remapping is enabled (the default), Tableau Server first attempts to bind to the default ports, or to user-configured ports if they are defined. If the ports are not available, Tableau Server attempts to remap most processes to other ports, starting at port 8000. When next restarted, Tableau Server will revert to using the default or configured ports.

The gateway port and SSL port are not dynamically remapped. If port 80 is not available when Tableau Server is first installed, the installation program will choose a different gateway port (usually 8000). This value will display on the General tab of the Configuration utility. Tableau Server will always use the port shown in the Configuration utility for the gateway process.

When dynamic port remapping is disabled, Tableau Server does not attempt to remap processes and if a conflict is detected, Tableau Server will not start.

Note: Port conflicts can affect how JMX ports are determined. For more information, see Enable the JMX Ports on page 600.

You can disable dynamic port remapping using the `tabadmin set service.port_remapping.enabled` command. For more information, see `tabadmin set options` on page 1294.

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/U-DP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>All</td>
<td>Distributed</td>
</tr>
<tr>
<td>80</td>
<td>TCP</td>
<td>Gateway</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSL. When Tableau Server is configured</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>All</td>
<td>Distributed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

- **Port 2233**
  - **Protocol:** UDP
  - **Used by:** Server Resource Manager UDP port used for communication between Tableau Server processes. The Server Resource Manager monitors memory and CPU usage of Tableau Server processes (backgrounder.exe, dataserver.exe, tabprotosrv.exe, tdeserver.exe, vizportal.exe, vizqlserver.exe).

- **Port 3729**
  - **Protocol:** TCP
  - **Used by:** Tableau Server setup

- **Ports 373-0**
  - **Protocol:** TCP
  - **Used by:** Tableau worker servers in distributed environment

For SSL, the application server redirects requests to this port.
<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>3731</td>
<td>TCP</td>
<td>distributed and highly available environments (the primary Tableau Server does not listen on these ports).</td>
<td>All Dis-tributed</td>
</tr>
<tr>
<td>5000</td>
<td>UDP</td>
<td>Server Worker Manager process (tabadmwrk.exe) that is used for auto-discovery of worker servers in a distributed environment.</td>
<td>X</td>
</tr>
<tr>
<td>6379</td>
<td>TCP</td>
<td>Cache Server process (redis-server.exe). Base port 6379. Consecutive ports after 6379 are used, up to the number of processes.</td>
<td>X</td>
</tr>
<tr>
<td>8060</td>
<td>TCP</td>
<td>PostgreSQL database</td>
<td>X</td>
</tr>
<tr>
<td>8061</td>
<td>TCP</td>
<td>PostgreSQL database. Used for verifying integrity of</td>
<td>X</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>8062</td>
<td>TCP</td>
<td>PostgreSQL database</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>database for restoring.</td>
<td></td>
</tr>
<tr>
<td>8080</td>
<td>TCP</td>
<td>Solr, Tomcat HTTP, and Repository processes</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8085</td>
<td>TCP</td>
<td>Tomcat HTTP</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8250</td>
<td>TCP</td>
<td>Background tasks</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8350</td>
<td>TCP</td>
<td>Background tasks</td>
<td>All</td>
</tr>
<tr>
<td>8600</td>
<td>TCP</td>
<td>Application Server process (vizportal.exe). Base port 8600. Consecutive ports after 8600 are used, up to the number of processes.</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8700</td>
<td>TCP</td>
<td>Application Server process (vizportal.exe)</td>
<td>All</td>
</tr>
<tr>
<td>8755</td>
<td>TCP</td>
<td>Tableau Administrative process</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>All</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>910-0–9199</td>
<td>TCP</td>
<td>VizQL Server process (base port 9100). Consecutive ports after 9100, up to</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the number of processes, are also used. By default, Tableau Server installs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>with two VizQL Server processes (ports 9100 and 9101).</td>
<td></td>
</tr>
<tr>
<td>9200, 9400</td>
<td>TCP</td>
<td>VizQL Server process</td>
<td>X</td>
</tr>
<tr>
<td>9345</td>
<td>TCP</td>
<td>File Store service</td>
<td>X</td>
</tr>
<tr>
<td>9346</td>
<td>TCP</td>
<td>File Store status service</td>
<td>X</td>
</tr>
<tr>
<td>970-0–9899</td>
<td>TCP</td>
<td>Data Server process (base port 9700). Consecutive ports after 9700, up to</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the number of processes, are also used. By default,</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/U-DP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>---------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tableau Server installs with two Data Server processes (ports 9700 and 9701).</td>
<td>All</td>
</tr>
<tr>
<td>9800, 1000-0</td>
<td>TCP</td>
<td>Data Server process</td>
<td></td>
</tr>
<tr>
<td>1100-0</td>
<td>TCP</td>
<td>Search server</td>
<td></td>
</tr>
<tr>
<td>1110-0</td>
<td>TCP</td>
<td>Search server</td>
<td></td>
</tr>
<tr>
<td>1200-0</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) client port</td>
<td></td>
</tr>
<tr>
<td>1201-2</td>
<td>TCP</td>
<td>Cluster Controller process</td>
<td></td>
</tr>
<tr>
<td>1300-0</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) leader port</td>
<td></td>
</tr>
<tr>
<td>1400-0</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) leader election port</td>
<td></td>
</tr>
<tr>
<td>2700-0-</td>
<td>TCP</td>
<td>Workers and</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>2700-9</td>
<td>TCP</td>
<td>primary server to communicate licensing information in distributed and highly available environments.</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td>One additional port is dynamically chosen for workers and the primary server to communicate licensing information in distributed and highly available environments. Instead, you can specify a fixed port (27010 is recommended). See the Tableau Knowledge Base for details.</td>
<td>X</td>
</tr>
<tr>
<td>2704-2</td>
<td>TCP</td>
<td>Data Engine process. Tableau Server installs with one</td>
<td>X</td>
</tr>
</tbody>
</table>
**User Management in Active Directory Deployments**

This topic describes important technical details that you should be familiar with if you use Active Directory to authenticate users for Tableau Server.

**Note:** This topic assumes that you are familiar with Active Directory user management and basic Active Directory schema and domain concepts.

**Active Directory user authentication and Tableau Server**

Tableau Server stores all user names in the Tableau Server identity store, which is managed by the **Tableau Server Repository** on page 653. If Tableau Server is configured to use Active Directory for authentication, you must first import user identities from Active Directory to the identity store. When users sign in to Tableau Server, their credentials are passed to Active Directory, which is responsible for authenticating the user; Tableau Server does not perform this authentication. (By default, NTLM is used for authentication, but you can enable Kerberos or SAML for single sign-on functionality—however, in all these cases, authentication is left to Active Directory.) However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

**Active Directory user name attributes and Tableau Server**

Active Directory uniquely identifies user objects using several attributes. (For details, see **User Naming Attributes** on the MSDN website.) Tableau Server relies on two Active Directory user naming attributes:
• `sAMAccountName`. This attribute specifies the logon name that was originally designed for use with older versions of Windows. In many organizations, this name is combined with the NetBIOS name for authentication, using a format like `example\jsmith`, where `example` is the NetBIOS name and `jsmith` is the `sAMAccountName` value. Due to the original design in Windows, the `sAMAccountName` value must be less than 20 characters.

In the Windows Active Directory Users and Computers administrative console, this value is in the field labeled User logon name (pre-Windows 2000) on the Account tab of the user object.

• `userPrincipalName` (UPN). This attribute specifies a user name in the format `j smith@example.com`, where `jsmith` is the UPN prefix and `@example.com` is the UPN suffix.

In the Windows Active Directory Users and Computers administrative console, the UPN is a concatenation of two fields on the Account tab of the user object: the User logon name field, and the domain drop-down list next to it.

**Adding users from Active Directory**

You can add users individually from Active Directory, either by typing them in the server environment or by creating a CSV file and importing the users. You can also add Active Directory users by creating a group via Active Directory and importing all of the group's users. The result can be different depending on which approach you're using.

**Adding users individually**

In most cases, Tableau Server uses the `sAMAccountName` value for the user name. When you import users individually from Active Directory (either by typing in their names or by using a CSV file), Tableau queries Active Directory with the user name that you provide. If a is matched is found, then that name is imported into Tableau Server and it becomes the name that the user enters in order to sign in to Tableau Server.

The user name that Tableau Server will import into the identity store will be the `sAMAccountName` value unless one of the following is true:

- If the user name that you specify is longer than 20 characters.
- If the user name that you specify contains an @ character.

If the user name you enter meets either of these conditions, then Tableau will import the `userPrincipalName` attribute, which will become the user's Tableau logon user name.

If user names were inadvertently imported using UPN names, you can delete the accounts in Tableau Server and then reimport those accounts using the `sAMAccountName` value for the user name, as shown in User logon name (pre-Windows 2000) in the Windows Active Directory Users and Computers administrative console.
Adding user groups

If you import an Active Directory user group, Tableau will import all users from the group using the `sAMAccountName`.

**Sync behavior when removing users from Active Directory**

Users cannot be automatically removed from Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that you can audit and reassign the user’s content before removing the user’s account completely.

However, Tableau Server will act upon user objects differently based how the status of that user object changes in Active Directory. There are two scenarios: deleting/disabling users in Active Directory or removing users from synchronized groups in Active Directory.

When you delete or disable a user in Active Directory and then synchronize that user’s group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user’s role is set to “unlicensed.”
- The user will still belong to the All Users group.
- The user is unable to sign in to Tableau Server.

When you remove a user from a group in Active Directory and then synchronize that group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user’s role is retained: it is not set to “unlicensed.”
- The user will still belong to the All Users group.
- The user will still have permission to the Tableau Server with access to everything that the All Users group is granted permission to use.

In both instances, to remove a user from Tableau Server, the server administrator must delete the user from the Server Users page in Tableau Server.

**Domain nicknames**

In Tableau Server, domain nickname is equivalent to the Windows NetBIOS domain name. In a Windows Active Directory forest, a fully qualified domain name (FQDN) can have an arbitrary NetBIOS name. The NetBIOS name is used as the domain identifier when a user logs in to Active Directory.

For example, the FQDN `west.na.corp.lan` might be configured with a NetBIOS name (nickname) of `SEATTLE`. The user `jsmith` in that domain could log on to Windows using either of the following user names:

- `west.na.corp.example.com\jsmith`
- `SEATTLE\jsmith`
If you want your users to sign in to Tableau Server with a NetBIOS name instead of the FQDN, then you’ll need to verify that the nickname value for each domain where users log in is set. See editdomain on page 1331 for information on how to view and set the nickname value for each domain.

Support for multiple domains
You can add users from a domain that's different from the domain of the Tableau Server computer in these cases:

- Two-way trust has been established between the server’s domain and the users’ domain.
- The server's domain trusts the users’ domain (one-way trust).

The first time you add a user from the non-server domain, use the fully-qualified domain name with the user name. Any additional users you add from that domain can be added using the domain’s nickname, provided the nickname matches the NetBIOS name.

Sign in to Tableau Server with NetBIOS name
Users can sign in to Tableau Server using the domain nickname (NetBIOS name), for example, SEATTLE\jsmith.

Tableau Server cannot query for NetBIOS name for a given FQDN. As a result, Tableau sets the nickname of a given FQDN according to the first entry in the namespace. For example, given the FQDN west.na.corp.lan, Tableau sets the nickname to west.

Therefore, you might need to update the domain nickname on Tableau Server before users can sign in using the nickname. If you do not update the nickname, users will have to sign in using a fully qualified domain name. For more information, see Users From New Domain Unable to Log In and Do Not Appear in User List in the Tableau Knowledge Base.

Command Line Utilities
Tableau Server has two built-in Windows-based command line utilities for scripting and automating various server tasks: tabadmin and tabcmd.

The tabadmin utility is used for administrative configuration tasks such as changing settings and customizing Tableau Server. This utility must be run from the primary Tableau Server computer.

The tabcmd utility is used for tasks that can also be performed from within the Tableau Server interface, such as removing users, forcing refreshes, and pulling reports. This utility can be run from the primary Tableau Server machine or installed remotely and run from another computer.
tabadmin

You can perform certain administrative tasks and change Tableau Server configuration settings using the tabadmin command line tool. It installs with Tableau Server by default and cannot be installed on other computers. For more information, see the following topics.

How to Use tabadmin

tabadmin allows you to perform administrative tasks from the command line on Tableau Server. It installs with Tableau Server by default and cannot be installed on other machines.

Note: You should only run tabadmin on the primary Tableau Server node, not on worker nodes.

The first step to using tabadmin is to open a command prompt as an administrator:

![Command Prompt]

Next, navigate to Tableau Server's bin directory by entering the following:

```
cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
```

You're now ready to enter tabadmin commands.

Change Tableau Server's Configuration from the Command Line

When you enter a command that changes the server’s configuration (a tabadmin set command for example), you need to follow a sequence of commands:

1. **Stop the server** before issuing the command.
2. Enter the appropriate command to make the configuration change.
3. Run `tabadmin config` to push the change out to all of the server's configuration files.

4. Start Tableau Server again.

Example

Change the server's configuration using the `tabadmin set` command:

```bash
tabadmin stop

tabadmin set [option-name value]

tabadmin config

tabadmin start
```

Display Command Line Help

Use the `tabadmin` built-in help to get a quick description of a command.

To display help for all `tabadmin` commands enter:

```bash
tabadmin help commands
```

To see help for a specific command, enter `tabadmin help <command>`. For example:

```bash
tabadmin help set
```

**tabadmin Commands**

**Note:** You should only run `tabadmin` on the primary Tableau Server node, not on worker nodes.

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

- **activate** on page 1258
- **administrator** on page 1259
- **assetkeys** on page 1259
- **autostart** on page 1261
- **backup** on page 1262
- **cleanup** on page 1263
- **clearcache** on page 1265
- **configure** on page 1265
- **customize** on page 1266
- **dbpass** on page 1268
decommission on page 1270
delete_webdataconnector on page 1270
exportsite on page 1271
failoverprimary on page 1273
failoverrepository on page 1274
get_openid_redirect_url on page 1274
importsite on page 1275
importsite_verified on page 1277
import_webdataconnector on page 1277
licenses on page 1279
list_webdataconnectors on page 1279
manage_global_credentials on page 1280
passwd on page 1281
recommission on page 1281
regenerate_internal_tokens on page 1282
reindex on page 1283
reset on page 1283
reset_openid_sub on page 1284
restart on page 1284
restore on page 1285
set on page 1286
sitestate on page 1286
start on page 1287
status on page 1288
activate
Activates or returns a Tableau Server license online or offline.

