

Table of Contents

Installing Tableau Server	1
Using Tableau Server to Facilitate Fact-Based Team Collaboration	35
Automating Server with Tableau's Command Line Tools	75

Installing Tableau Server

The more you have in your cup, the more likely people are to want a drink.

SETH GODIN¹

In the first eight chapters you learned how to use Tableau to connect to data, analyze data, visualize data, and build dashboards. Chapters 9, 10, and 11 are about installing, managing, and maintaining Tableau's information sharing products. Tableau provides three different tools for sharing information—Tableau Public, Tableau Online, and Tableau Server.

Tableau Public is a free cloud-hosted service aimed at bloggers, students, or data visualization enthusiasts that want to share their work publically. It is not designed for enterprise environments that require data security. In fact, anything published on Tableau Public is freely available for anyone to download. This tool is generally not used in enterprise environments that need to control access to the information.

Tableau Server is for customers that need to control where and how the data is stored and managed. It can be installed on hardware behind your firewall or on cloud services that you contract with directly. You can license it by named-user, or based on the server hardware that is installed.

Tableau Online (released in 2013) is another cloud-based information sharing environment (managed by Tableau Software) that provides data security without the need for installing any software or managing hardware. Your data is stored in a secure environment managed by Tableau Software. To start using Tableau Online just requires signing-up for the service and assigning access to your users based on your security needs.

Because Tableau Online is a service that doesn't require that you provision hardware and install any software, the majority of this chapter will be about your choices for installing and configuring Tableau Server. In the sections of the chapter related to setting-up security, differences between Tableau Server and Tableau Online will be discussed.

THE REASONS TO DEPLOY TABLEAU SERVER

Server's architecture provides the flexibility to scale from a single box to large multi-server deployments. User-friendly tools are provided for setup and maintenance of access rights, scheduling, and notification. Once setup is complete, continuing administration is normally minimal. Downloading and installing Tableau Server normally can be done in less than two hours.

A new feature in Tableau Server Version 8 provides users with the additional flexibility of editing or creating new reports and analysis. This functionality isn't a replacement for Tableau Desktop, but does allow staff that doesn't have a Tableau Desktop license to build and modify reports.

There are three primary reasons to deploy Tableau Server:

- Data governance (security)
- Efficiency (time savings)
- Flexibility (consumption and editing options)

DATA GOVERNANCE

Securing proprietary or confidential data is not only a business need, but also a legal requirement. Information managed by health care providers, insurance companies, and government entities is controlled by law. Businesses have a legal obligation to ensure that private employee and customer data is kept confidential and secure. Questions regarding policy implementation, data stewardship, and adaptability have to be balanced against this need for security.

Technology staff must be concerned about the accuracy and consistency of the data being consumed by staff without being overly controlling. Tableau Server balances these different needs very well by supporting data governance best practices. It allows information technology staff to maintain control over datasources (providing a single version of the truth) while simultaneously providing information consumers with the ability to adapt reports to their own purposes—without the need for additional technical staff or needing to resort to creating new (unauthorized) datasources.

EFFICIENCY

Sharing reports is very easy via Tableau's free desktop report consumption tool—Tableau Reader. However, this approach doesn't scale well and provides only limited means for securing the underlying data. Updating desktop reports is easy, but can be time consuming if you have dozens of weekly reports to

deliver. Tableau Server provides a secure environment for report consumption and can automatically update reports and inform users of new report availability via Server's subscription service. Administrators can monitor report consumption, server utilization, and performance.

Tableau Online provides similar benefits at a lower price point but requires publishing your Tableau reports outside of your firewall.

Many times the datasource doesn't include all of the information desired. Domain experts inside your entity may create initial workbooks that address aggregation needs, dimension grouping, and other particulars that would be desirable to share with everyone using Tableau. Server facilitates this sharing by allowing users to publish that metadata via datasource files through the server—saving everyone time and insuring report consistency. If those datasource files are modified, then changes are automatically propagated to everyone using that datasource.

Personnel consuming reports don't need to install any software to view reports because everything is viewed via a web browser. Internet Explorer, Firefox, Chrome, and Safari are all supported.

FLEXIBILITY

While Tableau connects directly to a wide variety of sources, if you desire to run reports from data extracts, extract updates can be scheduled to run automatically at any time interval desired. Do not underestimate the level of demand that Tableau generates. In short order, your deployment may go from a few users to hundreds, and then thousands. The number of reports will increase as well. Server provides the users with an easy-to-navigate consumption environment that allows them to ask questions and get answers. It provides administrators with the means for managing and updating reporting needs without the need for daily manual intervention.

Reports published via Server can include direct connections to datasources or data extracts that are automatically updated by Server. Administrators can assign rights for publishing, consuming, and even modifying reports. Reports can be embedded into existing company websites, and Tableau can pass through the security layer without requiring the user to enter redundant login information. Information consumers can securely view and edit reports via the web on their desktop, laptop, and on iOS or Android tablet devices.

Tableau Server is a robust environment that provides technology managers with the tools to secure and maintain the environment while also providing information consumers with fast access to the information they need.

LICENSING OPTIONS FOR TABLEAU SERVER AND TABLEAU ONLINE

Tableau Server can be licensed two different ways:

- Per-named-user basis
- Server core license

Core licensing provides unlimited access to any number of users. Pricing is based on the number of processor cores contained on the physical box or multiple boxes on which you deploy the software. Per-named-user licensing starts with a minimum of ten users. Core licensing requires an eight-core minimum. Although many factors can affect performance in a server deployment (hardware, network traffic, dashboard design), an eight-core configuration can support up to 225 concurrent users.

Tableau Online is a named-user license that requires a one-year commitment. You can start with a single license and add more as your needs grow.

DETERMINING YOUR HARDWARE AND SOFTWARE NEEDS

Tableau Server is a scalable system that is capable of meeting the demands of the most intense enterprise environments. Proper planning is an important first step before you settle on the appropriate hardware configuration and licensing options. At a minimum, you should consider the following details when planning your deployment:

- User count
- User concurrency rate
- Workbook complexity
- User Locations
- Database Locations
- Database size
- Extract Usage—number and size

The user count and user concurrency rate provide an expectation of the volume of requests that Server will be handling. This is normally fairly easy to estimate. User count represents the number of licensed users on Tableau Server that are able to make requests to the server. User concurrency rates represent the percentage of the licensed users that will be making requests at any single

moment. For example, a deployment anticipating 1,000 licensed users with an expected concurrency rate of 10 percent implies that approximately 100 users would be active in the system at any moment.

Workbook complexity is more difficult to anticipate. For this reason, before you plan your Server environment it may be advisable to identify a core group of report designers, train them, and have them build some initial reports that can serve as a basis for planning. This typically doesn't require more than a month to accomplish and doesn't need to involve many staff. Not all requests made to Tableau Server are equivalent. Server will spend more resources to render dashboards with complex designs and large volumes of data than dashboards with simple designs and low record counts. Poorly designed dashboards are the most common cause of poor performance in Tableau Server.

If you have users across many locations or database services deployed across multiple geographies, you may need to have a correspondingly larger number of Tableau Servers to support local demands if a central service isn't able to provide the desired responsiveness.

The amount of data you have, as well as the type of database sources you are using, must also be considered. Massive data or heavy demand along with a database that wasn't designed for the analytical loads can create the need for shifting some of the analytical burden to Tableau Server from the database by utilizing more Tableau Data Extract files.

DETERMINING WHAT KIND OF SERVER LICENSE TO PURCHASE

If you don't require that your data and reporting be within your company network—behind your firewall, Tableau Online provides a very convenient option. Tableau Software manages the hardware and is responsible for maintaining network performance. It is a very good option if you are comfortable with Software-as-a-Service (SaaS) models. The administrator of a Tableau Online deployment is only directly responsible for controlling the access by setting permissions for publishing and viewing the data.

If your organization is unable to reside your data in the cloud outside of your firewall, Tableau Server's named-user licensing or Core-Server licensing allow you to directly control every aspect of Tableau Server's setup and configuration—inside or outside of your company's firewall. For most large enterprise customers, Tableau Server offers the most flexibility.

Tableau Server named-user licensing is exactly what it sounds like—one license purchased per user—meaning that a license must be purchased for each

individual user of the system. If there are ten distinct employees that need access to Tableau Server, then all ten of them must have a named-user license.

A question that many people ask is whether or not Tableau can be deployed on any kind of multiplexing device so that individual users can share the per-named-user license. The answer is no. Licenses are transferable but this is not a practical way to split a single named-user license among an active user base. Named-user licenses are also referred to as Interactor licenses.

Core licensing allows customers to license Tableau Server by the server processor core—avoiding the need to purchase licenses for specific named-users. Core licensing provides greater flexibility, allowing for as many users as a server can support from a resource perspective. These licenses are typically sold in multi-core quantities in eight-core multiples. Pricing for core licensing reflects the fact that a single core can support many users. It also provides the option of enabling a special guest account to enable unrestricted access to reports assigned by the administrator.

The number of users you anticipate accessing the system normally determines which licensing model you choose. Smaller entities with low user counts typically find that named-user licensing provides a better value. Tableau Online will also appeal to this segment if externally-hosted security is permitted. Large organizations with user counts exceeding 250 normally find core licensing more cost effective.

In some cases mixed licensing models might be desirable since hardware limitations imposed by the core licensing model can be alleviated through the selective use of named-user licensing and/or Tableau Online.

TABLEAU SERVER'S ARCHITECTURE

Tableau Server is comprised of several processes operating together. These may run concurrently, but typically all processes won't be running all of the time. These include:

- Application Server (wgserver.exe)
- VizQL Server (vizqlserver.exe)
- Data Engine (tdeserver.exe, tdeserver64.exe)
- Backgrounder (backgrounder.exe)
- Data Server (dataserver.exe)
- Repository (postgres.exe)

The application server handles requests to the web application such as searching, browsing, logging in, generating static images and managing subscriptions. The VizQL server handles the task of loading and rendering requested views. The data engine receives queries made to Tableau Data Extracts present on the server. These queries come from the VizQL processes. To service these queries, the Data Engine loads the Tableau Data Extracts into memory and returns the requested record set. The backgrounder runs maintenance tasks and data extract refreshes. The data server handles requests to Tableau Data Sources. These requests can come from the Tableau Server or from Tableau Desktop users. The repository is the Postgres database Tableau Server uses to store settings, metadata, usage statistics and workbooks.

With the notable exception of the data engine, all of these processes are 32-bit processes. The data engine has a 64-bit executable code, which will be used by default if a 64-bit architecture is detected. All processes except the backgrounder are multi-threaded.

SIZING THE SERVER HARDWARE

Tableau Server runs well within a variety of hardware configurations. It can be deployed for small organizations on a relatively inexpensive single system. It can also be deployed for large organizations with thousands of users on clusters containing many powerful machines. You get what you pay for in terms of performance from hardware expenditures. The current minimum recommended hardware configuration for Tableau Server is a single machine with 32-gigabyte of memory and 8-CPU cores. Specific recommendations regarding the size and configuration of your deployment are affected by many factors including the complexity and size of the dashboards, the datasources, the timing and frequency of usage, the network, and the hardware configuration running the software. For these reasons specific benchmarks are not provided. Consult with Tableau Software's technical staff or a qualified Tableau Software Partner to obtain specific recommendations.

As your deployment grows you can increase capacity by scaling-up to a more powerful single server, or by splitting the increased demand across multiple physical servers.

A Scale-Up Scenario

To scale Tableau Server up on a single system, choose a platform that can provide more physical CPU cores and more system memory. At this time, major hardware manufacturers are shipping servers that support up to 32 physical CPU cores and far more memory than Tableau Server will require. The above ratio of CPU cores to system memory (1-CPU to 4-GB memory) is a good general

guideline to follow—plan for more memory when use of very large Tableau Data Extracts are expected. The data engine will hold data extracts in memory if possible. This improves query performance.

Disk performance is a secondary consideration when planning for Tableau Server in most cases. The major exception being situations in which there is heavy use of the data engine with extracts that will not fit into memory. In this case the data engine is forced to go to disk frequently—making faster I/O potentially worthwhile. Otherwise, even with heavy use of the data engine, Tableau Server does not benefit a great deal from more exotic I/O setups such as arrays of Solid State Drives (SSD).

