



Tableau Server Administrator Guide

Version 6.1

Welcome to Tableau™ Server. This guide will get you started using Tableau Server—including installing and configuring the server; setting up distributed servers; managing users, groups, and licensing; backing up and restoring the database; and embedding views.

In addition to this guide you will find more help on the server itself. Open Tableau Server and click Help in the upper right corner to see topics specific to the page you are viewing.

Know before you start...

The machine must:

- ☐ Run a supported operating system:
Windows Server 2003 (SP1 or higher), Windows Server 2008, Windows Server 2008 R2, Windows XP (SP2 or higher), Windows Vista, Windows 7
- ☐ Have an administrative account with permission to install software and services.
- ☐ Have at least one dual core processor with 2 GB of RAM.
- ☐ Optionally have a user account that the service can use.
(Useful if you're using NT Authentication with data sources, refer to [Appendix A](#) for more information.)
- ☐ Not run Internet Information Services (IIS) if you want to use the default port 80.
(You can modify the gateway port number to avoid conflict with IIS. See [TCP/IP Ports](#) to learn how.)

Configuration Information

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

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

Ports

By default Tableau Server requires the following TCP/IP ports to be available to the server: 80, 8080, 8085, 8060, 8200, 8000 (plus the specified number of application server processes), and 8100 (plus the specified number of VizQL™ server processes). The default configuration can be changed if there is a conflict. Refer to [TCP/IP Ports](#) to learn how.

Drivers

You may need to install additional database drivers. Download drivers from:
www.tableausoftware.com/community/support/drivers.

Contents

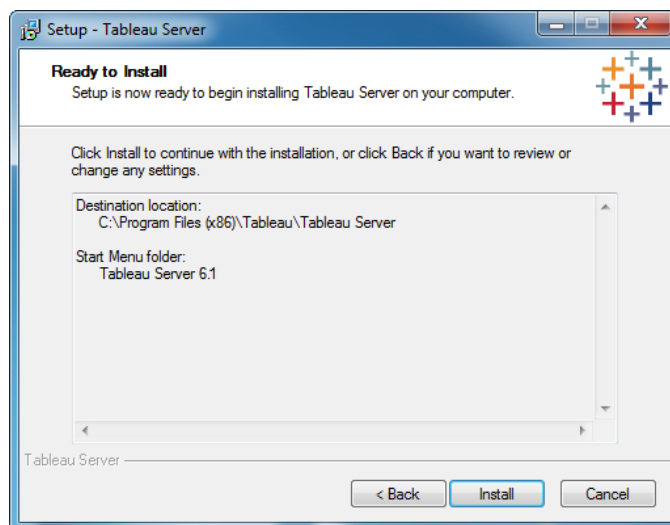
Installing.....	4
Activating the Product	5
Configuring the Server.....	10
Setting Up Distributed Servers	14
Configuring SSL	18
Publishing with Extracts.....	20
Adding Users	21
Licensing.....	24
Managing Tableau Services.....	29
Schedules and Tasks	44
Managing Domains.....	50
Managing Data Connections	51
Backing Up and Restoring the Database	52
Customizing Tableau Server	53
TCP/IP Ports.....	56
Logs and Temporary Files	58
Sharing Views	60
Embedding Views.....	61
Tableau Server Trusted Authentication	79
Appendix A: Configuring Run As User.....	87
Appendix B: Configuring SQL Server Impersonation.....	93
Appendix C: Reconfiguring the Server	99
Index	102

Installing

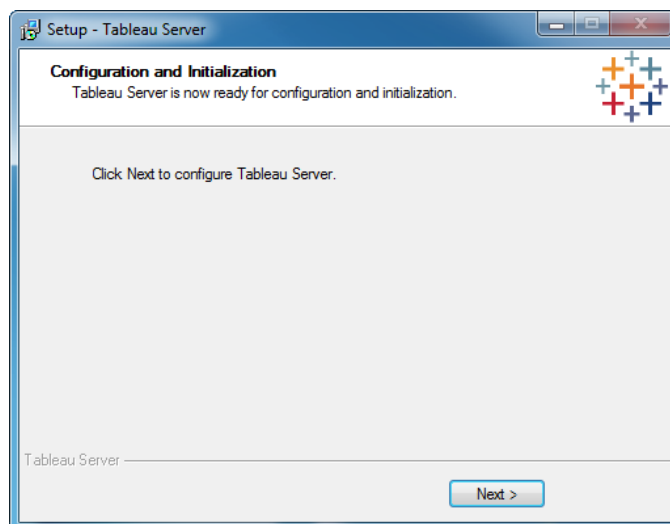
After you download the Tableau Server installation file, follow the instructions below to install the server.

To install Tableau Server:

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



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



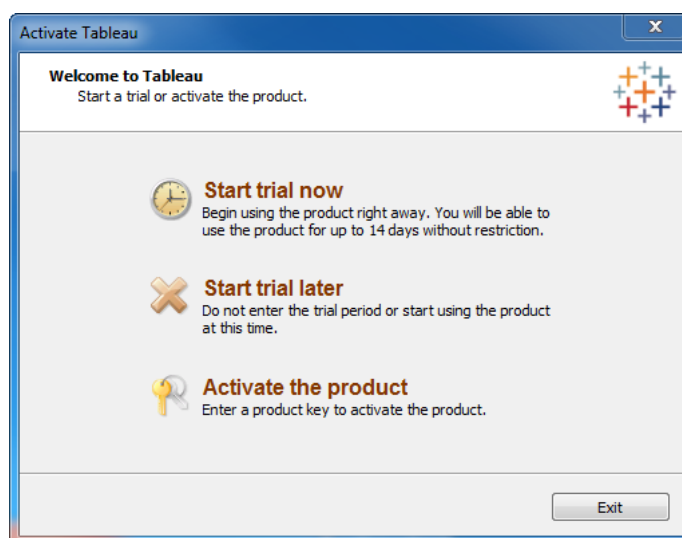
Note: If you need to support characters that are not the Latin-1 set, make sure to install the Windows Language Packs via Control Panel > Regional and Language Options. The language packs will need to be installed on the primary server as well as any worker machines.

Activating the Product

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](#).

Activate and Register

After installing and configuring the server, the product key manager automatically opens so you can enter your product key and register the product. Select **Activate** and then paste in your product key.

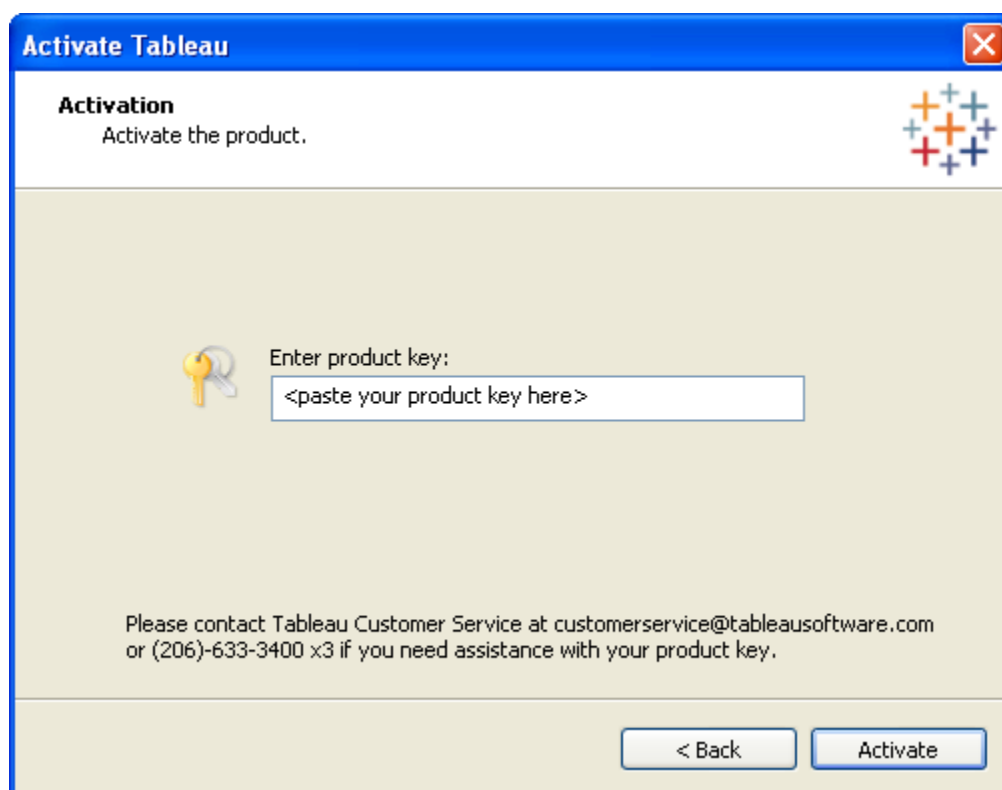


Refer to the [download help](#) page on the web site for step-by-step instructions. If you need to activate the product on a computer that is offline, see [Offline Activation](#).

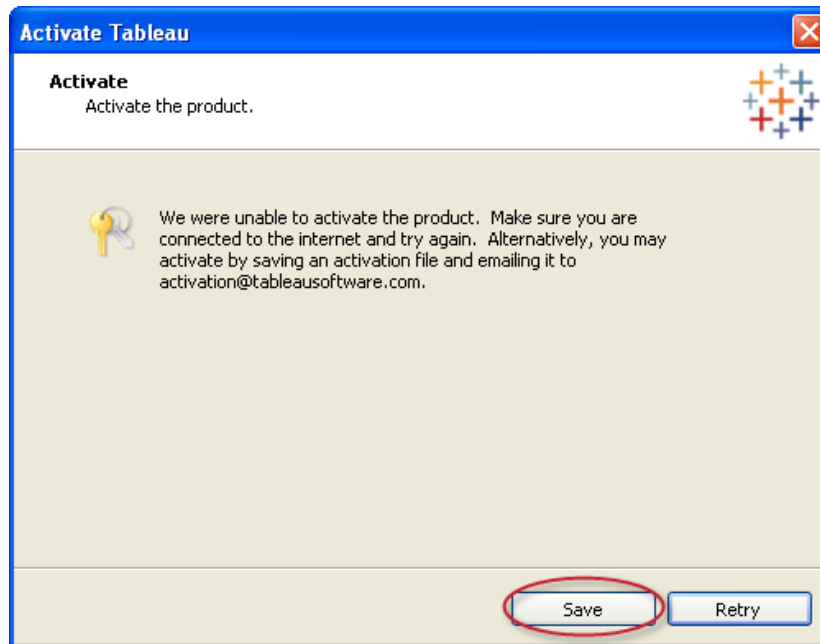
Offline Activation

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

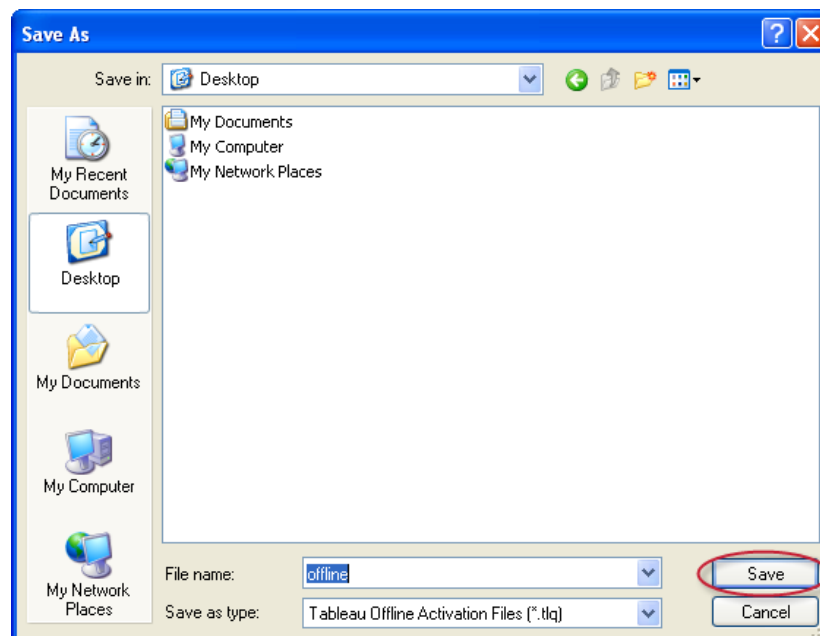
1. When the product key manager opens click **Activate the product.**
2. Paste your server product key into the corresponding text box and click Activate. You can get your product key from the [Customer Account Center](#) on Tableau's web site.



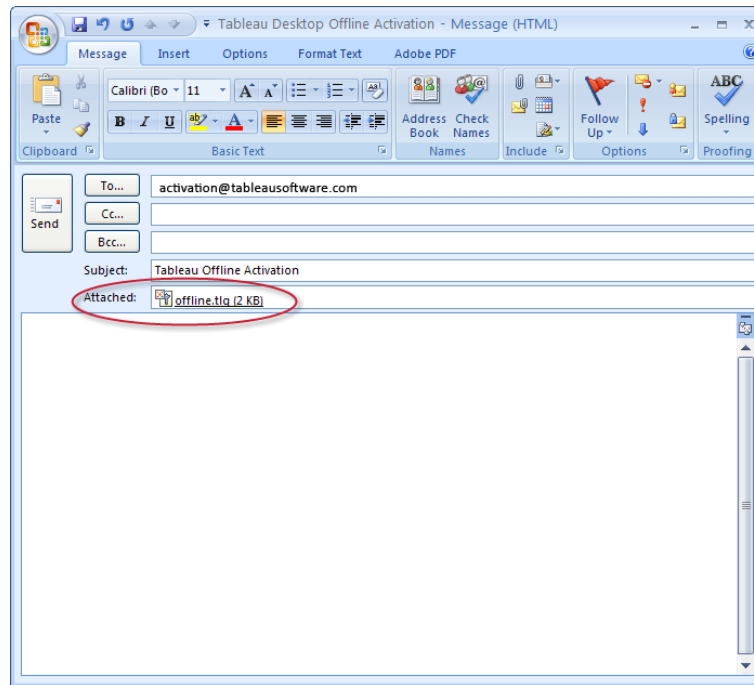
- When you are offline, activation will fail and you are given the option to save a file that you can use for offline activation. Click **Save**.



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



5. Back in Tableau click **Exit** to close the Activation dialog box.
6. Move the file to a computer that is online and open an email editor. Create a new email to activation@tableausoftware.com. Attach the file to the email and click **Send**.



7. The Tableau Customer Service Team will email you back a file called *offlinerresponse.tlf*. Move this file to the computer where you are installing Tableau Server. If you have Tableau Desktop installed on the computer you can then **double-click** the new file to complete activation. If you do not have Tableau Desktop installed continue to steps 8 and 9.



8. On the computer where you are installing Tableau Server, open a command prompt as an administrator and run the following command:

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

9. Then type the following command:

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

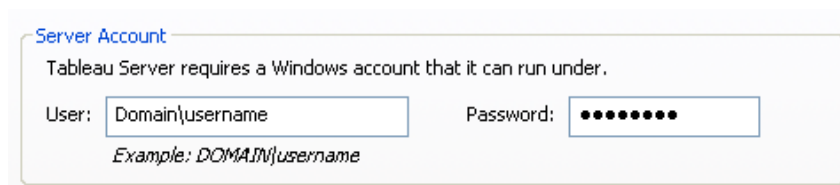
Replace the file path above with the path to where you've saved the response file Tableau emailed to you.

If you need additional assistance, please contact the Tableau Customer Service team at customerservice@tableausoftware.com.

Configuring the Server

General Configuration

1. By default, Tableau Server runs using the Network Service account. If you want to use another user account to accommodate NT Authentication with data sources, specify the user name and password in the appropriate text boxes. The user name should include the domain name. Refer to [Appendix A](#) to learn more about using a specific user account.



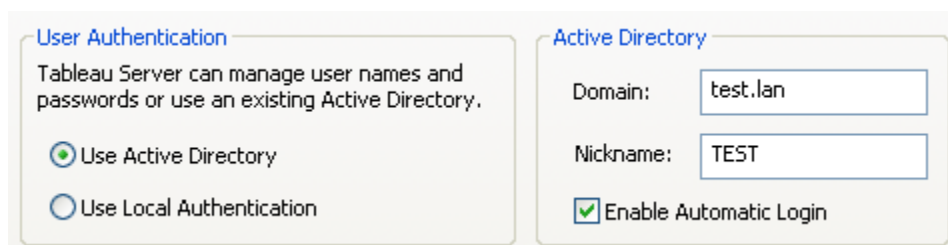
Server Account

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

User: Password:

Example: DOMAIN\username

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



User Authentication

Tableau Server can manage user names and passwords or use an existing Active Directory.

☒ Use Active Directory
☐ Use Local Authentication

Active Directory

Domain:

Nickname:

☒ Enable Automatic Login

If you use Active Directory:

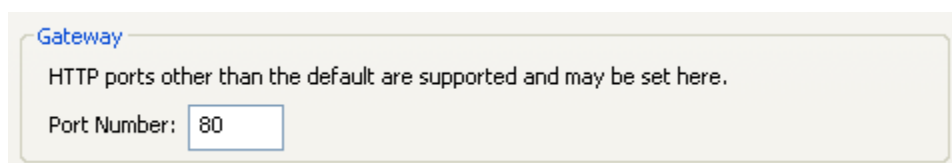
- You can optionally **Enable Automatic Login**, which uses Microsoft SSPI to automatically log in your users based on their Windows username and password.
- Be sure to type the fully qualified domain name (FQDN) and nickname.

To determine the FQDN: Select **Start > Run** then type **sysdm.cpl** in the Run textbox. In the System Properties dialog box, select the **Computer Name** tab. The FQDN is shown near the middle of the dialog box.

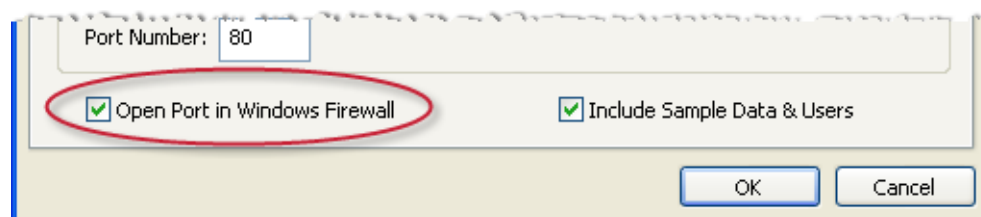
The first time your users log in, they will need to use the fully qualified domain name (for example, `test.lan\jsmith`). On subsequent logins, they can use the nickname (`test\jsmith`).

3. The default port for web access to Tableau Server (via HTTP) is port 80.

You may need to change the port number if you have another server running on port 80 or other networking needs. For example, you may need to change the port number if you have a hardware firewall or proxy in front of the Tableau Server host and don't want the back-end system running on port 80.



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



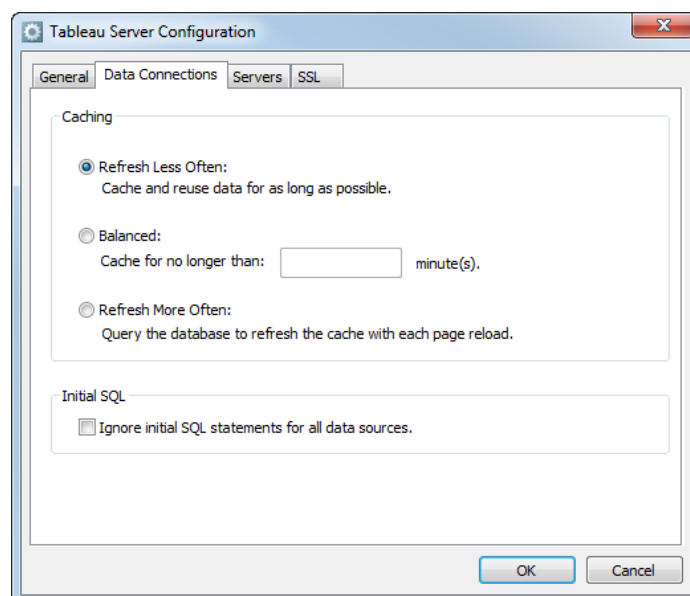
5. Select whether to include sample data and users. The sample data can help you get familiar with Tableau Server, especially if you are installing a trial version of the product. Initially the sample user uses one interactor license. You can change this user to unlicensed in order to reclaim the license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

Username	Password
Tableau Software	test

6. Optionally continue to the next page to configure Caching. If you do not want to configure caching options click **OK** to continue to Setup Tasks.

Data Connection Configuration

Use the Data Connection tab to configure aspects of caching and initial SQL statement usage that will apply to all data connections.



Views published to Tableau Server are interactive and often 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.

To configure caching:

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

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

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

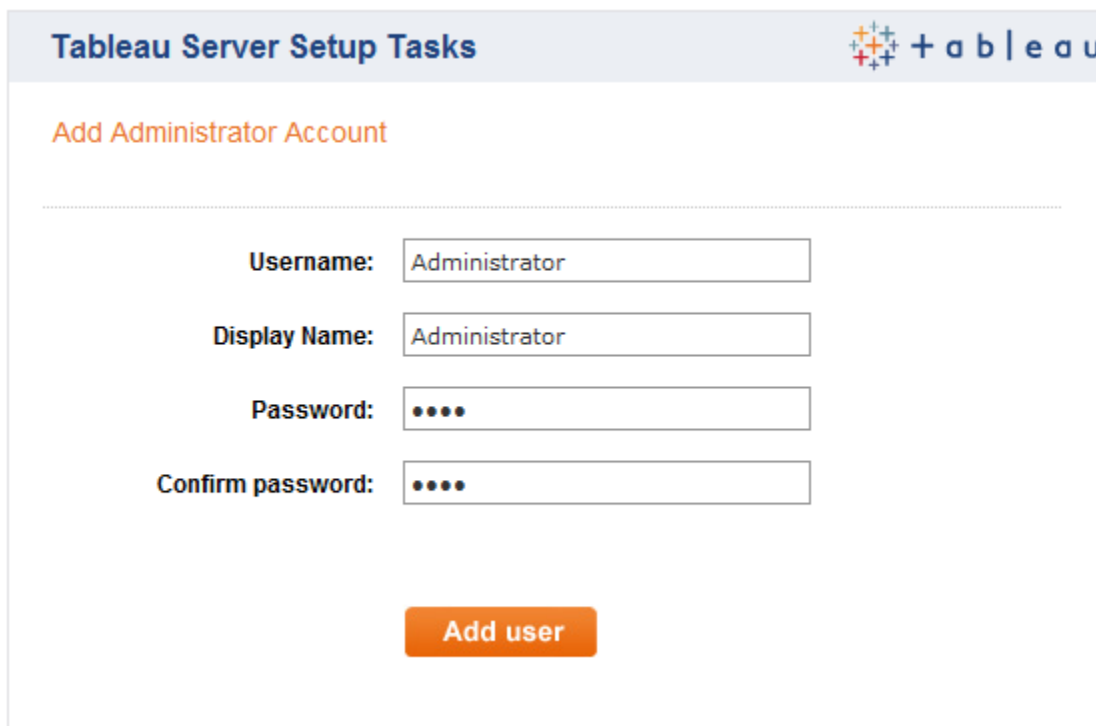
To disable initial SQL functionality:

1. Select the Data Connections tab in the Tableau Server Configuration dialog box.
2. Select the **Ignore initial SQL statements for all data sources** checkbox.

Workbooks created with initial SQL statements will still open but the initial SQL commands will not be sent.

Setup Tasks

The final step in activating Tableau Server is to add an administrator account. If you are using Active Directory, type the username and password for the Active Directory user who will be the administrator. If you are using Local Authentication, create an administrative account by typing a username, name, and password (twice) of your choosing. Then click **Add User**.



The screenshot shows the 'Tableau Server Setup Tasks' window with the 'Add Administrator Account' section. It contains four input fields: 'Username' (Administrator), 'Display Name' (Administrator), 'Password' (masked with dots), and 'Confirm password' (masked with dots). An orange 'Add user' button is at the bottom.

Note: By default, Tableau Server opens your default web browser to <http://localhost> to complete the setup tasks. Depending on your security settings, this may cause a blocked content message in your browser.