Examples
Activate a license offline:
```
tabadmin activate --tlf <file.tlf>
```
Return a license offline:
```
tabadmin activate --tlr <file.tlr>
```
Activate a license online:
```
tabadmin activate --activate <license>
```
Return a license online:
```
tabadmin activate --return <license>
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--tlf</td>
<td>FILE</td>
<td></td>
<td>For offline activation. If you are offline during Setup, you are prompted to save a .tlg file, which you submit to Tableau. Tableau sends you a .tlf file. You use this .tlf file to activate Tableau Server.</td>
</tr>
<tr>
<td>--tlr</td>
<td>FILE</td>
<td></td>
<td>For offline deactivation. The file you use as the argument is the .tlr file that you receive from Tableau.</td>
</tr>
<tr>
<td>--activate</td>
<td></td>
<td></td>
<td>Activate the specified license.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--return</td>
<td></td>
<td></td>
<td>Return the specified license.</td>
</tr>
</tbody>
</table>

**See Also**

*Activate Tableau Offline* on page 604

**administrator**

Grants or removes the system administrator capability to the named user. This command does not apply to site administrators.

**Examples**

Remove the system administrator capability from user *hwilson*:

```
tabadmin administrator hwilson false
```

Give the system administrator capability to user *jsmith*:

```
tabadmin administrator jsmith true
```

**assetkeys**

Creates a new key to encrypt sensitive information, such as credentials for external databases, stored within the Tableau repository, which is a PostgreSQL database that Tableau Server uses internally. The key you create with this command can contain either a passphrase that you specify or one that's randomly generated.

*Note:* Tableau Server must be running when you issue this command.

If you specify your key's passphrase, it's a best practice for it to be at least eight characters long. You should also take character sets into consideration. A strong passphrase should contain characters from at least three of the following character sets:

- Lowercase a-z
- Uppercase A-Z
• Digits 0-9
• Non-alphabetic characters

The new key is encrypted and stored in the following key file: `asset_keys.yml` (ProgramData\Tableau\Tableau Server\data\tabsvc\config). If the key file is lost or corrupted, you can use the `assetkeys --validate` command to recreate it.

If you use the `assetkeys` command then later create and restore a backup file (.tsbak), you will need to run the `tabadmin assetkeys --validate` command after restoring the backup file. By design, backup files do not contain custom encryption keys—even though some data may be encrypted with them. This protects the encrypted values in case the backup file falls into the wrong hands. When you run `tabadmin assetkeys --validate` after a backup restore, you are prompted to enter the key's passphrase.

**Examples**

Have Tableau Server generate a key and passphrase for you:

```
tabadmin assetkeys --auto_create
```

Generate a key using a passphrase that you specify. You are prompted to enter a passphrase, which will not be displayed as you type:

```
tabadmin assetkeys --create
```

Use the contents of a file as the passphrase:

```
tabadmin assetkeys --create_from_file C:\test\key\password.txt
```

Confirm that the key file `asset_keys.yml` in ProgramData\Tableau\Tableau Server\data\tabsvc\config is valid and consistent with the metadata in the Tableau Repository:

```
tabadmin assetkeys --validate
```

Recreate the file `asset_keys.yml` which is now corrupted or missing from ProgramData\Tableau\Tableau Server\data\tabsvc\config:

```
tabadmin assetkeys --validate
```

You will be prompted for the passphrase.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--auto_create</td>
<td>--auto_create</td>
<td>[length]</td>
<td>Generates a random passphrase to generate the key. Takes an optional argument for the length of the passphrase. You should record the passphrase and keep it in a safe place, as it will be required by --</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>validate</td>
<td>validate if assetkeys.yml is lost or corrupted.</td>
</tr>
<tr>
<td>--create</td>
<td></td>
<td>FILE</td>
<td>Generates a key using the passphrase you provide. You are prompted for the passphrase and it will not display as you type it. Your passphrase should be at least 10 characters long and not based on words found in the dictionary.</td>
</tr>
<tr>
<td>--create_from_file</td>
<td></td>
<td></td>
<td>Generates a key using the contents of a file that you provide as the passphrase.</td>
</tr>
<tr>
<td>--validate</td>
<td></td>
<td></td>
<td>Confirms that all asset keys being used internally by Tableau Server are up-to-date. If you lose the asset_keys.yml file (for example, due to file corruption), you can use the --validate option to recreate it. You are prompted for and must enter the passphrase that was used to generate the current asset keys in order to successfully recreate the key file.</td>
</tr>
</tbody>
</table>

See Also

Security on page 953

autostart

Specifies whether Tableau Server starts at system start-up time. By default, Tableau Server starts when the computer on which it's installed starts. If autostart is set to off, you will need to start Tableau Server either using tabadmin start or the Start menu.

Example

Display Tableau Server's auto-start status:

```
tabadmin autostart
```

Start Tableau Server when the operating system starts:

```
tabadmin autostart on
```
Do not start Tableau Server when the operating system starts:

`tabadmin autostart off`

**backup**

Creates a backup of the data managed by Tableau Server. This data includes Tableau's own PostgreSQL database, which contains workbook and user metadata, data extract (.tde) files, and configuration data. If you have imported web data connectors using the `import_webdataconnector` command, the backup process saves copies of the connectors as well. You do not need to stop Tableau Server before you create a backup file.

By default, the backup file is put into the directory where you are running the `tabadmin backup` command. To put the backup file into a specific location, you can include full path with the backup file name. You can also use the `--userdir` option to put the backup file into a known location.

**Note:** The command adds the .tsbak extension to the file name that you specify unless the name already contains that extension.

**Examples**

Create a backup file in the current directory named `tabserv.tsbak`:

```
tabadmin backup tabserv.tsbak
```

Create a backup file in the C:\backups\tableau folder named `tabserv.tsbak`:

```
tabadmin backup C:\backups\tableau\tabserv.tsbak
```

Append the current date to the backup file name and put temporary files created during the backup process in C:\mytemp\tableau. The backup file `tabserv.tsbak` is created in the directory where you are running the command from:

```
tabadmin backup tabserv.tsbak -d -t C:\mytemp\tableau
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>--date</td>
<td></td>
<td>Appends the current date to the backup file name.</td>
</tr>
<tr>
<td>-u</td>
<td>--user-dir</td>
<td></td>
<td>Places the backup file in the ProgramData\Tableau\Tableau Server folder.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-t</td>
<td>--tempdir</td>
<td>PATH</td>
<td>Specifies the location for temporary files created during the backup or when verifying database integrity.</td>
</tr>
<tr>
<td>-v</td>
<td>--verify</td>
<td></td>
<td>Verifies the integrity of the database. Available beginning with version 9.3.</td>
</tr>
</tbody>
</table>

See Also

**Back Up Tableau Server Data** on page 1144

cleanup

Reduces the disk space consumed by Tableau Server. Running `tabadmin cleanup` removes log files, temporary files, and select rows in Tableau Server’s PostgreSQL database. If Tableau Server is installed on multiple computers in a cluster, the command can also reset the information maintained by the coordination server that is used to synchronize between nodes and to manage failover.

The effect of the `cleanup` command depends on whether the server is running or stopped. For more information, see **Remove Unneeded Files** on page 1152.

**Examples**

Remove log files, temporary files, and HTTP request entries in the PostreSQL database:

```
    tabadmin cleanup
```

Remove log files and temporary files (leave HTTP request entries in the database untouched):

```
    tabadmin cleanup --restart
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>--restart</td>
<td></td>
<td>Stops Tableau Server, runs the cleanup command, and starts the server again.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>--reset-coordination</td>
<td></td>
<td></td>
<td>In addition to performing a normal cleanup, removes log files, transaction logs, and snapshots that are maintained by the Tableau Server coordination service (zookeeper) when Tableau Server is running on multiple computers in a cluster. Note that using this option completely resets the coordination service, meaning all state maintained by the coordination service is removed. This option also does the equivalent of a tabadmin configure command. For guidelines about when to</td>
</tr>
</tbody>
</table>
### Option (short) | Option (long) | Argument | Description
--- | --- | --- | ---
 |  | reset |  | reset the coordination service, see Troubleshoot Server Processes on page 1212.

#### See Also

Remove Unneeded Files on page 1152

---

clearcache

Clears the information being cached by the Cache Server process (redis-server.exe). The cache stores information used to render views in order to help speed rendering. Clearing the cache is useful if metadata about views or data sources that might be cached has changed, and those changes should take effect before the resource is removed from the cache in the normal course of server processing. For example, clearing the cache can be useful if you change permissions on a workbook or view and it’s important that the changed permissions take effect immediately.

You must stop the server before you run this command.

#### Examples

tabadmin clearcache

#### See Also

Tableau Server Processes on page 1240

---

configure

Updates Tableau Server’s configuration by forcing an update to all the files in ProgramData\Tableau\Tableau Server\data\tabsvc\<area>. This update includes refreshing the master service configuration file, workgroup.yml (ProgramData\Tableau\Tableau Server\data\tabsvc\config). When you make a configuration change, it’s a best practice to run
tabadmin configure (or tabadmin config) to ensure that all files affecting the server’s configuration are completely updated.

If you are running Tableau Server in a distributed environment and if you have imported web data connectors using the import_webdataconnector on page 1277 command or deleted them using the delete_webdataconnector on page 1270 command, the configure command makes sure that any web data connectors are correctly distributed (imported or deleted) in all nodes where the gateway process is running.

Examples

```
tabadmin configure
```

```
tabadmin config
```

See Also

- **Reconfigure the Server** on page 641
- **set** on page 1286
- **tabadmin set options** on page 1294

**customize**

Customizes the server name that’s displayed in tooltips and messages, and the logos that are used by Tableau Server. Note that even if you use this command, the bottom of every server page lists Tableau's copyright information.

Image files you use for logos can be in GIF, JPEG, or PNG format.

**Examples**

**Name**

Change the product name used in tooltips from “Tableau Server” to "My Company":

```
tabadmin customize name "My Company"
```

Reset the product name to the default:

```
tabadmin customize name -d
```

**Header logo**

Customize the main server header logo. The image can be up to 160 by 160 pixels, but not smaller than 32 by 32 pixels. For best results use an image that's 125 by 35 pixels. If the image is larger than 160 by 160 pixels, it is clipped.

```
tabadmin customize header_logo "C:\My Pictures\example.png"
```
Reset the header logo to the default:

```
tabadmin customize header_logo -d
```

**Sign-in logo**

Customize the sign-in page logo. The image can be up to 3000 by 3000 pixels.

```
tabadmin customize sign_in_logo "C:\My Pictures\example.png"
```

Reset the sign-in logo to the default:

```
tabadmin customize sign_in_logo -d
```

**Small logo**

Customize the web authoring header logo. The image can be up to 32 by 32 pixels. For best results use an image that's 32 by 32 pixels.

```
tabadmin customize smalllogo "C:\My Pictures\example.png"
```

Reset the header logo to the default:

```
tabadmin customize smalllogo -d
```

**Logo**

Set the main server header and the sign-in page logo to the same image. The image can be up to 160 by 160 pixels, but not smaller than 32 by 32 pixels. If the image is larger than 160 by 160 pixels, it is clipped.

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

Reset the logo to the default:

```
tabadmin customize logo -d
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>--default</td>
<td>name</td>
<td>header_logo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NAME</td>
<td>Sets the name to the value in the argument. The default is &quot;Tableau Server&quot;.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FILE</td>
<td>Sets both the header logo and</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>the sign-in page logo to the image referenced in the file path. If the image is larger than 160 by 160 pixels, it is clipped.</td>
</tr>
<tr>
<td>header_logo</td>
<td>FILE</td>
<td></td>
<td>Sets the logo to the image referenced in the file path. For best results use an image that's 125 by 35 pixels. If the image is larger than 160 by 160 pixels, it is clipped.</td>
</tr>
<tr>
<td>sign_in_logo</td>
<td>FILE</td>
<td></td>
<td>Sets the logo to the image referenced in the file path. The image can be up to 3000 by 3000 pixels.</td>
</tr>
<tr>
<td>smalllogo</td>
<td>FILE</td>
<td></td>
<td>Sets the logo to the image referenced in the file path. For best results use an image that's 32 by 32 pixels.</td>
</tr>
</tbody>
</table>

See Also

**Change the Name or Logo** on page 655

dbpass

Enables external access to Tableau’s PostgreSQL database (the repository). After you use the dbpass command to allow access to the database, you can connect to and query it using Tableau Desktop to create your own administrative views.

tabadmin dbpass [--disable] [--username <username>] [password]

**Note:** The --username option is valid starting with Tableau Server 8.2.5. In earlier versions dbpass only enabled the "tableau" user and you could not specify the user. 8.2.5 added a second user called "readonly" and introduced the ability to specify the user you are enabling access for.
Examples

Enable access for the **tableau** user and set the password to p@ssword:
```
tabadmin dbpass p@ssword
```

Enable access for the **readonly** user and set the password to p@ssword:
```
tabadmin dbpass --username readonly p@ssword
```

Disable external access for the default **(tableau)** user:
```
tabadmin dbpass --disable
```
or
```
tabadmin dbpass --disable --username tableau
```

Disable external access for the **readonly** user:
```
tabadmin dbpass --disable --username readonly
```

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>--disable</strong></td>
<td></td>
<td>Disable external access to Tableau’s PostgreSQL database for the default remote user <strong>(tableau)</strong> or, starting in 8.2.5, if a user name is specified, disable remote access for that user.</td>
</tr>
<tr>
<td><strong>--username</strong></td>
<td>tableau or readonly</td>
<td>Change the password for the specified user, or, if used with the <strong>--disable</strong> option, disable access for the specified user. Options for users are <strong>tableau</strong> and <strong>readonly</strong>. This option is valid in Tableau Server 8.2.5 or higher.</td>
</tr>
<tr>
<td></td>
<td>password provided by user</td>
<td>Enable remote access to Tableau’s PostgreSQL database for the default remote user <strong>(tableau)</strong> or, starting in 8.2.5, if a user name is specified, enable access for that user with the given password.</td>
</tr>
</tbody>
</table>

See Also

**Collect Data with the Tableau Server Repository** on page 1117
decommission

Prepares Tableau Server File Store nodes for removal from the distributed installation. This command puts the specified nodes into read-only mode so new content cannot be added to the File Store, and makes sure that all content on the node also exists on another File Store node. This command can be run while Tableau Server is running.

**Note:** Remove a decommissioned File Store node before restarting Tableau Server. Restarting automatically re-activates any decommissioned File Store nodes.

tabadmin decommission <node1 node2 ...>

**Examples**
Decommission worker2:

```
tabadmin decommission worker2
```

Decommission two nodes by IP address:

```
tabadmin decommission 10.32.139.30 10.32.139.22
```

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;node1 node 2 node 3...&gt;</td>
<td>List of File Store nodes (servers) to decommission. Separate multiple nodes with a space.</td>
</tr>
</tbody>
</table>

**See Also**

- [Distributed Environments](#) on page 694
- [Maintain a Distributed Environment](#) on page 705

**delete_webdataconnector**

Removes the specified web data connector from the server, or removes all web data connectors. If the web data connector is installed on a cluster, this command removes the specified connector or all connectors from all computers in the cluster.

**Note:** If the server is running in a distributed environment and the delete process is partially successful, users can still access the connector. For more information, see [Web Data Connectors in Tableau Server](#) on page 899.
Examples

tabadmin delete_webdataconnector connector1.html

tabadmin delete_webdataconnector --all

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--all</td>
<td>--all</td>
<td></td>
<td>Removes all web data connectors from Tableau Server. When you use this option, you do not specify a connector name. If the server is configured as a cluster, the command removes all connectors from all the nodes where they are installed.</td>
</tr>
</tbody>
</table>

See Also

import_webdataconnector on page 1277
list_webdataconnectors on page 1279
Web Data Connectors in Tableau Server on page 899
Tableau webdataconnector page on GitHub

exportsite

Exports a Tableau Server site, including its users, workbooks, projects, extracts, and data connections, and places it in a file with a .zip file extension. You can then use the exported site file to provision a new site by using the importsite on page 1275 and importsite_verified on page 1277 commands.

You don't need to stop Tableau Server before you use the exportsite command. Tableau Server will lock the site being exported during the export process.

Notes: When you import a site that you exported earlier, each user and schedule that is being imported must match an existing user and schedule. For suggestions about how to manage the export and import process to match users and schedules, see Tips for importing to a target with fewer users or schedules than the source site.

If your source site has workbooks that use published data sources, the target site name must match the source site name. The data connections for the workbooks will continue to refer to the source site name and can't be updated on the new site.

Examples
tabadmin exportsite <site ID> --file <PATH>

or

tabadmin exportsite <site ID> --file <FILE>

Export the site whose site ID is `finance` to a file named `finance_export.zip` and place it in `Program Files\Tableau\Tableau Server\10.0\bin`:

```bash
tabadmin exportsite finance --file finance_export
```

Export the Default site. The site ID for the Default site is "" (double quotes, no space).

```bash
tabadmin exportsite "" --file finance_export
```

If you are using Windows PowerShell to run the command, enclose the double quotes for the Default site within single quotes ("""). For example:

```bash
tabadmin exportsite '""' --file finance_export
```

Export the Default site to a file named `finance_export.zip` and place it in `C:\temp\exported sites` instead of in the Tableau Server bin directory. Because the path contains a space, it’s contained by quotes:

```bash
tabadmin exportsite '"" --file "C:\temp\exported sites\finance_export"
```

Export the site whose site ID is `finance`, name the export site file `financesite.zip`, place the file in `C:\sites\exported`, and write temporary run-time files to `C:\temp_files`:

```bash
tabadmin exportsite finance --file C:\sites\exported\financesite --tempdir C:\temp_files
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--file</td>
<td>FILE or PATH</td>
<td></td>
<td>The name or name and location (path) of the exported site file to be created. If you don’t specify a path, Tableau Server’s bin directory is the assumed location (Program Files\Tableau\Tableau Server\10.0\bin).</td>
</tr>
<tr>
<td>--tempdir</td>
<td></td>
<td></td>
<td>The location of temporary files created during export. Use this option if you don’t have write access to the Tableau Server installation directory. This option does not determine where the export site file is created.</td>
</tr>
</tbody>
</table>
failoverprimary

Identifies a second installation of the primary Tableau Server as the backup primary, or if the primary has failed, identify the backup primary as the new primary and the former primary as the new backup.

**Note:** If you run this command without providing an option, the current computer is assumed to be the primary and no backup primary is identified.

**Example**

```
tabadmin failoverprimary --primary "<computer name(s) or IPv4 address(es)>"
```

The following command specifies the primary Tableau Server computer (10.32.139.22) and the backup primary (10.32.139.50):

```
tabadmin failoverprimary --primary "10.32.139.22,10.32.139.50"
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-- primary</td>
<td>Computer name(s) or IPv4 address(es)</td>
<td>The Tableau Server machine that's acting as the cluster's primary.</td>
</tr>
</tbody>
</table>

**See Also**

- [Understanding High Availability on page 714](#)
- [Configure for Failover and Multiple Gateways on page 720](#)
- [Use a Backup Primary on page 732](#)
failoverrepository

Manually identifies a second, passive installation of the PostGRES repository as the active repository.