An example of a scale-up configuration for Tableau Server is a single machine with 24-CPU cores and 96-GB of memory. Based on the current Tableau Server scalability tests, it's expected that this server could handle somewhere between 108 and 378 concurrent requests depending on workbook complexity.

A Scale-Out Scenario

To scale Tableau Server out, multiple servers will need to be provisioned and the server processes will be split across them. In this case, the servers are not required to be configured identically. It is a common pattern to tailor each machine in a cluster to the processes running on it. Deploying Tableau Server on multiple servers will be discussed in greater later in this chapter in the section on High Availability Environments.

An example of a scale-out configuration for Tableau Server is a cluster consisting of three machines each configured with 8-CPU cores and 32-GB of memory. This configuration will provide slightly lower performance than the sample scale-up configuration due to the server communication overhead introduced by the cluster. A fourth machine can be used to run the gateway server. If this is done, any machine running gateway services exclusively is not counted against the Tableau Server Core license.

Regardless if you plan to scale-up or scale-out, if you decide to purchase under the core-license model you need to determine the number of cores that you'll be required to purchase. Do this by counting the number of physical cores across all of the machines that will be running Tableau Server processes excluding servers that are running gateway services only.

ENVIRONMENTAL FACTORS THAT CAN AFFECT PERFORMANCE

There are many environmental factors that can affect performance of Tableau Server. Normally the most significant factors relate to network performance, the browser, and resource contention.

Network Performance

Users will be connecting to Tableau Server either through an internal network or via the public Internet. Any poor performing network links in between users and the Tableau Server can cause erratic behavior of dashboards, and slow loading. Internal networks are not normally a problem. Spotty Internet connections are a common cause of long dashboard load times. If you do experience slow connection speeds the best solution is to increase the available bandwidth of the connection.

Browser

The user experience of Tableau Server is heavily dependent on JavaScript. As such, some browsers can cause Tableau Server to feel unresponsive or sluggish because of their sub-par JavaScript performance. Internet Explorer 7 is a major offender in this case. Chrome, Firefox, Safari and modern versions of Internet Explorer all have superior JavaScript performance. If it frequently takes a few clicks to get a Quick Filter drop down selection to apply, you might be running into a browser performance issue.

Resource Contention

Tableau Server will not perform well in environments with other resource hungry applications and services running on the same machine. Resource contention can cause slowness in each component process of Tableau Server. To get the most out of your Tableau Server license expenditure, ensure that Tableau Server is the only application running on the machine(s).

CONFIGURING TABLEAU SERVER FOR THE FIRST TIME

When installing Tableau Server there are many configuration options to evaluate. Most of these options can be adjusted after the installation, but some of these options cannot be changed without reinstalling the software. It is therefore important to place careful consideration behind the configuration options below.

GENERAL: SERVER RUN AS USER

Server Run As User refers to the Windows username that the Tableau Server service (tabsvc) will run under. By default, this is configured as the Network Service account. This can be changed to either a local machine account or a domain account. If choosing a domain account, specify the domain with the username. One reason to use a domain account is to provide access to datasources that

require Windows NT authentication without prompting users for credentials. In this case the account specified here logs into the datasource.

GENERAL: USER AUTHENTICATION AND ACTIVE DIRECTORY

Tableau Server can be configured to authenticate users in one of two ways:

- A Local Authentication
- An Active Directory Authentication (ADA)

It is very important that you choose the authentication method carefully because this cannot be changed once the server is installed. In the Local Authentication option users are added to the server by configuring a username and a password. In the Active Directory authentication option, users who are added to the Tableau Server must already exist within Active Directory. Active Directory manages the user's password. Selecting Active Directory authentication allows clients to reuse their existing security structure.

Be sure to enter the domain name and nickname when choosing to authenticate with Active Directory. This domain name must be a fully qualified domain name. Using the (ADA) method allows an additional option—Enable Automatic Log-on. This option enables users to automatically log in to Tableau Server with the currently logged in Windows account credentials via the Microsoft Security Support Provider Interface (SSPI). Additionally, Automatic Log-on should not be enabled if the guest account is enabled or trusted ticket authentication is used.

General: Port Number

By default, Tableau Server accepts requests on port 80. If this needs to be changed for networking reasons, reset the port number using this option.

General: Open Port in Windows Firewall

This opens the above port number in the Windows Firewall to ensure that requests can be received on the specified port. This setting normally shouldn't need to be edited unless you have changed from the default port 80.

Data Connections Caching

The caching options within Tableau Server dictate how often cached data will be reused and how frequently data will be queried from the datasource. These are the options:

- Refresh Less Often.
- Data in the cache is reused for as long as possible.

- Balanced (data is removed from the cache after a specified number of minutes).
- Refresh More Often.
- Data in the cache is refreshed on each page reload by a query to the datasource.

Caching option selections can significantly affect performance. Reading from the cache is much quicker than querying the datasource directly. In most cases leaving this option set to Refresh Less Often will provide the best performance. The main reason to change to Balanced or Refresh More Often, is to prevent old data from being reported from when you have a rapidly changing datasource.

Data Connections: Initial SQL

When connecting to a database, an initial SQL statement can be sent to set up an environment. For security reasons some administrators may want to disable the Initial SQL setting. Selecting the “Ignore initial SQL Statements for all datasources” setting will cause the workbooks created using the initial SQL statement to open, but the initial SQL command will not be sent.

Server: Number of Processes per Server

The server tab of the configuration dialog box allows the user to configure the machines in the Tableau Server cluster and the number of each type of process per server. These are the default configurations for an 8-core instance:

- VizQL—Two processes
- Application—Two processes
- Background—One process
- Extract engine—One process

Use the Edit dialog box to adjust the numbers of processes if needed. You should plan on at least one CPU core and one gigabyte of memory per process at a minimum. These settings are made in the configuration dialog box after installation. Cluster configurations will be covered in the High Availability section later in this chapter.

E-mail Alerts for Administrators

To have an e-mail notification sent to a specified administrator when the server detects problems, enable Send E-mail Alerts. Add the e-mail address of the person to be notified.

Subscriptions

Use the Enable E-mail Subscriptions option to allow users to subscribe to workbooks on Tableau Server so that they receive e-mail notification when the workbook is updated. This subscription will send the most recent version of a workbook to users at scheduled intervals.

If either e-mail notification is enabled (for problem alerts to the administrator or for subscriptions) the relevant SMTP Server information must be provided.

Secure Sockets Layer (SSL)

Select the Use SSL for Server Communication setting to enable Tableau Server to use SSL to secure communications. If this setting is enabled, the required certificate files must be provided. Tableau Server currently uses SSL only over Port 443. For more information about the requirements of the SSL certificate files, check the Configuring SSL reference in the Tableau Server Administrator's Guide.

To adjust these settings after installation, select the Configure Tableau Server shortcut under the Tableau Server folder in All Programs.

SETTING-UP SECURITY RIGHTS

Tableau Server has a robust system for managing security. To fully grasp it, you must understand the hierarchy of objects that contain reports and data within Tableau's environment.

Workbook

The Workbook object represents the Tableau workbook file published from Tableau Desktop. It contains dashboards and worksheets, which in terms of Tableau Server are all known as Views. Permissions can be applied to specific Views within a Workbook or at the whole Workbook level. Workbooks and Views can belong to projects and must be published to a site.

User

The User object represents a named-user who has access to the Tableau Server. Users must be granted a licensing level of Interactor or Viewer to log in to the server. It's possible to leave a user account on the server in an effectively disabled state by setting its licensing level to unlicensed. This can be useful for audit purposes. Users can be granted access to Views, Workbooks, Projects, and Sites. They can also be placed into groups. Also note that unlicensed Tableau Server users (that have been given publishing rights) can publish workbooks to Server even when they cannot view the published results on the server.

Project

The Project is an object used to organize and manage access to Workbooks. Workbooks are placed into Projects within a Site. This can be used as an organization tool by placing Workbooks with similar content into a single project. It can be used as an access restriction tool by granting access to a Project to a user or group and then publishing Workbooks into that Project.

Group

The Group is an object used to organize users in Sites on the Tableau Server. Users can be placed into Groups and these Groups can in turn be given permissions to objects on the server. Groups can be created locally on the Tableau Server or, if Active Directory authentication is in use, they can be imported from an Active Directory Group. Groups make managing user permissions within Tableau Server much easier.

Site

The Site is the top level of the security hierarchy. Sites are essentially completely separate Tableau Server instances from the user perspective. Users cannot log in to, or view, any information about Sites to which they do not have access. The base Tableau Server site is known as the Default Site. Users that belong to more than one site must choose which site they want to see when they log in. Additional Tableau Server sites are accessed using a name extension string appended on the URL using this format: (/t/ [name]).

Permissions

Where sites define separate work environments in Tableau Server, permissions define what users or groups are permitted to do within a site. Tableau Server comes with several standard permission roles that can be assigned to Users or Groups.

The Interactor role represents the common user who can access and use Objects but not edit them. The Publisher role allows users to publish reports from Tableau Desktop to the server. The Editor role allows the user to make changes to Workbooks. Users can also be granted several more specialized roles. Additionally there are two administrative permissions that can be granted at the Site and Instance level—System Administrator and Site Administrator. These last two permission types allow high-level control of the particular Instance.

When the standard roles aren't enough, it is possible to add very specific permissions to Groups or Users. More detail around specific permissions that are available can be found in the Tableau Server Administrator's Guide.

Using Groups and Projects to manage access is much easier than assigning user permissions to workbooks individually. Depending on the sensitivity of data contained in workbooks on the server, some organizations choose to make heavier use of individual Sites rather than Projects. It is important to understand that moving content between projects is very easy, but moving content between sites requires republishing the workbook. A common example of this is having separate Sites configured for departments such as human resources. Another common usage of Sites is using the default instance as production and creating an alternate test site for development and testing on Server.

ENABLING ROW-LEVEL SECURITY VIA FILTERS

Row-Level security is the ability to restrict access to specific data elements within a datasource to specified users. It is enabled by employing user filters. Using the Superstore sample data you will see an example that restricts access to records using the region dimension. Figure 9-1 shows a map visualization of Superstore that uses color to define regions.

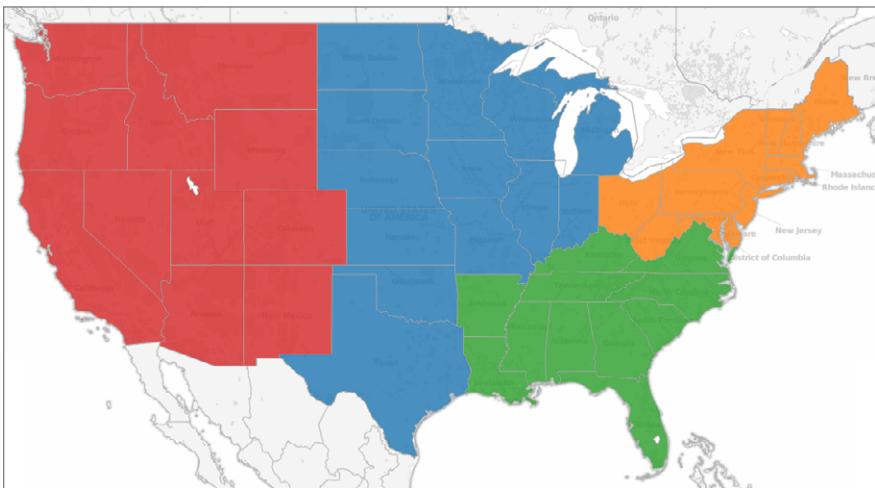


FIGURE 9-1 Map visualization colored by region

Start by creating a user filter in Tableau Desktop by selecting Server > Create User Filter > Region from the main menu. Users created for this example include the East Region, West Region, and the All Region groups. East and West can see only their regions. The All Region user will be able to view everything in the data set. Clicking the OK button defines the filter. You can see the members selected for the All Region option in Figure 9-2.