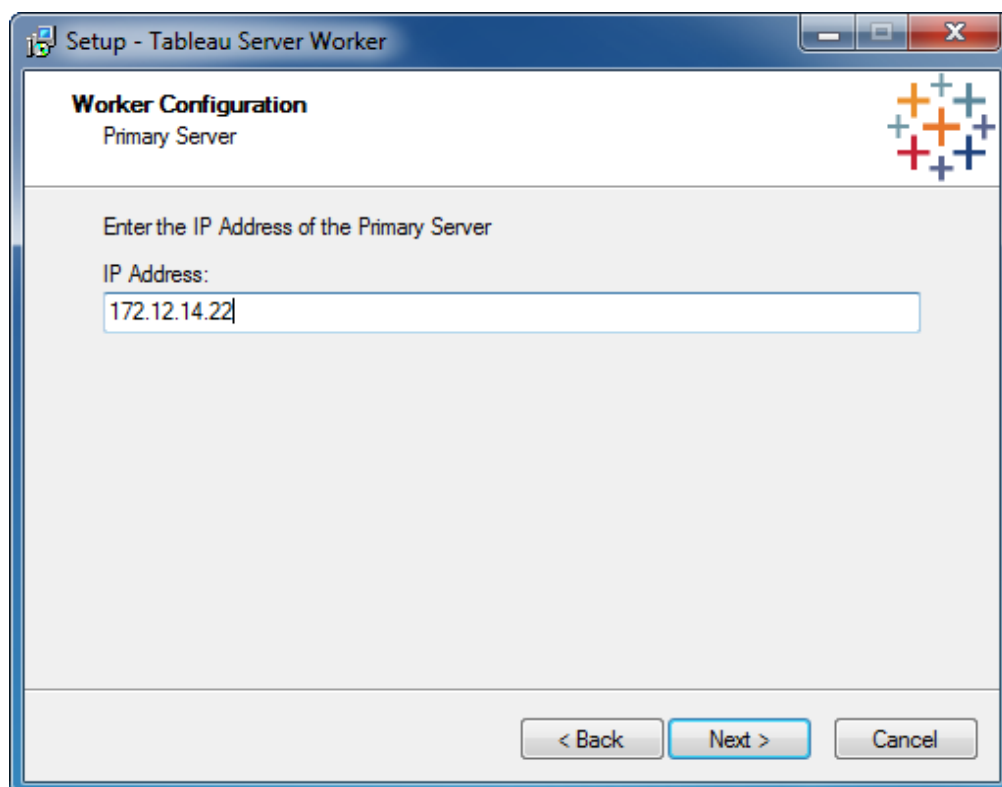
Setting Up Distributed Servers

After you complete the initial configuration, you can set up Tableau Server to run on multiple machines and you can fine tune which Tableau Server processes are run on each machine, including on the primary server. This type of distributed environment can help you support more users, improve view interaction and browsing, as well as optimize the handling of server background tasks.

For example, if you expect your users will spend more time interacting with views than browsing and searching, you could set up an environment with several machines dedicated as VizQL servers. On the other hand, if you expect more browsing and searching you could dedicate a number of machines to the server's web application processes. If your site handles a large number of extract refresh tasks, you can dedicate a machine to running background task processes.

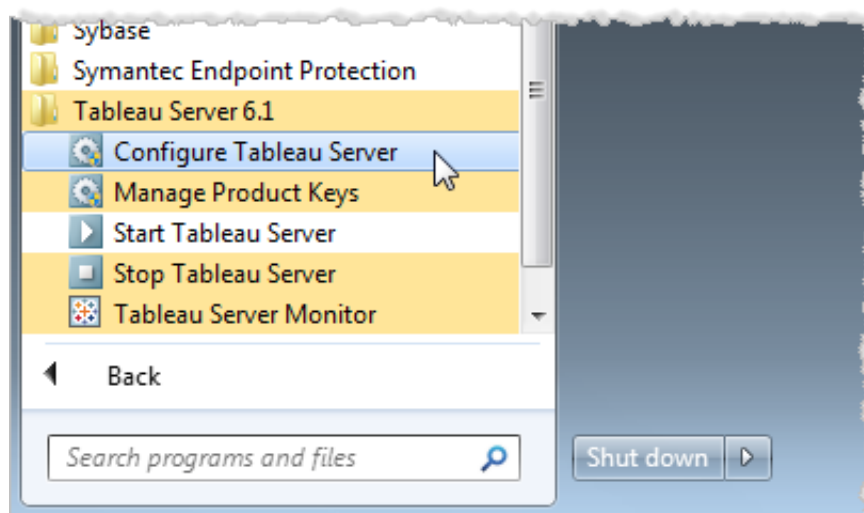
To set up distributed servers:

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

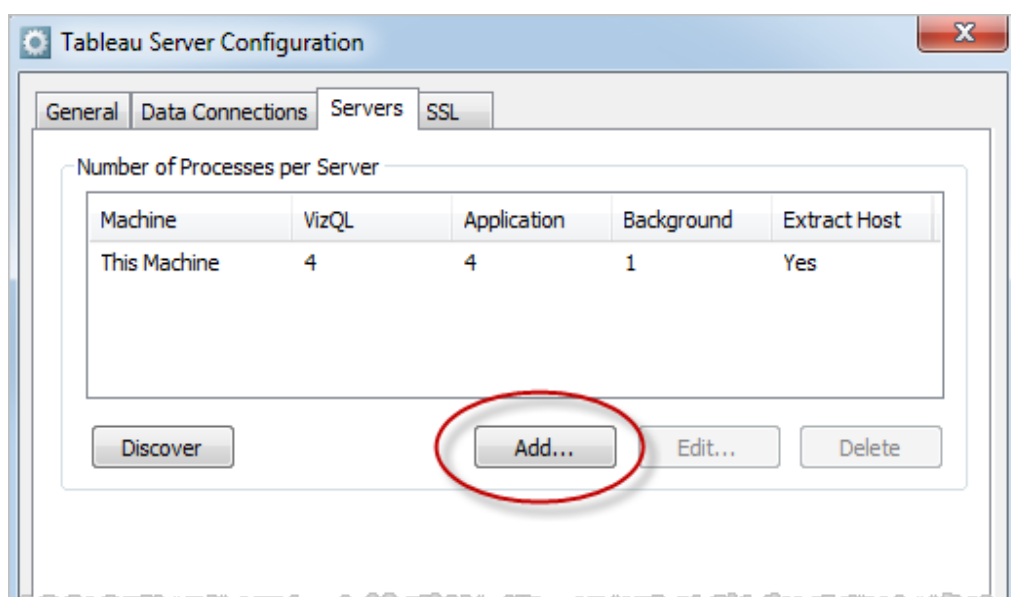


Note: To install the Worker software, port 3730 must be open. If that port is in use, the installation will fail. On the Primary server machine, the following ports must be open: 1070-1821 (for distributed Ruby) and ports 27000-27009 (for licensing communication).

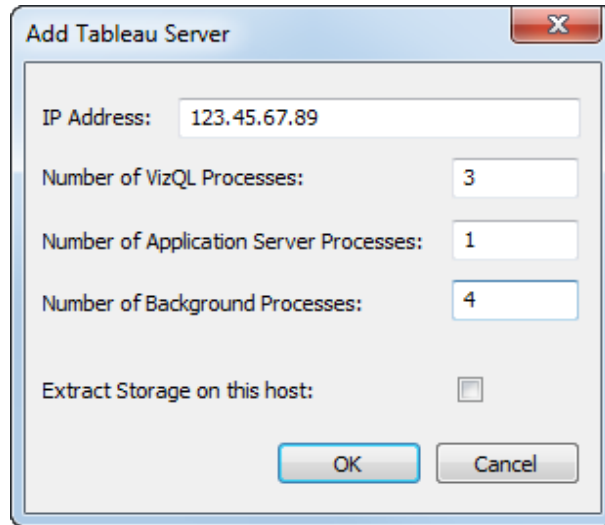
- Once the Worker software is installed on worker machines, return to the primary server and open the configuration utility by selecting **Tableau Server 6.1 > Configure Tableau Server** on the Start menu.



- In the Configuration Utility, select the **Servers** tab and click the **Add** button.



7. In the next dialog box, type the **IP Address** for one of the worker machines and specify the number of **VizQL**, **Application Server**, and **Background** processes to allocate to the machine.

A screenshot of the "Add Tableau Server" dialog box. It has a title bar with a close button (X). The dialog contains four input fields: "IP Address:" with the value "123.45.67.89", "Number of VizQL Processes:" with the value "3", "Number of Application Server Processes:" with the value "1", and "Number of Background Processes:" with the value "4". Below these is a checkbox labeled "Extract Storage on this host:" which is currently unchecked. At the bottom are "OK" and "Cancel" buttons.

IP Address:	123.45.67.89
Number of VizQL Processes:	3
Number of Application Server Processes:	1
Number of Background Processes:	4
Extract Storage on this host:	<input type="checkbox"/>

To assign one or more background processes to a server, you must also assign it at least one application server process:

8. Select the **Extract Storage on this host** checkbox if you want to use this server for extract storage. By default, extract storage is hosted on the primary machine.
9. Click **OK**.
10. Repeat these steps for each machine you want to add to the distributed environment.
11. When you're finished adding workers, click **OK** again to save the changes, then start the server on the primary machine.

Server Updates and Maintenance

After the first time you set up the workers you can perform all configuration and updates from the primary machine. Specifically, you should make changes using the command line tools and configuration utility on the primary server. Updates will be pushed to the workers automatically.

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

Database Drivers

The Tableau Server and Tableau Server Worker installers automatically install drivers for Oracle and Oracle Essbase databases. If you plan to publish workbooks that connect to other databases you will need to ensure that both primary and worker machines have the corresponding drivers.

Only workers that have VizQL processes need these database drivers. For example, if you have a worker dedicated as a VizQL server and another one dedicated as an Application server, you only need to install drivers onto the VizQL server.

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

Machine	Repository	Server Web Application	VizQL Server	Background Tasks	Web Server
172.24.15.23	✓	✓✓✓✓✓✓✓✓	✓✓✓✓✓✓✓✓	✓	
172.24.15.24		✓✓✓✓	✓✓✓✓	✓	✓
172.24.15.25		✓✓✓✓	✓✓✓✓	✓	✓

Configuring SSL

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all HTTP traffic. Setting up SSL ensures that access to the web application is secure and that sensitive information passed between the web browser and the server or Tableau Desktop and the server is protected. To configure the server for SSL you must first acquire a certificate from a trusted authority then import the certificate files into Tableau Server. Follow the steps below to configure Tableau Server to use SSL.

To configure Tableau Server to use SSL:

1. Acquire an Apache SSL certificate from a trusted authority (e.g., Verisign, Thawte, Comodo, GoDaddy, etc.). You can also use an internal certificate issued by your company.

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

2. Place the certificate files in a folder named SSL, parallel to the Tableau Server 6.1 folder. For example:

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

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

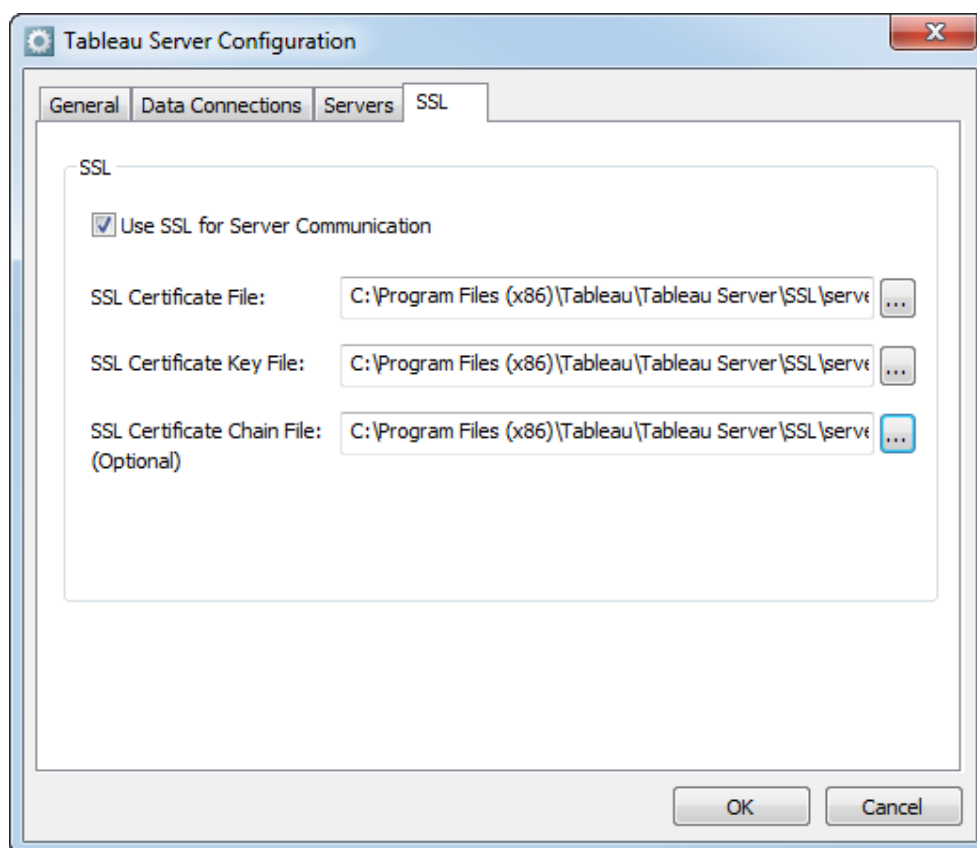
3. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 6.1 > Configure Tableau Server** on the Start menu.
4. In the Configuration Tableau Server dialog box, select the **SSL** tab.

5. Select **Use SSL for Server Communication**, and provide the location for each of the following certificate files:

SSL Certificate File - must be a valid PEM encoded x509 certificate with the extension .crt

SSL Certificate Key File - must be a valid RSA or DSA key that is not password protected with the file extension .key

SSL Certificate Chain File (Optional) - Some certificate providers issue two certificates for Apache. The second certificate is the chain file that contains information about the provider. If your provider has issued this second certificate you can enter it here.



6. Click **OK**.

The changes will take effect the next time the server is restarted. When the server is configured for SSL, it accepts requests to the non-SSL port (default is port 80) and automatically redirects to the SSL port 443. SSL errors are logged in the install directory at the following location. Use this log to troubleshoot validation and encryption issues.

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

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

Publishing with Extracts

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

When you publish a workbook that uses an extract, you can assign it to a recurring refresh schedule. At an interval you determine, Tableau refreshes all the data in the extract. You can also optionally define an incremental update for an extract, where you identify a column in your extract that has a numerical value, such as a timestamp. Tableau uses this column to identify new rows that need to be added to your extract. This is called an incremental refresh and it can also be scheduled.

For example, let's say you have a workbook with a view that focuses on new orders. It connects to a large data warehouse that is updated daily. Instead of publishing a workbook that queries the live data, you create an extract that includes only the data necessary for the view, including values for the date column, as well as some others, such as order ID, and product name. You also define an incremental extract on the date column. When you publish the workbook, you schedule the extract to be fully refreshed every Saturday night. You also schedule an incremental refresh for every weekday morning. This means that every weekday morning, Tableau connects to the data warehouse, identifies records in the date column with a timestamp newer than the most recent one in the extract, and adds those records to the extract. Every Saturday night, the entire extract is updated.

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

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

Adding Users

Everyone who needs to access Tableau Server, whether it's to publish, browse, or administer, must be added as a user. In addition, users must be assigned a [license level](#). Depending on how the server has been configured you can add users using the internal user management system (local authentication) or you can import from Active Directory.

Adding Local Users

If the server is configured to use Local Authentication, you can add individual users or import several users from a comma separated value (CSV) file. You cannot add local users to a server that is configured to use Active Directory.

To add local users:

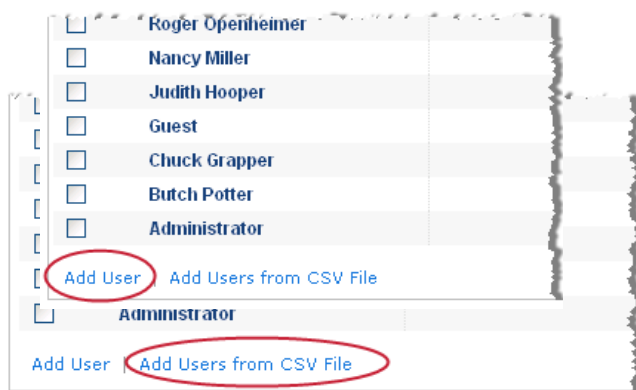
1. Log into Tableau Server using your administrator user name and password.
2. Click **Users** in the Administration area on the left side of the page.



3. Click one of the following links at the bottom of the list of users:

Add User - to add users one at a time by specifying a user name and password.

Add Users From CSV File - to add multiple users contained in a CSV file.



4. If you are adding a single user, specify the following:

Required User Information:

Username:

jsmith

Full Name:

John Smith

Password:

••••••••

Confirm Password:

••••••••

License Level for this user:

☐ Unlicensed

☐ Viewer

☒ Interactor

Assign user rights:

☒ Publish

☒ Administrator

☐ Content ☐ System

Add User

- **Username** - type a username for the user (e.g., jsmith, johnsmith). The username can only contain letters and numbers.
- **Full Name** - type a display name for the user (e.g., John Smith).
- **Password** - type a password for the user.
- **Confirm** - retype the password.
- **License Level** - select a license level. Refer to [License Levels & Permissions](#) to learn more.
- **User Rights** - select whether the user can publish workbooks and assign administrator rights. Refer to [User Rights](#) to learn more.

When finished, click **Add User**.

Otherwise, if you are importing a CSV file, **Browse** and select the file, then click **Import File**. CSV files can have the following columns in the order shown. Columns you include must also have column headers:

- Username*
- Password*
- Full Name
- License Level (Interactor, Viewer, or Unlicensed)
- Administrator (System, Content, or None)
- Publisher (yes/true/1 or no/false/0)

* These columns are required.

If the file contains usernames that already exist, they will be updated with the information in the file.

Adding Users from Active Directory

The *easiest* way to add users when using Active Directory authentication is to import an Active Directory group. Refer to the *Managing Groups* topic in the online help to learn more about groups on Tableau Server. You can also add users one at a time as described below.

To add users from Active Directory:

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



3. Click **Add User** at the bottom of the list of users.
4. Specify the following:

Required User Information:

Username:

jsmith

License Level for this user:

☐ Unlicensed
 ☐ Viewer
 ☒ Interactor

Assign user rights:

☒ Publish
 ☒ Administrator

☒ Content
 ☐ System

Add User

- **Username** - type a username for the user (e.g., jsmith, domain.lan\jsmith, jsmith@domain.lan).

If the Active Directory domain is different than the server domain you need to include the fully qualified domain name for the first user you add. Subsequent users can use the domain nickname. Refer to [Managing Domains on the Server](#) to learn how to modify the nickname.

- **License Level** - select a license level. Refer to [License Levels & Permissions](#) to learn more.
- **User Rights** - select whether the user can publish workbooks and assign administrator rights. Refer to [User Rights](#) to learn more.

When finished, click **Add User**.

Licensing

All users on Tableau Server must be assigned a license level. License Levels control how much access the user has on the server. For example, users with the Viewer license level cannot interact with views.

License Levels

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

Unlicensed - The user cannot log in to the server. All users are added as unlicensed by default.

Viewer - The user can log in and see published views on the server but cannot interact with the views. Users with this level can only be given permission to view, add comments, and view comments.

Interactor - The user can log in, browse the server, and interact with the published views.

Guest - The guest license level is available to allow users without an account on the server see and interact with an embedded view. When enabled, the user can load a webpage containing an embedded visualization without logging in. This option is only available with a core-based server license.

User Rights

In addition to the license levels, you can also assign the following user rights:

Publish - Allows the user to connect to Tableau Server from Tableau Professional so that she can publish and download workbooks and data sources.

Admin - Makes the user an administrator. There are two types of administrators: Content administrators and system administrators. Content administrators can manage users, groups, projects, workbooks, and data connections. System administrators have all the rights of a content administrator, and they can also administer the server itself. This includes handling maintenance, settings, schedules, and the search index. The Admin right can only be assigned to users with the Interactor license level and the Publish right.

Permissions

Finally, when someone publishes a workbook to the server, he or she can allow and deny specific capabilities to further control who can access the workbook. The following capabilities can be allowed or denied to individual users or entire groups:

View - See the workbook on the server.

Write - Edit and republish.

Delete - Delete the workbook from Tableau Server.

Filter - See and interact with filters that are published with each view.

Add Comment - Add comments to the view.

View Comments - See comments associated with the view.

View Underlying Data - See the raw data behind each view.

Export Image - Export and save the view as an image file.

Export Data - Export the aggregated data as a comma separated value (CSV) file.

Download File - Open the workbook from the server using Tableau Professional.

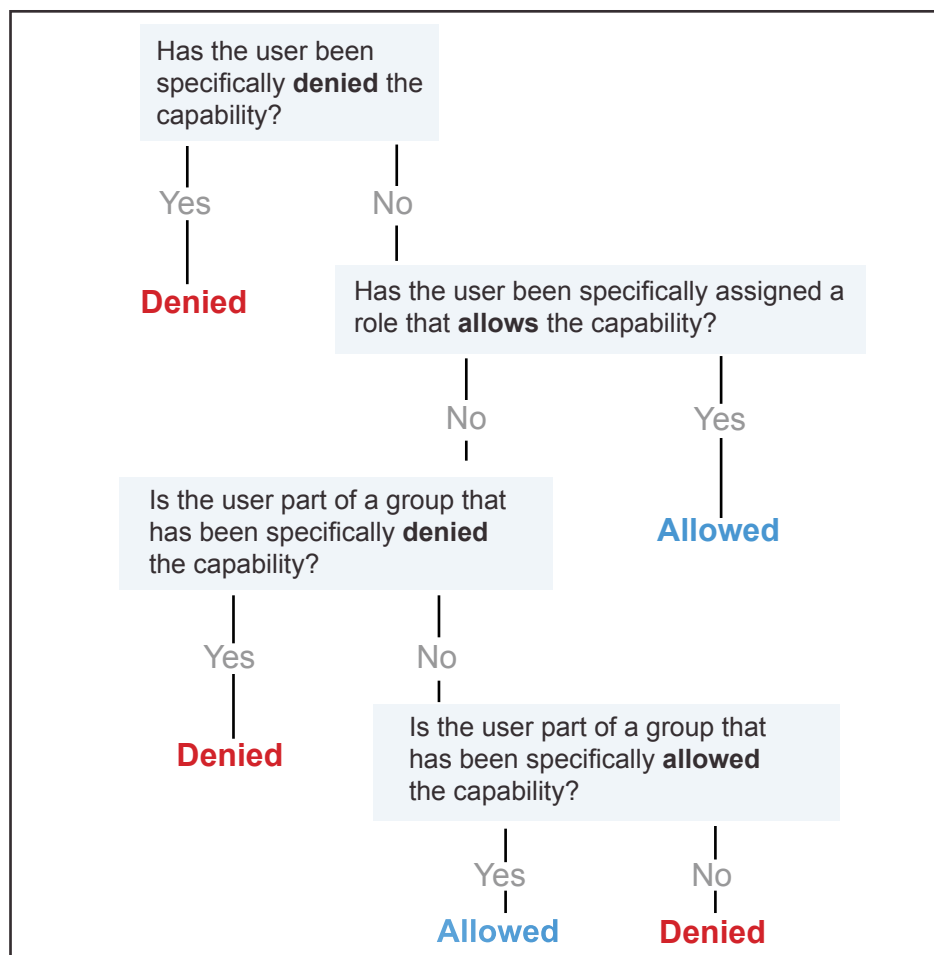
Share Customized - Make saved customizations to a view public for others to see (users can create custom views using the **Remember my changes** option).

Move - Move the workbook between projects.

Set Permissions - Modify the permissions.

To make it easy to assign common combinations of these capabilities, Tableau Server comes with some pre-defined permission roles. Select one of these roles when adding permissions to see the capabilities it allows. Refer to the *Setting Permissions* topic in the online help for more information about assigning permissions.

The diagram below explains how permissions are evaluated for a view or workbook.



Note: If the workbook is configured to show sheets as tabs, all sheets inherit the workbook permissions even if different permissions are specified on an individual sheet.

Assigning License Levels and Rights

When you add users to Tableau Server, you can optionally assign them license levels and user rights. By default users are [unlicensed](#) and have no user rights. That means that they cannot log in to the server and cannot publish workbooks. You can assign different license levels and user rights to further control access to the server.

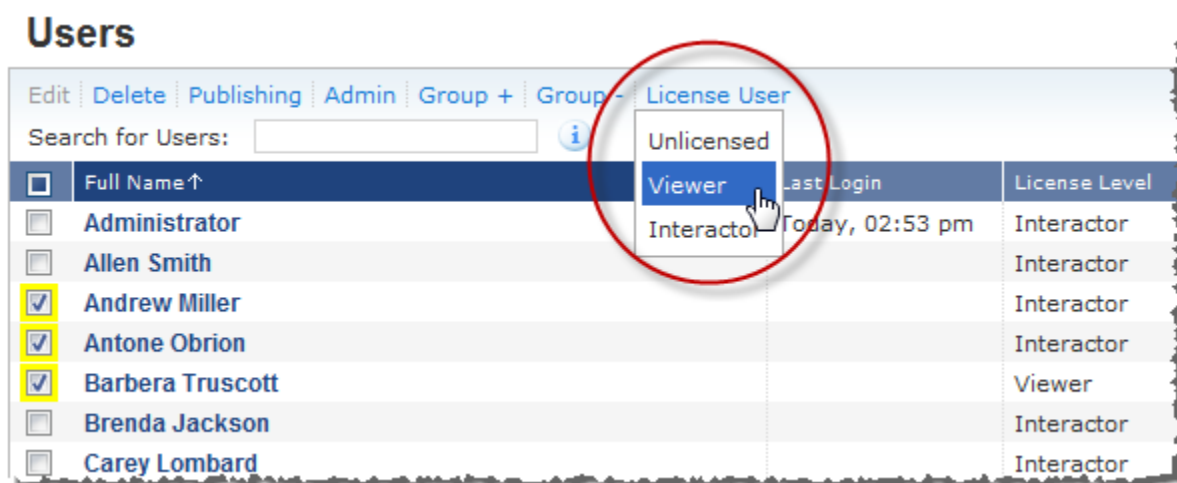
To assign license levels:

1. Log into Tableau Server using your administrator user name and password.
2. Click **Users** in the Administration area on the left side of the page.
3. Select one or more users you want to assign license levels to.
4. Click the **License User** link in the Actions toolbar along the top of the list.
5. Select the license level to assign to the selected users.