If Tableau Server is configured for high availability, failover of the repository is automatic. Use the failoverrepository command to manually fail over the repository (for example, if Tableau Server is configured for manual repository failover using the `tabadmin set clustercontroller.pgsql.failover false` command).

Tableau Server must be running when you run the failoverrepository command.

```
tabadmin failoverrepository --target <computer name or IPv4 address> | --preferred
```

Example

```
tabadmin failoverrepository --target worker_server2
```

**Note:** This command is persistent. The failover repository remains the active repository until you issue the command again. If you have a preferred active repository configured, use the --preferred option to switch back to that repository.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--target</td>
<td>--target</td>
<td>Computer name or IPv4 address</td>
<td>The Tableau Server repository node to failover to.</td>
</tr>
<tr>
<td>--preferred</td>
<td>--preferred</td>
<td></td>
<td>Failover to the repository node that is specified as the preferred active repository.</td>
</tr>
</tbody>
</table>

See Also

**Understanding High Availability** on page 714

**Configure for Failover and Multiple Gateways** on page 720

**Use a Backup Primary** on page 732

get_openid_redirect_url

If Tableau Server is configured to use OpenID Connect for authentication, gets the URL that is used to redirect users from the identity provider (IdP) to Tableau Server after a successful sign-
Example

tabadmin get_openid_redirect_url

See Also

OpenID Connect on page 1050
Configure Tableau Server for OpenID Connect on page 1054

importsite

Imports a site into Tableau Server. The importsite command is the first of two commands you use to import a site into Tableau Server. To run this command, you need the following:

- An exported site file. Tableau Server administrators create this file using the exportsite on page 1271 command. If you have a site on Tableau Online and you want to import it into your own on-premises installation of Tableau Server, request an exported site file from Tableau Customer Support.

- The site ID for the target site. The target site is the Tableau Server site into which you want to import. The target site must already exist when you run the importsite command; you can’t create it as part of the command. The site ID for Tableau Server’s default site is "" (double quotes, no space).

The contents of the site that you import will replace (not amend) the contents of the target site. For example, if your target site has a workbook named MyDashboard.twbx and the site you are importing does not have this workbook, the import process will remove MyDashboard.twbx from the target site.

When you run the importsite command, the command creates a temporary directory containing mapping files in comma-separated-value (CSV) format that define how the exported site’s assets (users, workbooks, projects, extracts, and data sources) will be mapped when the site has been imported. It is important that you verify these details. Use a text editor or Microsoft Excel to open the mapping files and make any changes. Any entries with ??? (question marks) represent mappings that couldn’t be handled and must be edited. After you verify the mappings, finish the import process using the importsite_verified on page 1277 command.

Note: When you import a site that you exported earlier, each user and schedule that is being imported must match an existing user and schedule. For suggestions about how to manage the export and import process to match users and schedules, see Tips for importing to a target with fewer users or schedules than the source site.
Examples

```bash
tabadmin importsite <site ID> --file <PATH>
```

or

```bash
tabadmin importsite <site ID> --file <FILE>
```

Import the file `sales_site.zip` located in `C:\tableau\exported` to a site whose site ID is `wsales`:

```bash
tabadmin importsite wsales --file C:\tableau\exported\sales_site.zip
```

Import the file `sales_site.zip`, which is located in `C:\Program Files\Tableau\Tableau Server\10.0\bin`, to the Default site. The site ID for the Default site is "" (double quotes, no space).

```bash
tabadmin importsite "" --file sales_site.zip
```

The mapping files for you to verify are placed in `ProgramData\Tableau\Tableau Server\data\tabsvc\temp\import_<site ID>_<datetime>\mappings`. To specify a different directory, use the `--tempdir` option.

Place the files to be verified in `C:\temp\site_to_import`:

```
Skip the verification step (not recommended):
```

```bash
tabadmin importsite wsales --file "C:\tableau\exported\sales_site.zip" -no-verify
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--file</code></td>
<td></td>
<td>PATH</td>
<td>The name and location of the exported site file you are importing. If you don’t specify a path, Tableau Server’s bin directory is the assumed location (Program Files\Tableau\Tableau Server\10.0\bin).</td>
</tr>
<tr>
<td><code>--no-verify</code></td>
<td></td>
<td></td>
<td>Skips the verification step and imports the exported site file directly to its new location in your Tableau Server installation. If you choose this option, you do not need to use the <code>importsite_verified</code> command.</td>
</tr>
<tr>
<td><code>--tempdir</code></td>
<td></td>
<td>PATH</td>
<td>The directory where you will verify that the site files have the correct mappings. If you don’t specify this</td>
</tr>
</tbody>
</table>
importsite_verified

Performs the second part of an import process for a site on Tableau Server. Before you can use importsite_verified, you must first use importsite on page 1275.

The importsite_verified command reads from a directory containing CSV files that you have verified, and imports a new site into Tableau Server based on how the site’s assets are mapped in the CSV files. The site that receives the import (the target site) must already exist on Tableau Server.

During the import process, Tableau Server locks the site receiving the import.

**Examples**

```
tabadmin importsite_verified <target site ID> --importjobdir <PATH>
```

Import files from the directory C:\temp\site_to_import to the site whose site ID is **esale**:

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--import-jobdir</td>
<td>PATH</td>
<td></td>
<td>The directory containing CSV files whose mappings you have verified.</td>
</tr>
</tbody>
</table>

See Also

**Import or Export a Site** on page 745

import_webdataconnector

Installs a web data connector on the server. Users who create workbooks can then reference the web data connector as a data source.
Note: Starting with version 10.0 of Tableau Server, the recommended way to make web data connectors available on Tableau Server is to add them to a safe list. For more information, see tabadmin Commands on page 1256.

Important: Before you import a web data connector, make sure that the JavaScript code in the connector does not implement any functionality that should not be on your server.

When the `import_webdataconnector` command finishes importing the connector, the command displays the server URL of the connector. When users want to reference the web data connector as a data source, they need to know this URL. (You can also view the URLs of connectors on your server by using the `list_webdataconnectors` command.)

If the web data connector includes references to an external file, such as to a .css file or .js file, you must make sure that the external file is available from the server, either over the web or as a local file. If the connector references a local file, the local file must be in the same folder as the connector’s .html file relative paths to subdirectories are not supported for imported web connectors. (Make sure that the `<link>` or `<script>` element in the connector correctly references the file as a peer of the connector file.) If the external file is local, you must use the `import_webdataconnector` command to import the external file separately.

If the server includes multiple computers in a cluster, the web data connector is imported to each computer where a gateway process is running.

Examples

```
tabadmin import_webdataconnector connector1.html

tabadmin import_webdataconnector c:\webdataconnectors\connector1.html --overwrite

tabadmin import_webdataconnector \\myshare\webdataconnectors\connector2.html --overwrite

tabadmin import_webdataconnector connector1.css
```

Note: The connector name can contain only these characters: a-zA-Z0-9()~_.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--overwrite</td>
<td>--overwrite</td>
<td></td>
<td>Overwrites any existing file on the server that has the same name as the file that you are</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>importing.</td>
</tr>
</tbody>
</table>

See Also

delete_webdataconnector on page 1270
list_webdataconnectors below
Web Data Connectors in Tableau Server on page 899

licenses
Displays license information for Tableau Server.

Examples

tabadmin licenses

```
tabadmin licenses
```

```
tabadmin licenses -p
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-p</td>
<td>--processor_cores</td>
<td></td>
<td>Display the physical core count for the current machine.</td>
</tr>
</tbody>
</table>

list_webdataconnectors
Displays the names or URLs of web data connectors that are installed on the server.

Examples

List the names of the web data connectors.

```
tabadmin list_webdataconnectors
```

List the URLs of the web data connectors.

```
tabadmin list_webdataconnectors --urls
```
**manage_global_credentials**

Manages credentials for delegated data access on Tableau Server. Use this command to specify the credentials for a proxy user that is used to access a data source that does not support single-sign on via Kerberos.

**Examples**

tabadmin manage_global_credentials --add --server <server> --user <username> --password <password>

Add credentials for a server named my-server.

tabadmin manage_global_credentials --add --server my-server --user jsmith --password p@ssword

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--add</td>
<td></td>
<td></td>
<td>Add credentials for the specified server.</td>
</tr>
<tr>
<td>--remove</td>
<td></td>
<td></td>
<td>Remove credentials</td>
</tr>
<tr>
<td>--show</td>
<td></td>
<td></td>
<td>Show current credentials</td>
</tr>
<tr>
<td>-s</td>
<td>--server</td>
<td>server</td>
<td>Server for which credentials are being managed</td>
</tr>
<tr>
<td>-u</td>
<td>--username</td>
<td>user</td>
<td>User name for connecting to a server</td>
</tr>
<tr>
<td>-p</td>
<td>--password</td>
<td>password</td>
<td>Password for connecting to a server</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>word</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-o</td>
<td>--override</td>
<td></td>
<td>Override existing credentials</td>
</tr>
</tbody>
</table>

**See Also**

Enabling Delegation for Cloudera Impala in the Tableau Knowledge Base.

**passwd**

Resets the password for a Tableau Server account. After typing the command, you are prompted to enter a new password for the user.

You can only use this command if Tableau Server's user authentication is set to Local Authentication. When authentication is set to Active Directory, passwords are handled by Active Directory, not Tableau Server.

**Examples**

```
tabadmin passwd <username>
```

Reset the **password for server user jsmith:**

```
tabadmin passwd jsmith
```

**See Also**

Configure General Server Options on page 608

**recommission**

Reverts a decommissioned file store node in read-only mode to an active read/write state. Use spaces to separate multiple nodes.

**Examples**

```
tabadmin recommission <computer name(s) or IPv4 address(es)>
```

Recommission file store node by IP address:

```
tabadmin recommission 10.32.139.29
```
regenerate_internal_tokens

Creates new security tokens that Tableau Server uses internally. These tokens include the passwords used by Tableau Server to access the repository, and the certificates used to validate internal SSL connections between Tableau Server components and the repository.

Running this command stops Tableau Server, so you will need to restart Tableau Server after you run the command.

Example

```
tabadmin regenerate_internal_tokens --passwords
```
```
tabadmin regenerate_internal_tokens --certs
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--certs</td>
<td>None</td>
<td>Regenerates key pair for internal SSL connections.</td>
<td></td>
</tr>
<tr>
<td>--passwords</td>
<td>None</td>
<td>Regenerates passwords for the Postgres database.</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>Regenerates key pair for internal SSL connections and passwords for Postgres database. <strong>Note</strong>: The key pair is regenerated only if internal SSL is configured.</td>
<td></td>
</tr>
<tr>
<td>--restart</td>
<td>None</td>
<td>Restart Tableau Server after regenerating tokens.</td>
<td></td>
</tr>
</tbody>
</table>

See Also

**Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)** on page 961

**Security** on page 953
reindex

Rebuilds the search index for Tableau Server. In rare instances, you may need to rebuild the index if searches on the server return incomplete or incorrect results, or if the Search & Browse process is down for an extended period. You can use this command if users cannot sign in to the server because no sites are listed after they enter their credentials.

**Note:** The recommended way to reindex Search is to run this command while Tableau Server is stopped. Reindexing while the server is running can result in content, including sites and projects, temporarily disappearing.

**Examples**

tabadmin reindex

Reindex the server

**See Also**

*Rebuild the Search Index* on page 1168

reset

Resets the Tableau Server administrator account. This command will reset the server so that you will need to set up an administrator account.

**Example**

tabadmin reset

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--destroy-sessions</td>
<td>None</td>
<td>Destroys all existing sessions. All users will be forced to sign in again.</td>
</tr>
<tr>
<td></td>
<td>--silent</td>
<td>None</td>
<td>Suppresses normal verbose mode. This is useful if you are creating a chain of several automated steps.</td>
</tr>
</tbody>
</table>

**See Also**

*Add an Administrator Account* on page 637
reset_openid_sub

Cleans the user identifier (sub value) that binds a user identity in Tableau Server to a specific OpenID Connect identity provider (IdP).

If Tableau Server is configured to use OpenID Connect for authentication, the first time a user signs in to Tableau Server using the IdP, Tableau stores the sub value sent by the IdP with the user information in Tableau Server. The sub provides a unique identity for that user with the IdP. If you change IdPs for OpenID Connect, you must remove the sub value for the user. That way, when the user signs in using the new IdP, Tableau can store a new sub value.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-u</td>
<td>--username</td>
<td>username</td>
<td>Removes the sub value for the specified user.</td>
</tr>
<tr>
<td>-a</td>
<td>--all</td>
<td>None</td>
<td>Removes the sub value for all users.</td>
</tr>
</tbody>
</table>

Example

tabadmin reset_openid_sub --username Alice

This command cleans the sub value for the user named Alice.

tabadmin reset_openid_sub --all

This command cleans the sub value for all users on the server.

See Also

OpenID Connect on page 1050

Changing IdPs in Tableau Server for OpenID Connect on page 1058

restart

Stops and starts all Tableau Server processes. The restart command also does a configuration so you do not need to do a tabadmin config if you are doing a restart (a config will not do any harm).

Example

tabadmin restart
**restore**

Restores a Tableau Server backup file (.tsbak) to a Tableau Server installation. When you restore a .tsbak file, the contents of the Tableau PostgreSQL database, data extracts, and configuration files are overwritten with the content in the backup file. If the backup was made after web data connectors were imported to the server using the `import_webdataconnector` on page 1277 command, the restore process restores the connectors as well. Using the `--no-config` option restores everything but the server’s configuration.

**Examples**

Restore a file named `tabserv.tsbak` located in C:\mybackups and then restart the server:

```
tabadmin restore C:\mybackups\tabserv.tsbak --restart
```

Restore a file named `tabserv.tsbak` located in the Tableau Server bin directory and then restart the server:

```
tabadmin restore tabserv.tsbak --restart
```

Restore a file named `tabserv.tsbak` located in C:\mybackups, retaining everything but the server’s configuration, but don’t restart the server:

```
tabadmin restore --no-config C:\mybackups\tabserv.tsbak
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--no-config</code></td>
<td><code>--no-config</code></td>
<td></td>
<td>Restore the Tableau Server backup file including the data but excluding the server’s configuration.</td>
</tr>
<tr>
<td><code>--parallel-pg-restore</code></td>
<td><code>--parallel-pg-restore</code></td>
<td></td>
<td>Run the restore process for the PostgreSQL repository as a parallel job.</td>
</tr>
<tr>
<td><code>--password</code></td>
<td><code>&lt;password&gt;</code></td>
<td></td>
<td>Restore the Tableau Server backup file using the Run As User password.</td>
</tr>
<tr>
<td><code>--password-file</code></td>
<td><code>File</code></td>
<td></td>
<td>Restore the Tableau Server backup file, reading the password from the specified file.</td>
</tr>
<tr>
<td><code>--restart</code></td>
<td><code>--restart</code></td>
<td></td>
<td>Restart the service when the restore process has completed.</td>
</tr>
</tbody>
</table>
See Also

**Restore from a Backup** on page 1150

**Recover Extracts from a Backup** on page 1152

---

**set**

Allows you to change the value of Tableau Server configuration options. If the parameter you’re setting begins with a hyphen, enclose the parameter’s value in both double- and single-quotes.