Once the user filter is created it will appear on the Set Shelf at the bottom left of the desktop. Dragging the set to the Filter Shelf applies the filter. You can simulate the results of the filter for other users by using the toggle at the bottom right of the Tableau Desktop interface by selecting other users. This will change the view to simulate how it will appear for each user you select so that you can verify that the filter produces the desired result. Figure 9–3 shows the region filter set placed on the Filter Shelf. Notice that the All Region user is being simulated using the drop down filter at the bottom right.

When a user filter is placed on a worksheet, the Publish Workbook to Tableau Server dialog box changes to include a new Generate Thumbnails As User option. You can see it in the lower right of Figure 9–4. This option allows you to select what thumbnail will display for each user in Tableau Server—ensuring that sensitive data isn't seen by unauthorized users. The View Permissions dialog box can contain many users with different view filters.

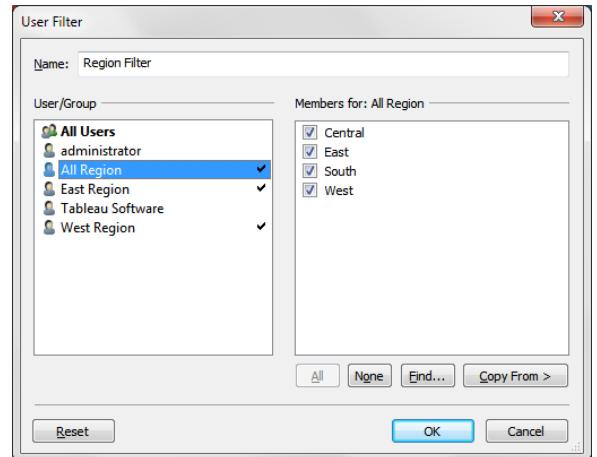


FIGURE 9–2 Tableau Desktop user filter dialog box

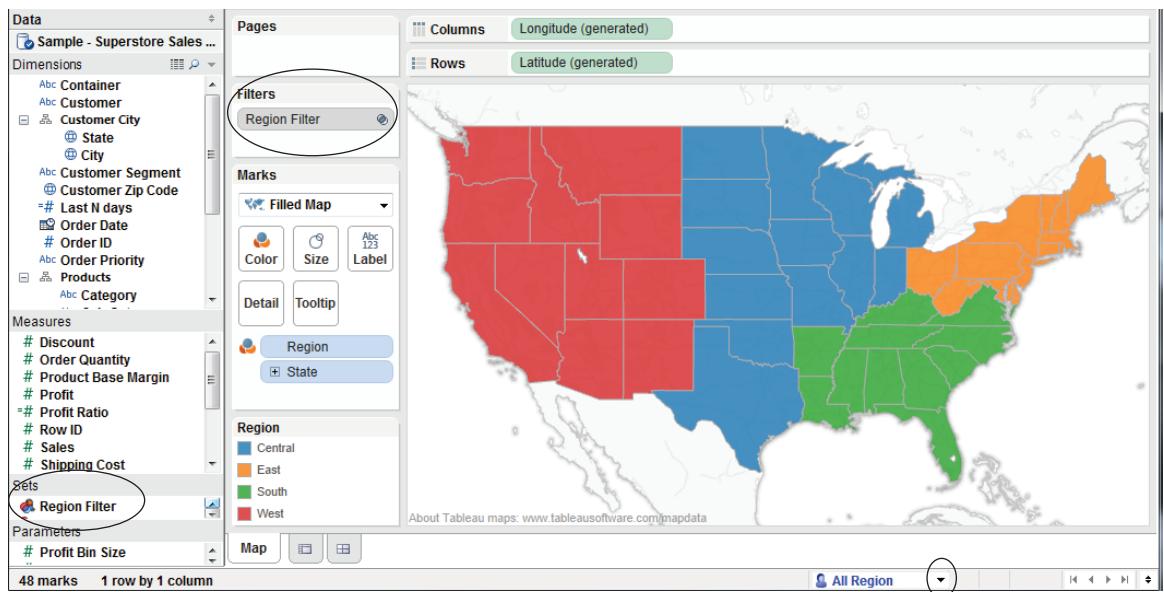


FIGURE 9–3 Filter shelf with region filter added

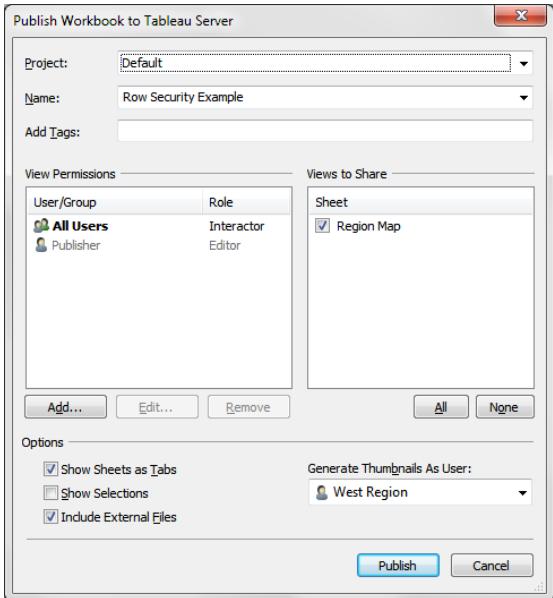


FIGURE 9–4 *Thumbnail filtering*

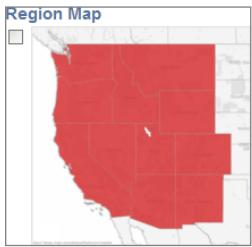


FIGURE 9–5 *Thumbnail generated for a West Region user*

In Figure 9–5 you see how the thumbnail in Tableau Server appears to a West Region user.

Logging into Tableau Server as an East Region user results in the thumbnail view being restricted for access to the eastern states in that territory. When the user selects the View, the actual map view will look like Figure 9–6, and include only the states that comprise that region.

An All Region user logging in will see every state as you see in Figure 9–7.

User filters are an effective and simple method of implementing Row-level security within Tableau Server. However, if the users are allowed to download the workbook from Tableau Server—and open it in Tableau Desktop—they can remove the filter and expose the unfiltered report. Keep this in mind to prevent unrestricted data access. Also remember that users with publisher access can republish reports after removing the filter and provide unrestricted access to the report. Ensure that individuals with publishing rights on Server are trained to prevent unauthorized distribution of sensitive data. Using datasource user filters can also be helpful for limiting the quantity and scope of data exposed.

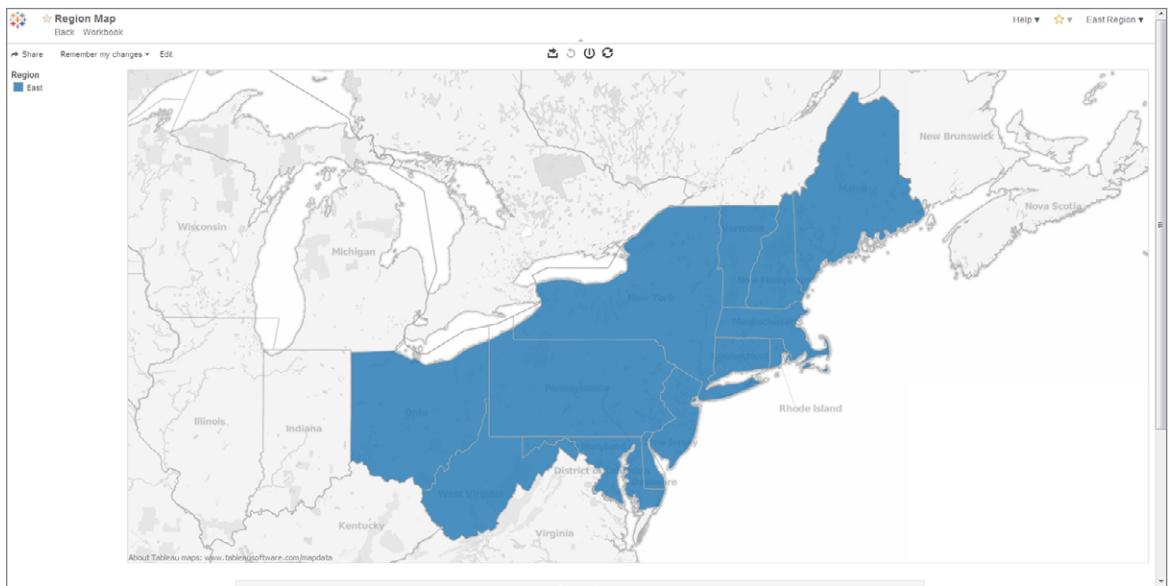


FIGURE 9–6 Map visualization restricted by the user filter

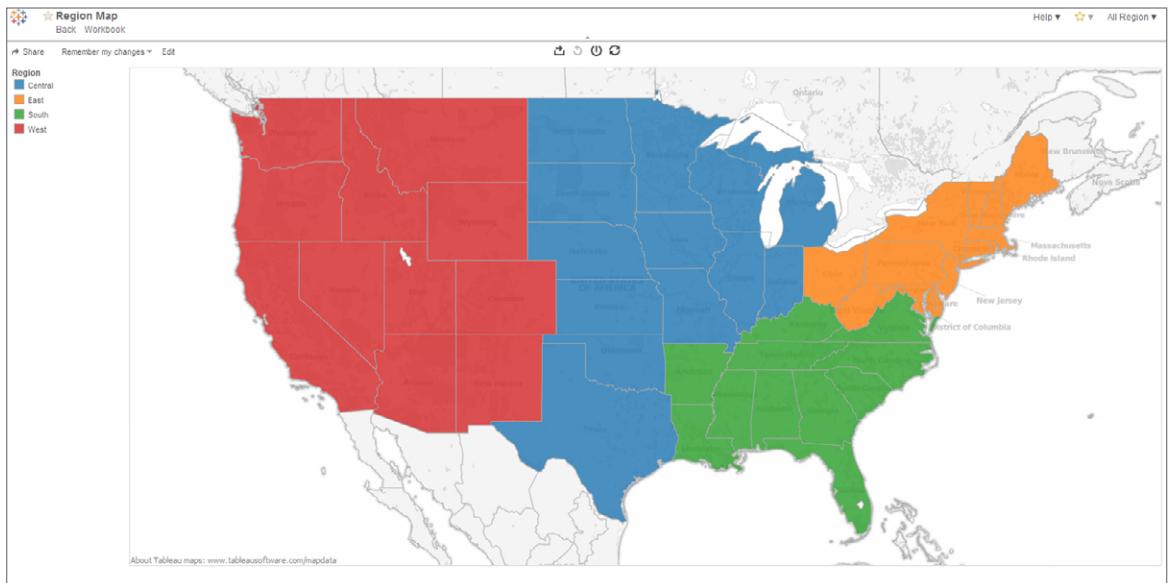


FIGURE 9–7 Map filter for an All Region user

There are other more complex methods of implementing Row-Level security. Be mindful that unless the data is secured at the datasource level, it isn't possible to prevent unrestricted access if the users are allowed to download the workbook and those people have access to Tableau Desktop. Implementing Row-Level filtering requires diligent user permission management to ensure no unexpected data access occurs.

WHEN AND HOW TO DEPLOY SERVER ON MULTIPLE PHYSICAL MACHINES

Earlier in this chapter you read about considerations for sizing hardware for Tableau Server—specifically the concepts of scaling-up and scaling-out. Scaling-up refers to using more powerful single server hardware. Scaling-out refers to bringing in more machines to help carry the workload. Clustering, distributed environments, and scaling-out all refer to the same concept: running Tableau Server on more than one machine, to spread the workload.

The decision to scale-out Tableau Server in a cluster is normally made when a single server cannot support the expected workload, and when adding additional machines represents a lower expected cost than scaling-up to a substantially more powerful single machine. Tableau's multiple processes can be assigned to different machines in the cluster to achieve efficient division of the workload.

For instance, an environment that makes use of very large data extracts could devote an entire machine in the cluster for running data extract engine processes. This machine could include a larger amount of system memory and fast I/O to support the need to quickly load and query many data extracts. In addition, another machine with very fast CPU cores could be dedicated to VizQL processes if high numbers of concurrent view requests are anticipated. Clustering Tableau Server can also provide high availability capabilities by creating redundant core processes on multiple machines. High availability configurations will be in the next section of this chapter.