The Licensed Level column in the list of users is the updated to reflect the changes. Refer to [License Levels](#) to learn more about each level.

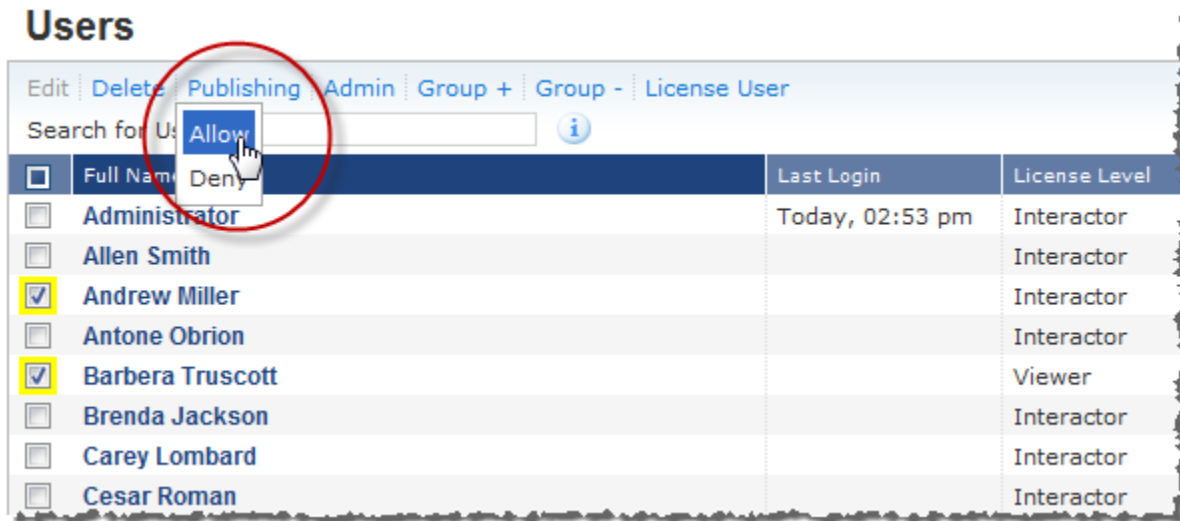
To allow or deny user rights:

1. Log into Tableau Server using your administrator user name and password.
2. Click **Users** in the Administration area on the left side of the page.
3. Select one or more users you want to assign user rights to.



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

5. Select **Allow** or **Deny** to change the Publishing right for the selected users.



6. Select **System** or **Content** or **None** to change the Admin right for the selected users.
The Admin and Publish columns in the list of users are updated to reflect the changes. Refer to [User Rights](#) to learn more about each right.

Managing Tableau Services

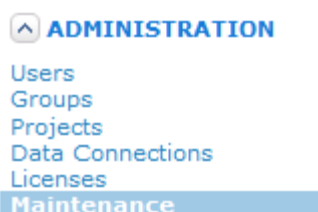
As an administrator you have access to several tools that help you monitor and manage Tableau Services. For maintenance from a remote machine you can use the maintenance tools accessed through the web application. If you are working directly on the server you can use Tableau Server Monitor and the command line administrative tools.

Maintenance Tools

In addition to Tableau Server Monitor, you also have access to maintenance tools on Tableau Server. There you can see detailed status for each service and process, monitor server and user activity, perform administrative tasks, and specify settings to customize the server. Refer to the Server Maintenance topic in the online help for more information regarding these tools.

To access the administrative tools on the server:

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



Viewing Service Status

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

<div> ✔ Service up, waiting for request 🟡 Service up, currently handling request ❌ Service down </div>					
Status					
Machine	Repository	Server Web Application	VizQL Server	Background Tasks	Web Server
Primary	✔	✔✔✔✔	✔✔✔✔	✔	

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

```
<?xml version="1.0" encoding="UTF-8" ?>
- <systeminfo>
- <machines>
- <machine name="Primary">
  <repository worker="Primary:8080" status="OK" />
  <serverwebapplication worker="Primary:8000" status="OK" />
  <serverwebapplication worker="Primary:8001" status="Busy" />
  <serverwebapplication worker="Primary:8002" status="OK" />
  <serverwebapplication worker="Primary:8003" status="OK" />
  <vizqlserver worker="Primary:8100" status="OK" />
  <vizqlserver worker="Primary:8101" status="OK" />
  <vizqlserver worker="Primary:8102" status="OK" />
  <vizqlserver worker="Primary:8103" status="OK" />
  <backgroundtasks worker="Primary" status="OK" />
</machine>
</machines>
</systeminfo>
```

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

Accessing Service Status Remotely

As the Tableau administrator, only you can see the tools on the Maintenance page, including the Status table. You can, however, make the machine-readable version of the Status table available to non-admin users and to computers other than the one that's hosting Tableau Server—for example, as part of a remote monitoring process.

To grant remote access to Tableau Service status:

1. On the computer running the primary Tableau Server, open the Tableau Server config file: `ProgramData\Tableau\Tableau Server\config\tabsvc.yml`
2. Add the line `wgserver.systeminfo.allow_referrer_ips: <IP address>` to `tabsvc.yml`, where `<IP address>` is the IP address of the computer you'd like to add. If you are granting service status access to multiple computers, use commas (no spaces) to separate each IP address.

For example:

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

3. Save and close `tabsvc.yml`.
4. Open a command prompt as an administrator and type:

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

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

```
tabadmin restart
```

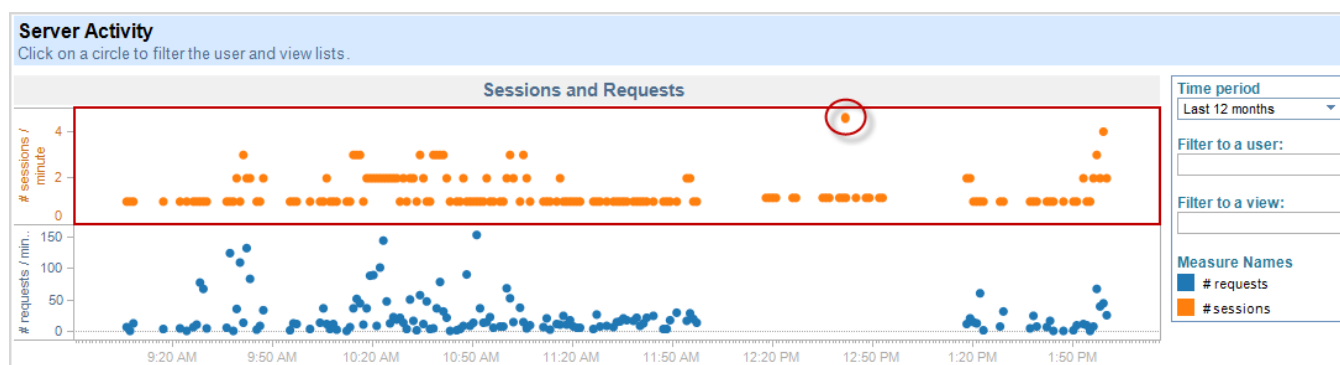
Now, users at computers whose IP addresses are added to `tabsvc.yml` can view Tableau service status by entering the URL `http://<server>/admin/systeminfo` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo`). This functionality can also be used as part of an automated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

Administrative Views

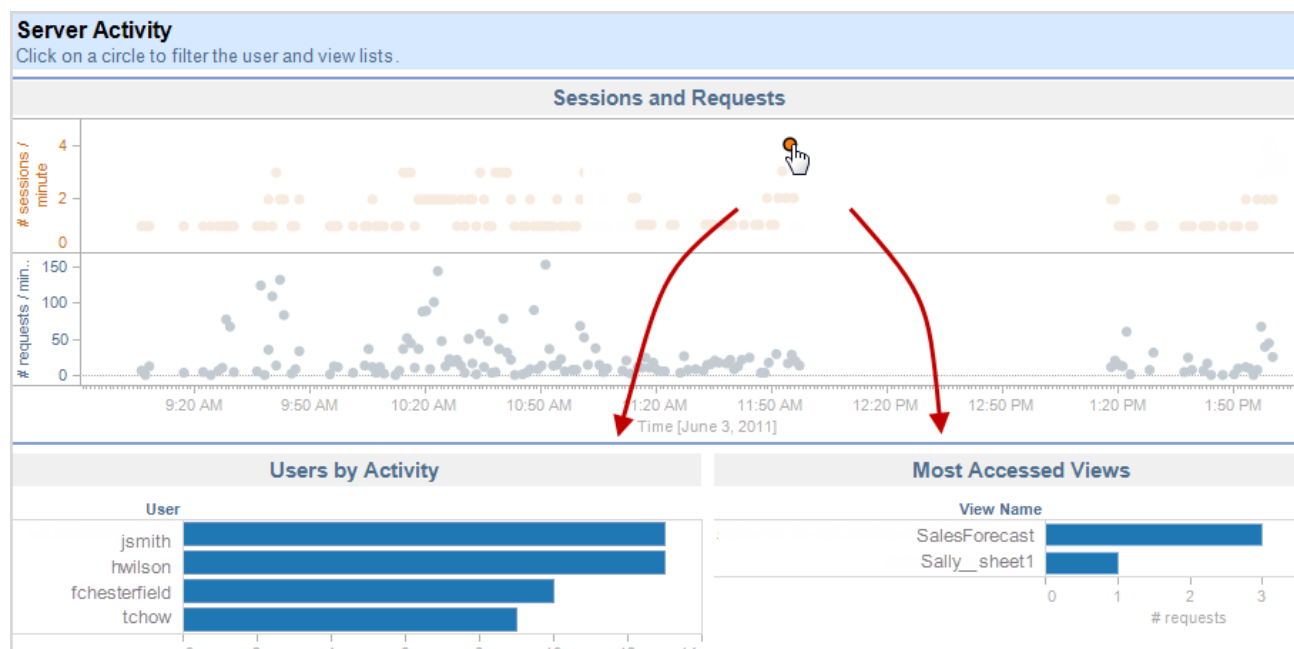
The Analysis table on the Maintenance page gives you access to several administrative views that can help you monitor activity on Tableau Server: [Server Activity](#), [User Activity](#), [Performance History](#), [Background Tasks](#), [Space Usage](#), and [Customized Views](#).

Server Activity

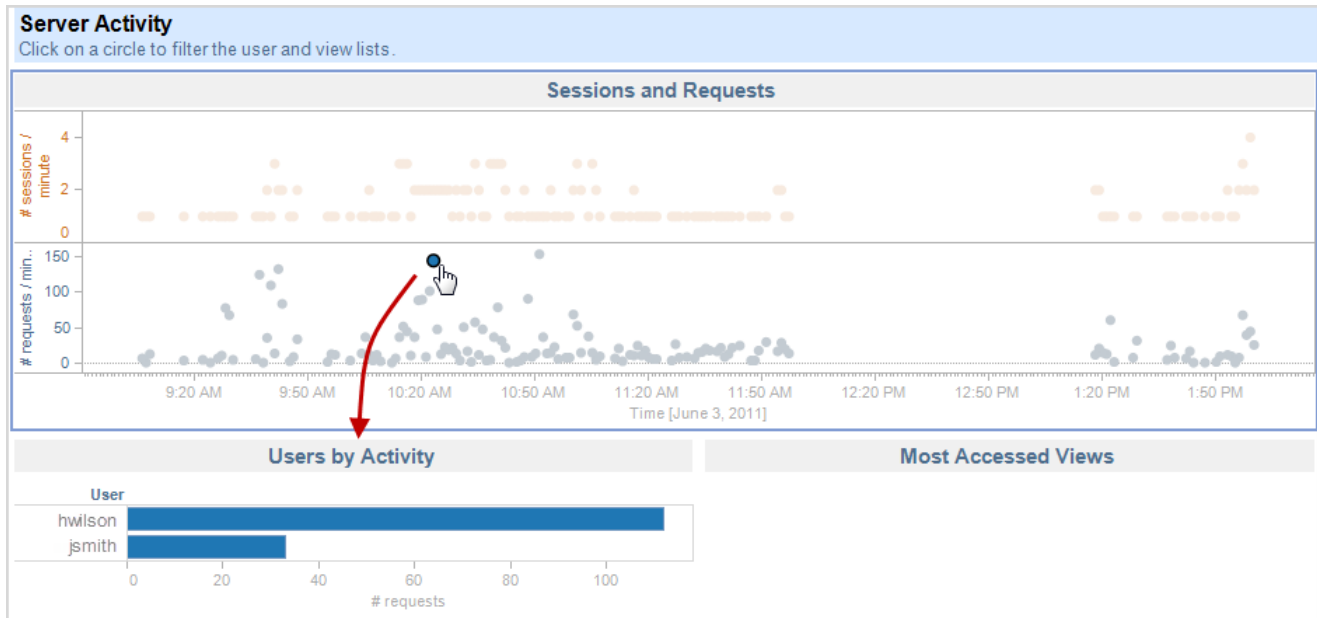
Server Activity gives you a basic view of Tableau Server activity for every minute within the selected time period. In **Sessions and Requests**, each orange circle represents a minute during which there was at least one server session (logged in user). A highly placed orange circle indicates a large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



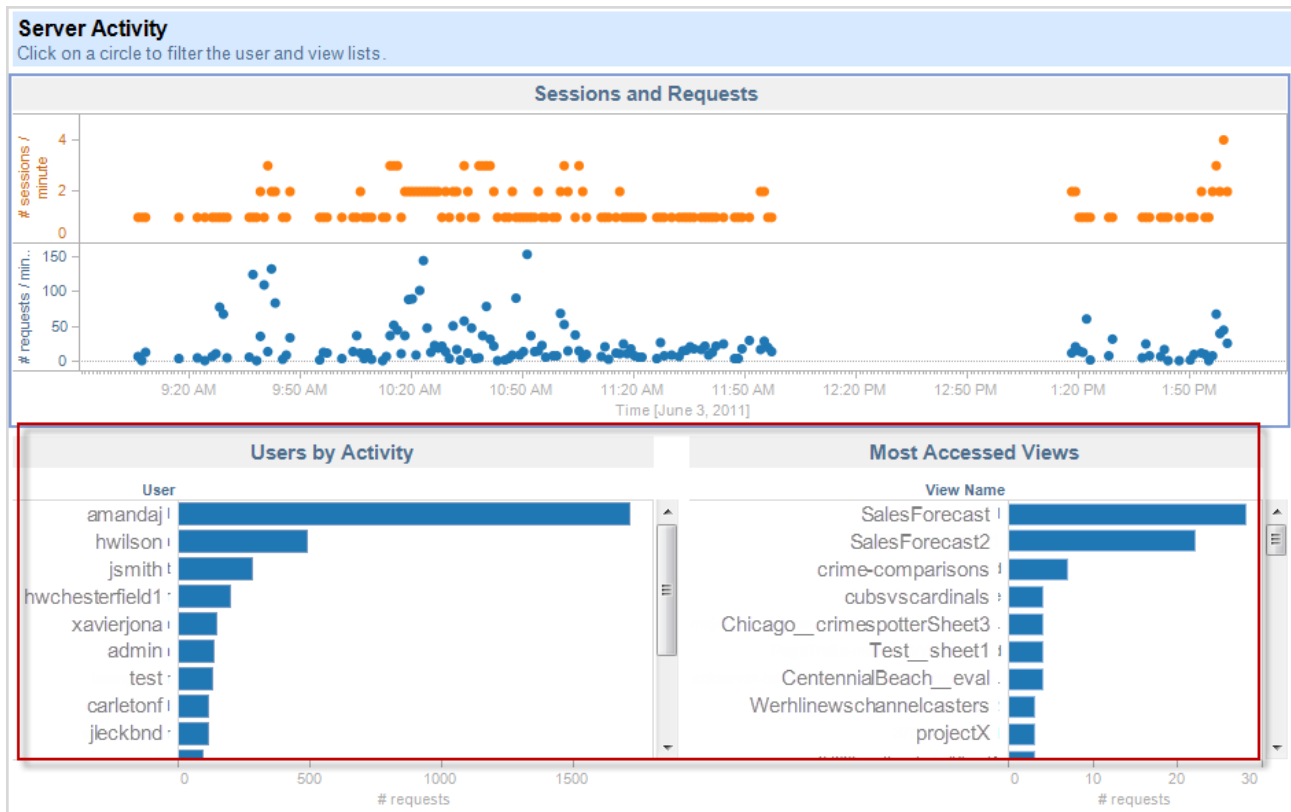
Click a circle to see who was active during that minute and which views they were accessing, if any:



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



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



User Activity

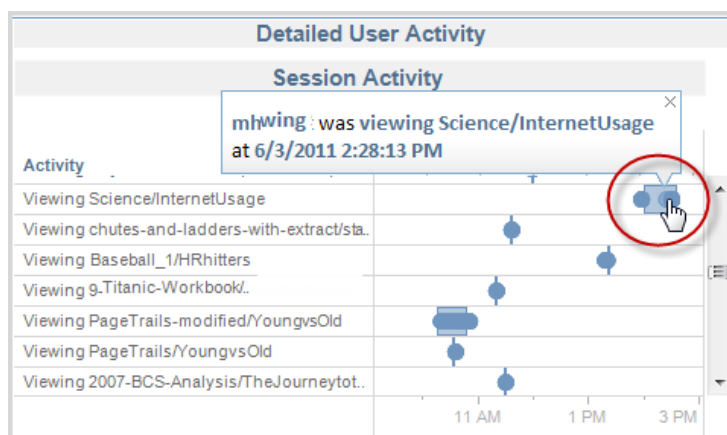
The User Activity view can help you gauge how heavily your Tableau Server installation is being used and whether you may need to buy additional licenses. Specifically, this view shows you who is logged into Tableau Server, from where, and when they last interacted with the server. If a user is logged in from multiple browsers, that will be displayed as well. For example, if a user logs in once from Internet Explorer and once from Mozilla Firefox, their name appears twice. If a user logs in twice from Mozilla Firefox, their name appears once.

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

Currently Active means that the user interacted with the server during the past five minutes.

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

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



Performance History

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

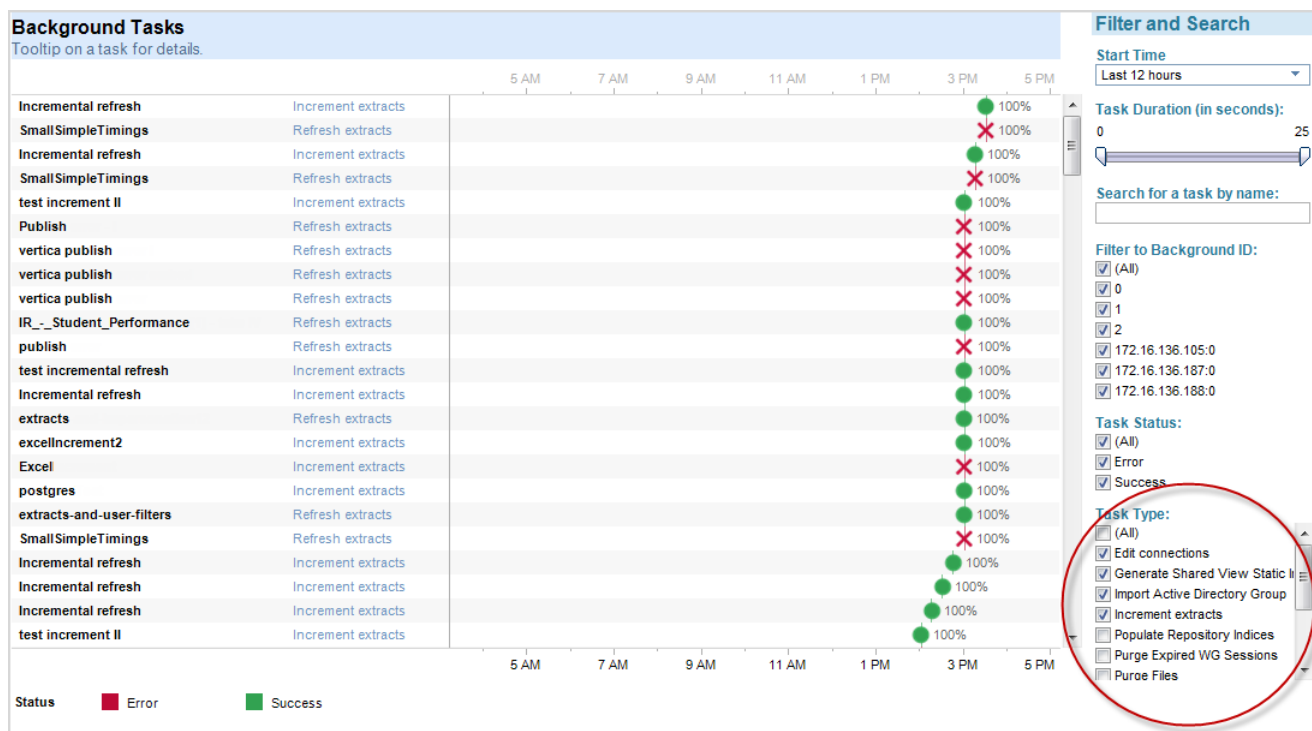
There are two different requests associated with views: Initial load requests, in orange, and compute requests, in blue. The latter are anything that causes Tableau Server to recompute what the user is seeing. This includes reload requests, and selecting and filtering items within a view.

Outlier marks represent requests with the biggest impact on server performance:



Background Tasks

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



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

Icon	Description
✗	Error – Server was unable to complete the task.
●	Success – Server completed the task.
▶	In process – Server is currently completing the task.
◆	Pending – A task that the server has not yet started.

For details on a task, hover over its icon:



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

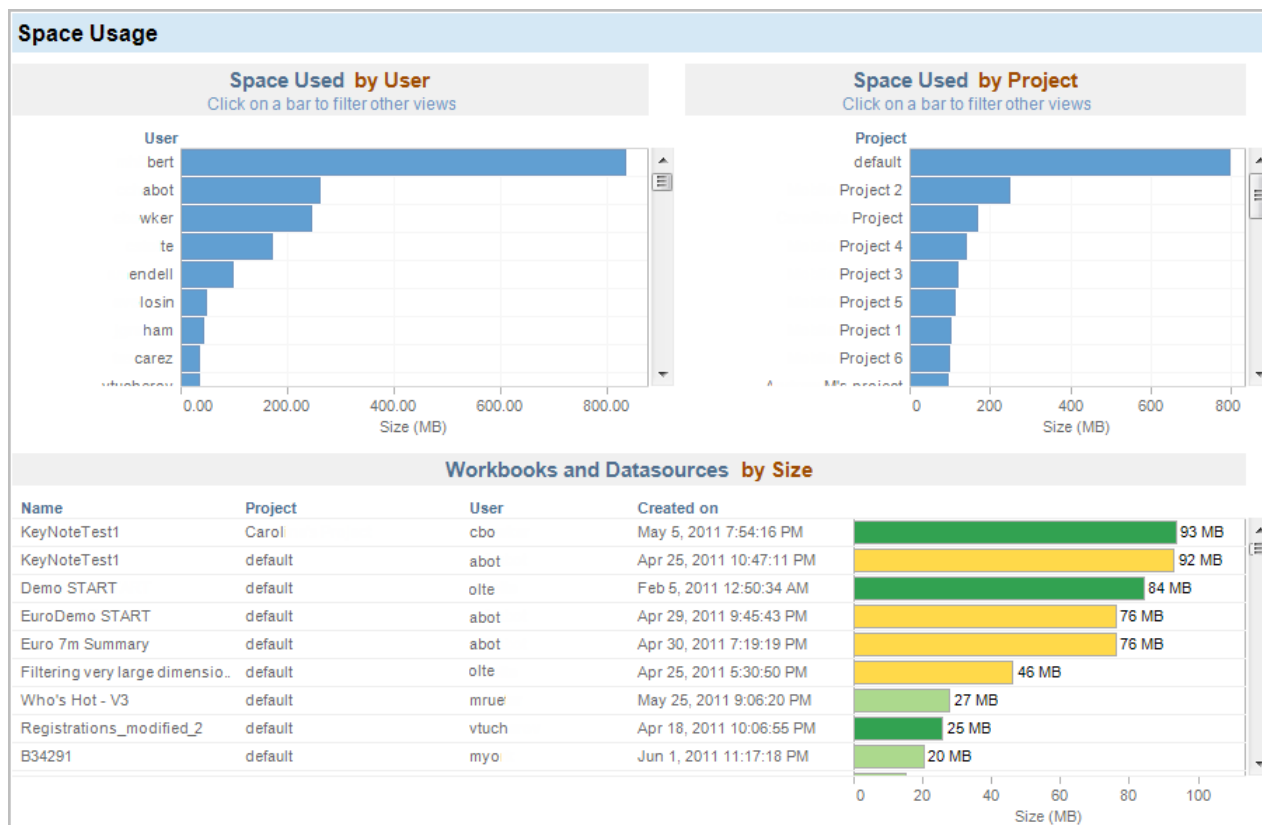
Filter to Background ID:

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

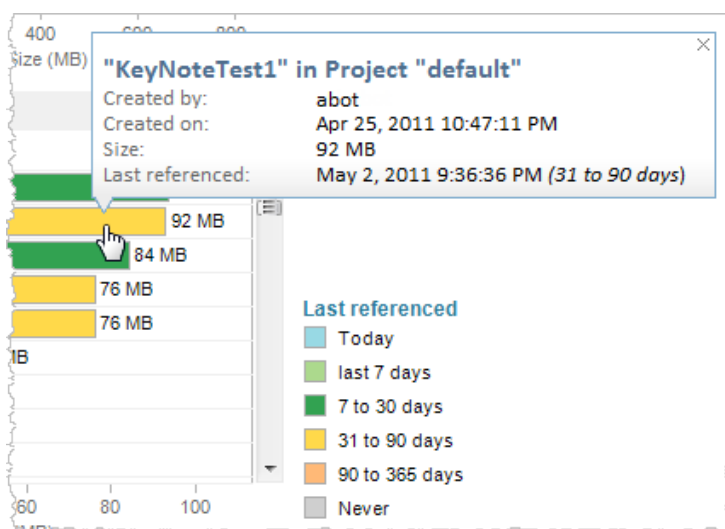
A multi-core machine running more than one background process will be listed with <IP address>:0 for the first process, <IP address>:1 for the second, and so on.