**Examples**

```bash
tabadmin set [option-name value]
```

Set the backgrounder query limit to 2.5 hours (9000 seconds):

```bash
tabadmin set backgrounder.querylimit 9000
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>default</td>
<td></td>
<td>Reset the parameter to its default value.</td>
</tr>
</tbody>
</table>

See Also

**tabadmin set options** on page 1294

---

**sitestate**

Activates (unlocks) or suspends a site. You can use this command to activate a site that was locked because of a site import failure. When a site is suspended, the only Tableau Server user who can access it is the system administrator.

**Note:** To specify the default site, use "" for the site ID.

**Examples**

```bash
tabadmin sitestate <site ID> --status <active|suspended>
```

Activate a site whose site ID is *wsales*:

```bash
tabadmin sitestate wsales --status active
```
Activate the Default site. The site ID for the Default site is "" (double quotes, no space).

```
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--status</td>
<td>active or suspended</td>
<td>Specifies whether to activate or suspend the specified site.</td>
</tr>
</tbody>
</table>
```

```
tabadmin sitestate "" --status active
```

**start**

Starts all Tableau Server processes. The start command also does a configuration so you do not need to do a `tabadmin config` if you are doing a start (a config will not do any harm).

To use `tabadmin start`:

1. Open a command prompt as an administrator:

   ![Command Prompt as Administrator](image)

2. Type the following:

   ```
cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"
```
3. Type the following to start the server:

```
tabadmin start
```

**Examples**

```
tabadmin start
```

```
tabadmin start --wait 1200
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--wait</td>
<td>--wait</td>
<td>number of seconds</td>
<td>Number of seconds after starting after which Tableau Server is ready to accept client requests. The default is 600 seconds.</td>
</tr>
</tbody>
</table>

**status**

Tells you whether or not Tableau Server is running and, if you use the `--verbose` option, gives you details on individual server process status, including whether a process is running and its process ID. The `tabadmin status` command obtains its information by connecting to the Windows Service tabsvc.exe, which in turn queries the tabspawn executables for each process. Because of this, it can sometimes display different information for the server processes than the status table on the Maintenance page, which queries the processes directly.

**Examples**

```
tabadmin status
```

```
tabadmin status --verbose
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>--verbose</td>
<td></td>
<td>Returns a list of all the Tableau Server processes, their process IDs, and their status.</td>
</tr>
</tbody>
</table>

**See Also**

*Server Settings (General)* on page 1177

*Tableau Server Processes* on page 1240
**stop**

Stops all Tableau Server processes. To use `tabadmin stop`:

1. Open a command prompt as an administrator:

   ![Command Prompt](image)

   - Open `cmd.exe` as an administrator.
   - Type `cd "C:\Program Files\Tableau\Tableau Server\10.0\bin"`

2. Type the following:
   ```
   tabadmin stop
   ```

**validate**

Confirms whether your Tableau Server environment meets the minimum requirements for running Tableau Server.

**Example**

tabadmin validate

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>-skiptem</code></td>
<td></td>
<td>Skip validating that temporary IPv6</td>
</tr>
</tbody>
</table>
**verify_database**

Verifies that a backup of the PostgreSQL database that serves as the Tableau Server repository will restore successfully.

*Note: The verify_database command is available beginning with Tableau Server version 9.3.*

If you specify a backup file (.tsbak) as an option, the command restores the file to a temporary database in order to verify the backup. If you do not specify a backup file, a temporary backup of the running database is created and then restored to a temporary database. If verification fails, errors are displayed on the command line and are also logged in the `tabadmin.log` file. Until the errors are addressed, You cannot restore a .tsback file that fails verification. If verification of the database fails, contact Tableau Support for assistance.

*Note: A running PostgreSQL database can have errors that don’t impact use but would cause a failure when you tried to restore a backup. This means that you may be able to continue to use a running database, but you cannot back it up and restore the backup. As a best practice, verify your database before taking a backup (prior to an upgrade, for example).*

**Example**

Verify a backup file in the C:\backups\tableau folder named **tabserv.tsbak**:

```
tabadmin verify_database --file C:\backups\tableau\tabserv.tsbak
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>--file</td>
<td>FILE</td>
<td>Backup file to verify. If no file is specified, the command verifies the running PostgreSQL database by making a temporary backup of it.</td>
</tr>
<tr>
<td>-t</td>
<td>--temp</td>
<td>PATH</td>
<td>Location of the temporary folder to use while doing verification. The default is the Tableau temp folder. Temporary files are removed after verification completes.</td>
</tr>
</tbody>
</table>
See Also

backup on page 1262
Verify the Tableau Postgres Database on page 1148

**warmup**

Causes every VizQL server process to load the vizql DLL file, resulting in faster load times when server users first load views. Administrators can run this command, or script it to be run, after a Tableau Server restart.

**Example**

tabadmin warmup

**whitelist_webdataconnector**

Add a web data connector to the safe list (whitelist) for an installation of Tableau Server. Users who create workbooks can then reference the web data connector as a data source. Tableau Server only uses the safe list if the `tabadmin webdataconnector.whitelist.mode` property is set to fixed or mixed.

The safe list includes the URLs of hosted web data connectors that you have vetted and that you want to allow Tableau Server users to connect to. For more information, see Web Data Connectors in Tableau Server on page 899.

**Important:** Before you add a web data connector to the safe list, check the functionality of the connector. For more information, see Testing and Vetting Web Data Connectors on page 906.

The URLs for connectors in the safe list are case sensitive. When a connector is on the safe list, data sources and workbooks that are associated with the connector can be refreshed on Tableau Server. After you make changes to the safe list, you must restart Tableau Server for the changes to take effect.

Optionally, you can also configure a secondary safe list for each connector on the safe list. This secondary safe list determines which domains the connector can send requests to and receive requests from. You might want to set the secondary safe list to ensure that connectors do not send information to untrusted domains. If you do not specify a secondary safe list for a connector, then that connector can connect to any domain.
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>--add</td>
<td>Connector URL</td>
<td>Add the URL of a hosted web data connector to the safe list.</td>
</tr>
<tr>
<td>-d</td>
<td>--delete</td>
<td>Connector URL</td>
<td>Remove the URL of a web data connector from the safe list.</td>
</tr>
<tr>
<td>-l</td>
<td>--list</td>
<td></td>
<td>List all of the connector URLs on the safe list.</td>
</tr>
<tr>
<td>-r</td>
<td>--reset</td>
<td></td>
<td>Clear the safe list.</td>
</tr>
<tr>
<td>-s</td>
<td>--add_secondary whitelist</td>
<td>Connector URL and comma separated list of URLs the connector can make requests to</td>
<td>Optional. Add a list of domains that a particular connector can make requests to. The first argument is a connector that has already been added to the safe list. The second argument is a comma separated list of domains or resources. You can include regular expressions in the list of domains. If you do not specify a secondary safe list for a connector, then that connector can connect to any domain.</td>
</tr>
<tr>
<td>-p</td>
<td>--print_secondary whitelist</td>
<td>Connector URL</td>
<td>Print the secondary safe list for a given connector URL.</td>
</tr>
<tr>
<td>-w</td>
<td>--reset_secondary whitelist</td>
<td>Connector URL</td>
<td>Clear the secondary safe list for a given connector URL.</td>
</tr>
</tbody>
</table>

**Note:** If you use Windows PowerShell to run these commands, you might need to include quotes around the argument. For example, when you add a secondary safe list with regular expressions, you need to include quotes.

**Examples**

```bash
tabadmin whitelist_webdataconnector -a https://example.com/myconnector.html

tabadmin whitelist_webdataconnector -a http://example.com:8080/myconnector.html
```
tabadmin whitelist_webdataconnector -d https://example.com/myconnector.html


tabadmin whitelist_webdataconnector -p https://example.com/myconnector.html

See Also

Web Data Connectors in Tableau Server on page 899

Tableau webdataconnector page on GitHub

ziplogs

Creates an archive (.zip) containing Tableau Server log files, without removing the log files themselves. If you are running a Tableau Server cluster, log files from worker servers are included in the archive that’s created.

Examples

Create an archive in the Tableau Server bin directory named logs.zip:

tabadmin ziplogs

Create an archive in the Tableau Server bin directory named mylogs.zip:

tabadmin ziplogs mylogs.zip

Create an archive in the Tableau Server bin directory named mylogs.zip that includes logs dated January 31, 2014 up to the present, excluding earlier logs:

tabadmin ziplogs -d 01/31/2014 mylogs.zip

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>--with-net-stat-info</td>
<td></td>
<td>Include information about the server environment in the .zip file.</td>
</tr>
<tr>
<td>-p</td>
<td>--with-postgresql-data</td>
<td></td>
<td>Include data from Tableau Server's PostgreSQL database. If Tableau Server is stopped, make a copy of the psql\data</td>
</tr>
</tbody>
</table>
## Archive Logs on Command Line (tabadmin)

Use the table below to learn more about Tableau Server options you can configure using the `set` command. See Tableau Server Ports on page 1244 for a complete list of ports.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api.server.enabled</td>
<td>true</td>
<td>Allows access to the Tableau Server REST API. By default, this functionality is enabled.</td>
</tr>
<tr>
<td>auditing.enabled</td>
<td>true</td>
<td>Allows access to the PostgreSQL (Tableau Server's own database) historical auditing tables. See Collect Data with the Tableau Server Repository on page 1117 for details.</td>
</tr>
</tbody>
</table>

### See Also

- Work with Log Files on page 1179
- Archive Logs on Command Line (tabadmin) on page 1189

### tabadmin set options

Use the table below to learn more about Tableau Server options you can configure using the `set` command. See Tableau Server Ports on page 1244 for a complete list of ports.

<table>
<thead>
<tr>
<th>Option</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-l</td>
<td>--with-latest-dump</td>
<td>Limit the included log files to only the most recent ones to help reduce file size. By default, the 10 most recent log files are included.</td>
</tr>
<tr>
<td>-f</td>
<td>--force</td>
<td>Overwrites the existing log file of the same name.</td>
</tr>
<tr>
<td>-d</td>
<td>--minimumdate [mm/dd/yyyy]</td>
<td>Log files with this date, up to the present, are included in the .zip file. Logs dated earlier are excluded from the file. If not specified, up to seven days worth of data is included.</td>
</tr>
<tr>
<td>-a</td>
<td>--all</td>
<td>Include all log files in the .zip file. Data from Tableau Server's PostgreSQL database is still excluded.</td>
</tr>
</tbody>
</table>

### See Also

- Work with Log Files on page 1179
- Archive Logs on Command Line (tabadmin) on page 1189

### tabadmin set options

Use the table below to learn more about Tableau Server options you can configure using the `set` command. See Tableau Server Ports on page 1244 for a complete list of ports.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api.server.enabled</td>
<td>true</td>
<td>Allows access to the Tableau Server REST API. By default, this functionality is enabled.</td>
</tr>
<tr>
<td>auditing.enabled</td>
<td>true</td>
<td>Allows access to the PostgreSQL (Tableau Server's own database) historical auditing tables. See Collect Data with the Tableau Server Repository on page 1117 for details.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>backgrounder.extra_timeout_in_seconds</td>
<td>1800</td>
<td>The number of seconds beyond the setting in backgrounder.querylimit before a background task is canceled. This setting makes sure that tasks do not hold up subsequent jobs if they are stalled. The setting applies to processes listed in backgrounder.timeout_tasks. To disable backgrounder timeouts, set the value of backgrounder.extra_timeout_in_seconds to &quot;&quot; (an empty string).</td>
</tr>
<tr>
<td>backgrounder.failure_threshold_for_run_prevention</td>
<td>5</td>
<td>The number of consecutive failures of a subscription or extract job before that job is suspended. Suspending continuously failing jobs helps preserve backgrounder resources for other jobs. Note: To reenable a suspended job, click Try again from the alert menu, or republish the data source or a workbook using the data source, or change the connection properties of the data source.</td>
</tr>
<tr>
<td>backgrounder.querylimit</td>
<td>7200</td>
<td>Longest allowable time, in seconds, for completing a single extract refresh task or subscription task. 7200 seconds = 2 hours. Note: If a background task reaches this time limit, it may continue to run for an additional several minutes while being canceled.</td>
</tr>
<tr>
<td>backgrounder.reset_schedules_on_startup</td>
<td>true</td>
<td>Controls when to run background tasks that were scheduled to run at a time when the server was stopped. When set to true (the default), tasks are run at their next scheduled time. When set to false, all tasks that were scheduled to run when the server was stopped are run, simultaneously, at server startup, including times when the Tableau Server backup file (.tsbak) is restored.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>backgrounder.send_email_on_refresh_failure</td>
<td>true</td>
<td>Controls whether extract refresh alerts are enabled for all sites on the server. By default alerts are enabled. To disable extract refresh alerts for all sites on a server, set this to false. Extract alerts can be enabled or disabled on a site basis by site administrators in site settings, or at the user level in user settings.</td>
</tr>
<tr>
<td>backgrounder.sort_jobs_by_run_time_history_observable_hours</td>
<td>-1</td>
<td>Controls the time window used when determining duration of the last full extract job. Tableau Server can sort full extract refresh jobs so they are executed based on the duration of their &quot;last run,&quot; executing the fastest full extract refresh jobs first. The &quot;last run&quot; duration of a particular job is determined from a random sample of a single instance of the full extract refresh job in last (&lt;n&gt;) hours. Full extract jobs are then prioritized to run in order from shortest to longest based on their &quot;last&quot; run duration. By default this is sorting is disabled (-1). If enabling this, the suggested value is 36 (hours).</td>
</tr>
<tr>
<td>backgrounder.sort_jobs_by_type_schedule_boundary_heuristics_milliSeconds</td>
<td>60000</td>
<td>Controls the time window that identifies backgrounder jobs which are determined to have the same scheduled start time. The backgrounder process orders work that is scheduled at the same time to be executed by job type, running the fastest category of jobs first: Subscriptions, then Incremental Extracts, then Full Extracts. Jobs are batched to determine which jobs are scheduled at the “same time”. A value 60,000 milliseconds (the default) indicates jobs for schedules starting within a 1 minute window should be classified in the same batch and so are ordered by type within that batch.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>backgrounder.subscription_image_caching</td>
<td>true</td>
<td>Controls whether backgrounder will cache images that are generated for subscriptions. Cached images do not have to be regenerated each time so caching improves subscription performance. By default image caching is enabled. To disable image caching for all sites on a server, set this to false.</td>
</tr>
<tr>
<td>backgrounder.timeout_tasks</td>
<td>refreshextracts, incrementextracts, subscription_notify, single_subscription_notify</td>
<td>The list of tasks that can be canceled if they run longer than the combined values in backgrounder.querylimit and backgrounder.extra_timeout_in_seconds. The list of tasks is delimited with commas. The default list represents all the possible values for this setting.</td>
</tr>
<tr>
<td>clustercontroller.pgsql.failover</td>
<td>true</td>
<td>In a high availability environment, controls whether failover of the PostGRES repository occurs automatically (the default). When set to false, failover to the passive repository only occurs when you to run the failoverrepository command.</td>
</tr>
<tr>
<td>clustercontroller.zk_ses-sion_timeout_ms</td>
<td>300000</td>
<td>The length of time, in milliseconds, that Cluster Controller will wait for the Coordination Service (ZooKeeper), before determining that failover is required.</td>
</tr>
<tr>
<td>dataengine.port</td>
<td>27042</td>
<td>Port that the data engine runs on.</td>
</tr>
<tr>
<td>dataserver.port</td>
<td>9700</td>
<td>Port that the data server runs on.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DataServer-RefreshMetadataPer-Session</td>
<td>false</td>
<td>Determines whether Tableau Server will make additional queries to get updated schema data for a published data source when there have been changes in the underlying schema structure. This is disabled by default for performance reasons, and there is a delay in the display of schema changes. If you want changes in the schema of a live published data source to be reflected quickly, or if you see errors (for example, &quot;An error occurred while communicating with the data source: Invalid column name. Statement could not be prepared.&quot;) set this to True. When set to true, Tableau Server makes additional queries to update the schema.</td>
</tr>
<tr>
<td>features.DesktopReporting</td>
<td>false</td>
<td>Controls whether Desktop License Reporting is enabled on the server. When set to false (the default), no Administrative Views related to desktop licenses are available. Set this to true to enable license reporting and make license usage and expiration Administrative Views visible on the Server Status page.</td>
</tr>
</tbody>
</table>
| gateway.http.request_size_limit            | 16380         | The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.  