In Tableau Server clustered environments, the first machine Tableau Server is installed on is known as the Primary Tableau Server, or, the Gateway. All other machines are known as Workers. The Gateway handles all of the requests to the Tableau Server and communicates with the workers to satisfy those requests. To set up a distributed cluster environment follow these steps:

1. Install Tableau Server on the primary machine. (Note the IP address of this machine.)
2. Stop the Tableau Server service on the primary machine.

3. Install the Tableau Server worker software on all of the Worker machines.
4. Return to the primary (Gateway) server and open the configuration utility.
5. Select the Servers tab and click the Add button.
6. Type the IP address of one of the Worker machines in the dialog box.
7. Specify the number of, and each type of, processes to deploy on the Worker.
8. Click OK.
9. Repeat the same steps for each Worker machine.

Once all of the Workers are added to the cluster, save the changes within the configuration utility and restart the Tableau Server service on the primary machine. For more information about clustered Tableau Server deployments see the Distributed Environments section of Tableau Software's Tableau Server Administrators Guide.

DEPLOYING TABLEAU SERVER IN HIGH AVAILABILITY ENVIRONMENTS

As Tableau Server usage increases, the need to ensure its continuous availability also rises. Strategies to guarantee constant availability are broadly referred to as high availability. These strategies necessitate that core components of Tableau Server be redundant to minimize the chance of unplanned downtime. Realizing this goal requires deployment in a distributed environment and running redundant critical processes on separate servers.

Achieving significant redundancy can be realized using a three-server cluster, but to achieve a fully redundant configuration, at least four servers are necessary.

THE THREE NODE CLUSTER

In this configuration, one node hosts the Gateway, which routes requests to the other two Worker servers. The two Worker servers both run all of the server processes. Even though all of Tableau Server's processes should be made redundant, the three processes that must be made redundant are the gateway, the data engine process, and the repository process. Prior to Tableau Server Version 8, both Workers have instances of the repository and data engine processes, but only one of the two Workers was actively accepting requests. Now both data engine processes actively accept requests, even though one is considered primary. In prior releases, the secondary Worker has standby

copies of the processes and is automatically promoted to active status if the main Worker fails. This promotion is called fail-over.

The loss of a Worker machine can occur without making the cluster inaccessible. However, since there is only a single Gateway machine, should that server go offline the cluster will be inaccessible to users. To have complete fault tolerance, a four-node cluster is required.

THE FOUR NODE CLUSTER

In a four-node cluster, a second Gateway machine is added to make that critical node redundant. However, this standby Gateway server must be promoted to active status manually. There is currently no automatic fail-over for Gateway machines.

The high availability setup process is similar to the basic cluster configuration detailed in the When and How to Deploy Server on Multiple Physical Machines section of this chapter.

The steps to set up a high availability configuration are:

1. Install Tableau Server on the primary machine (note the IP address of this machine).
2. Stop the Tableau Server service on the primary machine.
3. Run the Tableau Server Worker installer on the other machines included in the cluster (the primary server IP is needed for this step).
4. Open the configuration utility.
5. Select the Servers tab and click the Add button.
6. In the Add Tableau Server dialog box type the IP address of the first of the Workers.
7. Specify the number of each type of process.
8. Ensure that both the extract storage and repository storage are included on this host's settings. Click OK.
9. Start the Tableau Server service on the primary machine.
10. View the server status and observe that the instances of the extract engine and repository on the new worker appear to be down. This will be resolved once the primary server has transmitted all data for these processes to the new worker machine.

- 11.** After the worker extract engine and repository processes switch from Service Down to Service Standing By, stop the Tableau Server service on the primary machine again.
- 12.** Open the configuration utility on the primary server.
- 13.** Clear the extract storage in the configuration utility on this host and the repository storage on This Host check boxes for the primary server. Remove all other processes to configure this machine as a Gateway only. Click OK.
- 14.** Click the Add button on the servers tab.
- 15.** In the Add Tableau Server dialog box, type the IP address of the second Worker and specify the number of each type of process. Be sure to check both the Extract Storage and Repository Storage on this host's settings. Click OK.
- 16.** As an optional step, you can configure e-mail alerts about the cluster status from the E-mail Alerts tab in the configuration utility.
- 17.** Close the configuration utility and restart the Tableau Server service.
- 18.** Once the service comes back up check the status of the cluster from the Tableau Server maintenance page. You should see the IP address of the primary server listed with only the Gateway service. You should also see the two Worker server IP addresses listed with the remaining Tableau Server processes. One Worker will have an active data engine and repository and the other Worker will have standby copies of these processes.

The three-node configuration presented earlier may be augmented with a redundant Gateway server to increase reliability. For information about making the Gateway redundant and the manual fail-over process see the “Configuring a Highly Available Gateway” section of the Tableau Server Administrator’s guide.

LEVERAGE EXISTING SECURITY WITH TRUSTED AUTHENTICATION

Tableau Server is frequently deployed in landscapes that contain legacy systems that already contain security protocols to prevent unauthorized access. These systems may include internal portals, content management systems, or existing reporting interfaces. Is it possible to embed an interactive Tableau visualization into a site that already contains a legacy security protocol? The answer is yes. This is commonly referred to as Single Sign-on. The Tableau Server system for enabling this is called Trusted Authentication.

When using Trusted Authentication, it is assumed that the web server containing the embedded views will handle the user authentication. The person attempting to access the embedded view must be a valid user on both the web page and Tableau Server. The web page server passes the username of the person that has logged in to the Tableau Server. So, the usernames must match or be programmatically transformed to match.

Tableau Server must also be configured to acknowledge the web page server as a trusted server. This is configured using the Tableau Server Administration Tabadmin tool. The web page server must also be able to perform a POST request and transform the response into a URL. This means that static web pages that are not supported by a scripting language will not be able to support these requirements.

If the web page server uses Security Support Provider Interface (SSPI), configuring Trusted Authentication is unnecessary as long the users are valid members in Active Directory. In that case, Tableau Server authenticates the user via Active Directory as long as the users are also licensed to access Tableau Server. The flowchart in Figure 9–8 illustrates how security data travels between each component.

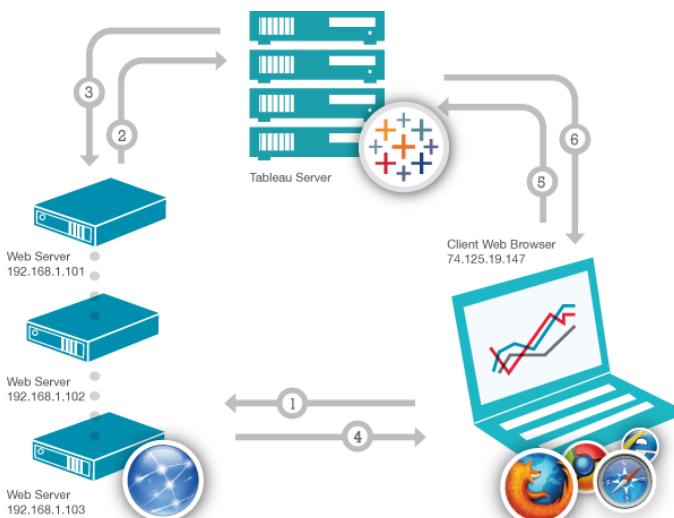


FIGURE 9–8 Trusted Authentication

If all of the above requirements are met then Trusted Authentication works in the following way:

1. A user visits the web page—When a user visits the web page 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. If the server is running multiple sites and the view is on a site other than the Default site, the POST request must include a target site parameter.
3. Tableau Server creates a ticket—Tableau Server checks the IP address of the web server that sends the POST request. If it is set up as a trusted host, then Tableau Server creates a ticket in the format of a unique nine-digit string. Tableau Server responds to the POST request with that ticket. If there is an error and the ticket cannot be created Tableau Server responds with a value of -1.
4. Web server passes the URL to the browser—The web server constructs a temporary URL for the view using either the view's URL or its object tag (if the view is embedded) and inserts it into the HTML for the page. The ticket will include a temporary address (for example: `http://tabserver/trusted/<ticket>/views/requestedviewname`). The web server passes the HTML for the page back to the client's web browser.
5. Browser requests a view from Tableau Server—The client web browser sends a request to Tableau Server using a GET request that includes the URL with the ticket.
6. Tableau Server redeems the ticket—Tableau Server sees that the web browser requested a URL with a ticket in it and redeems the ticket. Tickets must be redeemed within three minutes after they are issued. Once the ticket is redeemed, Tableau Server logs the user in, removes the ticket from the URL, and sends back the final URL for the embedded view.

The Tableau Server installation manual provides examples of the code required for the web server to handle the POST to Tableau Server, converting the ticket into a URL and embedding the view in many languages. These examples are included as a part of the Tableau Server installation. Navigate to the following directly to view them:

`C:\Program Files (x86)\Tableau\Tableau Server\8.0\extras\embedding.`

For tips on using Trusted Ticket Authentication with views that you wish to embed in other websites, see the section on “Use Trusted Ticket Authentication as an Alternative Single Sign-On Method” in Chapter 10.

DEPLOYING TABLEAU SERVER IN MULTI-NATIONAL ENTITIES

Tableau Desktop and Server support a wide range of locales and languages. This makes it easy to deploy in organizations with diverse nationalities. Language settings refer to the translation of text in the user interface elements within Tableau. Locale refers to the format of numbers and dates. Tableau Server supports English, German, French, Portuguese, Japanese, Chinese, and Korean. It also has support for more than three hundred locales.

Default language and locale options can be configured at the server level by users with system administrator permission. This option is located on the Maintenance page of the Tableau Server web interface. The default language is initially determined by the language settings of the server on which the software is installed. Figure 9–9 shows the dialog box.



FIGURE 9–9 The language and locale dialog box

Users can also configure their individual language and locale settings from the User Account page. However, users must do this from their view of the User Account page. Administrators cannot set language and locale options for a specific user. When a user changes these settings, this overrides the default language and locale settings designated by the administrator.

If the user does not have a language and locale specified on their user account page, those settings can also be taken from the user’s web browser—if the browser is using a language that Tableau supports. Also, keep in mind that the author of a workbook in Tableau Desktop can specify language and locale setting there as well. Settings specified in the workbook take precedence over all other language and locale settings.

The order of precedence—from highest to lowest priority—is designated as follows:

1. The Tableau workbook
2. The user preferences page
3. The locale specified by a user’s browser
4. The Tableau Server maintenance page
5. The computer on which Tableau Server is installed

Keep in mind that language options do not translate any report text—only Tableau user interface elements.

USING PERFORMANCE RECORDER TO IMPROVE PERFORMANCE

Earlier, at the end of Chapter 8 you learned how to use Tableau’s Performance Recorder to improve workbook performance in Tableau Desktop. There is also a separate Performance Recorder that allows you to record and view information about Tableau Server performance at the workbook level.

Prior to Tableau Version 8, this data had to be collected and analyzed manually from log files or via a third party application that was created by InterWorks. The Performance Recorder basically creates a Tableau workbook of your Tableau workbook’s performance. Information about the following events is captured and displayed visually:

- Query execution
- Geocoding
- Connections to datasources
- Layout computations
- Extract generation
- Data blending
- Server rendering

Performance Recorder is disabled on Tableau Server by default. To begin using it you must enable it on a per site basis. To activate Performance Recorder on the server, navigate to the Administration-Sites page and check the site you wish to enable. Click Edit. In the Edit Site dialog, check the Allow Performance Recording check box and click OK. Figure 9–10 shows the edit dialog box properly checked.

To use the Performance Recorder on a view you must append the following code ?:record_performance=yes to the view URL. If everything is working correctly, you should see a Show Performance Recording command in the view status bar. Clicking this link will open a view that is generated from the recorded performance data. This view does not automatically update. To see the most current data, close and open the view again. The

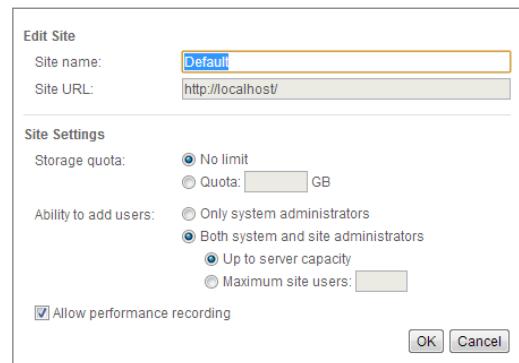


FIGURE 9–10 Enabling performance recording

Performance Recording will continue capturing data about interactions with the view until the user navigates away or removes the string from the URL. Figure 9-11 displays an example of the information available in the Performance Recording summary display.