Space Usage

The Space Usage view can help you identify which workbooks and data sources are taking up the most disk space on your server. Disk space usage is displayed by user, project, and by the size of the workbook or data source:

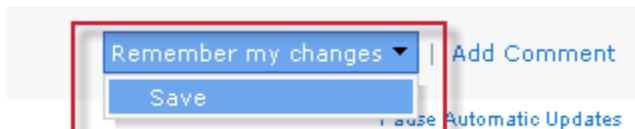


Move your cursor over any size bar to display usage details:



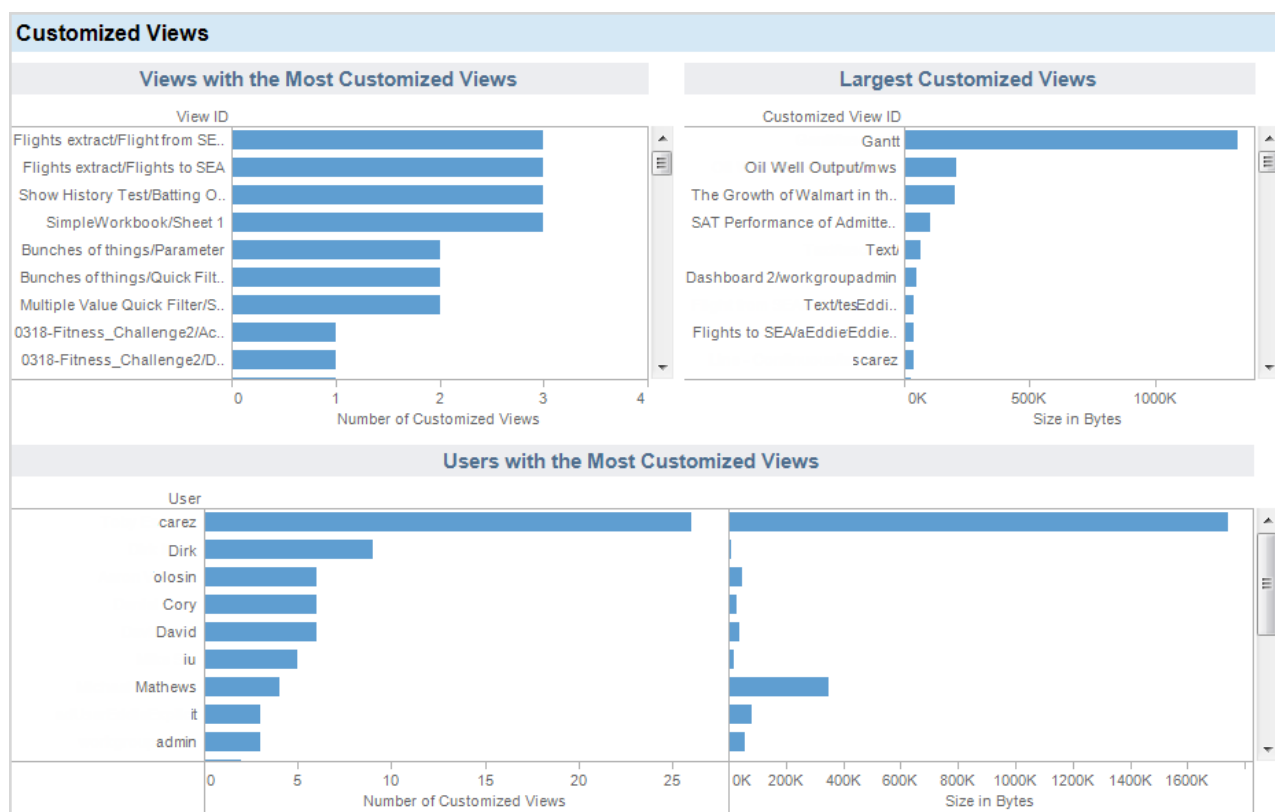
Customized Views

People working with views can use the **Remember my changes** option to save their customized views:



In addition, publishers can allow or prevent the sharing of customized views (see [Permissions](#)).

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



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. The Tableau Server repository has several database views set up that you can connect to and query. To access these views you must first use the command line tool to enable external access to the database.

To enable access to the Tableau Server database:

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

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Then use the following command to enable external access to the database for the user “tableau” with a password that you specify.

```
tabadmin dbpass [password]
```

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

3. Restart the server.

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

After you’ve enabled external access to the database you can connect to and query the database. Follow the steps below to use Tableau Desktop to connect.

To connect to and query the Tableau Server database:

1. In Tableau Desktop select **Data > Connect to Data**, then select **PostgreSQL** as the database to connect to. You may need to install the PostgreSQL database drivers. You can download drivers from www.tableausoftware.com/drivers.
2. In the PostgreSQL Connection dialog box, type the name or URL for Tableau Server.
3. Type 8060 for the port number.*
4. Type workgroup as the database to connect to.
5. Connect using the following username and password:

Username: tableau

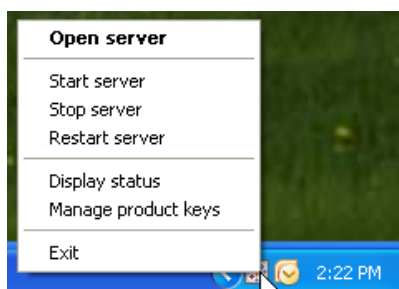
Password: the password you specified when you enabled access to the Tableau Server database.

6. Select a table to connect to and click **OK**. The “tableau” user has access to all of the tables the start with an underscore. For example, you can connect to “_background_tasks,” “_datasources,” and so on.

***Note:** You should connect using the port you have set up for the pgsqll.port, which is 8060 by default. Refer to [TCP/IP Ports](#) to learn more about managing and changing ports.

Tableau Server Monitor

Tableau Server Monitor is installed as part of Tableau Server and can be accessed in the Windows System Tray. Using this tool you can start and stop the services, launch Tableau Server, and display server status.



Open the Server

This command launches Tableau Server in your web browser. This is an easy way to access the web application and the associated maintenance tools.

Start/Stop the Server

You can start and stop the server using these commands. When you stop the server you make it unavailable to all of your users and terminate any sessions that are currently in progress. If someone is publishing a workbook when the server is stopped, the process is aborted. As a result, only some of the worksheets in the workbook may be published to the server. Because stopping the server can be very disruptive to your users, make sure to warn them prior to this operation or plan maintenance during non-business hours.

Restart 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 tool on Tableau Server.

Manage product keys

This command opens the product key manager where you can add and remove product keys.

Exit

This command closes Tableau Server Monitor. This command does not stop Tableau Server. You can re-open the application by selecting **All Programs > Tableau Server 6.1 > Tableau Server Monitor** on the Windows Start menu.

Command Line Tool - Tabadmin

When you install Tableau Server, a command line administrative tool, called `tabadmin.exe`, is also installed. You can use this tool to start, and stop the server in addition to several other commands. The instructions below describe the general syntax for using `tabadmin`.

At a command prompt type:

32-bit: `cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"`

64-bit: `cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"`

From there you can execute commands by typing: `tabadmin command options`

You can execute the following commands:

- `help` - shows help for `tabadmin.exe` commands
- `activate` - activates a license via online or offline activation.
- `administrator` - grant or revoke the Administrator right to a user
- `autostart` - display or set the auto-start behavior of the server
- `backup` - creates a back up of Tableau Server's data and configuration
- `cleanup` - cleans up service log files
- `configure` - updates the Tableau Server configuration (refer to [Appendix D](#) for options)
- `customize` - changes the name or logo used on the server
- `dbpass` - enables external access to the database for building administrative views
- `install` - install the Tableau Server service application
- `licenses` - shows information on the active licenses for Tableau Server
- `passwd` - resets the password for a Tableau Server account
- `prep_workers` - prepare a new worker for service by installing and configuring the software
- `reset` - resets the server back its initial state where an Administrator account must be set up
- `restart` - restarts the Tableau Services
- `restore` - restores a backup of Tableau Server's data and configuration
- `set` - sets a configuration parameter to a specified value or to its default value
- `start` - starts Tableau Services
- `status` - shows the current running status of Tableau Server
- `stop` - stops Tableau Services
- `uninstall` - uninstalls the Tableau Server service application
- `upgrade` - upgrades the service configuration and data to the current version of Server
- `warmup` - warms up the server by requesting a lightweight view from each VizQL process
- `ziplogs` - creates a zip file containing all of the log files

You can display detailed help and additional options for all of the commands described above by adding the `-h` option after the command (e.g., `tabadmin backup -h`).

You can use `tabadmin.exe` as part of scheduled tasks such as a regular backup or the Clean Up command, which clears out unnecessary logs and temporary files. Refer to [Logs and Temporary Files](#) to learn more.

Command Line Tool - Tabcmd

In addition to the tabadmin tool, Tableau Server comes with another tool to help you automate common tasks including batch publishing workbooks and user/group administration. This tool is called tabcmd.exe and is installed with Tableau Server. You can install this tool on other machines using the installer located at:

```
C:\Program Files (x86)\Tableau\Tableau Server\6.1\extras\TabcmdInstaller.exe
```

The instructions below describe the general syntax for using tabcmd.exe.

At a command prompt type:

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

From there you can execute commands by typing:

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

You can execute the following commands:

`help` - shows help for tabcmd.exe commands

`addusers` - adds users to a group

`creategroup` - creates a local group on the server

`createusers` - creates new users on the server

`delete` - deletes the given workbook from the server

`deletegroup` - removes a group

`deleteusers` - deletes users from the server

`get` - returns a file from the server

`login` - logs in to the server

`logout` - logs off the server

`publish` - publishes a workbook to the server

`refreshextract` - refreshes the extracts of a workbook on the server

`removeusers` - removes users from a group

`runschedule` - runs a schedule on the server

`set` - specifies a setting on the server

`syncgroup` - synchronizes the server with an Active Directory group

`version` - prints the version information

Refer to the full guide on this tool available on the [Product Manuals](#) page on the Tableau website.

Schedules and Tasks

When you publish a workbook that has an extract, you can schedule the extract to be automatically refreshed. There are two types of extract refreshes: full and incremental. An incremental refresh uses a specific column in your data extract to determine if new rows need to be added to the extract. A full refresh updates the entire extract.

Each type of refresh must be on a different schedule. For example, the full refresh can be performed once a week and the incremental can be handled each weekday morning. With this approach, you gain the performance benefits of Tableau's fast data engine, you restrict updating from the live database to only occur when database usage is low (for example, on Saturday evenings), and the data in your view stays fresh.

Enabling Scheduling

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

To enable scheduling:

Select **Scheduling** in the Settings table at the bottom of the Maintenance page.



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

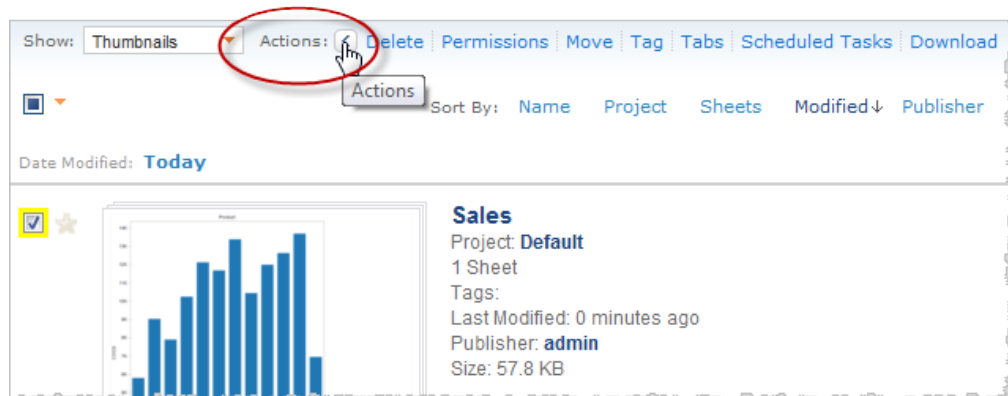
Note: Because database passwords may be required to refresh the extract, you must enable **Embedded Credentials** in order to allow scheduling.

Adding Workbooks to Schedules

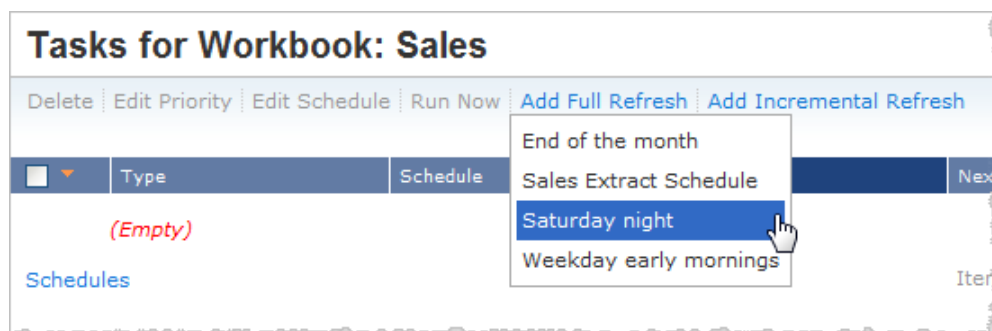
Once you've enabled schedules you can add workbooks to schedules from the Workbooks list. By default Tableau Server has three built in schedules. See [Managing Schedules](#) to learn how to create your own schedules.

To add a workbook to a schedule:

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



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



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

Managing Schedules

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

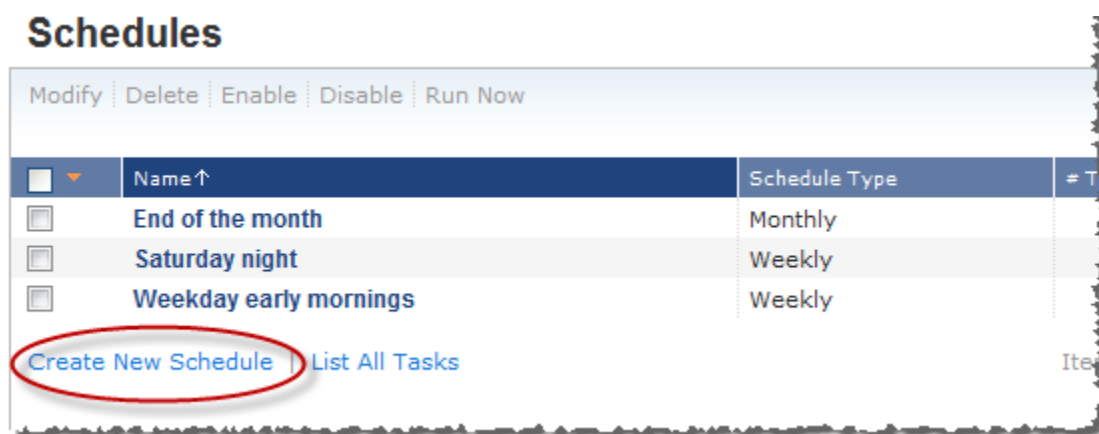
Tableau Server comes with three schedules. You can use the schedules as they are, modify them, or create your own. The Schedules page shows a list of schedules including their name, type, number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

To open the Schedules page:

- Click the **Maintenance** link in the Administration section on the left side of the page. Then click the **Schedules** link on the Maintenance page.

To create a new schedule:

- Click **Create New Schedule** at the bottom of the Schedules list.



- Specify a descriptive **Name** for the schedule (e.g., Every Saturday Morning, End of the Month).
- Optionally define a **Default Priority** from 0 to 100. This is the priority that will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the highest priority runs first. See [Managing Tasks](#) to learn more about modifying a task's priority.
- Choose whether the jobs in the schedule will run at the same time (concurrently, the default) or one after the other (sequentially).

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

Create New Schedule

Schedule Properties:

Schedule name:

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

Serialization: ☒ Jobs in schedule can run concurrently
☐ Run jobs sequentially

Schedule Definition:

☐ Hourly every from : to :

☐ Daily at :

☒ Weekly ☒ Sunday at :
☐ Monday
☐ Tuesday
☐ Wednesday
☐ Thursday
☐ Friday
☐ Saturday

☐ Monthly on the day of the month at :

[Create Schedule](#)

[Return to Schedules](#)

6. Click **Create Schedule**.

To modify a schedule:

1. On the **Schedules** page, select the schedule you want to modify. You can only modify one schedule at a time.
2. Click **Modify** in the toolbar.
3. Change the schedule as needed. When you're finished, click **Save Schedule**.

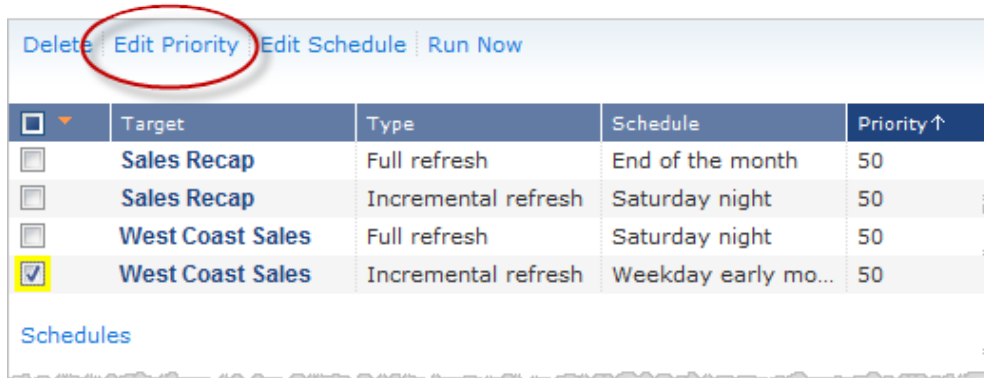
You can also enable, disable, delete, or run schedules by selecting one or more schedules in the list and selecting an option on the toolbar.

Managing Tasks

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

To change a task's priority:

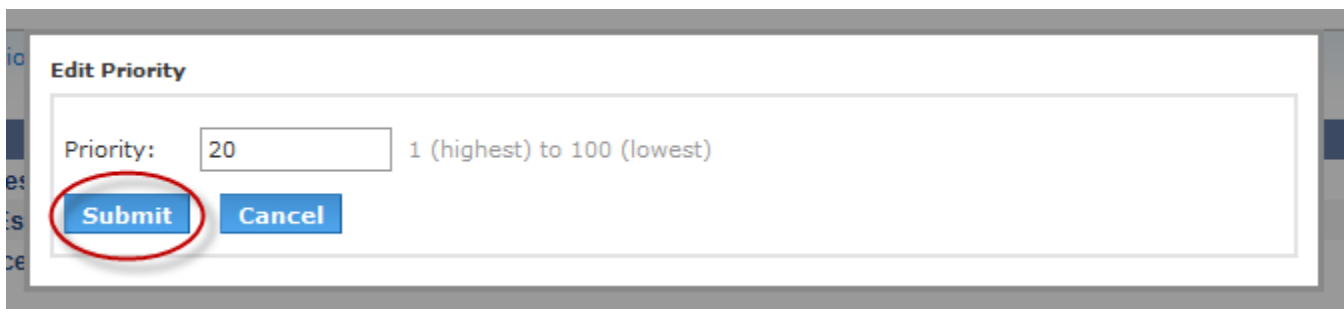
1. On the Tasks page select one or more tasks to modify.
2. Click **Edit Priority** on the toolbar.



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

Schedules

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



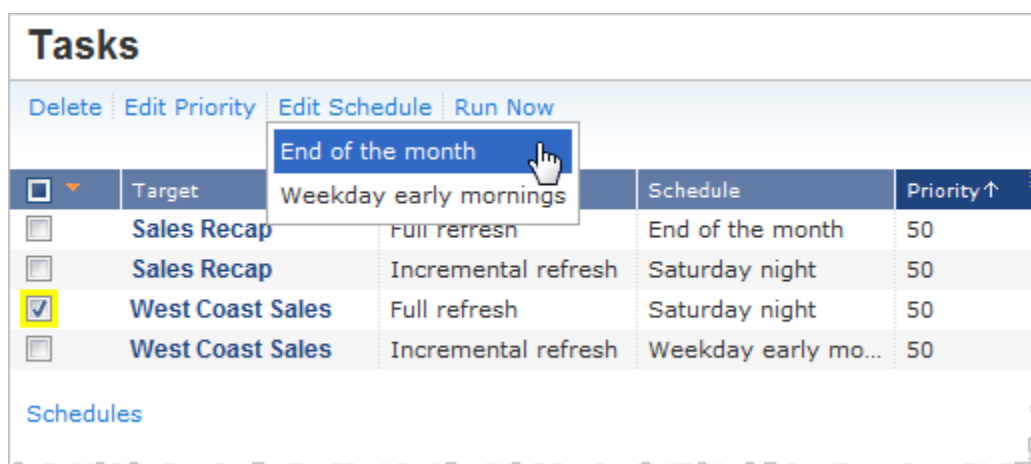
Edit Priority

Priority: 1 (highest) to 100 (lowest)

Submit **Cancel**

To move a task to a new schedule:

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



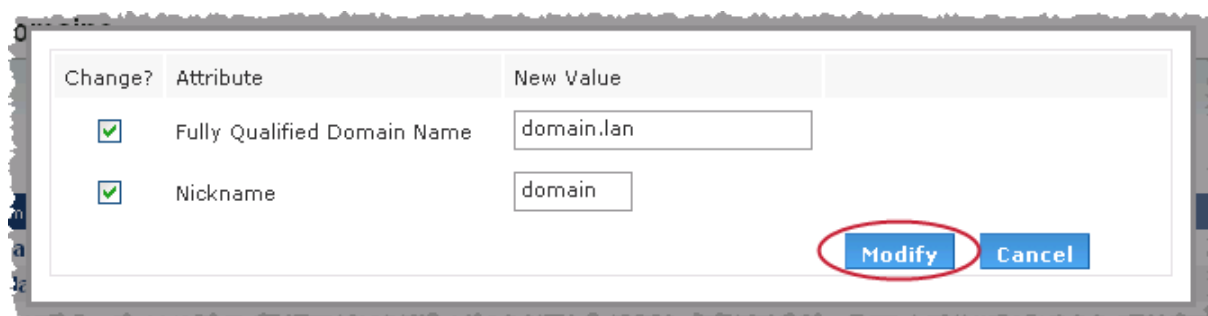
You can also delete and run tasks by selecting one or more tasks in the list and selecting an option on the toolbar.

Managing Domains

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

To access the list of domains:

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



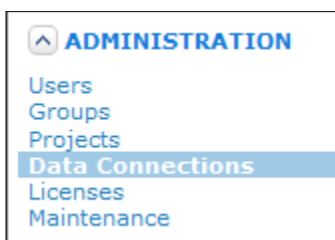
Change?	Attribute	New Value
<input checked="" type="checkbox"/>	Fully Qualified Domain Name	domain.lan
<input checked="" type="checkbox"/>	Nickname	domain

Modify **Cancel**

Note: You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to log in. However, if the user name that you are currently logged in with exists in both the current domain and the new domain you can modify the full name for the current domain.