**A low value for** gateway.http.request_size_limit **may result in authentication errors.** Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production. |
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>tomcat.http.maxrequestsize</strong></td>
<td></td>
<td>We recommend setting <code>tomcat.http.maxrequestsize</code> option to the same value that you set for this option.</td>
</tr>
<tr>
<td><strong>gateway.public.host</strong></td>
<td>Name of the machine</td>
<td>The name (URL) of the server, used for external access to Tableau Server. If Tableau Server is configured to work with a proxy server or external load balancer, it is the name entered in a browser address bar to reach Tableau Server. For example, if Tableau Server is reached by entering <code>tableau.example.com</code>, the name for <code>gateway.public.host</code> is <code>tableau.example.com</code>.</td>
</tr>
<tr>
<td><strong>gateway.public.port</strong></td>
<td>80 (443 if SSL)</td>
<td>Applies to proxy server environments only. The external port the proxy server listens on.</td>
</tr>
<tr>
<td><strong>gateway.slow_post_protection.enabled</strong></td>
<td>false</td>
<td>Enabling this can provide some help in protecting against slow POST (Denial-of-Service) attacks by timing out POST requests that transfer data at extremely slow rates. <strong>Note:</strong> This will not eliminate the threat of such attacks, and could have the unintended impact of terminating slow connections.</td>
</tr>
<tr>
<td><strong>gateway.timeout</strong></td>
<td>1800</td>
<td>Longest amount of time, in seconds, that the gateway will wait for certain events before failing a request (1800 seconds = 30 minutes).</td>
</tr>
<tr>
<td><strong>gateway.trusted</strong></td>
<td>IP address of proxy server machine</td>
<td>Applies to proxy server environments only. The IP address(es) or host name(s) of the proxy server.</td>
</tr>
<tr>
<td><strong>gateway.trusted_hosts</strong></td>
<td>Alternate name(s) of proxy server</td>
<td>Applies to proxy server environments only. Any alternate host name(s) for the proxy server.</td>
</tr>
<tr>
<td><strong>install.firewall.allowedprograms.manage</strong></td>
<td>true</td>
<td>Controls whether Tableau Server can add firewall rules. When set to <code>true</code> (the default), Tableau Server will add new firewall rules to</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>allow</td>
<td></td>
<td>allow its processes to make connections through Windows Firewall. Change this to false if you want to manage all firewall rules yourself and do not want Tableau Server to add new rules.</td>
</tr>
<tr>
<td>java.heap.size</td>
<td>128m</td>
<td>Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.</td>
</tr>
<tr>
<td>monitor-dataengine.connection.timeout</td>
<td>30000</td>
<td>The length of time, in milliseconds, that Cluster Controller will wait for the data engine, before determining that a connection timeout occurred. The default is 30,000 milliseconds (30 seconds).</td>
</tr>
<tr>
<td>native_api.-connection.limit.&lt;connection class&gt;</td>
<td></td>
<td>Set parallel query limit for the specified data source (connection class). This overrides the global limit for the data source. For information about specific connection class strings, see the Tableau Knowledge Base.</td>
</tr>
<tr>
<td>native_api.-connection.limit.globallimit</td>
<td>16</td>
<td>Global limit for parallel queries. Default is 16 except for Amazon Redshift which has a default of 2. For information about configuring parallel queries in Tableau Server, see the Tableau Knowledge Base.</td>
</tr>
<tr>
<td>postgresql.port</td>
<td>8060</td>
<td>Port that PostgreSQL listens on.</td>
</tr>
<tr>
<td>postgresql.verify_restore.port</td>
<td>8061</td>
<td>Port used to verify the integrity of the PostgreSQL database. See Verify the Tableau Postgres Database on page 1148 for more information.</td>
</tr>
<tr>
<td>rsync.timeout</td>
<td>600</td>
<td>Longest allowable time, in seconds, for completing file synchronization (600 seconds = 10 minutes). File synchronization occurs as part of configuring high availability, or moving the data engine and repository processes.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>schedules.display_schedule_description_as_name</td>
<td>false</td>
<td>Controls whether a schedule name displays when creating a subscription or extract refresh (the default), or the &quot;schedule frequency description&quot; name describing the time and frequency of the schedule displays. To configure Tableau Server to display timezone-sensitive names for schedules, set this value to true. When true, the &quot;schedule frequency description&quot; is also displayed after the schedule name on the schedule list page.</td>
</tr>
<tr>
<td>schedules.display_schedules_in_client_timezone</td>
<td>true</td>
<td>Shows the &quot;schedule frequency description&quot; in the timezone of the user when true (uses the client browser timezone to calculate the &quot;schedule frequency description&quot;).</td>
</tr>
<tr>
<td>server.log.level</td>
<td>info</td>
<td>The logging level for logs written to ProgramData\Tableau\Tableau Server-data\tabsvc\logs\vizqlserver\Logs*.txt. Set to debug for more information. When set to debug, logging is set to pre-8.2 verbosity. Using the debug setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 1197 for more information.</td>
</tr>
<tr>
<td>service.jmx_enabled</td>
<td>false</td>
<td>Setting to true enables JMX ports for optional monitoring and troubleshooting. See Enable the JMX Ports on page 600 for details.</td>
</tr>
<tr>
<td>service.max_procs</td>
<td># of processes</td>
<td>Maximum number of server processes.</td>
</tr>
<tr>
<td>service.port_remappping.enabled</td>
<td>true</td>
<td>Determines whether or not Tableau Server will attempt to dynamically remap ports when the default or configured ports are unavailable.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Setting to <code>false</code> disables dynamic port remapping. See Tableau Server Ports on page 1244 for more information.</td>
</tr>
<tr>
<td>session.ipsticky</td>
<td><code>false</code></td>
<td>Makes client sessions valid only for the IP address that was used to sign in. If a request is made from an IP address different from that associated with the session token, the session token is considered invalid. In certain circumstances—for example, when Tableau Server is being accessed by computers with known and static IP addresses—this setting can yield improved security.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note</strong>: Consider carefully whether this setting will help your server security. This setting requires that the client have a unique IP address and an IP address that stays the same for the duration of the session. For example, different users who are behind a proxy might look like they have the same IP address (namely, the IP address of the proxy); in that case, one user might have access to another user’s session. In other circumstances, users might have a dynamic IP address, and their address might change during the course of the session. If so, the user has to sign in again.</td>
</tr>
<tr>
<td>solr.rebuild_index_timeout</td>
<td>3600</td>
<td>When Tableau Server is upgraded or when a .tsbak file is restored, the background task rebuilds the search index. This setting controls the timeout setting for that task (3600 seconds = 60 minutes).</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ssl.client_certificate_login.fallback_to_password</td>
<td>false</td>
<td>Specifies if Tableau Server should use user name and password for authentication if SSL authentication fails. Valid options are false (the default) and true. By default, when configured for mutual SSL, Tableau Server does not allow a connection if SSL authentication fails. Set this to true to allow user name and password authentication if SSL authentication fails.</td>
</tr>
</tbody>
</table>
| ssl.client_certificate_login.mapping_strategy | UPN or LDAP    | Specifies the method to be used for retrieving the user name from the certificate. Options are LDAP, UPN, or CN. The default depends on how Tableau Server is configured for user authentication:  
- When Tableau Server authentication is configured for Local Authentication, the default is UPN (User Principal Name).  
- When Tableau Server authentication is configured for Active Directory (AD), the default is LDAP (Lightweight Directory Access Protocol). CN (Common Name) is an option the administrator can set for either authentication type. |
<p>| ssl.revocation.file                          |               | Specifies the file path for an SSL CA Certificate Revocation List (CRL) file. Example: <code>tabadmin set ssl.revocation.file &quot;c:\Program Files\Tableau\Tableau Server\SSL\ca-bundle-client.crl</code> |
| subscriptions.enabled                       | false         | Controls whether subscriptions are configurable system-wide. See Manage Subscriptions on page 925.                                                                                                           |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriptions.timeout</td>
<td>1800</td>
<td>Longest allowable time, in seconds, for a single view in a workbook subscription task to be rendered before the task times out. This value applies separately to each view in the workbook, so the total length of time to render all the views in a workbook (the full subscription task) may exceed this timeout value. 1800 seconds = 30 minutes.</td>
</tr>
<tr>
<td>tomcat.http.maxrequestssize</td>
<td>16380</td>
<td>The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures. A low value for tomcat.http.maxrequestsizemay result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production. We recommend setting gateway.http.request_size_limit option to the same value that you set for this option.</td>
</tr>
<tr>
<td>tomcat.https.port</td>
<td>8443</td>
<td>SSL port for Tomcat (unused).</td>
</tr>
<tr>
<td>tomcat.server.port</td>
<td>8085</td>
<td>Port that tomcat listens on for shutdown messages.</td>
</tr>
<tr>
<td>vizportal.adsync.update_system_user</td>
<td>false</td>
<td>Specifies whether email addresses and display names of users are changed (even when changed in Active Directory) when an Active Directory group is synchronized in Tableau Server. To ensure that user email addresses</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| and display names are updated during syn-  |               | frowned, set `viz-  
| chronization, set `viz-portal.adsync.update_system_user` to `true`, and then restart the server.                                                                                                                                                                                                                                           |
| vizportal.csv_user_mgmt.index_site_users   | `true`        | Specifies whether indexing of site users is done user by user when importing or deleting users with a CSV file. When set to `true` (the default) indexing is done as each user is added or deleted. To delay the indexing of the site users until after the entire CSV file has been processed, set this to `false`. |
| vizportal.log.level                        | `info`        | The logging level for vizportal Java components. Logs are written to `ProgramData\Tableau\Tableau Server- 
|                                            |               | \data\t- 
<p>|                                            |               | absvc\logs\vizportal*.log. Set to <code>debug</code> for more information. Using the debug setting can significantly impact performance, so you should only use this setting when directed to do so by Tableau Support. See Change Logging Levels on page 1197 for more information. |
| vizqlserver.allow_insecure_scripts         | <code>false</code>       | Allows a workbook to be published to the server from Tableau Desktop, and to be opened from the server, even if the workbook contains SQL or R expressions that are potentially unsafe (for example, a SQL expression that could potentially allow SQL injection). When this setting is <code>false</code> (the default), publishing a workbook or opening it from the server results in an error message, and the workbook is blocked. You should set this value to <code>true</code> only if you want to use workbooks that contain SQL or R expressions that have been detected as potentially unsafe, and only if the |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>workbooks come from a safe source and you have verified that they do not contain an unsafe expression.</td>
</tr>
<tr>
<td>vizqlserver.browser.render</td>
<td>true</td>
<td>Views under the threshold set by vizqlserver.browser.render_threshold or vizqlserver.browser.render_threshold_mobile are rendered by the client web browser instead of by the server. See About Client-Side Rendering on page 1128 for details.</td>
</tr>
<tr>
<td>vizqlserver.browser.render</td>
<td>100</td>
<td>The default value (100) represents a high level of complexity for a view displayed on a PC. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the PC's web browser.</td>
</tr>
<tr>
<td>vizqlserver.browser.render</td>
<td>20</td>
<td>The default value (20) represents a high level of complexity for a view displayed on a tablet. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the tablet's web browser.</td>
</tr>
<tr>
<td>vizqlserver.clear_session_on_unload</td>
<td>false</td>
<td>Determines whether or not VizQL sessions are kept in memory when a user navigates away from a view or closes their browser. The default value (false) keeps sessions in memory. To close VizQL sessions on leaving a view or closing a browser, set this to true. See General Performance Guidelines on page 1087 for more information.</td>
</tr>
<tr>
<td>vizqlserver.geosearch_cache_size</td>
<td>5</td>
<td>Sets the maximum number of different geographic search locale/language data sets that can be loaded into server memory at the same</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>time. When the server receives a geographic search request for locale/language data set that is not in memory, it will load the set into memory. If loading the data set will exceed the specified limit, the least recently used locale/language data set is cleared from memory so the requested one can be loaded. The minimum value is 1. Each cache takes approximately 60 MB in memory (so if you set this to 10, the memory usage would be 600 MB (60 * 10)).</td>
</tr>
<tr>
<td>vizqlserver.log.level</td>
<td>info</td>
<td>The logging level for vizqlserver Java components. Logs are written to ProgramData\Tableau\Tableau Server-data\t- absvc\logs\vizqlserver*.log. Set to debug for more information. Using the debug setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 1197 for more information.</td>
</tr>
<tr>
<td>vizqlserver.port</td>
<td>9100</td>
<td>Base port for the VizQL servers.</td>
</tr>
<tr>
<td>vizqlserver.protect_sessions</td>
<td>true</td>
<td>When set to true (the default), prevents VizQL sessions from being reused after the original user signs out.</td>
</tr>
<tr>
<td>vizqlserver.querylimit</td>
<td>1800</td>
<td>Longest allowable time for updating a view, in seconds.</td>
</tr>
<tr>
<td>vizqlserver.rserve.host</td>
<td></td>
<td>Specifies an Rserve host. This setting, and the three settings immediately below, supports R functionality in workbooks. R is an open source software programming language and a software environment for statistical computing and graphics. In Tableau Desktop, you can</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>use a set of four functions</td>
<td></td>
<td>to pass R expressions to an Rserve server and obtain a result. If you upload a workbook that uses any of these functions, you should configure Tableau Server for an Rserve connection, by configuring this option and the three following. Otherwise, any worksheets that use R functionality will be unavailable. See Pass Expressions to External Services in the Tableau Desktop help for further details.</td>
</tr>
<tr>
<td>vizqlserver.rserve.port</td>
<td>6311</td>
<td>Specifies an Rserve port. This setting supports R functionality in workbooks.</td>
</tr>
<tr>
<td>vizqlserver.rserve.username</td>
<td></td>
<td>Specifies an Rserve username. This setting supports R functionality in workbooks. Not all Rserve hosts require a username and password.</td>
</tr>
<tr>
<td>vizqlserver.rserve.password</td>
<td></td>
<td>Specifies an Rserve password. This setting supports R functionality in workbooks. Not all Rserve hosts require a username and password.</td>
</tr>
<tr>
<td>vizqlserver.session.expiry.minimum</td>
<td>5</td>
<td>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.</td>
</tr>
<tr>
<td>vizqlserver.session.expiry.timeout</td>
<td>30</td>
<td>Number of minutes of idle time after which a VizQL session is discarded.</td>
</tr>
<tr>
<td>vizqlserver.showdownload</td>
<td>true</td>
<td>Controls the display of the Tableau Workbook option of the Download menu in views. When set to false, the Tableau Workbook option is unavailable.</td>
</tr>
<tr>
<td>vizqlserver.showshare</td>
<td>true</td>
<td>Controls the display of Share options in views.</td>
</tr>
</tbody>
</table>

**Note:** To let users control display with the "showShareOptions" JavaScript or...
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vizqlserver.trustedticket.log_level</td>
<td>info</td>
<td>The logging level for trusted authentication. The logs are written to ProgramData\Tableau\Tableau Server-data\tabsvc\logs\vizqlserver\vizql*.log. Set to debug for more information. Using the debug level can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels on page 1197 for more information.</td>
</tr>
<tr>
<td>vizqlserver.trustedticket.token_length</td>
<td>24</td>
<td>Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.</td>
</tr>
<tr>
<td>vizqlserver.trustedticket.use_deprecated_9digit_token</td>
<td>false</td>
<td>When set to true, tickets are 9 digits long (as in version 8.0 and earlier) and the setting vizqlserver.trustedticket.token_length is ignored.</td>
</tr>
<tr>
<td>vizqlserver.url_scheme_whitelist</td>
<td></td>
<td>Adds to the protocols to whitelist when using URL actions on views and dashboards. http, https, gopher, news, ftp, and mailto are whitelisted by default.</td>
</tr>
<tr>
<td>webdataconnector.enabled</td>
<td>true</td>
<td>When this setting is true, you can use tabadmin commands to manage web data connectors on the server, and web data connectors are included when you back up and restore the server. If the setting is false, web data connectors that are on the server are not included during backup and restore. For more</td>
</tr>
</tbody>
</table>