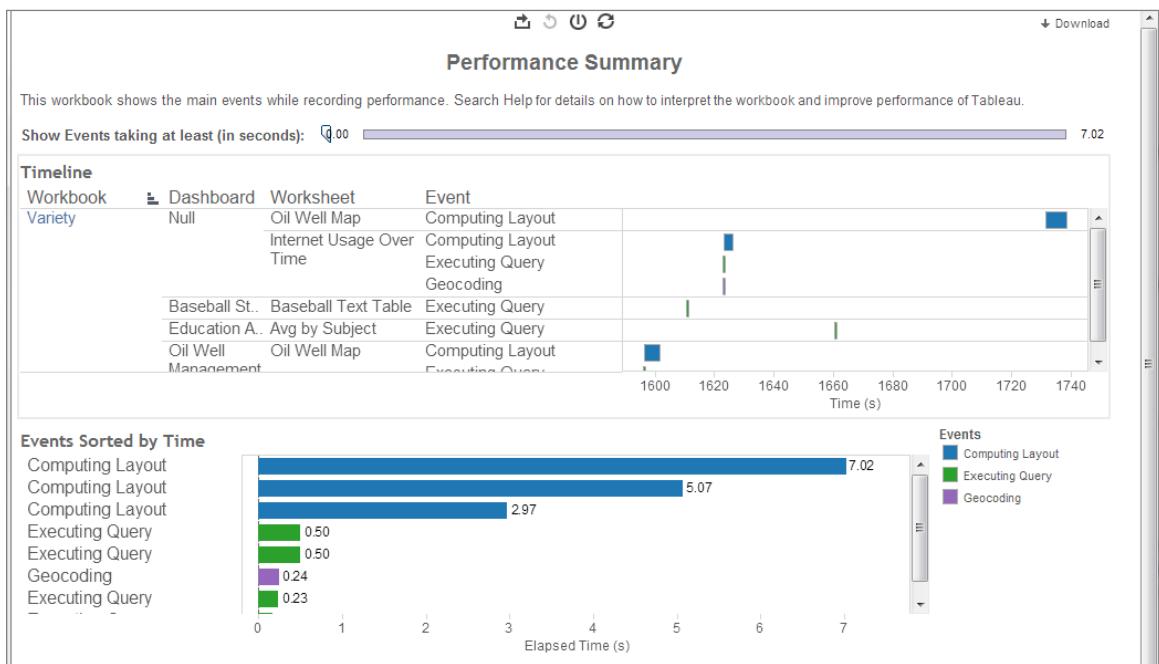


FIGURE 9-11 A Performance summary workbook

The example recording was taken while interacting with one of the sample Tableau Server workbooks supplied with your server license. The dashboard that the Performance Recorder generates contains three views:

- Timeline—A Gantt chart displaying event start time and duration
- Events Sorted by Time—A bar chart showing event duration by type
- Query Text—Optionally appears when clicking-on an Executing Query event in the bar chart

TIMELINE GANTT CHART

The Timeline Gantt Chart displays by workbook, dashboard, or worksheet when each event occurred. Event start time is indicated by the bar's horizontal position, and the duration of each event is indicated by the individual bar length.

THE EVENTS SORTED BY TIME

This section of the workbook shows the duration of recorded events in descending order. This is useful for observing the execution time of each event that occurred during the Performance Recording. This will help you identify any lengthy events that may be the cause of performance problems.

QUERY TEXT

Optionally, the workbook also displays the query text for any specific event that you want to examine in detail. You access the detail by clicking on any of the green Executing Query events in the bar chart. This is a handy feature that allows you to review any query text that may be of interest without having to leave the Tableau Performance Summary dashboard.

PERFORMANCE-TUNING TACTICS

The Performance Summary report generated by the Performance Recorder informs you about the specific events that may be contributing to slow performance. Once you understand the events most affecting performance try the following tactics to address the performance problem.

Query Execution

Query Execution represents the time that it takes for the datasource to execute a query and retrieve the data requested by the worksheet. If the datasource is a database, it is very helpful to see the queries issued by Tableau in order to identify inefficiencies. Common issues include poor indexing strategies, fragmented indexes, database contention, insufficient database resources and inefficient SQL queries. If the datasource is the Tableau data engine, there are fewer troubleshooting options.

Geocoding

Geocoding represents the time Tableau needs to locate geographical dimensions. If this event type is consuming too much time, consider geocoding your records in the source data set and passing a pre-calculated latitude and longitude to Tableau rather than having Tableau generate the geocodes when rendering the map view.

Connecting to the Data Source

Connecting to the Data Source conveys the time required for Tableau to connect to the datasource. This event is typically not a large percentage of total worksheet time. In rare cases there can be a network or datasource issues that

extend connection times. To rule out these issues examine the network topology between the Tableau Server and the datasource server.

Layout Computations

This is the time needed for Tableau Server to compute the visual layout of the worksheet in the Layout Computation event. This can be influenced by server resource contention as well as worksheet complexity. The more marks that are visualized within the workbook, the more time that workbook will require to load and refresh. It may be necessary to restrict the number of marks simultaneously displayed through techniques such as actions, filters, and aggregation. Large crosstabs can be particularly costly, and are not a good visual analytic technique. If all these tactics fail to result in noticeable improvement, it may be necessary to provide additional resources to the server.

Generating Extract

The amount of time that the data engine spends generating an extract is called the Generating Extract event. The size of the datasource (the numbers of rows and columns) along with the time Tableau spends compressing and sorting the data are the major factors affecting the time required to generate extract files.

If your extract file is taking too long to refresh in your environment, it may be possible to speed-up the process by removing unnecessary columns from the extract. This will reduce the time required for generating, sorting, and compressing the remaining columns. Should the problem persist, you may want to ensure that all fields have the appropriate data type assigned to them in the underlying database. Improperly defined field types in the source database can affect the performance during the extract creation, as well as any subsequent queries needed to be performed against the extract file.

If extract generation speeds are still not good enough, try running more data engine processes or placing them on their own Worker instance.

Blending Data

The amount of time that Tableau Server spends performing data blends is the Blending Data event. This event can take a long time when working with large amounts of data from the blended datasources. Filtering before the blend at the datasource level can be effective. If possible, consider moving data into a single datasource so that joins can be used instead of blending.

Server Rendering

The amount of time that Tableau Server spends rendering the computed layouts into a format to send to the client browser is the Server Rendering

event. This time it takes to complete this event can be impacted by the load on the VizQL processes as well as the complexity of the layouts. Refer to the Computing layouts event for guidance.

Whether specifically mentioned or not, most of these events can be quickened by restricting the amount of data visualized through filtering or aggregation. This can also be achieved by using faster hardware or adding more resources on Tableau Server. As far as workbook performances goes, if it doesn't perform well in Tableau Desktop, it won't perform well in Tableau Server either. For this reason you should use the Performance Recorder on the desktop to troubleshoot performance issues there before publishing an under-performing workbook to the server.

MANAGING TABLEAU SERVER IN THE CLOUD

Increasingly, organizations are choosing to move away from hosting on-premise servers by migrating to cloud-based solutions. Flexibility and decreased initial costs are two reasons for pushing software into the cloud.

WHAT DOES IT MEAN TO BE IN THE CLOUD?

Before discussing cloud-based Tableau Server hosting options, it might be helpful to define what we mean by cloud-based. The term, "In the Cloud" has become a catch-all term in recent years for any service that isn't hosted by an on-premise server. That definition doesn't quite capture the scaling implications of the cloud though. Cloud solutions are typically hosted and rapidly scalable systems. As mentioned at the beginning of this chapter, Tableau Software has two server versions that operate only in the cloud.

TABLEAU'S CLOUD-BASED VERSIONS OF SERVER

Tableau Public is a Tableau Server implementation hosted by Tableau Software that is free to use but comes with some caveats. Chief among these is that all workbooks and data hosted on Tableau Public are just that, public. This is probably a deal breaker for most organizations. However, if your organization wants to make data available to the public anyway, this is a great (free) solution. Other caveats with Tableau Public are:

- Datasources are limited to 1,000,000 rows per datasource.
- Only file-based datasources can be used.
- Data is limited to 50 megabytes per account.

Data extracts are the most common datasource used on Tableau Public.

For groups wanting to present reports to the public, but also needing to maintain control of their underlying data, Tableau has an offering called Tableau Public Premium. A fee-based service, Tableau Public Premium allows users to restrict access to the underlying data in a workbook and also prevent downloads of that workbook. Additionally, it removes the maximum row limit imposed by Tableau Public. As these two offerings can't restrict access to reports in any way, they are clearly not for organizations with sensitive data.

Tableau Online provides an added measure of control and security beyond Tableau Public. It is a cloud-based version of Tableau Server that is licensed on a per-named-user basis with no minimum requirement on the number of licenses. The software is installed and maintained by Tableau Software in a secure hosting facility. It is very easy to use Tableau Online. Once you have signed-up, you can start publishing workbooks for other licensed Tableau Online users to view.

There are a few differences between Tableau Online and Tableau Server including:

- Workbooks published to Tableau online must use Tableau Data Extract that must be refreshed regularly. Live connections to Amazon Redshift are supported as well.
- No guest access. Everyone using Tableau Online must be licensed to use the service.
- No custom branding in the Tableau Online environment
- Tableau Software creates and maintains your site.
- No minimum user requirement

At the beginning of this chapter, Tableau's three different server products: Tableau Server, Tableau Public, and Tableau Online were introduced. Currently, the majority of Tableau Server customers want to host Tableau Server on-premise, behind their company firewall. But, an increasing number of organizations are choosing to host Tableau Server in the cloud.

PUTTING TABLEAU SERVER IN THE CLOUD

Although Tableau Server is most frequently hosted within company networks, it too can be hosted in the cloud by utilizing Amazon EC2 instances and most other services that provide cloud-based Windows Server platforms. Amazon EC2 is not currently a platform supported by Tableau Software, but it does work. There are a few items to consider if you want to deploy Tableau Server using a cloud service provider. You are still fully responsible for the installation

and maintenance of Tableau Server deployed this way—unless you also want to farm-out this work on a contract basis to consultants.

Tableau Server needs to be accessible to your users, so make sure ports are opened in any firewalls and that the server will accept traffic from your users' network addresses. Active Directory integration can be tricky with these platforms so consider local authentication if you encounter issues.

When deploying Tableau Server in multi-node configurations, ensure that the IP addresses of the nodes are static so that node communication won't be impaired through system restarts. Also ensure that firewall rules are in place to allow nodes to communicate with each other. The most common issues with running Tableau Server in a cloud environment are networking related. Once the Tableau Server is installed and accessible, administering it is very similar to administering a locally-installed host.

MONITORING ACTIVITY ON TABLEAU SERVER

As your server deployment grows you can monitor usage activity to ensure the best experience for your users. Tableau server includes an administrative maintenance menu that displays most of the information you'll need in order to monitor the status of the processes running on each server. There are also a series of Tableau workbooks accessed via the analysis section that you can see in Figure 9-12.

The screenshot shows the Tableau Server maintenance interface. On the left is a navigation sidebar with links like Admin, Content, Maintenance, Sites, Licenses, Users, Groups, Projects, Data Connections, Schedules, Tasks, Subscriptions, Activities, and Settings. The main area has tabs for Status, Analysis, Activities, and Settings. The Status tab shows a table of server components: Machine (192.168.1.134), VizQL Server, Application Server, Background Tasks, Data Server, Data Engine, Repository, and Web Server. The Analysis tab lists various views: Server Activity, User Activity, View Performance History, Background Tasks, Space Usage, and Customized Views. The Activities tab shows options to Rebuild search index and Clear all saved data connection passwords. The Settings tab lists three checkboxes: Embedded Credentials (checked), Scheduling (checked), and Saved Passwords (unchecked). Below these are dropdowns for Default start page, Default language and locale, and a link to Reset all settings to their default values.