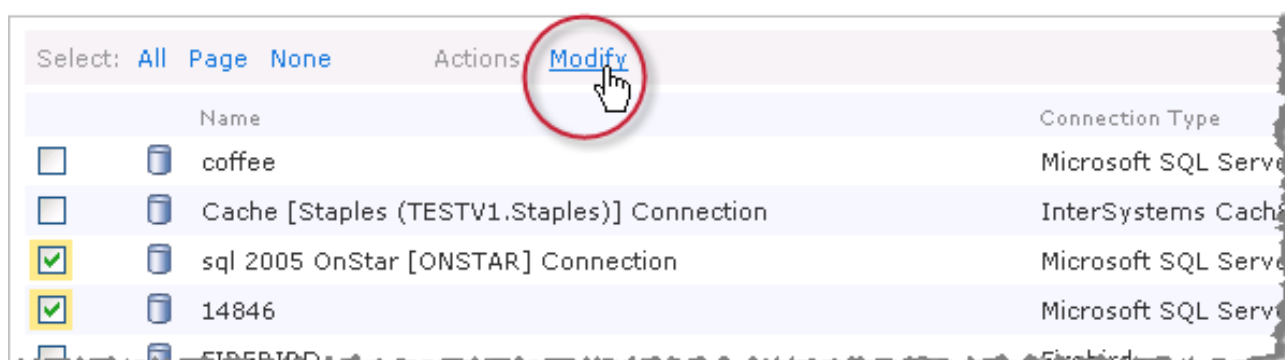
Managing Data Connections

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



Data connections are different from data sources in that each connection is associated with a single workbook and describes the attributes required for connecting to a data source (e.g., server name, database name, etc.). That means if you have three workbooks that connect to the same data source, you will still have three connections listed on the connections page.

The Data Connections page allows you to manage the connection information for all of the workbooks published to the server. For example, you may have a large number of workbooks that connect to a database on a specific server. If the name of the server changes, you can update all of the workbooks at once so they reference the new server name. Another example is if a workbook connects to a data source using a specific user name and password. You can quickly update all of the workbooks to use a different set of credentials.



Refer to the online help for more information about finding and modifying connections on the server. You can access the help by clicking the Help link at the top of the Data Connections page.

Backing Up and Restoring the Database

Use the command line tool to back up and restore the database. You can use the commands described below along with the built-in Windows task scheduler to automate back-ups on a regular schedule.

Backing Up

It is important to back up the database so you can restore the published views in the case of a system failure. When you back up the database a single file is created with the .tsbak file extension. This file contains the contents of the database and the configuration files.

To back up the database:

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

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"  
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Create a backup file by typing:

```
tabadmin backup filename --stop-server -d
```

In the above line, replace “filename” with what you want to name the backup file. The --stop-server option stops the server for the backup and then restarts it when it is done. If you are doing a simple backup you must either have this option or the --unsafe option to backup while the server is running. Backups that are part of a batch operation do not require these options.

The -d is optional; if included the current date is appended to the file name.

Restoring from a Backup File

When you restore, the contents of the database as well as configuration files are overwritten with the content in the backup file.

To restore from a backup file:

1. Stop the service by typing:

```
tabadmin stop
```

2. Restore the database from a backup file by typing:

```
tabadmin restore filename
```

In the above line, replace “filename” with the name of the backup file you want to restore from.

3. When finished, start the services by typing:

```
tabadmin start
```

Caution: If you are using an external extract storage database, you will need to back it up separately to avoid having to republish all of the workbooks in the case of a restore. The backup file does contain extract data stored in the built-in data engine. Refer to [Publishing with Extracts](#) to learn more.

Customizing Tableau Server

You can customize how Tableau Server looks to personalize it for your company or group.

Server Look and Feel

You can change the following elements:

Name - text that appears in screen tips, warning messages, and error messages. The default name is Tableau Server.

Logo - image that appears on login page and in the left column of most pages. The logo shown when you open an individual view cannot be changed.

To change the name:

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

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Change the name by typing the following:

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

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

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

```
tabadmin restart
```

To change the logo:

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

32-bit: `cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"`

64-bit: `cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"`

2. Change the logo by typing the following:

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

In the above line, replace “image” with the filename of the image that you want to appear as the name on the server. For best results, use an image that is 125px X 35px in size. The image can be .png, .jpg, or .gif file format.

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

```
tabadmin restart
```

To restore the default settings:

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

```
tabadmin customize parameter -d
```

In the above line, replace “parameter” with what you want to restore (either `name` or `logo`).

2. Restart the server for the change to take effect.

User Interaction

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



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

Embedded Credentials - Allow publishers to attach passwords to published workbooks that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks and are only accessible on server. That is, when the workbook is opened in Tableau Desktop, users will still need to enter a user name and password to connect to the data source. When this setting is turned off, all existing embedded passwords are saved but are not used for authentication. That way if you turn the setting back on, users don't have to re-embed the passwords.

Scheduling - Allow publishers to assign workbooks to schedules. This option is only available if you have configured [Extract Storage](#) on the server and **Embedded Credentials** is enabled.

Public User List - Allow web users to see a list of all users on the system. When this setting is turned on a link to a list of all users is added to the left navigation bar.

Saved Passwords - Allow users to save data source passwords across multiple visits and browsers. By default users can choose to "Remember my password until I log out," which lets them save their password during a single browser session. When the Saved Passwords setting is selected a user can instead choose to "Remember my password," which saves the password across multiple visits and browsers so users will be automatically authenticated regardless of the computer they are using. You, as an administrator, can clear all saved passwords at any time. In addition, users can clear their own saved passwords.

Enable Guest - Allow users to view and interact with embedded views without having to log into a Tableau Server account. Permission can be assigned to the Guest user account to control the interactivity allowed for each view. This option is only available if you have a core-based server license.

Note: If you clear the **Enable Guest** checkbox then can't select it again because it is grayed out, click the **Reset all settings to their defaults** link.

TCP/IP Ports

Tableau server uses a variety of TCP/IP ports by default:

80, 8060, 8080, 8085, 3730

8000 - This is the base port for the application server. In addition, the consecutive ports after 8000 up to the number of processes (specified when configuring Tableau Server) are used. For example, if the application server is configured to use four processes, ports 8000, 8001, 8002, and 8003 are used.

8100 - This is the base port for the VizQL server. In addition, the consecutive ports after 8100 up to the number of processes (specified when Configuring Tableau Server) are used. For example, if the VizQL server is configured to use 4 processes, ports 8100, 8101, 8102, and 8103 are used.

27000 – 27009 - This is the range of ports used for licensing communication between workers and the primary machine in a distributed environment.

1070 – 1821 - This is the range of ports used for Distributed Ruby in a distributed environment.

Editing the Default Ports

You can modify the default ports using the command line administrative tool, tabadmin.exe (refer to [Command Line Tool - Tabadmin](#) to learn about other commands available). Follow the steps below to change the Tableau Server port configuration.

To modify a default port setting:

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

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Modify a specific port value by typing the following:

```
tabadmin set parameter new_value
```

In the above line, “parameter” is one of values in the following table and “new_value” is the new port number you want use.

You can use the following parameters to modify the corresponding ports:

Port to Change	Parameter
80	gateway.port
8060	pgsql.port
8080	solr.port, tomcat.http.port, repository.port*
8085	tomcat.server.port
8000	wgserver.port
8100	vizqlserver.port

*These parameters should all be set to same value.

For example, if you have a conflict on port 8000 and would like to configure Tableau Server to use port 8020 you would type the following:

```
tabadmin set wgserver.port 8020
```

3. Make all of the necessary port configuration changes. Then restart the services by typing the following:

```
tabadmin restart
```

Caution: 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.

To restore the default values for any port:

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

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Restore the default by typing the following:

```
tabadmin set parameter --default
```

Where “parameter” is the before listed parameter that corresponds to the port setting you want to restore to the default value for.

3. When finished, restart the services by typing the following:

```
tabadmin restart
```

Logs and Temporary Files

The Tableau Service generates several logs and temporary files that can help you understand and track recent activity as well as debug any problems that may arise. If you need to save space on the hard drive, you can occasionally delete these files. Below is a description of where these files are located and what they are used for.

Tableau Service Logs

The following log files track activities related to the web application, database, and index:

Windows XP: `C:\Program Files\Tableau\Tableau Server\data\tabsvc\logs`

Windows Server 2008, Windows Vista, and Windows 7: `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:

Windows XP: `C:\Program Files\Tableau\Tableau Server\data\tabsvc\vizqlserver\Logs`

Windows Server 2008, Windows Vista, and Windows 7: `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.

Windows XP: `C:\Documents and Settings\username\Local Settings\Temp`

Windows Server 2008, Windows Vista, and Windows 7: `C:\ProgramData\Tableau\Tableau Server\temp`

Archiving the Logs

You can archive all of these log files using the `ziplogs` command. This command creates a zip file containing all of the Tableau Server Log files and is useful when you're working with Tableau support. The `ziplogs` command does not remove the log files, rather it copies them into a zip file.

To create a zip file containing all logs:

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

```
32-bit: cd "C:\Program Files\Tableau\Tableau Server\6.1\bin"
```

```
64-bit: cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Stop Tableau Server by typing:

```
tabadmin stop
```

3. Create the zip file by typing `tabadmin ziplogs -l <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 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 -d 03/12/2011
```

The above command creates a zipped file named `logs.zip` that includes logs dated March 12, 2011 up to the present; earlier logs are excluded. To see a list of all the `ziplogs` options, type `tabadmin ziplogs -h`.

4. Restart Tableau Server by typing:

```
tabadmin restart
```

You can find the zipped log file in the Tableau Server bin directory.

Clean Up

The `clean up` command removes service logs in order to save space. At a command prompt type the following:

```
tabadmin cleanup --restart
```

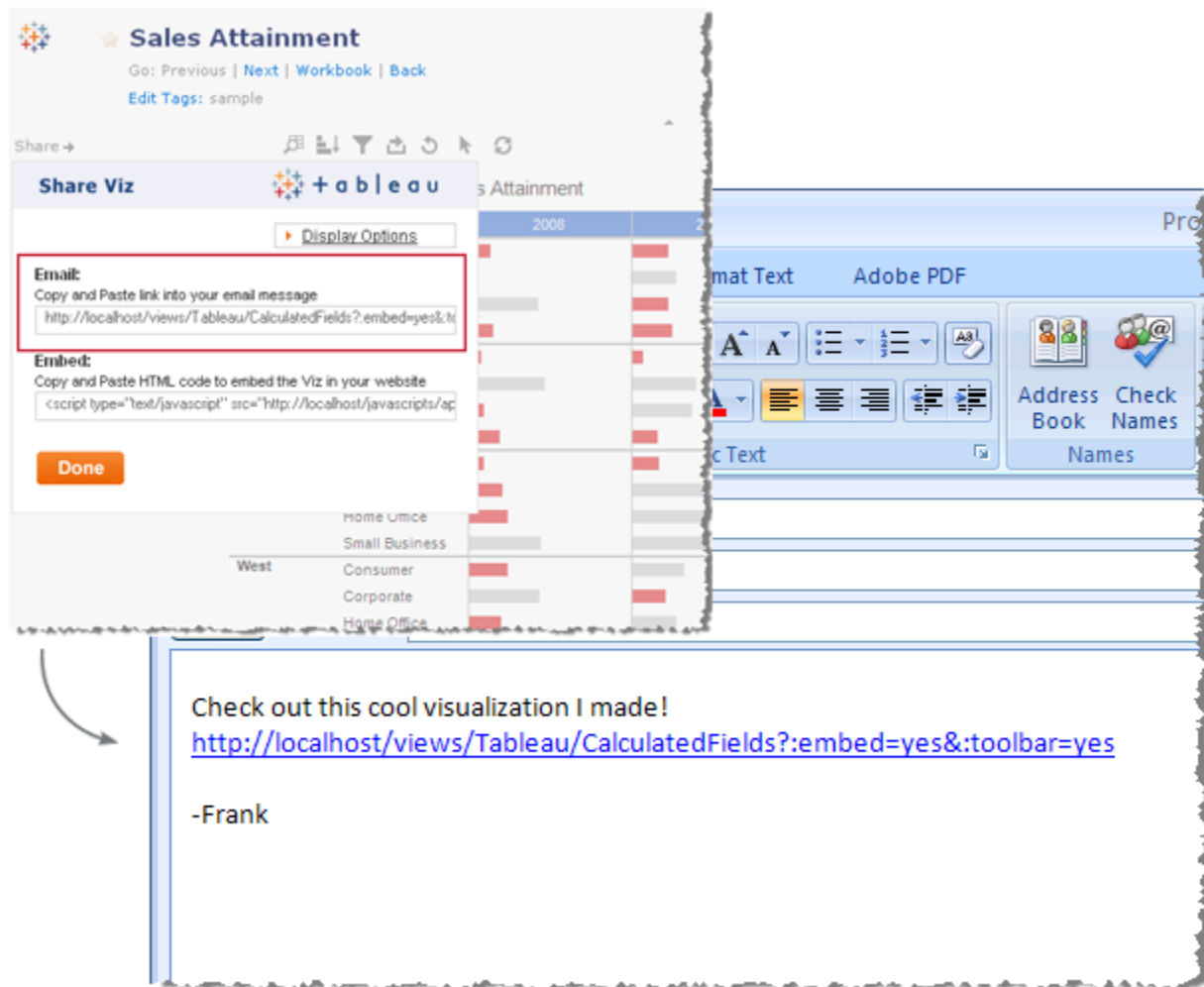
Note: In general you should shut down the server prior to running this command. However, if the server is running you should include the `--restart` option to ensure a successful clean up and restore.

Sharing Views

You can easily share your Tableau Server views with others using the Share feature. Clicking the Share button generates simple links that you can either embed in another blog or webpage or email to a friend or co-worker.

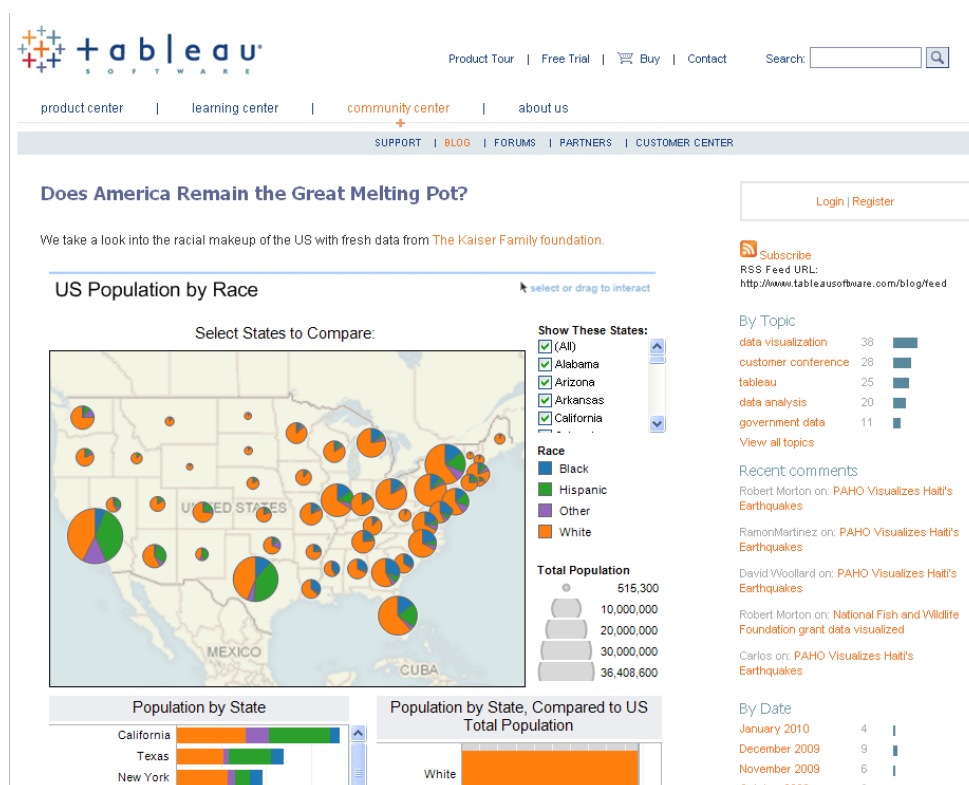
To email views:

1. Navigate to the view on Tableau Server that you want to email.
2. Click the **Share** link in the upper left corner of the view. This button is in the lower left corner for Guest users and embedded views.
3. Copy and paste the email link into your email message.



Embedding Views

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

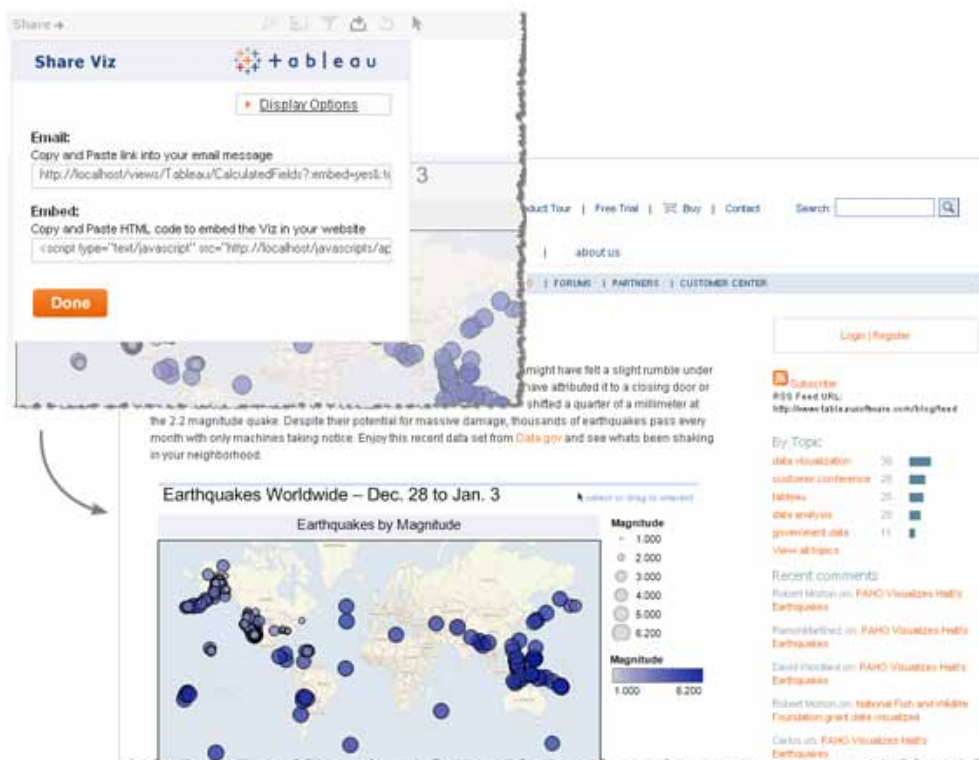


There are three ways you can embed views:

- **Use the Share embed code as-is:** The Share link in the upper left corner of each view provides automatically-generated embed code. All you have to do is copy the code and paste it into your webpage. See [To embed views](#) for steps.
- **Write your own embed code:** You can enhance the default embed code Tableau provides or you can build your own code. Either way you can add parameters that control comments, toolbars, and more. See [Writing Embed Code](#) for details.
- **Use the Tableau JavaScript API:** You can use Tableau JavaScript objects in your own web application code. See [Using the Tableau JavaScript API](#).

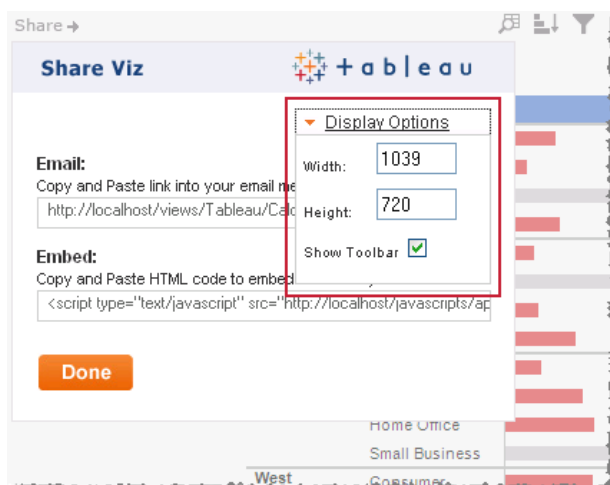
To embed views:

1. Navigate to the view on Tableau Server that you want to embed and click the **Share** link in the upper left corner of the view. For guest users and embedded views, this button is in the lower left.
2. Copy and paste the Embed code into your webpage.



Display Options

When emailing or embedding views, you can optionally adjust the display options including width, height and whether to show the toolbar. The width and height show the current size of the view by default.



Writing Embed Code

If you're writing your own embed code, you can take one of two approaches:

- **Use Tableau JavaScript:** This is the preferred approach. Just use the Share embed code as the starting point for your own code, adding or editing object parameters that control comments, toolbars, and more. The default embed code, which relies on a Tableau JavaScript file, is also the only way to control the load order of multiple embedded views.
- **Specify the View URL:** As with earlier versions of Tableau, you can embed a view using an IFrame or Image tag, where the source is the raw URL for the view. You may want to do this if you can't use JavaScript at your web site. There may also be situations where all you can specify is an URL—such as if you're embedding a view using SharePoint's Page Viewer Web Part.