URL parameter, you must set vizqlserver.showshare to false.
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>web-data-connector.refresh.enabled</td>
<td>true</td>
<td>When this setting is true, the server supports doing refreshes for web data connector-based data sources. For more information, see Web Data Connectors in Tableau Server on page 899.</td>
</tr>
<tr>
<td>web-data-connector.whitelist.mode</td>
<td>mixed</td>
<td>Determines how Tableau Server can run web data connectors. Supported modes are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- local. Users can run connectors that have been imported to Tableau Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- fixed. Users can run connectors that are on a safe list (whitelist) of URLs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- mixed. Users can run imported connectors or connectors on the safe list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- insecure. Users can run any connector.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Important:</strong> Use the <code>insecure</code> option only for development and testing. Because connectors run custom code, running connectors that have not been vetted can pose a security threat.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information about how to add connectors to a safe list and import connectors, see Web Data Connectors in Tableau Server on page 899.</td>
</tr>
<tr>
<td>wgserver.audit_history_expiration_days</td>
<td>183</td>
<td>Specifies the number of days after which historical events records are removed from the PostgreSQL database (the Tableau Server database). See Collect Data with the Tableau Server Repository on page 1117 for details.</td>
</tr>
<tr>
<td>wgserv-</td>
<td>false</td>
<td>Controls whether or not Tableau Desktop</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>er.authentication.desktop_nosaml</td>
<td></td>
<td>uses SAML for authentication. Use this option when your IdP does not use forms-based authentication. Valid options are true and false. By default this is not set, so the behavior is equivalent to setting it to false. Set this to true to disable SAML authentication for Tableau Desktop.</td>
</tr>
<tr>
<td>wgserv-er.authentication.app_nosaml</td>
<td>false</td>
<td>Serves as the above setting for the Tableau Mobile app.</td>
</tr>
<tr>
<td>wgserv-er.authentication.restricted</td>
<td>false</td>
<td>Controls whether users can sign in to Tableau Server using a Tableau Server username and password. This setting is useful in scenarios where users normally sign in to the server using single sign-on (SSO), such as by using SAML, OpenID Connect, or Kerberos. In these cases, the user also has a Tableau Server username and password. If wgserv-er.authentication.restricted is set to true, only system administrators can sign in to Tableau Server using a username and password; all other users must sign in to the server using SSO. Setting wgserv-er.authentication.restricted to true also has the effect of restricting user access to command-line tools like tabcmd and tabconfig. These tools do not support SSO, and therefore require a user to sign in using a Tableau Server. If the setting is true, users who are not system administrator cannot use these command-line tools.</td>
</tr>
<tr>
<td>wgserver.change_owner.enabled</td>
<td>true</td>
<td>Controls whether the ownership of a workbook, data source or project can be changed. Other options include false and adminonly. See Manage Ownership on page 784 for details.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.clickjack_defense.enabled</td>
<td>true</td>
<td>When set to true, helps prevents a malicious person from &quot;clickjacking&quot; a Tableau Server user. In a clickjack attack, the target page is displayed transparently over a second page, and the attacker gets the user to click or enter information in the target page while the user thinks he or she is interacting with the second page. For more information, see <a href="#">Clickjack Protection on page 964</a>.</td>
</tr>
<tr>
<td>wgserver.domain.fqdn</td>
<td>value of %USERDOMAIN%</td>
<td>The fully qualified domain name of the Active Directory server to use.</td>
</tr>
<tr>
<td>wgserver.restrict_options_method</td>
<td>true</td>
<td>Controls whether Tableau Server accepts HTTP OPTIONS requests. If this option is set to true, the server returns HTTP 405 (Method Not Allowed) for HTTP OPTIONS requests.</td>
</tr>
<tr>
<td>wgserv-er.saml.idpattribute.username</td>
<td>username</td>
<td>Specifies the attribute used by the IdP for SAML authentication. The default is username. For more information, see <a href="#">SAML on page 1010</a>.</td>
</tr>
<tr>
<td>wgserver.saml.logout.enabled</td>
<td>true</td>
<td>Specifies whether SAML logout is enabled for Tableau Server. The default is true. This setting only applies if SAML authentication is enabled for Tableau Server.</td>
</tr>
<tr>
<td>wgserver.saml.logout.redirect_url</td>
<td></td>
<td>Specifies the post-logout landing page for SAML authentication. The default is the standard server sign-in page. You can specify an absolute or a relative URL. For more information, see <a href="#">SAML Requirements</a>.</td>
</tr>
<tr>
<td>wgserver.saml.maxassertiontime</td>
<td>3000</td>
<td>Specifies the maximum number of seconds, from creation, that an assertion is usable.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.sam-l.maxauthenticationage</td>
<td>7200</td>
<td>Specifies the maximum number of seconds allowed between user's authentication and processing of the AuthNResponse message.</td>
</tr>
<tr>
<td>wgserver.saml.responseskew</td>
<td>180</td>
<td>Sets the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed.</td>
</tr>
<tr>
<td>wgserver.session.apply_lifetime_limit</td>
<td>false</td>
<td>Controls whether there is a session lifetime for server sessions. Set this to <strong>true</strong> to configure a server session lifetime.</td>
</tr>
<tr>
<td>wgserver.session.lifetime_limit</td>
<td>1440</td>
<td>The number of minutes a server session lasts if a session lifetime is set. The default is 1440 minutes (24 hours). If <strong>wgserver.session.apply_lifetime_limit</strong> is <strong>false</strong> (the default) this is ignored.</td>
</tr>
<tr>
<td>wgserver.session.idle_limit</td>
<td>240</td>
<td>The number of minutes of idle time before a sign-in to the web application times out.</td>
</tr>
<tr>
<td>workerX.gateway.port</td>
<td>80 (443 if SSL)</td>
<td>External port that Apache listens on for workerX. worker0.gateway.port is Tableau Server’s external port. In a distributed environment, worker0 is the primary Tableau Server.</td>
</tr>
<tr>
<td>workerX.vizqlserver.procs</td>
<td># of processes</td>
<td>Number of VizQL servers.</td>
</tr>
<tr>
<td>workerX.vizqlserver.port</td>
<td>9100</td>
<td>Base port for the vizQL server on workerX.</td>
</tr>
<tr>
<td>zookeeper.config.dataLogDir</td>
<td></td>
<td>Specifies the directory and file path for ZooKeeper transaction logs. By default ZooKeeper transaction logs are written to the Tableau data directory (for example c:\Tableau\Tableau Server\data\tabsvc\zookeeper\0\data). Use this option to specify a different location.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
|        |               | The drive and path apply to all nodes in a cluster. The location will be created if it does not exist. The drive must exist and be writable on all nodes. This should not be a UNC path to a share. 
ZooKeeper recommends that transaction logs be written to a dedicated drive to optimize performance. 
Example: `tabadmin set zookeeper.config.dataLogDir "d:\Tableau\Tableau Server\zookeeper"` |

**Restore a Setting to its Default Value**

You can restore the default value for a Tableau Server configuration setting by doing the following:

1. **Stop the server.**
2. Still in the bin directory, restore the default value for a particular setting by typing the following:
   ```shell
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:
   ```shell
tabadmin set vizqlserver.session.expiry.timeout --default
   ```
Alternatively, you can use the shorter `-d` command. For example:
   ```shell
tabadmin set vizqlserver.querylimit -d
   ```
3. **Next, run the configure command:**
   ```shell
tabadmin configure
   ```
4. **Start the server.**
**tabcmd**

The tabcmd utility is one of the two command line tools that installs with Tableau Server (the other is tabadmin on page 1255). The commands provided through tabcmd can help you automate common tasks, such as publishing workbooks in batches and administering users and groups. The tabcmd utility installs in the Tableau Server bin folder (C:\Program Files\Tableau Server\10.0\bin), but you can install and run tabcmd on another machine as well. For more information, see the following topics.

**Install tabcmd**

By default, the tabcmd command line utility installs with Tableau Server to the server’s bin folder (for example, C:\Program Files\Tableau\Tableau Server\10.0\bin). You can run it from there. For administrative flexibility, you can also install it on other computers.

If you installed the tabcmd command line utility on computers that are not running Tableau Server and you are upgrading Tableau Server to a new major version (version 9.3 to version 10.0 for example), Tableau recommends you also upgrade standalone installations of tabcmd to avoid any potential incompatibilities between versions.

To install tabcmd on another machine:

1. Navigate to the extras folder on Tableau Server:
   
   C:\Program Files\Tableau\Tableau Server-\10.0\extras\TabcmdInstaller.exe

2. Copy TabcmdInstaller.exe to the computer where you want to install it.

3. Double-click TabcmdInstaller.exe to run it.

4. Follow the prompts to install tabcmd.

Because tabcmd is a command line tool, and due to some limitations with the Windows operating system, Tableau recommends that you install tabcmd in a folder named tabcmd at the root of the C:\ drive (C:\tabcmd).

**Note:** Running the tabcmd Setup program does not automatically add tabcmd to the Windows PATH variable, you will need to either explicitly call tabcmd using its full path or add its directory to the PATH variable.

**How to Use tabcmd**

The basic steps for using tabcmd are as follows:

1. Open the Command Prompt as an administrator.
2. Change to the Tableau Server bin folder.
For example:

```
cd C:\Program Files\Tableau\Tableau Server\10.0\bin
```

Or you can include the location in the command.

3. Run the tabcmd command.

When you use tabcmd, you must establish an authenticated server session. The session identifies the Tableau Server and the Tableau Server user running the session. You can start a session first, and then specify your command next, or you can start a session and execute a command all at once. If you are using tabcmd to perform more than one task, you must run each task one after the other (serially), rather than in parallel.

Commands (such as `login`) and the options (such as `-s, -u, etc.) are not case sensitive, but the values you provide (such as `p@ssw0rd` or `User@Example.com`) are case sensitive.

**Examples**

The following command demonstrates starting a session with the Tableau Server named `tabserver.myco.com`:

```
tabcmd login -s http://tabserver.myco.com -u admin -p p@ssw0rd!
```

The next example shows a command that deletes a workbook named `Sales_Workbook`:

```
tabcmd delete "Sales_Workbook"
```

Here’s how to accomplish all of the above with one command—note that you do not need `login` here:

```
tabcmd delete "Sales_Workbook" -s http://tabserver.myco.com -u admin -p p@ssw0rd!
```

A Tableau Server can run multiple sites. When a workbook is on the Default site of a multi-site server you don’t need to specify Default, the above command is sufficient. However, if the command applies to something on a site other than Default, you need to specify the site ID for that site (see `login` on page 1338). Here’s the same command for a workbook that’s on the West Coast Sales site (site ID `wsales`):

```
tabcmd delete "Sales_Workbook" -s http://tabserver.myco.com -t wsales -u admin -p p@ssw0rd!
```

The options `-s, -t, -u, and -p` are among the tabcmd global variables, which can be used with any command.

For more information, see [tabcmd Commands on page 1319](#).

**Status messages and logs**

When a command is successful, tabcmd returns a status code of zero. A full error message for non-zero status codes is printed to `stderr`. In addition, informative or progress messages may be printed to `stdout`. 

- 1316 -
A full log named `tabcmd.log` that includes debugging, progress, and error messages is written to `C:\Users\<username>\AppData\Local\Tableau`.

**tabcmd Global Options**

The table below shows the options that are used by all commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-h</td>
<td><code>--help</code></td>
<td></td>
<td>Displays the help for the command.</td>
</tr>
<tr>
<td>-c</td>
<td><code>--use-certificate</code></td>
<td></td>
<td>Use client certificate to sign in. Required when mutual SSL is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For more information, see <a href="#">Configure External SSL on page 972</a>.</td>
</tr>
<tr>
<td>-s</td>
<td><code>--server</code></td>
<td>Tableau Server URL</td>
<td>Required at least once to begin session.</td>
</tr>
<tr>
<td>-u</td>
<td><code>--user</code></td>
<td>Tableau Server username</td>
<td>Required at least once to begin session.</td>
</tr>
<tr>
<td>-p</td>
<td><code>--password</code></td>
<td>Tableau Server password</td>
<td>Required at least once to begin session. You can alternatively use the <code>-P</code> option.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>--password-file filename.txt</code></td>
<td>Allows the password to be stored in the given file rather than the command line for increased security.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>site, the Default site is assumed. Applies only to servers with multiple sites.</td>
</tr>
<tr>
<td>-x</td>
<td>--proxy</td>
<td>Host:Port</td>
<td>Uses the specified HTTP proxy.</td>
</tr>
<tr>
<td></td>
<td>--no-prompt</td>
<td></td>
<td>When specified, the command will not prompt for a password. If no valid password is provided the command will fail.</td>
</tr>
<tr>
<td></td>
<td>--no-proxy</td>
<td></td>
<td>When specified, an HTTP proxy will not be used.</td>
</tr>
<tr>
<td></td>
<td>--no-cert-check</td>
<td></td>
<td>When specified, tabcmd (the client) does not validate the server’s SSL certificate.</td>
</tr>
<tr>
<td></td>
<td>--[no-]cookie</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>--timeout</td>
<td>seconds</td>
<td>Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.</td>
</tr>
<tr>
<td></td>
<td>--</td>
<td></td>
<td>Specifies the end of</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where --430105/SHEET1 is a required value for the export command. tabcmd export --csv -f &quot;D:\export10.csv&quot; -- -430105/SHEET1</td>
</tr>
</tbody>
</table>

**tabcmd Commands**

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

- addusers (to group)
- creategroup
- createproject
- createsite
- createsiteusers
- createusers
- delete workbook-name or datasource-name
- deletegroup
- deleteproject
addusers group-name

Adds users to the specified group.

Example

tabcmd addusers "Development" --users "users.csv"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--users</td>
<td>filename.csv</td>
<td>Add the users in the given file to the specified group. The file should be a simple list with one user name per line. User names are not case sensitive. The users</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-n</td>
<td>--name</td>
<td>name</td>
<td>Specify the name of the project that you want to create.</td>
</tr>
<tr>
<td>-d</td>
<td>--description</td>
<td>description</td>
<td>Specify a description for the project.</td>
</tr>
</tbody>
</table>

creategroup group-name

Creates a group. Use addusers (for local groups) and syncgroup (for Active Directory groups) commands to add users after the group has been created.

**Example**

tabcmd creategroup "Development"

createproject project-name

Creates a project.

**Example**

tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing quarterly sales reports."

createsite site-name

Creates a site.

**Examples**
Create a site named West Coast Sales. A site ID of WestCoastSales will be automatically created, the site will have no storage quota limit, and site administrators will be able to add and remove users:

```
tabcmd createsite "West Coast Sales"
```

Create a site named West Coast Sales with a site ID of wsales:

```
tabcmd createsite "West Coast Sales" -r "wcoast"
```

Prevent site administrators from adding users to the site:

```
tabcmd createsite "West Coast Sales" --no-site-mode
```

Set a storage quota, in MB:

```
tabcmd createsite "West Coast Sales" --storage-quota 100
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>--url</td>
<td>site ID</td>
<td>Used in URLs to specify the site. Different from the site name.</td>
</tr>
<tr>
<td>--user-quota</td>
<td></td>
<td>number of users</td>
<td>Maximum number of users that can be added to the site.</td>
</tr>
<tr>
<td>--[no-]site-mode</td>
<td></td>
<td></td>
<td>Allow or deny site administrators the ability to add users to or remove users from the site.</td>
</tr>
<tr>
<td>--storage-quota</td>
<td></td>
<td>number of MB</td>
<td>In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.</td>
</tr>
</tbody>
</table>

createsiteusers filename.csv

Adds users to a site, based on information supplied in a comma-separated values (CSV) file. If the user is not already created on the server, the command creates the user before adding that user to the site.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, role, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see CSV Import File Guidelines on page 810. As an alternative to including role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the --role option.
By default, users are added to the site that you are logged in to. To add users to a different site, include the global `--site` option and specify that site. (You must have permissions to create users on the site you specify.)

If the server contains multiple sites, you cannot assign the `ServerAdministrator` role to a user by using the `createsiteusers` command. (Use `createusers` instead.) If you specify the `ServerAdministrator` role for the `role` option, the command returns an error. If the CSV file includes `System` as value for administrator, the value is ignored and the user is assigned the `Unlicensed` role. However, if the server contains only one site (the default site), you can assign the `ServerAdministrator` role or specify system for the administrator value; in that case, the `createsiteusers` command works like the `createusers` command.

By default, this command creates users using a synchronous operation (it waits for all operations to complete before proceeding). You can use the `--no-wait` option to specify an asynchronous operation.

**Local authentication**

If the server is configured to use local authentication, the information in the CSV file is used to create users.

**Active Directory authentication**

If the server is configured to use Active Directory authentication, user information is imported from Active Directory to the server. In that case, any password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not guaranteed to be unique across domains, therefore you must include the domain as part of the user name (for example, `example\Adam` or `adam@example.com`.