FIGURE 9-12 Tableau Server maintenance menu

(c) OOBTech 2018

THE STATUS SECTION

The status section displays the current status of processes that are available on each machine deployed. The example in Figure 9–12 is for a single server. If you have a multi-cluster setup you will see each machine’s IP address shown on its own row.

THE ANALYSIS SECTION

The Analysis section provides links to embedded Tableau workbooks that provide metrics on these areas of interest:

- Server Activity—Displays thirty days of views with information on workbook counts, users, and sites
- User Activity—Shows user activity by server with detailed user activity available as well
- Performance History—Provides information on particularly taxing workloads impacting performance
- Background Tasks—Gives you views of the tasks, primarily those triggered by user actions
- Space Usage—Shows you how much space is being used by user, project, workbook, and by datasource
- Customized Views—Displays the user-modified views changed using the Remember My Changes feature

THE ACTIVITIES SECTION

There may be some problem in your server setup that causes Tableau Server’s search index to become corrupted. If this happens, users may not get the correct results when searching for workbooks or datasources. Run the Rebuild Search Index if searches are not bringing back the expected results.

If you enabled the Saved Passwords option for people accessing Server, this allows users to save passwords for datasources in multiple browsers and visits. If you need to force everyone to enter passwords again, the Cleared Saved Data Connection Password For All Users option will require everyone to enter a password the next time they visit. This can also be done more selectively for individual users via the user preferences menu.

THE SETTINGS SECTION

The options included in this section allow you to control how the behavior of credentials embedded in workbooks assign workbooks with embedded credentials to schedules, and provide a way for users to save datasource passwords for multiple browsers and visits. You can also change Tableau Server's default start page, default menu language, and locale for displaying numbers. Another feature is the ability to return any altered settings to the default values. See Tableau Server's on-line manual for more details.

Tableau has recognized the need to provide enterprise class management tools for monitoring activity, security, and performance. Tableau Version 8 provides significantly more tools than prior releases. In Chapter 10 you'll see how easy it is for users to publish and consume reports in Tableau Server.

NOTES

1. Godin, Seth: *Linchpin: Are You Indispensible?* Penguin Group (USA), 2010.
Print. Page 154.

Using Tableau Server to Facilitate Fact-Based Team Collaboration

All good-to-great companies began the process of finding a path to greatness by confronting the brutal facts.

JAMES COLLINS¹

Tableau Server facilitates information sharing and team collaboration by making interactive dashboards and views accessible to authorized individuals via any of the popular web-browsing software tools available today. Reports can be directly consumed via iOS (Apple) or Android tablets. Beginning with Tableau Server Version 8, authorized staff can also edit existing reports or create completely new analysis using Tableau Server.

Authorized Tableau Server users can also share metadata including joins, groupings, sets, name aliases, and other customized data by publishing Tableau Datasource files to Server. You will learn how to take advantage of these features and more in this chapter.

PUBLISHING DASHBOARDS IN TABLEAU SERVER

After Tableau Server is installed, those creating reports and analysis must be provided with publishing rights. Staff that will be consuming reports must be granted access rights. Once you've created a workbook containing at least one worksheet, you can publish that information to Tableau Server. Workbooks containing many different worksheets and dashboards can be published in full, or by selecting any combination of worksheets and dashboards. The Tableau Desktop menus used for publishing to Tableau Server are shown in Figure 10-1.



FIGURE 10–1 Publishing from Tableau Desktop to Server

Publishing a workbook requires three steps:

1. Open the workbook you wish to publish.
2. Select the Server menu and click on Publish Workbook.
3. Enter the server URL, your username, your password, click OK.

The Publish Workbook to Tableau Server dialog box will appear as you see in Figure 10–2.

Using this menu you define when, how, and what details will be published to the server. If your workbook's datasource is a Tableau Data Extract (tde) file, you can also schedule regular data updates using the Scheduling and Authentication button at the bottom of Figure 10–2.

Tableau organizes and secures published workbooks using a variety of methods:

- Project—Folders for grouping workbooks
- Name—Naming workbooks
- Tags—Allowing for user-defined tagging of workbooks
- Permissions—Controlling what users are permitted to do
- View—Hiding or sharing specific views

Creatively combining these entities facilitates secure access at the appropriate level for individuals, teams, work groups and projects. The specific purpose of each is explained in more detail below.

PROJECT

Projects are folders for organizing your reports and controlling access to those reports. Server comes with one Default project folder. Those with administrative rights can create additional projects. Figure 10–2 shows an additional project called Demo Dashboards, which was added to hold the reports being published.

NAME

You can accept the name assigned to the workbook when it was created in Tableau Desktop or choose to define a new name that will appear in Tableau Server when the workbook is published. You define new workbook names using the Name field seen in Figure 10–2.

TAGS

Tagging published workbooks is optional but provides another way to search for reports. They can be helpful if you publish a large number of reports. Enter each tag separated by a comma or space. If the tag you are entering contains a space, surround the tag by quotation marks (e.g., "Production Benchmarks").

VIEW PERMISSIONS

Those authorized to do so can optionally add, edit, or remove permissions for all users, groups, or individual users through the View Permissions tool. When editing permissions another dialog box opens, which you can see in Figure 10–3. This box allows you to edit permission types for different roles (viewer, interactor, editor, or custom).

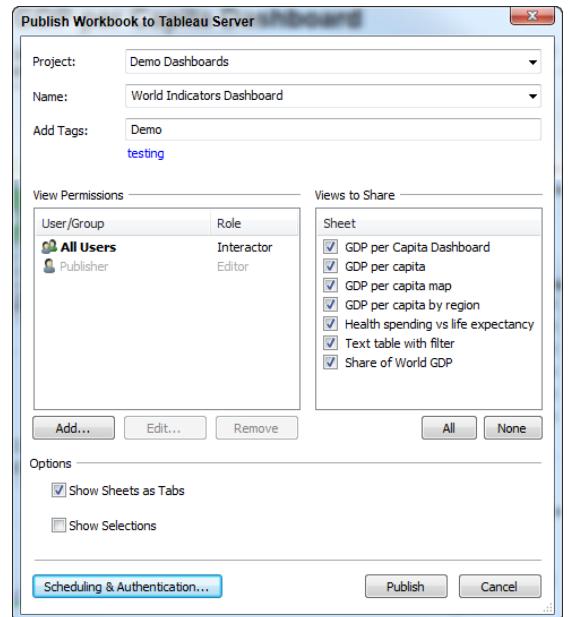


FIGURE 10–2 Publishing dialog box

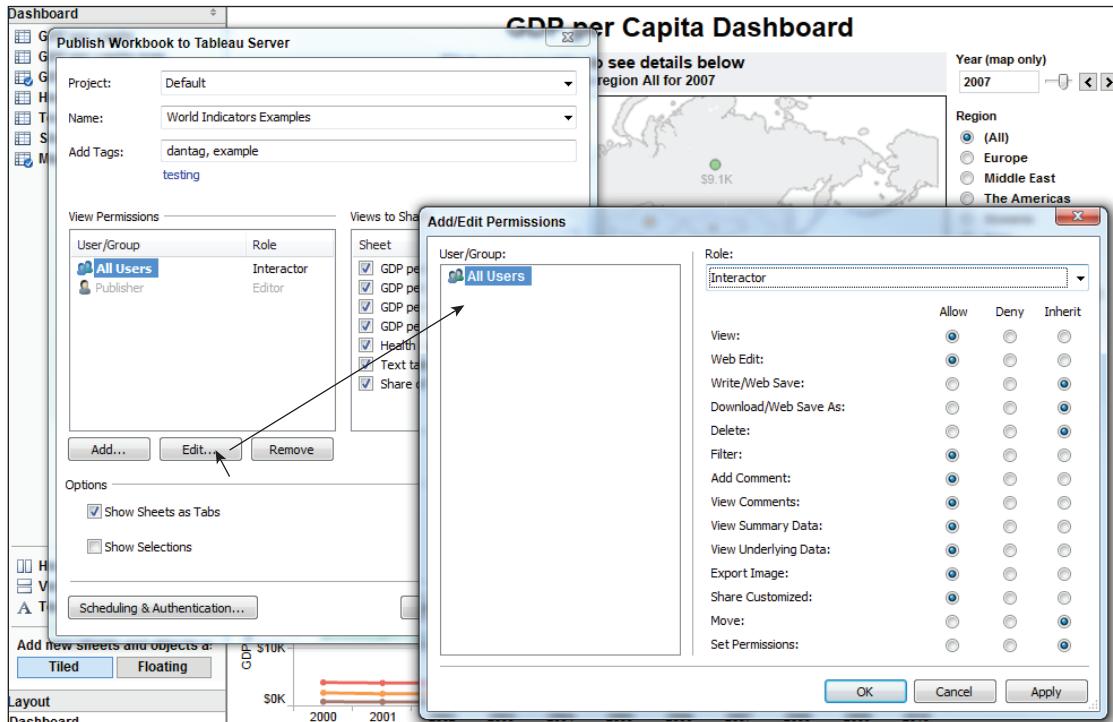


FIGURE 10–3 Adding and editing users and permissions

VIEWS TO SHARE

The Views to Share option allows you to select specific sheets you wish to share. Any sheets that are not selected are hidden on Tableau Server, but these are still available within the workbook if it is downloaded from Tableau Server.

OPTIONS

Appearing at the bottom of the publishing dialog box are more optional selections that control the appearance of what is published.

Checking the Shows Sheets as Tabs option will generate tabs when the report is published to Tableau Server—facilitating navigation between worksheets and dashboards in the published workbook. The Show Selections option allows selections you've made on a worksheet or dashboard to persist when the workbook is published to Server and will be displayed to users consuming the workbook.

If the datasource that you are using for the report being published comes from an external database or file, you will also see a check box for the inclusion of external files; checking that option generates a copy of the source file on Tableau Server. Custom image files used in any view will also be saved. If you have a live database or an extract file being utilized by the workbook, you should also see a button in the lower left for Scheduling and Authentication. Selecting that button allows you to set the refresh schedule for data extract sources or to change how a live database connection is authenticated on Tableau Server. Details regarding scheduling updates and authentication are covered in depth later in this chapter in the sections on “Sharing Connections, Data Models, and Data Extracts” and “Using Subscriptions to Deliver Reports via E-mail.”

Select the Publish button seen at the bottom of Figure 10-2 to initiate the upload to Tableau Server. Upon completion a pop-up will appear displaying the newly-published workbook.

If your Tableau Server instance is configured for multiple sites, you will also see a Select Site dialog box to define on which server site the workbook will be published. Tableau’s default is a single site. Multiple sites are partitions of the same physical server.

ORGANIZING REPORTS FOR CONSUMPTION

Publishing reports to the web is the first step in effectively sharing information on Tableau Server. As your user base grows and reports proliferate, finding reports you’re interested in requires organization. User security, group security, and site security were discussed in Chapter 9. Tableau Server provides two additional ways to organize reports, projects, and tags. Projects are virtual folders in which you publish workbooks and datasources. Tableau also provides built-in support for adding security to each project—allowing you to more easily manage security across multiple workbooks and datasources. Users can also tag particular workbooks or datasource files with keywords. This provides you with a user-defined search term that is helpful for locating files when there are a lot of published workbooks. Defining a sensible framework for projects and tag recommendations might be helpful to your user base and provide some consistency across your enterprise. These could be defined in advance, but you may allow users to define additional tags that meet their specific needs as well.

For example, you may define projects by business unit or function—leveraging tags and adding context to each search. In a university setting three different departments might be consuming reports:

- Admissions
- Financial Aid
- Career Services

The Admissions office might be concerned with tracking the number of students applying each year and whether they were accepted and enrolled. Financial Aid would like to track the amount of aid offered and accepted. Career Services might be interested in monitoring the progress of students that have graduated and are seeking employment. Setting up projects for each office could be advantageous because it will facilitate security while organizing the reports logically for the staff of each department.