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

Tableau JavaScript

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

```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<div class="tableauPlaceholder" style="width:800; height:600;">
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="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` object parameter is the only required parameter; all other parameters are optional. For examples, see the [Parameter List](#) table and the "Script Tag Examples" in the following pages.

View URL as the Source

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

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

You must specify the `embed` URL parameter and can optionally include parameters that control comments, toolbar, and revert, among others. You can also add filters to the URL that control the specific data that shows when a view is loaded. For examples, see the [Parameter List](#) table and the "Iframe Tag Examples" in the following pages.

Parameter List

Object Parameter	URL Parameter	Values	Description	Examples
comments	:comments	yes; no; ro	Hide and show comments. Comments are hidden by default if the parameter is not set. When set to "ro" value shows the comments as read-only.	<code><param name="comments" value="no"/></code> <code>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:comments=yes</code>
–	:embed	yes	Required for URL parameter. Hides the top navigation area, making the view blend into your web page better.	<code>http://tabserver/views/Date-Time/DateCalcs?:embed=yes</code>
filter	–	string	Customizes what is displayed when the view opens.	<code><param name="filter" value="Team=Blue"/></code>
linktarget	:linktarget	string	The target window name for external hyperlinks.	<code><param name="linktarget" value="_blank"/></code> <code>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:linktarget=_blank</code>
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.	<code><param name="load-order" value="2"/></code>
name	–	string	Required for object parameter. Workbook and sheet name.	<code><param name="name" value="MyCoSales/Sales"/></code>
path	–	string	For trusted authentication only, cannot be used with the "ticket" parameter. Overrides value of the "name" parameter and is used as the URL. See the Trusted Authentication Examples .	<code><param name="path" value="trusted/123456789/views/workbookQ4/SalesQ4"/></code> <code>http://tableauserver/trusted/123456789/views/workbookQ4/SalesQ4?:embed=yes&:tabs=yes</code>
–	:refresh	yes; no	Renders the page.	<code>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:refresh=yes</code>
–	:revert	yes; no	Return the page to its original form.	<code>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:revert=yes</code>
tabs	:tabs	yes; no	Displays or hides tabs.	<code><param name="tabs" value="yes"/></code>
ticket	–	number	For trusted authentication only, cannot be used with the "path" object parameter. Must be used with "name" object to construct the trusted ticket redemption URL. See the Trusted Authentication Examples .	<code><param name="name" value="workbookQ4/SalesQ4"/></code> <code><param name="ticket" value="123456789"/></code> <code>http://tableauserver/trusted/123456789/views/workbookQ4/SalesQ4?:embed=yes&:tabs=yes</code>
toolbar	:toolbar	yes; no	When yes the toolbar is included with the embedded view. The toolbar is shown by default if this parameter is not set.	<code><param name="toolbar" value="yes"/></code> <code>http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:toolbar=yes</code>

Adding Filters

In addition to the basic set of parameters for an embedded view, you can also pass filter values so the view opens showing just the data you want. For example, you may want to include a hyperlink from another part of your web application to an embedded sales performance view that only shows a specific region.

Script Tag Example

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

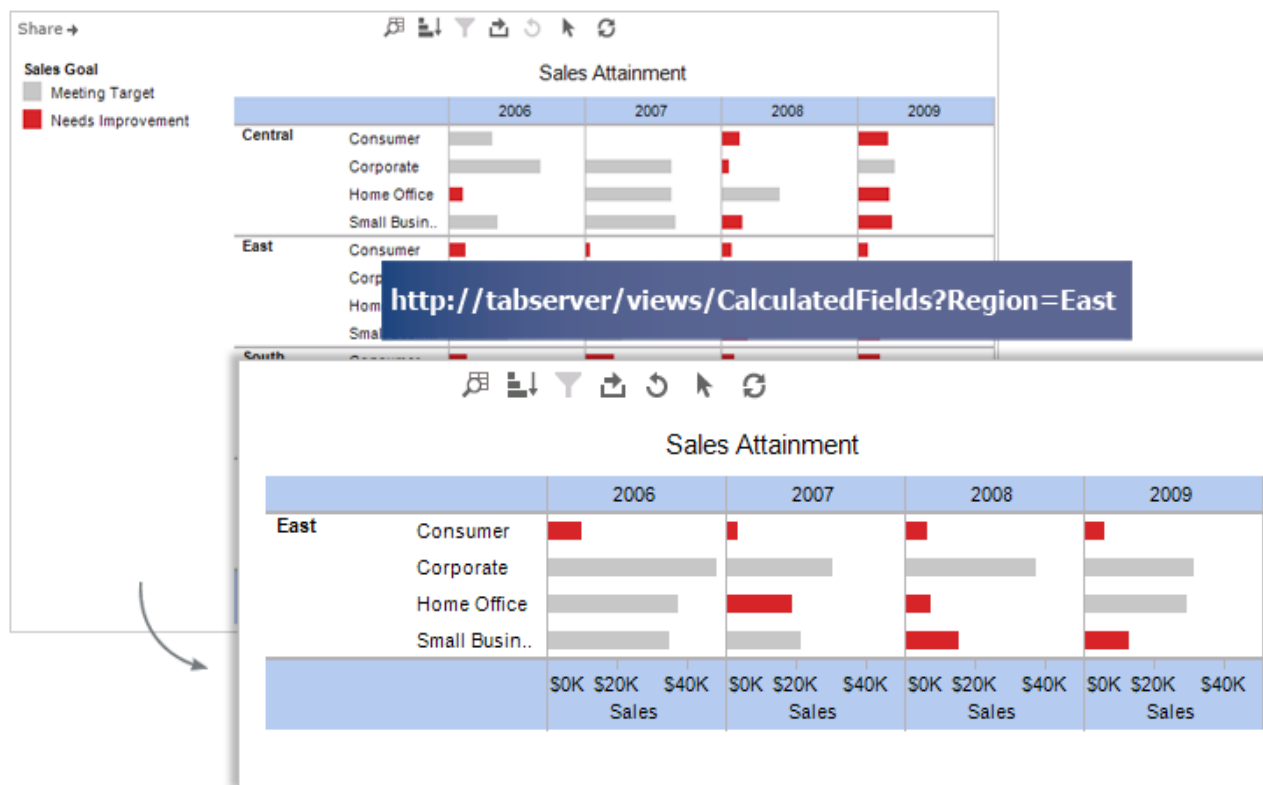
To pass through multiple filters, just separate each value with a comma. For example:

```
<param name="filter" value="Region=East,West" />
```

Iframe Tag Example

```
<iframe src="http://myserver/views/CalculatedFields?:embed=yes&Region=East"
width="800" height="600"></iframe>
```

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



Filtering on Multiple Fields

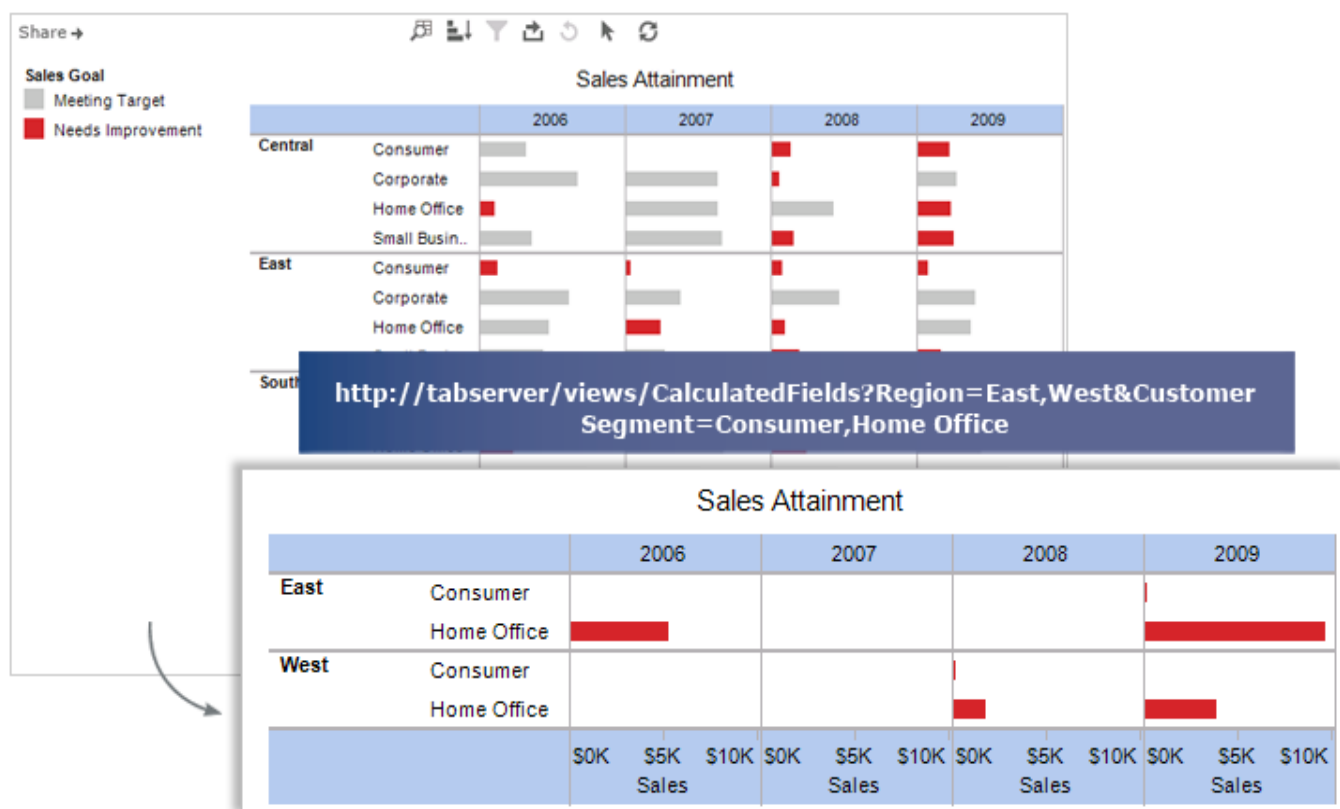
You can pass filters on as many fields as you want, including fields that are not in the original view.

Script Tag Example

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

Iframe Tag Example

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



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

Filtering Dates and Times

If you want to filter on a Date/Time field you should include the value using the default Tableau format shown below:

```
yyyy-mm-dd hh:mm:ss
```

The time part uses a 24-hour clock. Many databases store all date values as Datetime fields, so you may need to pass a time value along with your date.

Script Tag Example

```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="Sales/Sales-Performance" />
  <param name="filter" value="Date=2008-12-01" />
</object>
```

This example filters on both a date field and a datetime field:

```
<param name="filter" value="2008-12-01%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.

Filtering Measures

You can filter measures by including one or more values. There is no support for greater than, less than, or ranges. The example below filters to show only \$100 and \$200 sales.

Script Tag Example

```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="name" value="Sales/Sales-Performance" />
  <param name="filter" value="Profit=100, 200" />
</object>
```

Iframe Tag Example

```
<iframe src="http://myserver/Sales/Sales-Performance?:embed=yes&Profit=100,
200" width="900" height="700"></iframe>
```

Controlling Load Order for 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 have load order parameters, they are loaded in the order in which they appear on the page.

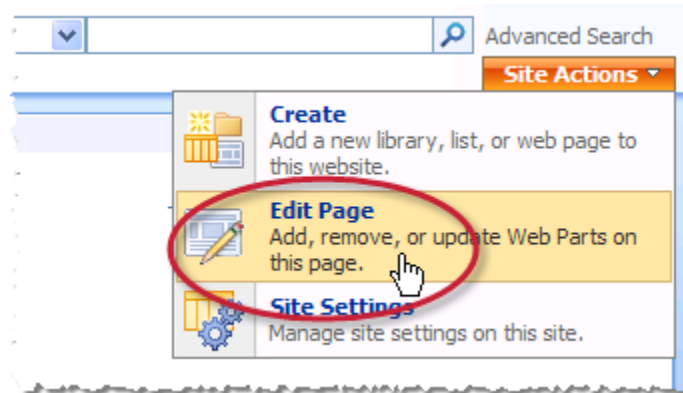
```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="600" height="400" style="display:none;">
  <param name="name" value="MyCoSales/TopPerformers" />
  <param name="tabs" value="yes" />
  <param name="toolbar" value="yes" />
  <param name="comments" value="no" />
  <param name="filter" value="Salesperson=Top 5" />
  <param name="load-order" value="0"
</object>
```

```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js">
</script>
<object class="tableauViz" width="600" height="400" style="display:none;">
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="tabs" value="yes" />
  <param name="toolbar" value="yes" />
  <param name="comments" value="no" />
  <param name="load-order" value="-1"
</object>
```

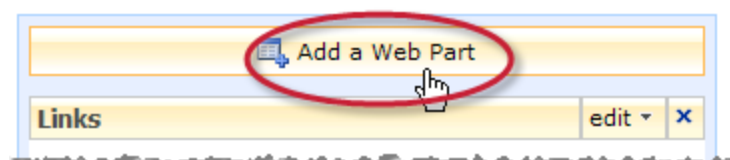
Example 1: Embedding Views into SharePoint (Microsoft SSPI)

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

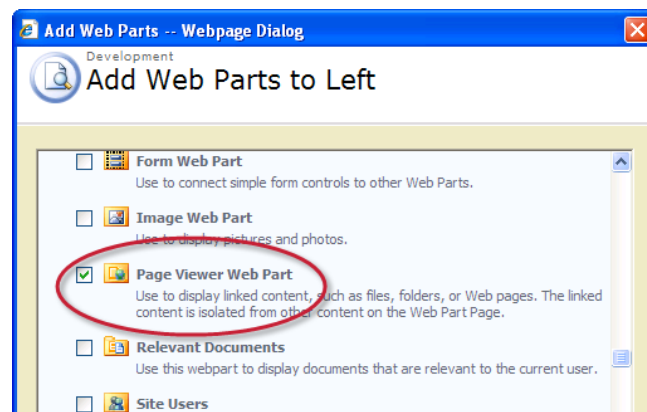
1. Navigate to the SharePoint page that you want to embed a view into.
2. On the Site Actions menu in the upper right corner of the page select **Edit Page**.



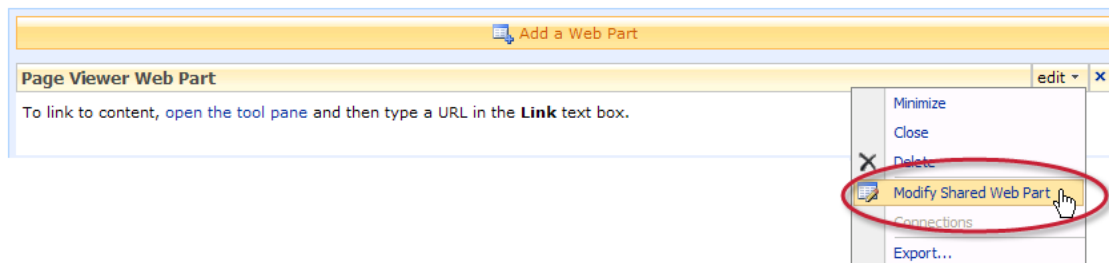
3. Click the **Add a Web Part** button in the section of the page where you want to embed the view.



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

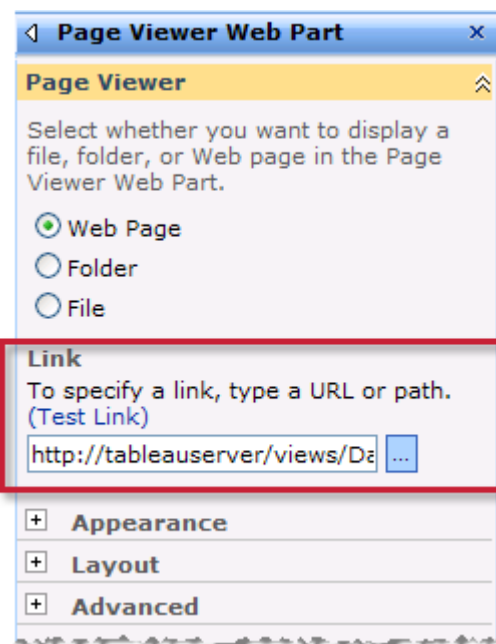


- Back on the SharePoint page select **Modify Shared Web Part** on the **Edit** menu for the new web part.

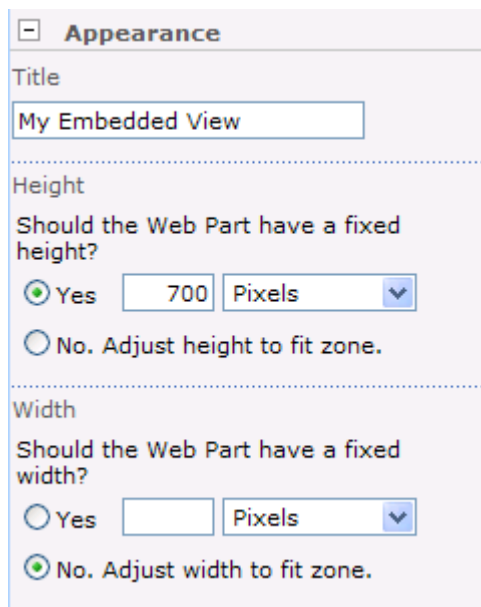


- On the right side of the page, you can specify the attributes of the Page View Web Part. Type the URL for the view you want to embed. Use the format specified in the Embedding Views section of this document. For example you may type:

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



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



The screenshot shows the 'Appearance' configuration panel for a Tableau web part. It has a title 'My Embedded View'. Under the 'Height' section, the option 'Should the Web Part have a fixed height?' is selected with 'Yes', and the value '700' is entered in the 'Pixels' dropdown. Under the 'Width' section, the option 'Should the Web Part have a fixed width?' is selected with 'No. Adjust width to fit zone.'.

- When finished, click **OK** to apply the changes and exit edit mode.

The view will be embedded into the web part that you just created. Your users will not need to log in to Tableau Server to see the embedded view, rather they will be automatically authenticated using Microsoft SSPI.

Example 2: Embedding Views into Wikis

You can easily embed a view into a wiki or other web page simply by putting the view inside an `<iframe>` tag.

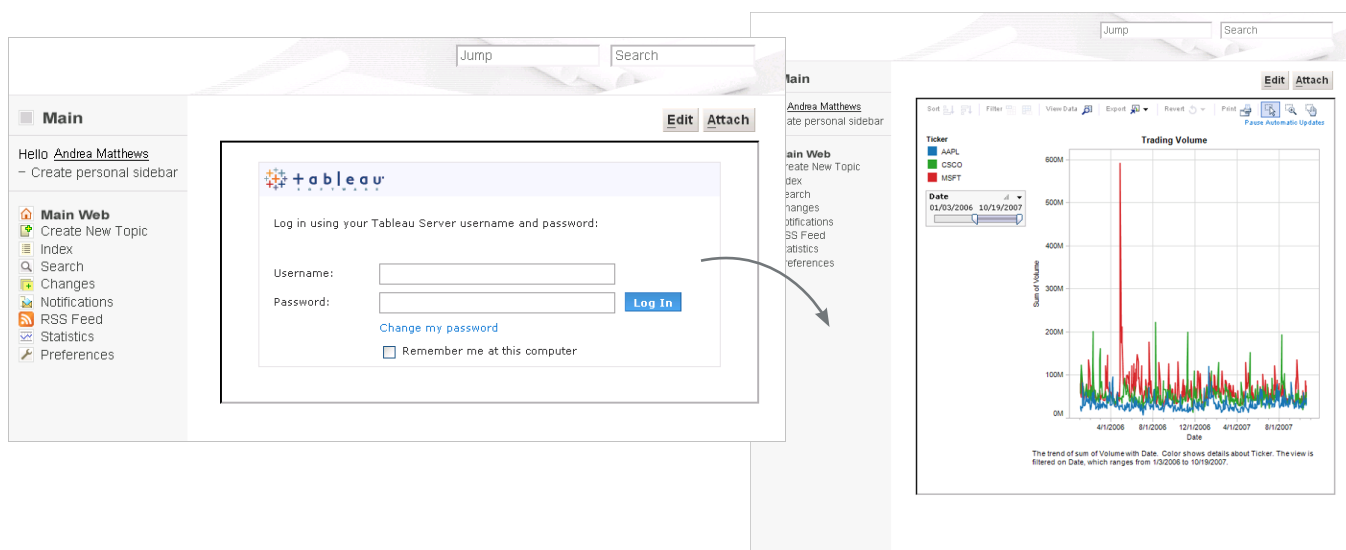
1. Navigate to the wiki page you want to embed a view into.
2. Edit the page and add an `<iframe>` where the source is the URL for the view. For example:

```
<iframe src="http://tableauserver/views/Date-Time/DateCalcs?:embed=yes&:comments=ro
&:toolbar=no" width="900" height="700"></iframe>
```

3. Save your changes.

The view is embedded into the wiki page. If both Tableau Server and the wiki are both configured to use Microsoft SSPI, users accessing an embedded view on the wiki will be automatically logged in so they can see the view.

If the server and the wiki are not using the same method for authentication, users will first be asked to log into the server before they can see the view.



Example 3: Embedding Images

In addition to embedding a view into a <script> or <iframe> tag you can also embed the view as an image. When you embed an image the view is not interactive, however, it is updated every time the page fully reloads. That way the image shows the latest data even if the underlying data changes.

1. Navigate to the page where you want to embed the image.
2. Edit the page and add an tag where the source is the URL for the view plus the .png file extension. For example:

```

```

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

Example 4: Embedding Views into SharePoint (Trusted Authentication)

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

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

```
C:\Program Files\Tableau\Tableau Server\6.1\extras\embedding\sharepoint\
```

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

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

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

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

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

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

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

- Copy the TableauEmbeddedView.dll file into your C:\Windows\assembly folder and delete it from the bin file you copied it into in step 2 above.

- Reopen the web.config file you opened in step 3 above and find the following line:

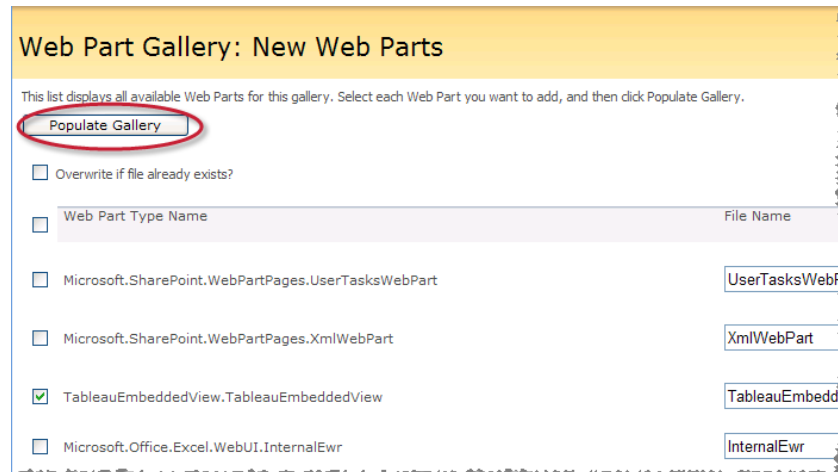
```
<trust level="WSS_minimal" originUrl="" />
```

Change the line above to the following:

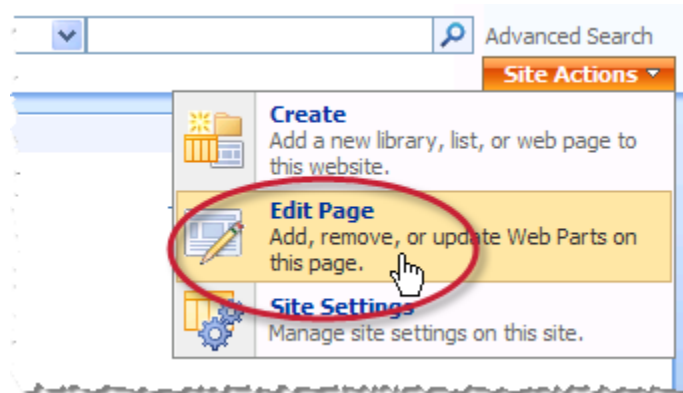
```
<trust level="Full" originUrl="" />
```

- Create a custom trust policy, which will grant full access to the TableauEmbeddedView.dll only. Refer to the [Microsoft Technical Article](#) to learn more about how to do this.

6. Open a browser and navigate to: `http://<your_sharepoint_server>/_layouts/newdwp.aspx`.
7. Select the entry titled `TableauEmbeddedView.TableauEmbeddedView` and click the **Populate Gallery** button.



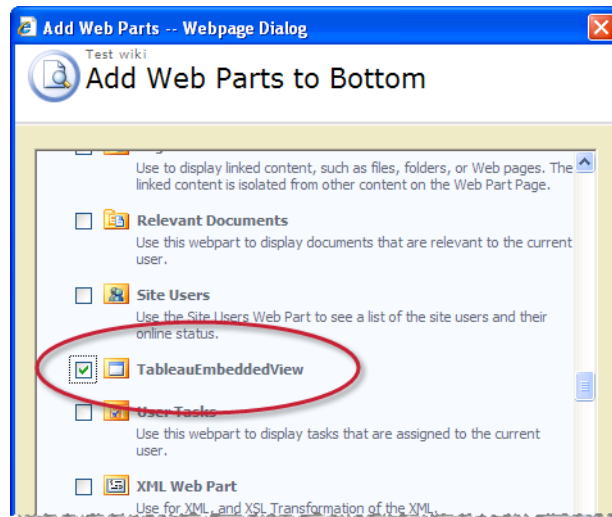
8. Navigate to the SharePoint page that you want to embed a view into.
9. On the Site Actions menu in the upper right corner of the page select **Edit Page**.



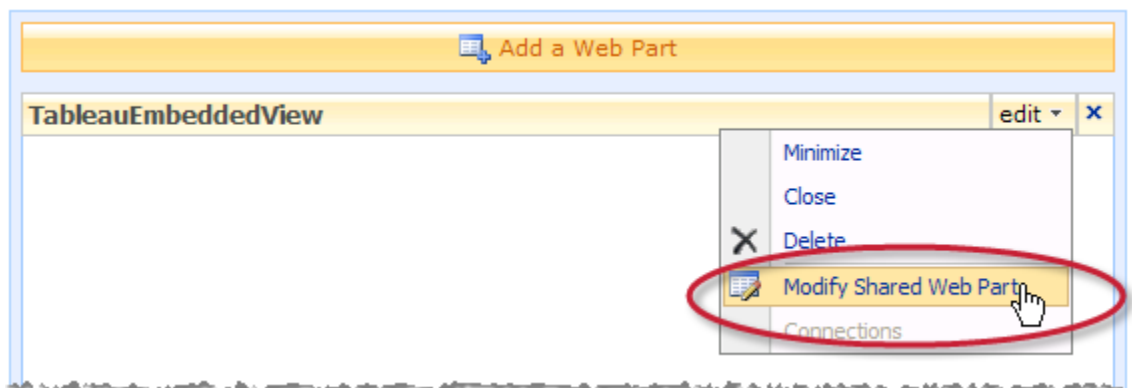
10. Click the **Add a Web Part** button in the section of the page where you want to embed the view.



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



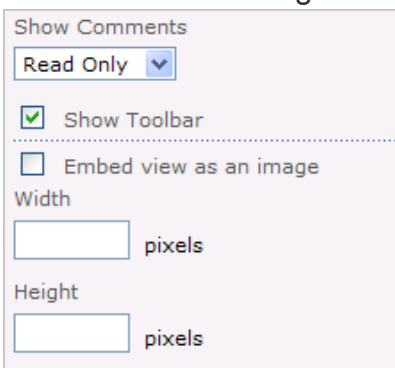
12. Back on the SharePoint page select **Modify Shared Web Part** on the **Edit** menu for the new web part.



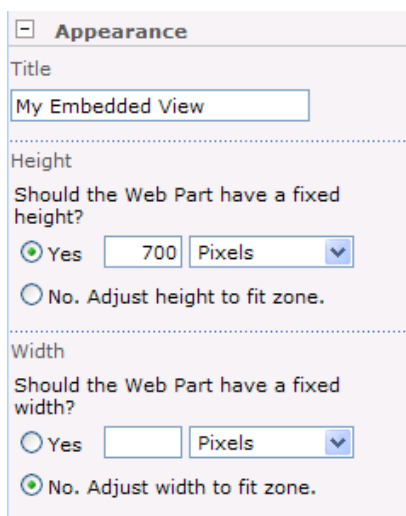
13. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part. Type the name of your Tableau Server.
14. Then type the path to the view you want to embed. For example you may type /views/Date-Time/DateCalcs.