While these can be sent either as `domain/username` or `username@domain.com`, we recommend using the `domain/username` format. See User Management in Active Directory Deployments on page 1251 for more information.

**Example**

```
tabcmd createsiteusers "users.csv" --role "Interactor"
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--admin-type</td>
<td>--admin-type</td>
<td>Site or None</td>
<td>(Deprecated. Use the <code>--role</code> option instead.) Assigns or removes the site administrator right for any user who does not already have an administrator</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>setting in the CSV file. The default is None for new users and unchanged for existing users. If the server contains multiple sites; system admin-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>istrators cannot be created or demoted using createsiteusers. (Use createusers instead.)</td>
</tr>
<tr>
<td>--complete</td>
<td></td>
<td></td>
<td>Requires that all rows be valid for any change to succeed. This is the default setting.</td>
</tr>
<tr>
<td>--license</td>
<td>Interactor, Viewer, or Unlicensed</td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Specifies the license level for any user who does not already have a license level setting in the CSV file. The default is Unlicensed for new users and unchanged for existing users.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting in Tableau Server 9.0.</td>
</tr>
<tr>
<td>--no-complete</td>
<td></td>
<td></td>
<td>Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.</td>
</tr>
<tr>
<td>--no-pub-</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>lisher</td>
<td></td>
<td></td>
<td>option instead.) Disallows publishing rights for any users who do not already have a publisher setting in the CSV file. This is a default value for new users.</td>
</tr>
<tr>
<td>--nowait</td>
<td></td>
<td></td>
<td>Do not wait for asynchronous jobs to complete.</td>
</tr>
<tr>
<td>--publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Assigns publishing rights for any user who does not already have a publisher setting in the CSV file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.</td>
</tr>
<tr>
<td>-r</td>
<td>--role</td>
<td>ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed</td>
<td>Specifies a site role for any user who does not already have a role specified in the CSV file. The default is Unlicensed for new users and unchanged for existing users. If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.</td>
</tr>
</tbody>
</table>

**Note:** You cannot assign the ServerAdministrator or role if the server has
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--silent-progress</td>
<td></td>
<td>Do not display progress messages for the command.</td>
</tr>
</tbody>
</table>

createusers filename.csv

Create users in Tableau Server, based on information supplied in a comma-separated values (CSV) file.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, role, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see [CSV Import File Guidelines on page 810](#). As an alternative to including role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the --role option.

If the server has only one site (the default site), the user is created and added to the site. If the server has multiple sites, the user is created but is not added to any site. To add users to a site, use createsiteusers.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

**Local authentication**

If the server is configured to use local authentication, the information in the CSV file is used to create users.
Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory to the server. In that case, any password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not guaranteed to be unique across domains, therefore you must include the domain as part of the user name (for example, example\Adam or adam@example.com).

While these can be sent either as domain/username or username@domain.com, we recommend using the domain/username format. See User Management in Active Directory Deployments on page 1251 for more information.

Example

tabcmd createusers "users.csv" --role "ServerAdministrator"

tabcmd createusers "users.csv"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--admin-type</td>
<td>Site or None</td>
<td>(Deprecated. Use the --role option instead.) Assigns or removes the site administrator right for any user who does not already have an administrator setting in the CSV file. The default is None for new users and unchanged for existing users.</td>
<td></td>
</tr>
<tr>
<td>--complete</td>
<td></td>
<td>Requires that all rows be valid for any change to succeed. This is the default setting.</td>
<td></td>
</tr>
<tr>
<td>--license</td>
<td>Interactor, Viewer, or Unlicensed</td>
<td>(Deprecated. Use the --role option instead.) Specifies the license level for any user who does not already have a license level setting in the CSV file. The default is Unlicensed for new users and unchanged for existing users.</td>
<td></td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting with Tableau Server 9.0.</td>
</tr>
<tr>
<td>--no-complete</td>
<td></td>
<td></td>
<td>Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.</td>
</tr>
<tr>
<td>--no-publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Disallows publishing rights for any users who do not already have a publisher setting in the CSV file. This is a default value for new users.</td>
</tr>
<tr>
<td>--nowait</td>
<td></td>
<td></td>
<td>Do not wait for asynchronous jobs to complete.</td>
</tr>
<tr>
<td>--publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Use the --role option instead.) Assigns publishing rights for any user who does not already have a publisher setting in the CSV file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.</td>
</tr>
<tr>
<td>-r --role</td>
<td>ServerAdministrator, SiteAdministrator, Publisher, Interactor,</td>
<td></td>
<td>Specifies a role for any user who does not already have a role specified in the CSV file.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed</td>
<td></td>
<td>The default is Unlicensed for new users and unchanged for existing users. On a multi-site server, the command does not assign the user to a site. Therefore, the only roles that the command will assign are ServerAdministrator and Unlicensed. In that case, if you specify a different role (like Publisher or Viewer), the command assigns the Unlicensed role. On a single-site server, the user is created and added to the default site using the role that you specify. If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user. <strong>Note:</strong> If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.</td>
</tr>
<tr>
<td></td>
<td>--silent--</td>
<td></td>
<td>Do not display progress mes-</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>sages for the command.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

`delete workbook-name or datasource-name`  
Deletes the specified workbook or data source from the server.  
This command takes the name of the workbook or data source as it is on the server, not the file name when it was published.  
**Example**  
`tabcmd delete "Sales_Analysis"`

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-r</code></td>
<td><code>-project</code></td>
<td>Project name</td>
<td>The name of the project containing the workbook or data source you want to delete. If not specified, the “Default” project is assumed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>-workbook</code></td>
<td>Workbook name</td>
<td>The name of the workbook you want to delete.</td>
</tr>
<tr>
<td></td>
<td><code>-datasource</code></td>
<td>Data source name</td>
<td>The name of the data source you want to delete.</td>
</tr>
</tbody>
</table>

`deletegroup group-name`  
Deletes the specified group from the server.  
**Example**  
`tabcmd deletegroup "Development"`

`deleteproject project-name`  
Deletes the specified project from the server.  
**Example**  
`tabcmd deleteproject "Designs"`

`deletesite site-name`  
Deletes the specified site from the server.
Example
tabcmd deletesite "Development"

deletesiteusers filename.csv
Removes users from from the site that you are logged in to. The users to be removed are specified in a file that contains a simple list of one user name per line. (No additional information is required beyond the user name.)

By default, if the server has only one site, or if the user belongs to only one site, the user is also removed from the server. On a Tableau Server Enterprise installation, if the server contains multiple sites, users who are assigned the role of Server Administrator are removed from the site but are not removed from the server.

If the user owns content, the user's role is change to Unlicensed, but the user is not removed from the server or the site. The content is still owned by that user. To remove the user completely, you must change the owner of the content and then try removing the user again.

If the user was imported from Active Directory, the user is removed from the site and possibly from the server. However, the user is not deleted from Active Directory.

Example
tabcmd deletesiteusers "users.csv"

deleteusers filename.csv
Deletes the users listed in the specified comma-separated values (CSV) file. The CSV file should contain a simple list of one user name per line.

Example
tabcmd deleteusers "users.csv"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c</td>
<td>--[no-- -]complete</td>
<td></td>
<td>When set to --complete this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.</td>
</tr>
</tbody>
</table>

editdomain
Changes the nickname or full domain name of an Active Directory domain on the server. A domain “nickname” is the Windows NetBIOS domain name.
You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to sign in. However, if the user name that you are currently signed in with exists in both the current domain and the new domain, you can modify the full name for the current domain.

Review User Management in Active Directory Deployments on page 1251 to understand how multiple domains, domain name mapping, and user names interact with Tableau Server.

To see a list of domains, use listdomains.

Examples

tabcmd editdomain --id 2 --nickname "new-nickname"
tabcmd editdomain --id 3 --name "new-name"

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--id</td>
<td>Domain ID</td>
<td>The ID of domain to change. To get a list of domain IDs, use listdomains.</td>
</tr>
<tr>
<td>--name</td>
<td>Domain name</td>
<td>The new name for the domain.</td>
</tr>
<tr>
<td>--nickname</td>
<td>Domain nick-name</td>
<td>The new nickname for the domain.</td>
</tr>
</tbody>
</table>

editsite site-name

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

Examples

tabcmd editsite wc_sales --site-name "West Coast Sales"
tabcmd editsite wc_sales --site-id "wsales"
tabcmd editsite wsales --status ACTIVE
tabcmd editsite wsales --user-quota 50

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--site-name</td>
<td>Name to change the site to</td>
<td>The name of the site that's displayed.</td>
</tr>
<tr>
<td>--site-id</td>
<td>The site ID to</td>
<td>Used in the URL to uniquely identify the site.</td>
</tr>
</tbody>
</table>
## Option (long) | Argument | Description
--- | --- | ---
 | change the site to |  |  
--user-quota | Number of users | Maximum number of users who can be members of the site.  
--[no-]site-mode |  | Allow or prevent site administrators from adding users to the site.  
--status | ACTIVE or SUSPENDED | Activate or suspend a site.  
--storage-quota | Number of MB | In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.  

### export

Exports a view or workbook from Tableau Server and saves it to a file. This command can also export just the data used for a view.

Note the following when you use this command:

- **Permissions**: To export, you must have the Export Image permission. By default, this permission is Allowed or Inherited for all roles, although permissions can be set per workbook or view.

- **Exporting data**: To export just the data for a view, use the --csv option. This exports the summary data used in a view to a .csv file.

- **Specifying the view, workbook, or data to export**: You specify this using the "workbook/view" string as it appears in the URL for the workbook or view, not using its "friendly name," and excluding the :iid=<n> session ID at the end of the URL. For example, to export the Tableau sample view Investment Growth from the Finance workbook, you would use the string Finance/InvestmentGrowth, not Finance/Investment Growth, or Finance/InvestmentGrowth?:iid=1. Use -t <site_id> if the server is running multiple sites and the view or workbook is on a site other than Default.

To export a workbook, you still include a valid view in the string you use. Using the above example, to export the Finance workbook, you would use the string Finance/InvestmentGrowth. Finally, to export a workbook, it must have been published with Show Sheets as Tabs selected in the Tableau Desktop Publish dialog box.

- **The saved file's format**: Your format options depend on what's being exported. A workbook can only be exported as a PDF using the --fullpdf argument. A view can
be exported as a PDF (--pdf) or a PNG (--png).

- **The saved file's name and location** (optional): If you don't provide a name, it will be derived from the view or workbook name. If you don't provide a location, the file will be saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

  Note: You must include a file name extension such as .csv or .pdf. The command does not automatically add an extension to the file name that you provide.

- **Dashboard web page objects not included in PDF exports**: A dashboard can optionally include a web page object. If you are performing an export to PDF of a dashboard that includes a web page object, the web page object won't be included in the PDF.

- **Non-English characters and PDF exports**: If you are exporting a view or workbook with a name that includes a non-English characters you need to URL encode the character.

  For example if your command includes the city Zürich, you need to URL encode it as Z%C3%BCrich:

  ```
  tabcmd export "/Cities/SHEET1?locationCity=Z%C3%BCrich" -fullpdf
  ```

**Clearing the Cache to Use Real-Time Data**

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using `tabcmd` with your own scripting and the refresh URL parameter is being used a great deal, this can have a negative impact on performance. It's recommended that you use refresh only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

**Examples**

**Views**

```
tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report.csv"
```

```
tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"
```

```
tabcmd export "Finance/InvestmentGrowth" --png
```

```
tabcmd export "Finance/InvestmentGrowth?:refresh=yes" --png
```

**Workbooks**
**tabcmd export "Q1Sales/Sales_Report" --fullpdf**

**tabcmd export "/Sales "Sales/Sales_Analysis" --fullpdf --pagesize tabloid -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"**

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>--filename</td>
<td>The name and extension to use for the saved file</td>
<td>Saves the file with the given filename.</td>
</tr>
<tr>
<td>--csv</td>
<td></td>
<td></td>
<td>View only. Export the view's data (summary data) in CSV format.</td>
</tr>
<tr>
<td>--pdf</td>
<td></td>
<td></td>
<td>View only. Export as a PDF.</td>
</tr>
<tr>
<td>--png</td>
<td></td>
<td></td>
<td>View only. Export as an image in PNG format.</td>
</tr>
<tr>
<td>--fullpdf</td>
<td></td>
<td></td>
<td>Workbook only. Export as a PDF. The workbook must have been published with <strong>Show Sheets as Tabs</strong> enabled.</td>
</tr>
<tr>
<td>--pagelayout</td>
<td>--landscape,</td>
<td></td>
<td>Sets the page orientation of the exported PDF. If not specified, its Tableau Desktop setting will be used.</td>
</tr>
<tr>
<td>--pagesize</td>
<td>--unspecified,</td>
<td></td>
<td>Sets the page size of the exported PDF. Default is letter.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>---------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>--width</td>
<td>-w</td>
<td>Number of pixels</td>
<td>Sets the width. Default is 800 px.</td>
</tr>
<tr>
<td>--height</td>
<td>-h</td>
<td>Number of pixels</td>
<td>Sets the height. Default is 600 px.</td>
</tr>
</tbody>
</table>

**get url**

Gets the resource from Tableau Server that's represented by the specified (partial) URL. The result is returned as a file.

Note the following when you use this command:

- **Permissions**: To get a file, you must have the **Download/Web Save As** permission. By default, this permission is allowed or inherited for all roles, although permissions can be set per workbook or view.

- **File extension**: The URL must include a file extension, for example, "/views/Finance/InvestmentGrowth.csv". The extension (.csv) determines what's returned. A view can be returned in PDF, PNG, or CSV (summary data only) format. A Tableau workbook is returned as a TWB if it connects to a published data source or uses a live connection, or a TWBX if it connects to a data extract.

To figure out the correct extension, you can use a web browser to navigate to the item on Tableau Server and add the file extension to the end of the URL.

When you type the URL for the GET request, exclude the session ID (:iid=<n>) that appears at the end of the file name. For example, use "/views/Finance/InvestmentGrowth.pdf" instead of "/views/Finance/InvestmentGrowth?:iid=3.pdf".

**Note**: If you are downloading a view to a PDF or PNG file, and if you include a **--filename** parameter that includes the .pdf or .png extension, you do not have to include a .pdf or .png extension in the URL.

- **The saved file's name and location** (optional): The name you use for **--filename** should include the file extension. If you don't provide a name and file extension, both will be derived from the URL string. If you don't provide a location, the file is saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

- **PNG size** (optional): If the saved file is a PNG, you can specify the size, in pixels, in the URL.

**Clearing the cache to use real-time data**
You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using tabcmd with your own scripting, using the `refresh` parameter a great deal can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of an entire workbook.

**Examples**

**Views**

```plaintext
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename "Weekly-Report.png"

tabcmd get "/views/Finance/InvestmentGrowth.pdf" -f "Q1Growth.pdf"

tabcmd get "/views/Finance/InvestmentGrowth" -f "Q1Growth.pdf"

tabcmd get "/views/Finance/InvestmentGrowth.csv"

tabcmd get "/views/Finance/InvestmentGrowth.png?:size=640,480" -f growth.png

tabcmd get "/views/Finance/InvestmentGrowth.png?:refresh=yes" -f growth.png
```

**Workbooks**

```plaintext
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Workbooks\Weekly-Reports.twb"
```

**initialuser**

Create the initial administrative user on a server that does not have an initial administrative user defined.

**Note:** The `tabcmd initialuser` command does not require authentication to Tableau Server, but you must run the command on the primary server node.

**Examples**

```plaintext
tabcmd initialuser --username "admin" --password "P@ssword!"

tabcmd initialuser --username "admin" --password "P@ssword!" --friendly "Tableau Admin"
```
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>--friendly</td>
<td>Display name for the user</td>
<td>Creates the initial administrative user with the display name.</td>
</tr>
</tbody>
</table>

**listdomains**

Displays a list of the Active Domain domains that are in use on the server, along with their nicknames and IDs. If the server is configured to use local authentication, the command returns only the domain name `local`.

**Example**

```
tabcmd listdomains
```

**listsites**

Returns a list of sites to which the logged in user belongs.

**Example**