Adding tags to each workbook could provide additional context regarding the details. Examples in a University might include:

- Admissions—Undergraduate, admissions, “accepted vs. denied,” enrolled, declined, graduate
- Financial Aid—Aid, grants, loans, scholarships, transfer scholarships, undergraduate, graduate
- Career Services—Offers, accepted offers, max salaries, median salaries, undergraduate, graduate

And please look closely at the admission tag example “accepted vs. denied.” When tags include spaces, they must be wrapped in quotations. Notice the same tags being used in different projects and workbooks. This allows a user to search for similar analysis performed across different departments.

For example, if the dean of a college wanted to quickly find all the reports available analyzing undergraduate students, the dean could search using the Undergraduate tag and quickly access reports related to admissions, financial aid, and career services.

ADDING TAGS TO WORKBOOKS

Users can add tags to any workbook they have the right to access. Figure 10–4 shows a Global tag being added to one workbook contained within a project.

On the left side of the window in Figure 10–4 you can see the view has been filtered for the Dan Playground project. Selecting a workbook and clicking on the tag menu option exposes existing tags that can be applied to the selected workbook. If none of the existing tags work, clicking the Add option opens a dialog box for entering a new tag. In Figure 10–4 you can see a Global tag being added.

Workbooks

- Admin
- Content
- Delete
- Permissions
- Move
- Tag
- Tabs
- Scheduled Tasks
- Download

Sort By: Name ↑ Project Sheet

Book Writing Plan Version 8 Testing

beta
blending
book
dan
extract
maps
play
Schedule
stuff
testing
Testing
Add...

Add Tags
Use commas or spaces to separate multiple tags.
Global

Image Layout Example

InterWorks Google Analytics Pagesviews

VA Administration - Veteran Pop Map 2013

World Bank DGM V8 Examples

FIGURE 10-4 Adding a tag to a workbook

Tags can also be added directly when publishing a workbook. This additional option is located in the menu that is presented when you publish a workbook.

CREATING A FAVORITE

Favorites are workbook views or dashboards that you use often and want to save for quick access. Favorites are accessible via a drop down menu in the upper right corner of the browser window or from the menu on the left side of the screen. Figure 10–5 displays a favorite menu showing a list of favorites. Favorites are indicated using a small star—visible in the upper right area of Figure 10-4 and Figure 10-5.

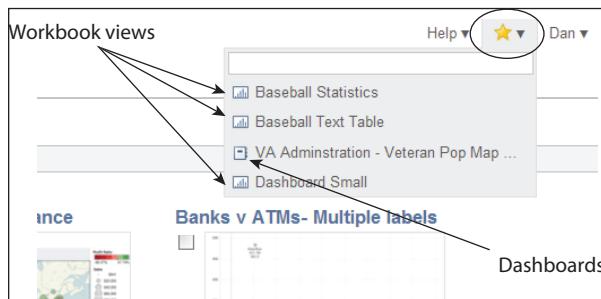


FIGURE 10–5 The favorites menu

Figure 10–5 shows three worksheets and one dashboard have been defined as favorites. Any workbook or worksheet can be made into a favorite from the thumbnail or list views. Figure 10–6 shows a list view of the reports.

Workbooks								
	Name	# Sheets	Size	Publisher	Modified	Project	Tabs	
<input type="checkbox"/>	Book Writing Plan Version 8 Testing	17	460.8 KB	Dan	Jan 22, 2013 10:41 AM	Dan Playground		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Figures for Chapter 5	5	5.5 MB	Dan	Feb 15, 2013 6:45 AM	Dan Playground		<input type="checkbox"/>
<input type="checkbox"/>	Image Layout Example	3	145.5 KB	Dan	Feb 14, 2013 1:56 PM	Dan Playground		<input type="checkbox"/>
<input type="checkbox"/>	InterWorks Google Analytics Pagesviews	3	746.9 KB	Dan	Jan 22, 2013 10:38 AM	Dan Playground		<input type="checkbox"/>
<input type="checkbox"/>	VA Administration - Veteran Pop Map 2013	9	4.4 MB	Dan	Jan 28, 2013 11:01 PM	Dan Playground		<input type="checkbox"/>
<input type="checkbox"/>	World Bank DGM V8 Examples	12	833.7 KB	Dan	Feb 26, 2013 2:03 PM	Dan Playground		<input type="checkbox"/>

Rows per page: 25 Pages: 1 / 1 Prev Next

FIGURE 10–6 List view with a favorite

To make any workbook a favorite select the star associated with the item. This will toggle the star—coloring it yellow—and will add the item to the favorites menu in the upper right as seen in Figure 10–5. You can also add a favorite from the thumbnail view. The most convenient way to access favorites is via the star drop-down menu that you see in the upper right of Figure 10–6. You can also filter for favorites by using the menu on the left. The filter menu also allows you to select specific projects, publishers, tags, or recently used items.

By combining user and group security with projects, favorites, and tags you can control access to sensitive information and allow users to set up their own means for facilitating easy access to the information that is most important to meet their particular needs.

OPTIONS FOR SECURING REPORTS

Managing the security of data and reports is an important consideration. With the exception of a core-licensed server (with guest accounts enabled) all users must log in to Tableau Server before they have access to view any information. Applying permissions at the project level, you can efficiently manage access to a large number of workbooks and datasources while still providing the flexibility to alter security for a single group or user at any time. Securing reports is done using a combination of application layer and data layer controls.

- The Application Layer—Tableau Server credentials
- The Data Layer—database security

THE APPLICATION LAYER

Tableau Server provides application layer security through user credentials. Users can be managed in one of three ways:

- Local authentication
- Microsoft Active Directory
- Trusted Ticket Authentication

In Chapter 9 you learned about details related to managing security for users. Once a user has been authenticated to access the Tableau Server environment, you specify which projects, workbooks, and datasources that user is permitted to see. This is called object-level security. Tableau supports the assignment of object-level permissions for any user group or user by utilizing any of the following objects:

- Project
- Workbook
- Datasource

Using a top-down approach, permissions can be assigned at the project level—which may be inherited by any workbook or datasource published to that project. Permissions assigned to a user group will automatically propagate to all users within the group unless a user has explicit permissions overriding the group settings. The publisher has ultimate control over whether to accept the default permissions or define customized permissions. Tableau server comes with three standard permission levels already defined. These are called Roles and include viewer, interactor, and editor. Figure 10–7 shows the interactor role permissions.

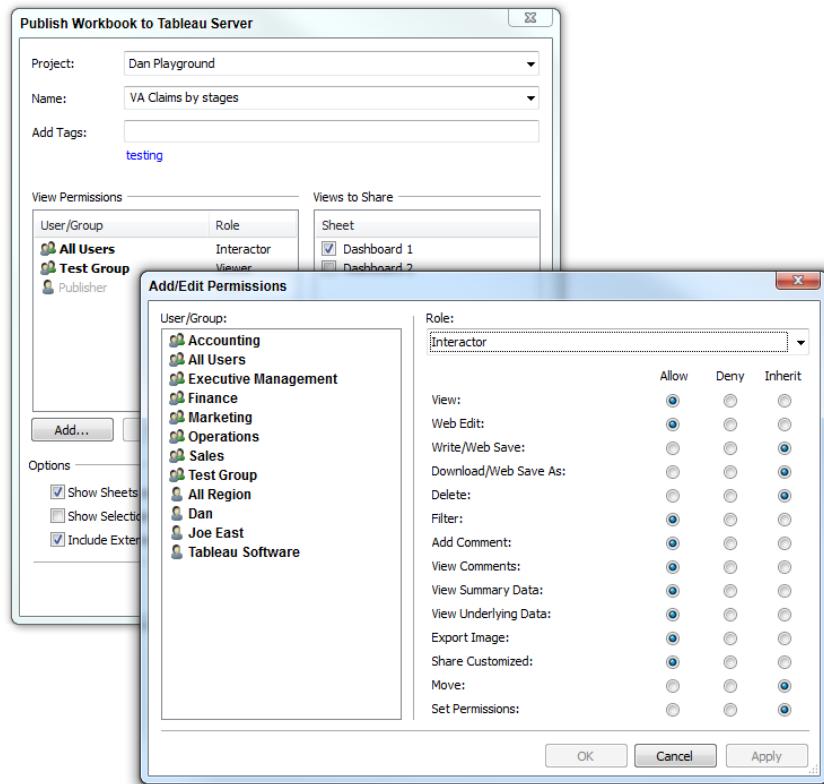


FIGURE 10–7 Interactor permission role menu

The permissions menu is accessed when your workbook is published by clicking the Add button below the View Permissions area seen in Figure 10–7. Other roles can be viewed by selecting the Role menu drop-down arrow. Custom roles can be defined by selecting a user or group, and then choosing the custom role option from the Role menu. This allows you to set customized permissions when assigning your custom role to a specific user or group. Tableau's manual provides step-by-step instructions for defining permissions. Access the appropriate section of the manual from the help menu in Tableau and search for Setting Permissions.

DEFINING CUSTOM ROLES

Customizing roles is done by defining the permissions for the role. Understanding the permissions that you allow is important. Depending on the selections made, you may grant the ability for people to republish reports,

change filters, redesign the workbook views, build new views, export data, download the workbook, share custom views, or even set new permissions. For a detailed description of each capability use Tableau's help menu and search for Permissions.

Care should be taken when granting permission to prevent the unauthorized dissemination of data. The list below categorizes permissions by risk level. High risk items provide the ability for the user to override permissions or disseminate data. Medium risk items convey the ability to alter or export views. Low risk permissions pertain to viewing and commenting capabilities.

High Risk Permissions

- Write/Web Save
- Download/Web Save As
- Move
- Set Permissions
- Connect

Medium Risk Permissions

- Web Edit
- View Summary Data
- View Underlying Data
- Export Image

Low Risk Permissions

- View
- Delete
- Filter
- Add Comment
- View Comments

These risk assessments are meant to be guidelines only. If your data is highly sensitive, care should be taken to mask confidential information at the datasource level to assure that confidential information is not inappropriately exposed.

A PERMISSION-SETTING EXAMPLE

Permissions can be defined so that it is possible to reuse a single workbook for groups with different access rights. For example, you may choose to group users by office. This was described in the university example mentioned earlier in this chapter (admissions, financial aid, and career services). Permissions for related projects could be set so that each office only gains access to the workbooks specifically related to their individual (office) groups.

At the same time, the university president's office could access the workbook, but with different permission settings that permit access to all of the projects and all of the related data details.

As a result, financial aid users won't see the admissions or career services reports. Instead, they will only see and have access to their financial aid reports. Yet, the university president's group will be able to view reports related to all three groups. Using this model, administrators can efficiently manage security for large and diverse entities.

The Data Layer

When employing a live database connection in Tableau Desktop you must provide credentials to authenticate the database server. This data-level security is persisted on Tableau Server as well. When publishing a workbook or datasource you must choose what type of authentication you'll associate with your live connection.

It's important to understand the difference between application layer and data layer security. When a user logs into Tableau Server, the user is authenticated at the application layer but not the data layer. When accessing any report that utilizes a live connection, the user must also be authenticated by that datasource. How the user is authenticated is predicated upon what settings you select when publishing the workbook or datasource. Your choices boil down to four options:

- Prompting the user to enter credentials
- Using embedded credentials
- Using a Server Run as Account
- Using SQL Server Impersonation (available for SQL Server only)

Embedded Credentials

Tableau also offers administrators the option to permit users to save their datasource credentials across multiple visits and browsers. This is enabled through

the Embedded Credentials settings option in the administrative maintenance screen in Tableau Server. Figure 10–8 shows the menu with the appropriate selection checked. You also have the option to embed the connection username and password for the database form within Tableau Desktop.

The screenshot shows the Tableau Server Maintenance interface. On the left, there's a sidebar with options like Admin, Content, Maintenance, Sites, Licenses, Users, Groups, Projects, Data Connections, Schedules, Tasks, Subscriptions, Analysis, Activities, and Settings. The 'Maintenance' tab is selected. In the main area, there's a 'Status' table with columns for Machine, VizQL Server, Application Server, Background Tasks, Data Server, Data Engine, Repository, and Web Server. Below the status table is an 'Analysis' section with several items listed. At the bottom is a 'Settings' section with three items: 'Setting', 'Description', and 'Setting'. The 'Setting' column has checkboxes for 'Embedded Credentials' (which is checked and circled), 'Scheduling', and 'Saved Passwords'. The 'Description' column provides details for each setting.