Tableau View Settings
Tableau Server Name
<input type="text" value="tableau-server"/>
View Path
<input type="text" value="/views/Date-Time/DateCalcs"/>

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



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



17. When finished, click **OK** to apply the changes and exit edit mode. Now the view is embedded in the page and users who access the view will be automatically logged in based on their user name and password for SharePoint. Anyone who accesses an embedded view needs to be a licensed user on Tableau Server and their user name on SharePoint must be the same as their user name on Tableau Server.

Note: This is an example of embedding views into SharePoint using the provided .dll file. You can also embed views into other types of web applications and even build your own .dll file. See [Using the Tableau JavaScript API](#) for more information.

Using the Tableau JavaScript API

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

```
<html>
<head></head>
<body>
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
<param name="name" value="Workbook/ViewName" />
<param name="tabs" value="yes" />
<param name="toolbar" value="yes" />
<param name="comments" value="no" />
</object>
<form>
</form>
<script type="text/javascript">
    function useVizAPI() {
        var viz = window.tableau.vizs[0];
        viz.hide();
        viz.show();
        //viz.refresh();
        //viz.revert();
        //viz.filter({Product: ['Mint', 'Green Tea']});
    }
    if (document.addEventListener) {
        window.addEventListener("load", useVizAPI, false);
    } else if (document.attachEvent) {
        window.attachEvent("onload", useVizAPI);
    }
</script>
</body>
</html>
```

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

Web application example code for setting up trusted authentication with SharePoint, PHP, Ruby, and Java applications is installed with Tableau Server and is located in (32-bit) C:\Program Files\Tableau\Tableau Server\6.1\extras\embedding and (64-bit) C:\Program Files (x86)\Tableau\Tableau Server\6.1\extras\embedding.

Tableau Server Trusted Authentication

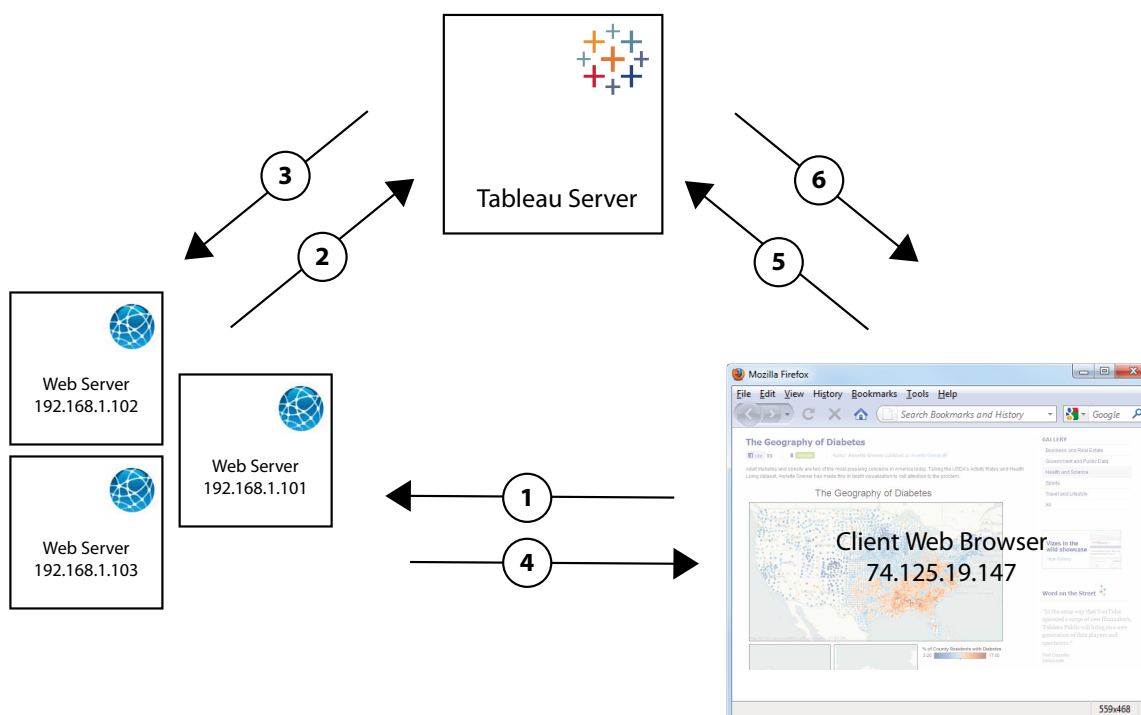
When you embed Tableau Server views into webpages, everyone who visits the page must be a licensed user on Tableau Server. When users visit the page they are prompted to log into Tableau Server before they can see the view. If you already have a way of authenticating users on the webpage or within your web application, you can avoid this prompt and save your users from having to log in twice by setting up trusted authentication.

Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from these trusted web servers it assumes that your web server has handled whatever authentication is necessary.

If your web server uses SSPI, 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. If you are not using SSPI or Active Directory, you will need to set up trusted authentication.

How Trusted Authentication Works

The diagram below describes how Trusted Authentication works between the client's web browser, your web server(s) and Tableau Server.



- 1 User visits the webpage**
 When a user visits the webpage with the embedded Tableau Server view, it sends a GET request to your web server for the HTML for that page.
- 2 Web server POSTS to Tableau Server**
 The web server sends a POST request to Tableau Server. That POST request must have a `username` parameter. The username value must be the username for a licensed Tableau Server user.*
- 3 Tableau Server creates a ticket**
 Tableau Server checks the IP address of the web server (192.168.1.XXX in the above diagram) that sent the POST request. If it is set up as a trusted host then Tableau Server creates a ticket in the form of a `unique_id`. Tableau Server responds to the POST request with that `unique_id`. If there is an error and the ticket cannot be created Tableau Server will respond with a value of -1.
- 4 Web server passes the URL to the browser**
 The web server then constructs the URL for the embedded view and inserts it into the HTML for the page. The URL includes the `unique_id` (e.g., `http://tabserver/trusted/<unique_id>/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 `unique_id`.
- 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 `unique_id` from the URL, and sends back the final URL for the embedded view.

Setting Up Trusted Authentication

Follow the steps below to set up trusted authentication between Tableau Server and one or more web servers.

Add Trusted IP Addresses to Tableau Server

To configure Tableau Server to recognize and trust requests from one or more web servers, you will use the `tabadmin` command line utility:

1. Open a command prompt window as an administrator and change directories to the location of Tableau Server's bin directory. The default location is `C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin`

2. Type the following command:

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

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

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

Note: The comma separated list should be within quotes with one space after each comma.

3. When you're finished, type the following command to save the configuration:

```
tabadmin config
```

4. Finally, type the following command to restart the server:

```
tabadmin restart
```

Configure your Web Server to Send Requests to Tableau Server

When your web server requests a view from Tableau Server, it must contact the server with a POST request. The POST must have the following parameter:

```
username=<username>
```

The `<username>` must be the username for a licensed Tableau Server user. If you are using Local Authentication the username can be a simple string.

If you are using Active Directory with multiple domains you must include the domain name with the user name (e.g., domain\username).

You can configure the server to match on client IP address to require the ticket to be redeemed by a specific user using the same machine that issued the original ticket. Because there may be several proxies between the web browser and your web server as well as between your web server and Tableau Server, the client IP address is not considered by default. Refer to [Optional: Configure Tableau Server for Client IP Matching](#) to learn more.

Set up your Web Server to Construct the URL with the Unique_Id

When Tableau Server responds to the POST request from the web server, it sends back a ticket in the form of a unique_id ([step 3 in the diagram](#)).

The web server needs to construct the appropriate URL that includes the unique_id and points at the requested view on Tableau Server. For example, the URL may look like this: `http://tabserver/trusted/<unique_id>/views/<workbook>/<view>`

You will need to provide the web server with functions to collect the following information:

- **Username:** The username for the licensed Tableau Server user.
- **URL:** The URL of the desired embedded view, including embedding option parameters. Here are some examples:

Script Tag Examples

```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600"
style="display:none;">
    <param name="name" value="MyCoSales 9/SalesScoreCard" />
    <param name="ticket" value="123456789" />
</object>
```

Another option is to state the full path of the view explicitly:

```
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600"
style="display:none;">
    <param name="name" value="MyCoSales/SalesScoreCard" />
    <param name="path" value="trusted/123456789/views/MyCoSales/
SalesScoreCard" />
</object>
```

Iframe Tag Example

```
<iframe src="http://tabserver/trusted/123456789/views/workbookQ4/
SalesQ4?embed=yes" width="900px" height="700px"></iframe>
```

- **IP address:** The IP address of the user's machine (only if configured for client trusted IP matching).

There are several code samples that you can use to set up your web server that are included with Tableau Server. Example code for Java, Ruby, and PHP are located at:

32-bit: C:\Program Files\Tableau\Tableau Server\6.1\extras\embedding
64-bit: C:\Program Files (x86)\Tableau\Tableau Server\6.1\extras\embedding

Optional: Configure Tableau Server for Client IP Matching

By default, Tableau Server does not consider the client web browser IP address when it creates or redeems tickets. You can change the default behavior and enforce that the ticket not only must be redeemed by a specific user but also that it must use the same machine that issued the original request. Follow the steps below to configure Tableau Server to enforce client IP address matching.

1. Open a command window and change directories to the location of Tableau Server's bin directory. The default location is `C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin`

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

```
tabadmin set wgserver.extended_trusted_ip_checking true
```

3. Then type the following command:

```
tabadmin configure
```

4. Finally, restart the server by typing the following:

```
tabadmin restart
```

Troubleshooting Trusted Authentication

Below are some common issues and errors you might encounter when you're configuring trusted authentication.

-1 Return Status from Tableau Server

Tableau Server returns -1 for the ticket value if it cannot issue the ticket. Check the following:

- **All Web Server IP Addresses are Added to Trusted Hosts**

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

- **Trusted Hosts List is Formatted Properly**

The list of trusted hosts on Tableau Server must be a comma separated list with a space after each comma. For example, the list should be something like: 192.168.1.101, 192.168.1.102, 192.168.1.103 and so on.

- **Username in POST Request is Valid Tableau Server User**

The username you send in the POST request must be a licensed Tableau Server user with a viewer or interactor license level. You can see a list of users and their license levels by logging into Tableau Server as an administrator and clicking the Licensing link on the left side of the page.

- **Username in POST Request Includes Domain**

If Tableau Server is configured to use Local Authentication, the username that you send in the POST can be a simple string. However, if the server is configured for Active Directory you must include the domain name with the user name (domain\username). For example, the username parameter might be: username=dev\jsmith

HTTP 401 - Not Authorized

If you receive a 401- Not Authorized error, you may have configured Tableau Server to use Active Directory with SSPI. 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#)

The example code for the SharePoint .dll references the following GET request:

```
SPContext.Current.Web.CurrentUser.Name
```

The above request will return the display name of the current Windows Active Directory user. If you want to use the login ID, then you will need to change the code to:

```
SPContext.Current.Web.CurrentUser.LoginName
```

After you make the change, recompile the SharePoint .dll.

Attempting to Retrieve the Ticket from the Wrong IP Address

The client web browser IP address is not considered by default when redeeming the ticket. If you have configured Tableau Server 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 Tableau Server for Client IP Matching](#) to learn how to configure Tableau Server to enforce client IP address matching.

Appendix A: Configuring Run As User

You can use a dedicated Active Directory (AD) user account for the Tableau Server service to run under, called a Run As User account. Some administrators choose to do this when published workbooks on Tableau Server connect to live data sources. The server's default Network Service account (NT AUTHORITY\NetworkService) doesn't have the correct permissions for connecting to data sources on other computers. A correctly-configured AD account does.

For data sources that require NT authentication, the AD account can also automatically handle the authentication process, thus shielding users from prompts for credentials when the workbook connects to the live data source. Finally, a Run As User AD account that is dedicated to a specific resource is often less problematic to manage than an AD account associated with a person.

To configure Tableau Server to use a Run As User account, follow the procedures under [Required Configuration Steps](#), below. The steps under [Settings to Confirm](#) may vary from site to site.

Note: If you are installing Tableau Server with your Run As User account in hand, before you run Setup, confirm that the Windows Secondary Login service has the correct values for Log On and Startup. See [Verify Tableau Service Settings](#) for more information.

Required Configuration Steps

Run As User Account

Your first step is to identify or create an Active Directory account for the Tableau Server service to run under. This will be the Tableau Server's Run As User account, and it should have the following:

- Permissions for connecting to the data source with at least read access.
- Credentials to allow Tableau Server to satisfy the NT authentication process with the data source. Microsoft data sources that perform NT authentication include Microsoft SQL Server and Microsoft Analytical Services (MSAS), but not Access or Excel.

Confirm Domain Two-Way Trust

Confirm that there is a two-way trust between domains if any of the following are true:

- The machines hosting the Tableau Server and the data source are on separate domains.
- Tableau Server users are on a separate domain from Tableau Server or the data source.

Verify Tableau Service Settings

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

1. Log on as administrator to the computer running Tableau Server.
2. On the Tableau Server computer, select **Start > Control Panel > Administrative Tools > Services**.
3. Open Services and Applications, then click Services. Confirm that the following services have the correct settings:

Service Name	Logon Value	Startup Value
FLEXnet Licensing Service	Local System	Manual
Secondary Login	Local System	Automatic
Tableau Server (tabsvc)	<domain>\<username> This is the Run As User account. See below.	Automatic
Tablicsvr	Local System	Automatic

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

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

Configure Local Security Policy

If your Run As User account isn't an administrator on the Tableau Server machine, you must configure the machine's local security policy so that the Tableau Server Run As User account can log onto the machine as a service and make configuration changes. To do this:

1. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.
2. In the Local Security Settings window, open **Local Policies**, highlight **User Rights Assignments**, then right-click **Log on as a service** and select **Properties**.
3. In the Log on as a service Properties window, click **Add User or Group**.
4. Type the <domain>\<username> for the Tableau Server Run As User account (for example: MYCO\tableau_server), and click **Check Names**.
5. When the account resolves correctly, it is underlined. Click **OK**.
6. Click **OK** to close the Local Security Settings windows.

Data Source Connection Settings

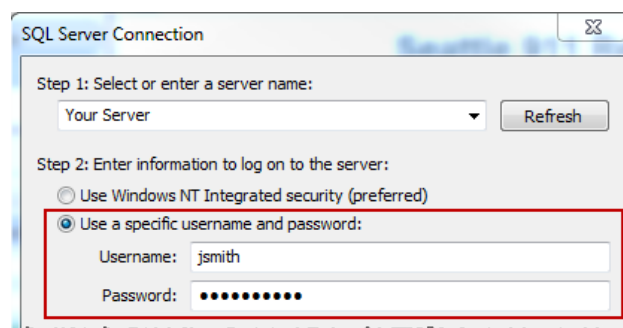
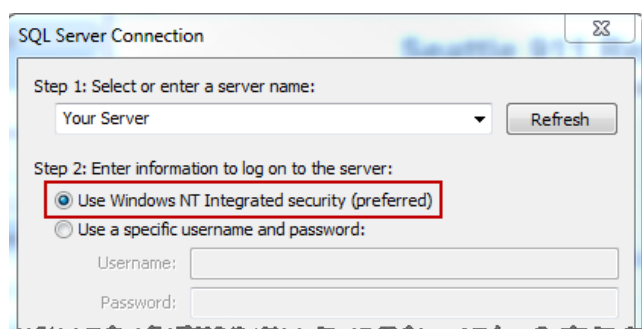
To automatically authenticate your users when the workbook they're accessing connects to a live, NT-authenticated data source, configure your Tableau data connection with the **Use Windows NT Integrated security** option selected:

Windows NT Integrated Security:

Authenticates with the server's Run As User account.

Username and Password:

Each Tableau Server user is prompted for database credentials.



Settings to Confirm

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

Granting Read and Execute Permissions

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

1. On the machine hosting Tableau Server, use Windows Explorer to right-click on **Local Disk (C:)** and select **Properties**.
2. In the Local Disk (C:) Properties Window, select the **Security** tab.
3. Click **Edit**, then **Add**.
4. Type the `<domain>\<username>` for the Tableau Server Run As User account.
5. Click **Check Names** to resolve the account, then **OK** to confirm.
6. With the Tableau Server Run As User account highlighted, confirm that it has **Read & execute** permissions. Selecting **Read & execute** automatically selects **List folder contents** and **Read**.
7. Click **OK** to exit.

Granting Modify Permissions

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

1. Navigate to the following folder(s):

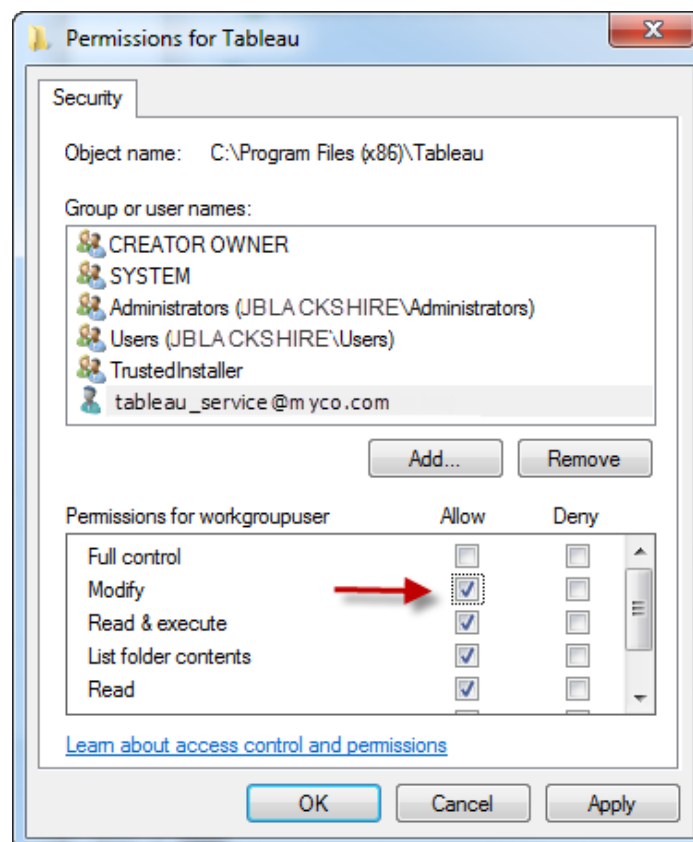
32-bit: `C:\Program Files\Tableau`

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

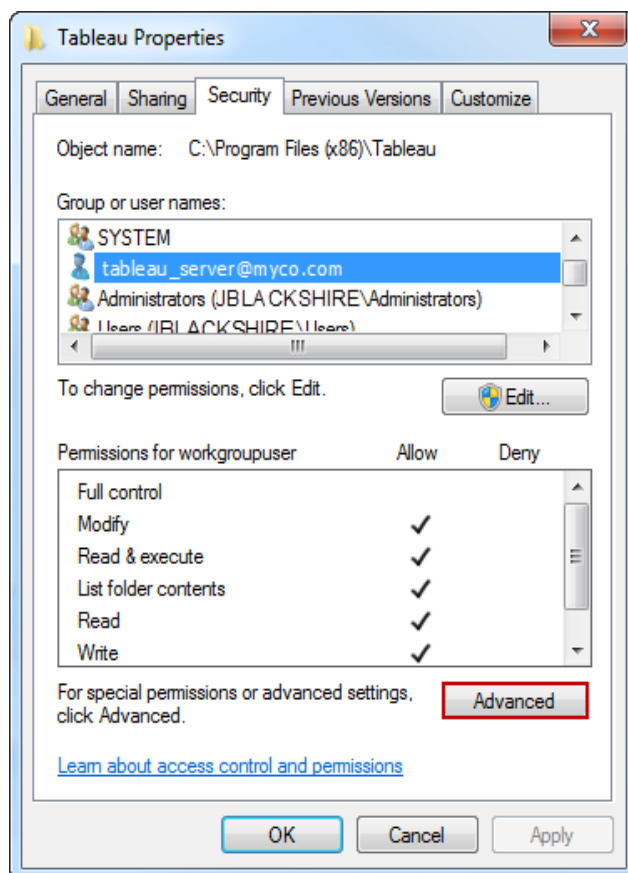
Windows Server 2008, Windows Vista, and Windows 7: `C:\ProgramData\Tableau`

2. Right-click the folder, select **Properties**, and click the **Security** tab:

- Click **Edit**, then **Add**.
- Type the `<domain>\<username>` for the Tableau Server Run As User account.
- Click **Check Names** to resolve the account, then **OK** to confirm.
- With the Tableau Server Run As User account highlighted, confirm that it has **Modify** permissions. Selecting **Modify** automatically grants all permissions except for **Full Control** and **Special Permissions**:

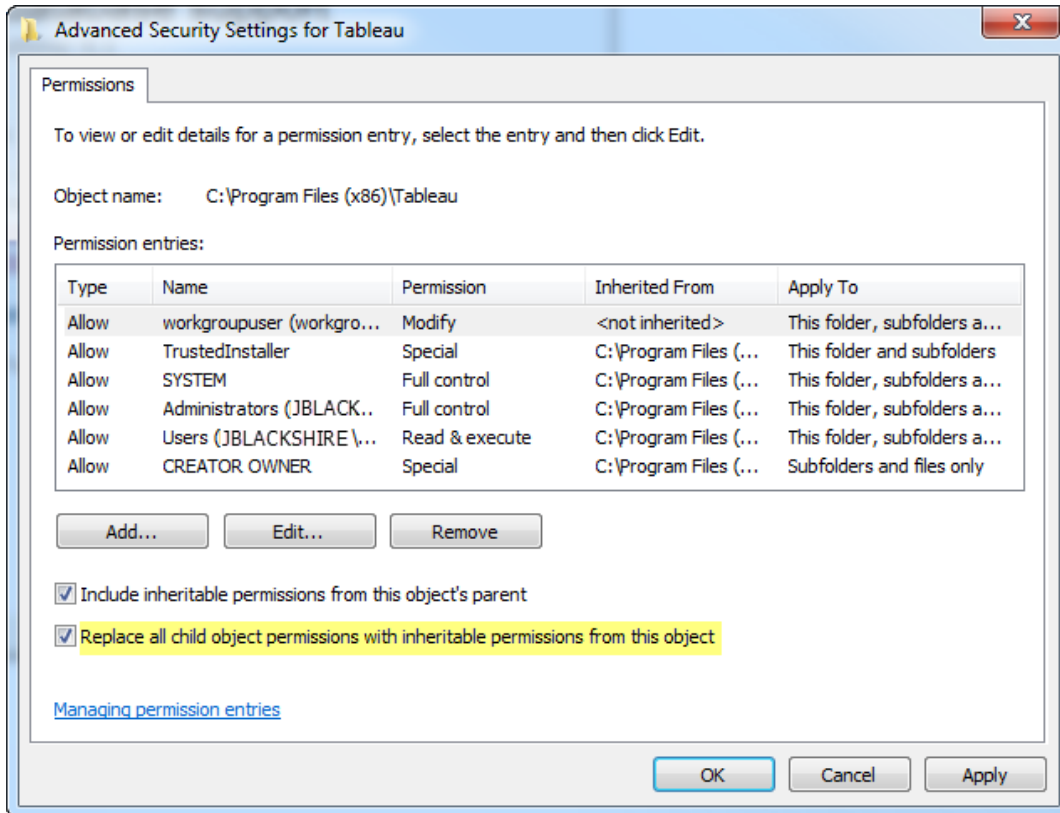


- For each folder, on the Tableau Properties Security tab, click **Advanced**:



- In the Advanced Security Settings for Tableau window, click **Change Permissions**.

5. In the Advanced Security Settings for Tableau dialog box, highlight the Run As User account and select the **Replace all child object permissions with inheritable permissions from this object** check box:



6. Click **OK** to apply changes to all subfolders and files – this may take a few minutes.
7. Click **OK** to confirm changes.

Appendix B: Configuring SQL Server Impersonation

Impersonation is when one user account acts on behalf of another user account. You can configure Tableau and Microsoft SQL Server to perform database user impersonation, so that the SQL Server database account used by Tableau Server queries on behalf of SQL Server database users, who are also Tableau users.

The main benefit of this feature is it allows administrators to implement and control their data security policy in one place: their databases. When Tableau users access a view with a live connection to a SQL Server database, the view only displays what the users' database permissions authorize them to see. An additional benefit is that the users don't have to respond to a database login prompt when they access the view. Also, workbook publishers don't have to rely on user-specific filters to restrict what's seen in views.