```
tabcmd listsites --username adam --password P@ssword!
```

**login**

Logs in a Tableau Server user.

Use the `--server, --site, --username, --password` global options to create a session.

**Note:** When you use the `tabcmd login` command, you cannot use SAML single sign-on (SSO), even if the server is configured to use SAML. To log in, you must pass the user name and password of a user who has been created on the server. You will have the permissions of the Tableau Server user that you're signed in as. For more information, see [Site Roles for Users](#) on page 788 and [Manage Permissions](#) on page 834.

If you want to log in using the same information you've already used to create a session, just specify the `--password` option. The server and user name stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify the port.

You need the `--site (-t)` option only if the server is running multiple sites and you are logging in to a site other than the Default site. If you do not provide a password you will be prompted for one. If the `--no-prompt` option is specified and no password is provided the command will fail.

Once you log in, the session will continue until it expires on the server or the `logout` command is run.
Example

Logs you in to the Tableau Server running on your local machine:

```
tabcmd login -s http://localhost -u jsmith -p p@ssw0rd!
```

Logs you in to the Sales site on sales-server:

```
tabcmd login -s http://sales-server -t Sales -u administrator -p p@ssw0rd!
```

```
tabcmd login -s http://sales-server:8000 -t Sales -u administrator -p p@ssw0rd!
```

Logs you in to the Sales site on sales-server using SSL but does not validate the server's SSL certificate:

```
tabcmd login --no-certcheck -s https://sales-server -t Sales -u administrator -p p@ssw0rd!
```

Establishes a forward proxy and port for localhost:

```
tabcmd login --proxy myfwdproxyserver:8888 -s http://localhost -u jsmith -p p@ssW0rd!
```

Logs you in to the reverse proxy using SSL:

```
tabcmd login -s https://myreverseproxy -u jsmith -p p@ssW0rd!
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-s</td>
<td>--server</td>
<td>server URL</td>
<td>If you are running the command from an on-premises Tableau Server computer, you can use <a href="http://localhost">http://localhost</a>. Otherwise, specify the computer's URL, such as <a href="http://bigbox.myco.com">http://bigbox.myco.com</a> or <a href="http://bigbox">http://bigbox</a>. For Tableau Online specify the URL <a href="https://online.tableau.com">https://online.tableau.com</a>.</td>
</tr>
<tr>
<td>-t</td>
<td>--site</td>
<td>site ID</td>
<td>Include this option if the server has multiple sites, and you are logging in to a site other than the Default site. The site ID is used in the URL to uniquely identify the site. For example, a site named</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>West Coast Sales might have a site ID of west-coast-sales.</td>
</tr>
<tr>
<td>-u</td>
<td>--username</td>
<td>user name</td>
<td>The user name of the user logging in. For Tableau Online, the user name is the user's email address.</td>
</tr>
<tr>
<td>-p</td>
<td>--password</td>
<td>password</td>
<td>Password for the user specified for --username. If you do not provide a password you will be prompted for one.</td>
</tr>
<tr>
<td></td>
<td>--password-file</td>
<td>filename.txt</td>
<td>Allows the password to be stored in the given file rather than the command line, for increased security.</td>
</tr>
<tr>
<td>-x</td>
<td>--proxy</td>
<td>Host:Port</td>
<td>Use to specify the HTTP proxy server and port for the tabcmd request.</td>
</tr>
<tr>
<td></td>
<td>--no-prompt</td>
<td></td>
<td>Do not prompt for a password. If no password is specified, the login command will fail.</td>
</tr>
<tr>
<td></td>
<td>--no-proxy</td>
<td></td>
<td>Do not use an HTTP proxy server.</td>
</tr>
<tr>
<td></td>
<td>--cookie</td>
<td></td>
<td>Saves the session ID on login. Subsequent commands will not require a login. This value is the default for the command.</td>
</tr>
<tr>
<td></td>
<td>--no-cookie</td>
<td></td>
<td>Do not save the session ID information after a successful login. Subsequent commands will require a login.</td>
</tr>
<tr>
<td></td>
<td>--timeout</td>
<td>Number of seconds</td>
<td>The number of seconds the server should wait before processing the login command. Default: 30 seconds.</td>
</tr>
</tbody>
</table>

**logout**

Logs out of the server.

**Example**

tabcmd logout
publish filename.twb(x), filename.tds(x), or filename.tde

Publishes the specified workbook (.twb(x)), data source (.tds(x)), or data extract (.tde) to Tableau Server.

If you are publishing a workbook, by default, all sheets in the workbook are published without database user names or passwords.

The permissions initially assigned to the workbook or data source are copied from the project that the file is published to. Permissions for the published resource can be changed after the file has been published.

If the workbook contains user filters, one of the thumbnail options must be specified.

Example

tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-username "jsmith" --db-password "p@ssw0rd"

tabcmd publish "analysis_sfdc.tde" -n "Sales Analysis" --oauth-username "username" --save-oauth

If the file is not in the same directory as tabcmd, include the full path to the file.

Example

tabcmd publish "C:\Tableau Workbooks\analysis.twbx" -n "Sales_Analysis" --db-username "jsmith" --db-password "p@ssw0rd"

tabcmd publish "C:\Tableau Workbooks\analysis_sfdc.tde" -n "Sales Analysis" --oauth-username "username" --save-oauth

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>--name</td>
<td>Name of the workbook or data source on the server</td>
<td>If omitted, the workbook, data source, or data extract will be named after filename.</td>
</tr>
<tr>
<td>-o</td>
<td>--overwrite</td>
<td></td>
<td>Overwrites the workbook, data source, or data extract if it already exists on the server.</td>
</tr>
<tr>
<td>-r</td>
<td>--project</td>
<td>Name of a project</td>
<td>Publishes the workbook, data source, or data extract into the specified project. Publishes to the “Default” project if not specified.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>--db-username</td>
<td>--db-username</td>
<td></td>
<td>Use this option to publish a database user name with the workbook, data source, or data extract.</td>
</tr>
<tr>
<td>--db-password</td>
<td>--db-password</td>
<td></td>
<td>Use this option to publish a database password with the workbook, data source, or data extract.</td>
</tr>
<tr>
<td>--save-db-password</td>
<td>--save-db-password</td>
<td></td>
<td>Stores the provided database password on the server.</td>
</tr>
<tr>
<td>--oauth-username</td>
<td>--oauth-username</td>
<td>Email address of the user account</td>
<td>Connects the user through a pre-configured OAuth connection, if the user already has a saved access token for the cloud data source specified in --name. Access tokens are managed in user preferences. For existing OAuth connections to the data source, use this option instead of --db-username and --db-password.</td>
</tr>
<tr>
<td>--save-oauth</td>
<td>--save-oauth</td>
<td></td>
<td>Saves the credential specified by --oauth-username as an embedded credential with the published workbook or data source. Subsequently, when the publisher or server administrator signs in to the server and edits the connection for that workbook or data source, the connection settings will show this OAuth credential as embedded in the content. If you want to schedule extract refreshes after publishing, you must include this option with --oauth-username. This is analogous to using --save-db-password with a traditional database connection.</td>
</tr>
<tr>
<td>--thumb-</td>
<td>--thumb-</td>
<td></td>
<td>If the workbook contains user filters, the</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>nail-user-name</td>
<td></td>
<td></td>
<td>thumbnails will be generated based on what the specified user can see. Cannot be specified when --thumbnail-group option is set.</td>
</tr>
<tr>
<td></td>
<td>--thumbnail-group</td>
<td></td>
<td>If the workbook contains user filters the thumbnails will be generated based on what the specified group can see. Cannot be specified when --thumbnail-user-name option is set.</td>
</tr>
<tr>
<td></td>
<td>--tabbed</td>
<td></td>
<td>When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.</td>
</tr>
<tr>
<td></td>
<td>--append</td>
<td></td>
<td>Append the extract file to the existing data source.</td>
</tr>
<tr>
<td></td>
<td>--replace</td>
<td></td>
<td>Use the extract file to replace the existing data source.</td>
</tr>
<tr>
<td></td>
<td>--disable-uploader</td>
<td></td>
<td>Disable the incremental file uploader.</td>
</tr>
<tr>
<td></td>
<td>--restart</td>
<td></td>
<td>Restart the file upload.</td>
</tr>
</tbody>
</table>

**refreshextracts workbook-name or datasource-name**

Performs a full or incremental refresh of extracts belonging to the specified workbook or data source.

This command takes the name of the workbook or data source as it appears on the server, not the file name when it was published. Only an administrator or the owner of the workbook or data source is allowed to perform this operation.

**Examples**

```
tabcmd refreshextracts --datasource sales_ds
```

```
tabcmd refreshextracts --workbook "My Workbook"
```

```
tabcmd refreshextracts --url SalesAnalysis
```
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--incremental</td>
<td></td>
<td></td>
<td>Runs the incremental refresh operation.</td>
</tr>
<tr>
<td>--synchronous</td>
<td></td>
<td></td>
<td>Adds the full refresh operation to the queue used by the Backgrounder process, to be run as soon as a Backgrounder process is available. If a Backgrounder process is available, the operation is run immediately. The refresh operation appears on the Background Tasks report. During a synchronous refresh, <code>tabcmd</code> maintains a live connection to the server while the refresh operation is underway, polling every second until the background job is done.</td>
</tr>
<tr>
<td>--workbook</td>
<td>Name of a workbook</td>
<td>The name of the workbook containing extracts to refresh. If the workbook has spaces in its name, enclose it in quotes.</td>
<td></td>
</tr>
<tr>
<td>--datasource</td>
<td>Name of a data source</td>
<td>The name of the data source containing extracts to refresh.</td>
<td></td>
</tr>
<tr>
<td>--project</td>
<td>Name of a project</td>
<td>Use with <code>--workbook</code> or <code>--datasource</code> to identify a workbook or data source in a project other than Default. If not specified, the Default project is assumed.</td>
<td></td>
</tr>
<tr>
<td>--url</td>
<td>URL name of a workbook</td>
<td>The name of the workbook as it appears in the URL. A workbook published as “Sales Analysis” has a URL name of “SalesAnalysis”.</td>
<td></td>
</tr>
</tbody>
</table>

`removeusers group-name`

Removes users from the specified group.

**Example**

`tabcmd removeusers "Development" --users "users.csv"`
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--users</td>
<td></td>
<td>filename.csv</td>
<td>Remove the users in the given file from the specified group. The file</td>
</tr>
<tr>
<td></td>
<td>--[no-</td>
<td></td>
<td>should be a simple list with one user name per line.</td>
</tr>
<tr>
<td></td>
<td>-]complete</td>
<td></td>
<td>Requires that all rows be valid for any change to succeed. If not specified</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>--complete is used.</td>
</tr>
</tbody>
</table>

**runschedule schedule-name**

Runs the specified schedule.

This command takes the name of the schedule as it is on the server.

For Tableau Online, the command can be run within the scope of a single site, using site administrator permissions.

**Example**

tabcmd runschedule "5AM Sales Refresh"

**set setting**

Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server.

Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- allow_scheduling
- embedded_credentials
- remember_passwords_forever

**Example**

tabcmd set embedded_credentials

**syncgroup group-name**

Synchronizes a Tableau Server group with an Active Directory group. If the Tableau Server group does not already exist, it is created and synchronized with the specified Active Directory group.

If the group name itself includes an "@" (other than as the domain separator) you need to refer to the symbol using the hex format "\0x40".
Example

tabcmd syncgroup "Development"
tabcmd syncgroup "Dev\0x40Fremont"

**Note:** If you synchronize a group that you are a member of, changes that you make using this command do not apply to your user. For example, if you use this command to remove the administrator right from users in a group that you are a member of, you are still an administrator when the command finishes.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--administrator</td>
<td>System, Site, or None</td>
<td>(Deprecated. Some operations may no longer work. Use the --role option instead.) Assigns or removes the administrator right for users in the group. The None option removes the administrator right from all users in the group (except you, if you are a member of the group that you are synchronizing). If you do not include this option, users who are added to the group after you run the command are not assigned the administrator right.</td>
</tr>
<tr>
<td></td>
<td>--license</td>
<td>Interactor, Viewer, or Unlicensed</td>
<td>(Deprecated. Some operations may no longer work. Use the --role option instead.) Specifies the license level for users in the group.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting in Tableau Server 9.0.</td>
</tr>
<tr>
<td>--no-publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Some operations may no longer work. Use the --role option instead.) Disallows publishing rights for users in the group.</td>
</tr>
<tr>
<td>--overwritesiterole</td>
<td></td>
<td></td>
<td>Allows a user's site role to be overwritten with a less privileged one when using --role. By default, a user site role can be promoted when using --role, but cannot be demoted. Because the --overwritesiterole option will demote user site roles, use it with caution.</td>
</tr>
<tr>
<td>--publisher</td>
<td></td>
<td></td>
<td>(Deprecated. Some operations may no longer work. Use the --role option instead.) Assigns publishing rights</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>--role</td>
<td>ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed</td>
<td>Specifies a role for users in the group. The default is Unlicensed. <strong>Note:</strong> If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.</td>
</tr>
<tr>
<td></td>
<td>--silent-progress</td>
<td></td>
<td>Do not display progress messages for the command.</td>
</tr>
</tbody>
</table>

**version**

Displays the version information for the current installation of the tabcmd utility.

**Example**

tabcmd version

**Server Administrator Reference**

This section provides reference material for server administrators.

- Tableau Server Processes on page 1240
- Tableau Server Ports on page 1244
Developer Resources

The Developer Portal on the Tableau Community is the place to go for everything related to extending and automating Tableau. There you can get access to the following:

- **JavaScript API**—Integrate Tableau views into your own web applications.
- **REST API**—Manage provisioning, permissions, and publishing on Tableau Server or Tableau Online via HTTP. The REST API gives you access to the functionality behind the data sources, projects, workbooks, site users, and sites. You can use this access to create custom applications or to script interactions with server resources.
- **Tableau SDK**—Use C, C++, Java, or Python to create extracts from any data and then to publish your extracts.
- **Web data connector**—Create a Tableau connection in JavaScript to almost any data that’s accessible over HTTP. This can include internal web services, JSON data, XML data, REST APIs, and many other sources.
- **ODBC connector**—Create a connection using ODBC (Open Database Connectivity), which is a data-access protocol that’s supported by a wide array of data sources. In Tableau Desktop, you can connect to any ODBC-compliant source using the built-in ODBC connector.

In addition to these resources, you can get complete documentation and examples, and collaborate with the Tableau developer community.

**Go to the Developer Portal**
Quick Starts

Quick Starts are one-page articles that let you quickly review how a feature works and that provide links to more detailed information.

This topic lists the Quick Start categories for Tableau Server. You can find Quick Starts for other Tableau products here:

- Tableau Desktop
- Tableau Online

Quick Starts for Tableau Server Users

- Access and Manage Your Content on page 72
- Quick Start: Stay Connected with Automatic Sign-In
- Quick Start: Quick Site Search on page 53
- Quick Start: Discover Popular Content on page 57
- Quick Start: Custom Views on page 133
- Quick Start: Keep Content Revisions on page 940
- Quick Start: Share Views on page 83
- Quick Start: Set Up Subscriptions on page 926
- Tableau Mobile: Analyze Data Anywhere
- Quick Start: Quick Site Search on page 53

Quick Starts for Site Administrators

- Quick Start: Permissions on page 842
- Quick Start: Lock Content Permissions to a Project on page 866
- Quick Start: Set Up Subscriptions on page 926

Quick Starts for Server Administrators

- Quick Start: Creating a Backup Primary on page 712
- Quick Start: Tableau Desktop License Reporting on page 1078
- Quick Start: Distributed Server on page 694
- Quick Start: Configuring Failover & Highly Available Gateways on page 709
- Quick Start: Disk Space Alerts on page 1091
• **Quick Start: Mutual (Two-Way) SSL Authentication** on page 970
• **Quick Start: Refresh Extracts on a Schedule** on page 919
• **Quick Start: Generate a Snapshot of Server Logs** on page 1184
• **Quick Start: Single Sign-On with Kerberos** on page 988
• **Quick Start: Single Sign-On with SAML** on page 1011
• **Quick Start: Synchronize All Active Directory Groups on a Schedule** on page 827
Contact Us

Directory of worldwide offices

Sales
Contact

Support

1. Search our support resources.
2. Review the search results to see if your question is answered.
3. If you can’t find what you need, scroll to the bottom of the search results, and click Continue and Create Case.
Copyright

© 2020 Tableau Software, Incorporated and its licensors. All rights reserved.

Additional information regarding Tableau’s copyrights, patents and trademarks can be found at www.tableau.com/ip.

For information about Third Party Code which may be distributed with the Software, please review the Tableau Software End User License Agreement.