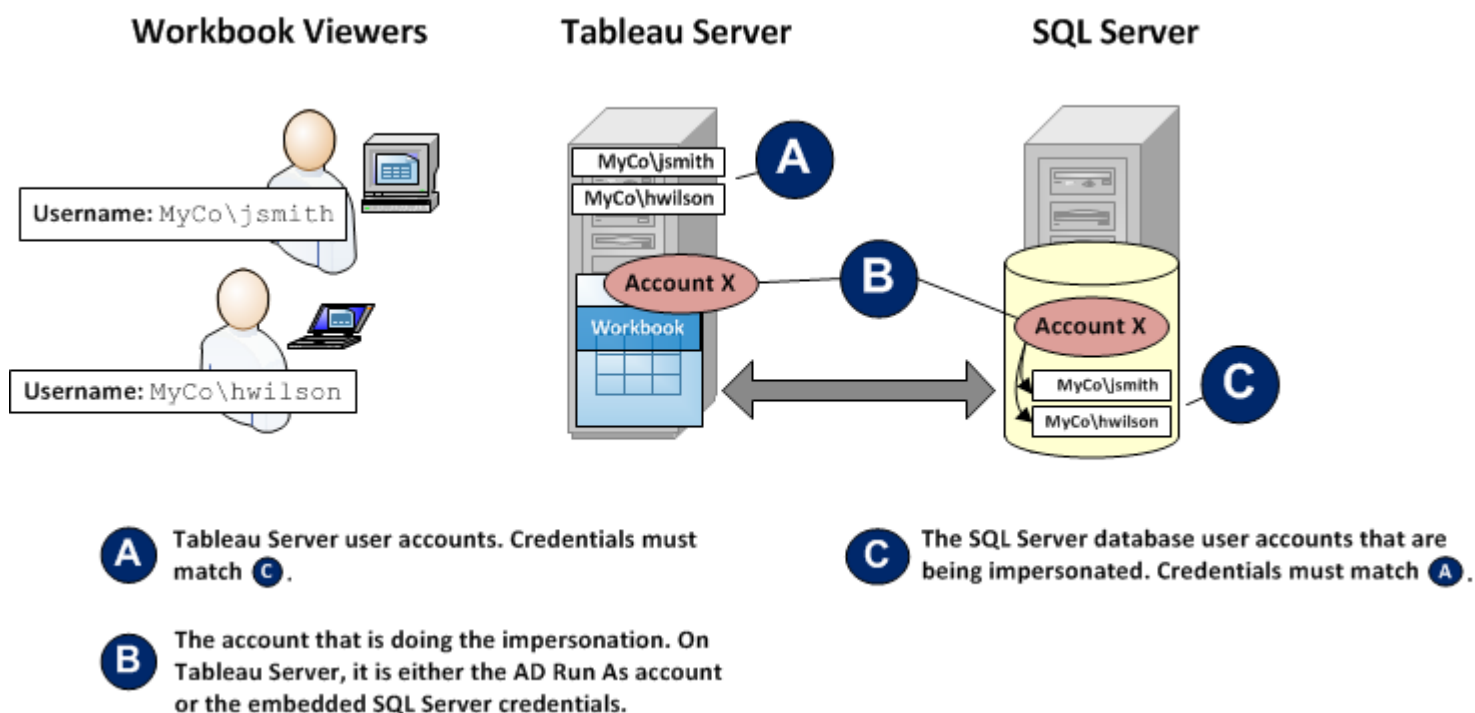
Requirements

Here's what you need to use this feature:

- **Live connections to SQL Server only:** Impersonation can only be used for views that have a live connection to a SQL Server database, version 2005 or newer.
- **Individual database accounts:** Each person who'll be accessing the view must have an explicit, individual account in the SQL Server database to which the view connects. Members of an Active Directory (AD) group cannot be impersonated. For example, if Jane Smith is a member of the AD group Sales, and her database administrator adds the Sales AD group to the SQL Server database, Jane cannot be impersonated.
- **Matching credentials and authentication type:** The credentials of each Tableau user's account and their Tableau user authentication type must match their credentials and authentication type in the SQL Server database. In other words, if Jane Smith's Tableau Server user account has a username of MyCo\jsmith and Tableau Server is using Active Directory for user authentication, her username on the SQL Server database must also be MyCo\jsmith and SQL Server must be using Windows Integrated Authentication.
- **SQL Server prerequisites:** In SQL Server you should have a data security table, a view that enforces data security, and you should require that your database users use the view.
- **SQL IMPERSONATE account:** You need a SQL Server database account that has IMPERSONATE permission for the above database users. This is either an account with the sysadmin role or one that has been granted IMPERSONATE permission for each individual user account (see the [MSDN article on EXECUTE AS](#)). This SQL Server account must also be one of two accounts on the Tableau side of things:
 - The Tableau Server Run As User account (see [Impersonating with a Run As User Account](#)).
 - The workbook publisher's account (see [Impersonating with Embedded SQL Credentials](#)).

How It Works

Here's an illustration of how database user impersonation works:



In the above illustration, Jane Smith (`MyCo\jsmith`) is a West Coast sales representative and Henry Wilson (`MyCo\hwilson`) covers the East. In the SQL Server database, the account permissions for Jane's account, `MyCo\jsmith`, only give her access to West Coast data. Henry's account, `MyCo\hwilson`, can only access data for the East Coast.

A view has been created that displays data for the entire country. It has a live connection to a SQL Server database. Both users log into Tableau Server and click the view. Tableau Server connects to SQL Server using a database account with IMPERSONATE permission for each user's database account. This account acts on behalf of each user's database account.

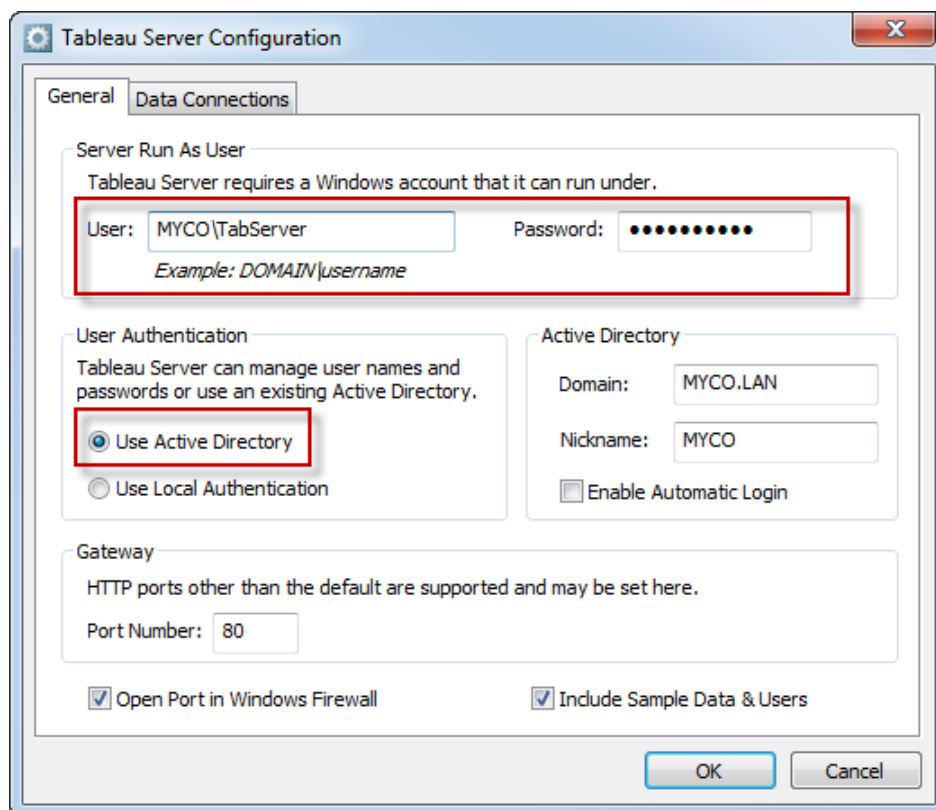
When the view displays, it is restricted by each user's individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.

Impersonating with a Run As User Account

Impersonating via a Run As User account is the recommended approach. The Run As User account is an AD account the Tableau Server service can run under on the machine hosting Tableau Server (see [Appendix A: Configuring Run As User](#)). This same account must have IMPERSONATE permission for the database user accounts in SQL Server. From a data security standpoint, using the Tableau Server Run As account for impersonation gives the administrator the most control.

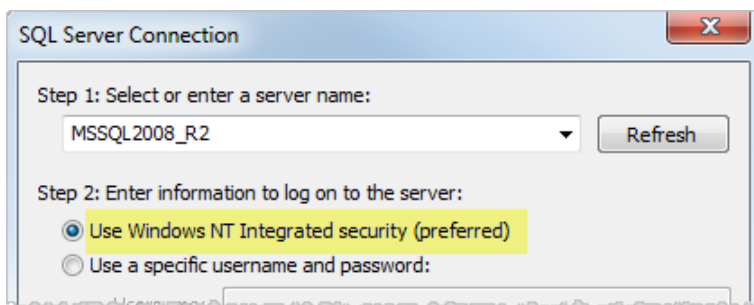
To impersonate via a Tableau Server Run As User account:

1. When you configure Tableau Server as part of Setup, under **Server Run As User**, enter the Run As User AD account that has IMPERSONATE permission for the user accounts. Under User Authentication, select **Use Active Directory**:

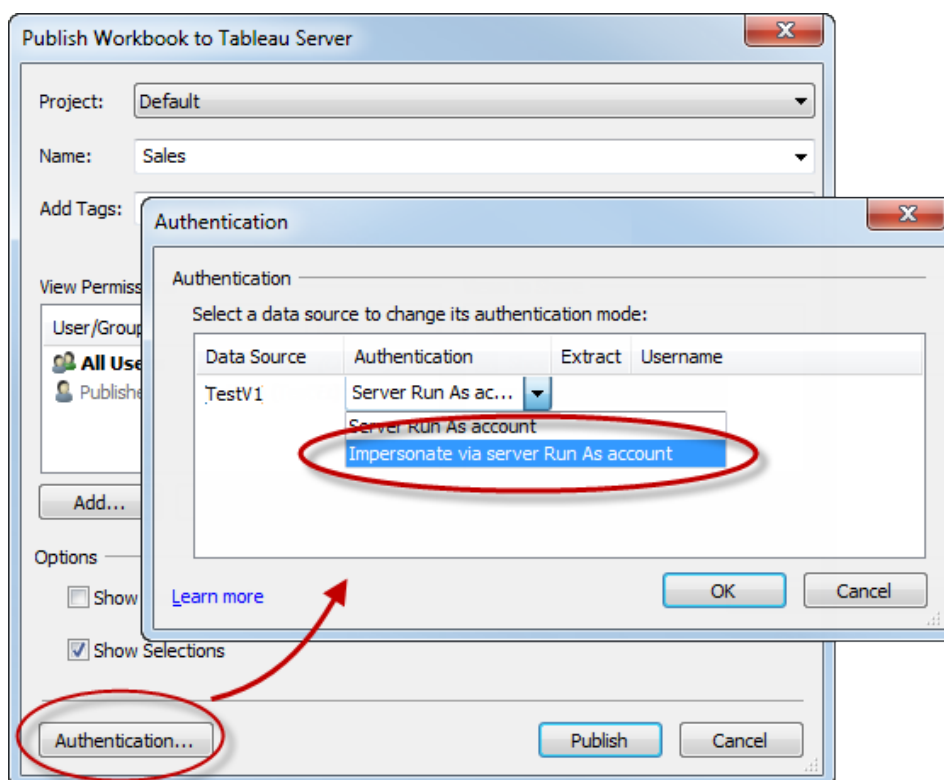


2. Click OK to finish configuration.

3. Create a workbook in Tableau Desktop. When you create the data connection, select **Use Windows NT Integrated security** for the workbook's live connection to a SQL Server database:



4. In Tableau Desktop, publish the workbook to Tableau Server (**Server > Publish Workbook**).
5. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via server Run As account** from the drop-down list:



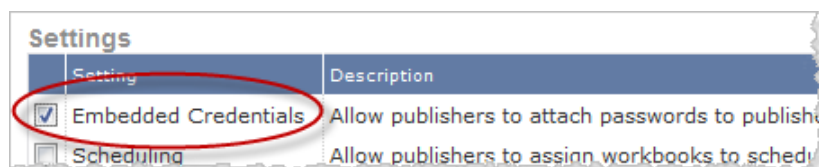
6. Click OK.
7. Test the connection by logging into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Impersonating with Embedded SQL Credentials

You can also perform impersonation by having the person who publishes a view embed their SQL Server account credentials in the view. Tableau Server can be running under any type of account, but it will use these credentials, supplied by the publisher, to connect to the database.

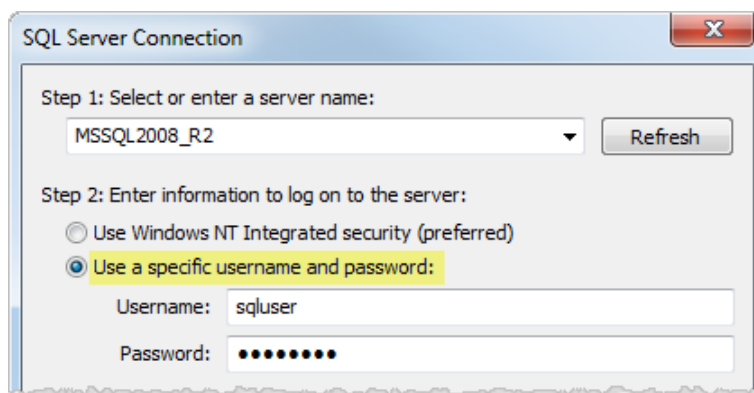
This may be the right choice for your site if the account that handles the impersonation cannot be an AD account and if you're comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

Note that to use this approach, **Embedded Credentials** must be enabled on Tableau Server:



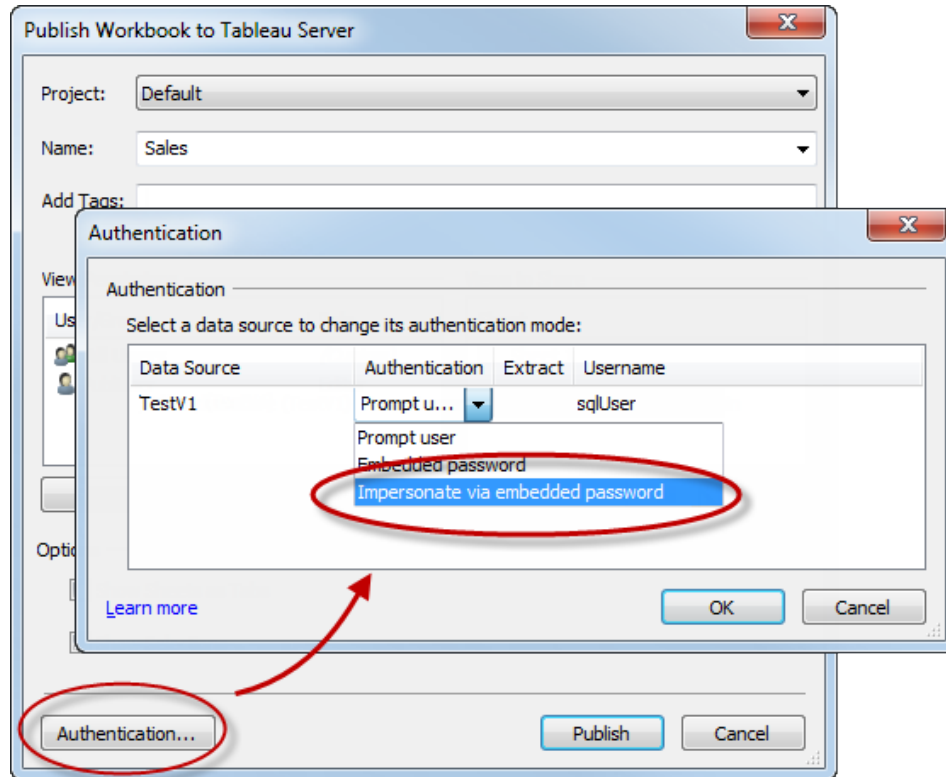
To impersonate via the workbook publisher's account:

1. In Tableau Desktop, create a workbook. When you create the data connection, select **Use a specific username and password** for the workbook's live connection to a SQL Server database:



2. Publish the workbook to Tableau Server (**Server > Publish Workbook**).

3. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via embedded password** from the drop-down list:



4. Click OK.
5. Test the connection by logging into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user account is authorized to see.

Appendix C: Reconfiguring the Server

You may need to reconfigure the server after Setup. Use the configuration tool or the command line tool to reconfigure the server.

Using the Configuration Tool

You can re-open the configuration tool by selecting **All Programs > Tableau Server 6.1 > Configure Tableau Server** on the Windows Start menu. There you can modify the number of processes and the ports used by the VizQL Server and Application.

Caution: You cannot switch between Active Directory and Local Authentication.

Other Configuration Options

You can configure other server settings using the command line tool. As an administrator, change directories using the command below.

```
cd C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin\
```

Then use the following command, substituting in the appropriate variables based on the option you want to change.

```
tabadmin set option-name value
```

Use the table below to learn more about each option you can configure.

Option	Default Value	Description
<code>workerX.gateway.port</code>	80	External port that Apache listens on.
<code>gateway.host</code>	Name of the machine	The host name for the server. The default host name is set to the machine running Tableau Server.
<code>workerX.backgrounder.port</code>	8200	Internal port the backgrounder listens on.
<code>java.heap.size</code>	128m	Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.
<code>pgsql.port</code>	8060	Port that PostgreSQL listens on.
<code>repository.port</code>	8080	Port the repository listens on. This must be the same value as <code>tomcat.http.port</code> .
<code>solr.port</code>	8080	Port that solr listens on. This must be the same value as <code>tomcat.http.port</code> .
<code>tomcat.http.port</code>	8080	Port that Tomcat runs on.
<code>tomcat.https.port</code>	8443	SSL port for Tomcat (unused).
<code>tomcat.server.port</code>	8085	Port that tomcat listens for shutdown messages on.
<code>vizqlserver.port</code>	8100	Base port for the VizQL servers.
<code>workerX.vizqlserver.procs</code>	# of processes	Number of VizQL servers.
<code>vizqlserver.querylimit</code>	30	Longest allowable time for updating a view, in seconds.
<code>vizqlserver.session.expiry.timeout</code>	120	Number of minutes of idle time after which a VizQL session is discarded.
<code>wgserver.domain.fqdn</code>	value of %USERDOMAIN%	The fully qualified domain name of the Active Directory server to use.
<code>workerX.wgserver.port</code>	8000	Base port for the web application servers.
<code>workerX.wgserver.procs</code>	# of processors	Number of web application server processes.
<code>wgserver.session.idle_limit</code>	240	The number of minutes of idle time before a login to the web application times out.
<code>wgserver.show_view_titles_not_names</code>	true	You can only use this option if you upgraded to version 6.1 from an earlier version. A value of true keeps the pre-6.1 behavior and causes Tableau Server to display view titles as their identifiers (for ex., in search results); false causes the server to display view names as their identifiers.
<code>wgserver.trusted_hosts</code>		This option takes a comma separated list of trusted IP addresses for the machine you want to accept trusted requests from. A common value is 127.0.0.1 if you want to put the webserver and Tableau Server on the same machine. This option is used when setting up a trusted relationship between the web server and Tableau Server when embedding views.

Applying the Changes

Modifying the configuration options edits the configuration file `tabsvc.yml`, which is located in the config directory. You can modify the options without impacting the operation of Tableau Server. The changes will take effect the next time you start the server. You can also force the configuration changes using the steps below.

To apply the changes to the server:

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

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Stop the server:

```
tabadmin stop
```

3. Next, configure the server:

```
tabadmin configure
```

4. Start the server again by typing the following:

```
tabadmin start
```

Restoring the Default Configuration

You can restore the default configuration for any of the options with the following steps:

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

```
cd "C:\Program Files (x86)\Tableau\Tableau Server\6.1\bin"
```

2. Then restore the default value for a particular option by typing the following:

```
tabadmin set option-name -default
```

Index

A

- Activation 5
- Active Directory
 - Automatic Login* 10
 - Find Domain* 89, 90
 - Set Domain* 10
 - Set Nickname* 10
- Add Users 21
- Administrative Views
 - Background Tasks* 36
 - Custom* 40
 - Customized Views* 39
 - Performance History* 35
 - Server Activity* 32
 - Space Usage* 38
 - User Activity* 34
- Administrator Account 12
- Administrator Right
 - Content* 24
 - System* 24
- Application Server Ports 56
- Authorization. *See* Permissions
- Automatic Login 10

B

- Background Processes 16
- Backup the Server 42, 52

C

- Caching 12
- Certificate
 - Chain File* 18
 - File* 18
 - Key File* 18
- Clean Up 59
- Comma Separated Value File 21
- Configuration Options 99
- Configuration Tool 99
- Configure the Server 10, 42, 99
 - Extract Storage* 20

- Content Administrator 24
- Custom Administrative Views 40
- Customer Account Center 5, 14
- Customize the Server 42, 53, 55

D

- Database Drivers 17
- Data Connections 51
- Data Engine 20
- Date Filters 67
- Display Status 41
- Distributed Servers 14
- Domains
 - Manage on Server* 50
 - Set Domain* 10
 - Set Nickname* 10

E

- Email Views 60
- Embedded Credentials 55
- Embedded Views 61
 - Images* 73
 - JavaScript API* 78
 - Manual* 63
 - SharePoint (SSPI)* 69
 - SharePoint (Trusted)* 74
 - Trusted authentication* 83
 - Wiki Pages* 72
- Enable Guest 55
- Exit the Server 41
- Extracts
 - Full* 20
 - Incremental* 20
 - Scheduling refresh* 44
- Extract Storage 20
 - Refresh Data* 44

F

- Firewall Port 11
- Fully Qualified Domain Name 89, 90

G

- Guest User 55

I

- Impersonation
 - Overview* 93
 - Run As account* 95
 - SQL account* 97
- Initial SQL 13
- Installation 4

L

- License Levels
 - Assigning to Users* 27
 - Guest* 24
 - Interactor* 24
 - Unlicensed* 24
 - Viewer* 24
- Local Authentication 10
- Logo; change 53
- Logs
 - Archiving* 59, 60
 - Remove & Clean up* 59
 - Service* 58
 - SSL Errors* 19
 - Temporary Files* 58
 - VizQL* 58

M

- Maintenance 17
- Manage product keys 41
- Manual Embed Views 63
 - Date Filters* 67
 - Measure Filters* 67

Measure Filters 67
 Microsoft SSPI 69
 Multiple Machines 14

N

Name; change 53

O

Offline Activation 6
 Online Activation 5

P

Passwords

Administrator 12
 Publish with Password 55
 Server Account 10

Performance 12

Permissions

Add Comment 25
 Delete 25
 Download File 25
 Export Data 25
 Export Image 25
 Filter 25
 Move 25
 Set Permissions 25
 Share Customized 25
 View 25
 View Comments 25
 View Underlying Data 25
 Write 25

Ports

Configuration Options 99
 Defaults 2
 Editing 56
 Resetting to Default 57

Primary Server 14

Product Activation 5

Public User List 55

Publish with Password 55

R

Refresh Extracts 44
 Registration 5
 Remember my changes (Custom-
 ized Views) 39
 Restart Server 41
 Restore Default Settings 54
 Restore the Server Database 42, 52
 Run As User 10, 87

S

Sample Data & Users 11
 Saved Passwords 55
 ScheduledTasks 44
 Schedules 44, 55
 Secured Sockets Layer (SSL) 18
 Server Account
 Permissions 87
 Set Username & Password
 Troubleshooting 87

Setup Wizard 4

SharePoint 69

SSPI 79

Start the Server 41

StartupTasks 12

Stop the Server 41

Storage 20

System Administrator 24

System Requirements 2

T

Tabadmin 42

Activate 42
 administrator 42
 Autostart 42
 Backup 42
 Cleanup 42
 Configure 42
 Customize 42
 Dbpass 42
 Help 42
 Install 42

Passwd 42
 prep_workers 42
 Reset 42
 Restart 42
 Restore 42
 Set 42
 Start 42
 Status 42
 Stop 42
 Uninstall 42
 Warmup 42
 Ziplogs 42

Tabcmd 43

Creategroup 43
 Createusers 43
 Delete 43
 Deletigroup 43
 Deleteusers 43
 Get 43
 Help 43
 Login 43
 Logout 43
 Publish 43
 Set 43
 Syncgroup 43
 Version 43

Tableau Services

Display Status 41
 Exit 41
 Machine-Readable Status 31
 Manage product keys 41
 Open the Server 41
 Restart Server 41
 Start the Server 41
 Stop the Server 41
 View Status 30

Tasks 44

TCP/IP Ports 56

Temporary Files 58

TLF File 8

TLQ File 7

Trusted Authentication 74, 79, 102

U

[Updates](#) 17

[User Rights](#)

Admin 24

Assigning to Users 27

Publish 24

[Users](#)

Adding (Active Directory) 23

Adding (local) 21

Licensing 24

Permissions 24

Rights 24

V

[VizQL Server Ports](#) 56

W

[Wiki Pages](#) 72

[Windows Firewall](#) 11

[Worker Machines](#) 14