Setting	Description
<input checked="" type="checkbox"/> Embedded Credentials	Allow publishers to attach passwords to published workbooks that will automatically authenticate web users.
<input type="checkbox"/> Scheduling	Allow publishers to assign workbooks to schedules. Embedded Credentials must also be enabled.
<input type="checkbox"/> Saved Passwords	Allow users to save data source passwords across multiple visits and browsers.

FIGURE 10–8 Enabling embedded credentials

By using this option, all users that utilize the connection will have the same level of access as the publisher of the workbook. This is a convenient feature for users that saves them from having to log-in a second time. However, enabling embedded credentials removes the ability to manage data-level access on a per-user basis.

Server Run As Account using Windows Active Directory

Tableau Server runs in Windows Server environments. Therefore, Tableau Server installations utilize an active directory service account to run. A beneficial consequence of this fact is that Windows Active Directory (AD) can be used to eliminate redundant logins for Tableau Server users.

When a report is viewed on Tableau Server using a data connection employing this method, the Server Run As Account will be used to authenticate against

the database. Your database administrator will need to ensure the Server Run As Account has the proper access to connect to and query the tables and views used in your connection. Use Tableau Server's online manual and search for Run As User to view the setup details for this feature.

SQL Server Impersonation

An option only available when connecting to a SQL Server database, impersonation is another way of eliminating the need for users to log in twice while still preserving the ability to manage data level access on a per user basis. This also allows the SQL Server database administrator to control security policy from the database and propagate those policies to Tableau Server.

To use SQL Server Impersonation each Tableau Server user will need individual accounts on SQL Server with credentials matching those on Tableau Server. For instance, if you have chosen to use Active Directory to manage your Tableau Server users, you must grant the same Active Directory accounts access to SQL Server. The user will either need to be the Server Run As Account or have their credentials embedded in the workbook during the publishing step by selecting the Impersonate Via Embedded Password option in the authentication menu.

When a user views a workbook that has implemented SQL Server Impersonation, they are authenticated using the Server Run As Account or via embedded SQL Server credentials. This account then impersonates the user connecting and accesses the database with their defined permissions. Search Tableau Server's online manual for SQL Server Impersonation for more details regarding setup and configuration.

Tableau Server provides a variety of ways to manage security. In the next section you'll find out how Tableau Server provides more flexibility and efficiency thought the Data Server.

IMPROVE EFFICIENCY WITH THE DATA SERVER

The Tableau Data Server provides a way to manage datasources that have been published to Tableau Server. These published sources can include direct connections to a database, or Tableau Data Extract files. Authorized staff can set permissions associated with the connections and also set refresh schedules for data extract files. The metadata associated with these published sources becomes available to any workbook that uses the datasource. Metadata includes:

- Custom calculated fields
- Ad hoc groupings
- Ad hoc hierarchies

- Field name aliases
- Custom fonts and colors

The Data Server is efficient because it provides a flexible way to spread heavy workloads by enabling Tableau Server to absorb some of the demand normally handled by the primary database server.

While using data extract files is not a requirement, data extract files frequently perform better than the host database. The Data Server also saves time—enabling the work of a single individual to be shared by many. Datasources published to server can be accessed by authorized Tableau Desktop users to create new analysis.

Next you'll learn how to publish a datasource to Tableau Server and then use the Data Server to centrally host and share files, schedule automatic updates, and leverage incremental extract refreshing for near real-time data.

Publishing a Datasource

Publishing a datasource file to the data server is done from Tableau Desktop by opening the workbook containing the datasource you wish to make available for others to use. From the workbook, access the menu for publishing the datasource by right-clicking in the data window containing the datasource in the upper left section of the worksheet as seen in Figure 10–9.

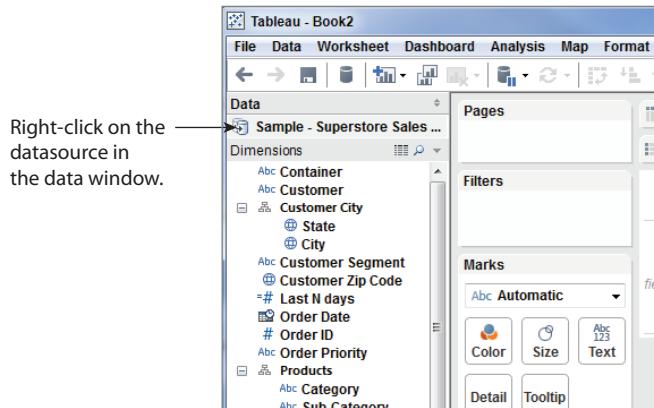


FIGURE 10–9 Publishing to the data server

After right-clicking and selecting publish to server, a server login dialog box appears. You will be required to enter the server URL, your username, and password to access the server. If you have a multiple site deployment you'll also

need to enter the site you want to publish to as well. Once the server login is completed a dialog box will appear as you see in Figure 10-10.

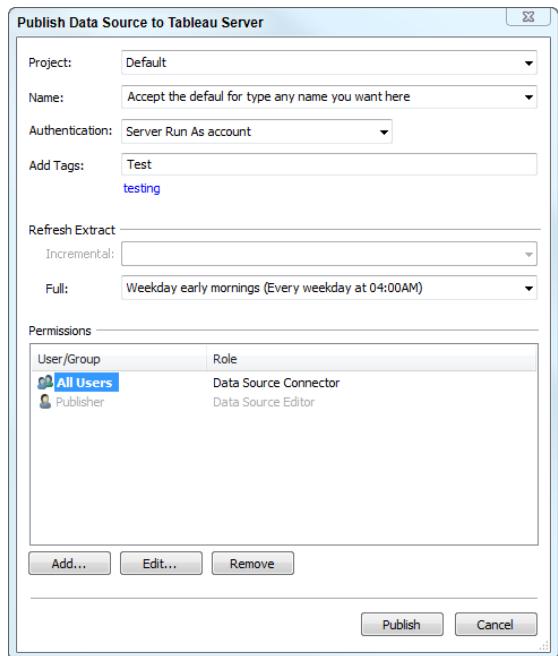


FIGURE 10-10 Dialog box for publishing a datasource

Define the parameters for publishing the datasource by selecting the project, the datasource name, the authentication method, tags, how and when you want server to refresh the extract, and finally what permissions you wish to assign to the extract. Most of these topics have already been covered in Chapter 9 or in earlier sections of this chapter. In the next two sections you will learn more about the options for updating datasource files and how to use incremental updates.

Manual vs. Automatic Updates

One potential benefit of using an extract—a portable copy of your original data set—can also be a drawback. The extract may not reflect the latest changes occurring in the datasource until the extract is refreshed. Tableau provides two different methods for updating extract files—manual and automatic updates.

Manual Updates Using Tableau Desktop

Manually updating data extracts can be done via the data menu or by right-clicking on the data menu. Follow these steps to refresh the datasource file:

1. Start Tableau Desktop if it is not already running.
2. Open the workbook containing the extract you wish to refresh.
3. Select the data menu and refresh all extracts (or add data from a file to append new data).
4. A dialog box will appear displaying the extracts that are available to update.
5. Click on the refresh button to update the extract files.

If your workbook contains multiple extract files they will all be updated using this method. You can also update individual extract files in the workbook by pointing at the datasource in the data window, right-clicking, and then selecting extract, then refresh.

This manual process is one way you can append data from a separate source file or database—assuming the separate source contains the same fields as the original datasource. To do so, follow the same steps as above but in the last step choose Add Data From File instead of Refresh. In Chapter 11 you'll see how Tableau Server's command line tools can be used to automate manual processes.

Automatic Updates Using Tableau Server

If you have many different datasources and workbooks using datasource files published to Server, manually updating large numbers of files would be impractical. Tableau Server comes with a pre-defined update schedule and allows you to create your own custom update schedules.

To schedule updates you'll need to first publish your extract to Tableau Server directly using Data Server or indirectly by publishing the workbook that uses an extract as its datasource. During the publishing process you have the option to select a refresh schedule to have Tableau Server automatically update the extract.

Tableau Server includes predefined schedules or your server administrator can define custom schedules set to recur at a monthly, weekly, daily, or hourly time interval. Schedules can also be defined to allow jobs to run concurrently or sequentially, with an option to change the priority of the schedule relative to others that may occur at the same time. Figure 10-10 presented earlier, shows the schedule option in the Refresh Extract section. The drop-down box next to Full contains the available options. You can see in the example that the extract will be refreshed during weekdays at 4:00AM.

Defining a Custom Refresh Schedule

For those users granted administrative rights, creating custom refresh schedules is done from the Tableau Server admin menu. You can see these schedules in Figure 10-11.

Accessing the admin/schedules menu provides a list of what is available and summary information regarding the available schedules—their type, scope, the number of times run, how they run, as well as the next scheduled run time. To define a new custom schedule, you must select the New menu option you can see above the check boxes in Figure 10-11. Selecting that exposes the custom schedule dialog box you see below in Figure 10-12.

Schedules							
	New	Modify	Delete	Enable	Disable	Run Now	
Maintenance	<input type="checkbox"/>		Name	↑	Schedule Type	Scope	# Tasks
Sites	<input type="checkbox"/>		End of the month		Monthly	Extract	0
Licenses	<input type="checkbox"/>		Every 8 Hours		Hourly	Extract	0
Users	<input type="checkbox"/>		Monday morning		Weekly	Subscription	Parallel
Groups	<input type="checkbox"/>		Saturday night		Weekly	Extract	0
Projects	<input type="checkbox"/>		Test Schedule		Weekly	Extract	0
Data Connections	<input type="checkbox"/>		Test Subscription		Weekly	Subscription	Parallel
Schedules	<input type="checkbox"/>		Weekday early mornings		Weekly	Extract	0
Tasks	<input type="checkbox"/>		Weekday mornings		Weekly	Subscription	Parallel
Subscriptions	<input type="checkbox"/>						

Rows per page: Pages: 1 / 1 Prev Next

FIGURE 10-11 Admin schedule menu

Create New Schedule

Schedule Properties:

Schedule name: Refresh every 2 hours

Schedule scope: Use this schedule for: Extracts/Refreshes

Default priority: 50 (from 1-100, 1 is highest priority, 100 is lowest)

Serialization: Jobs in schedule can run concurrently
 Run jobs sequentially

Schedule Definition:

Hourly every 2 hours from 12 : 00 AM to 12 : 00 AM

Daily at 12 : 00 AM

Weekly Sunday at 12 : 00 AM
 Monday
 Tuesday
 Wednesday
 Thursday
 Friday
 Saturday

Monthly on the last day of the month at 12 : 00 AM

Create Schedule

FIGURE 10-12 Creating a custom schedule

Give the schedule a clearly descriptive name and fill in the highlighted blanks. Then, click the Create Schedule button. This makes the schedule available for use. As you can see, there is plenty of flexibility for controlling when data extracts are refreshed.

INCREMENTAL UPDATES

What if you have a particularly large or very active datasource? Very large source files can take time to update. You can reduce the time required for extracting data by employing incremental updates. Typically when refreshing extracts, the current rows are truncated and completely replaced by a new copy of the data set. In contrast, incremental refreshes allow you to specify a date, date-time, or an integer value field contained in your data, to specifically identify new records in a datasource.

When an incremental refresh is used, Tableau will check for the maximum value of the field in your extract and compare that value to each row in the original datasource—importing only the rows with a later or higher value. This approach will reduce the time required to update your extract. The larger the source file, the more significant will be your potential time savings.

You define this option in Tableau Desktop when you build the extract definition by selecting the incremental refresh option, then selecting the field that you want to use to identify the new data. The field options you see in Figure 10–13 include the order date or ship date fields.

If you choose to use incremental refreshes, you are not excluding the option for a full refresh. On the contrary, you are only allowing for the additional choice of an incremental refresh when either performing a manual or automatic update. It is advisable to run full refreshes of the data on a regular basis because the incremental refresh may not capture all of the changes in the source data set.

CONSUMING INFORMATION IN TABLEAU SERVER

As your Tableau deployment matures you may have hundreds of reports and datasources being published, updated, and consumed. Facilitating access

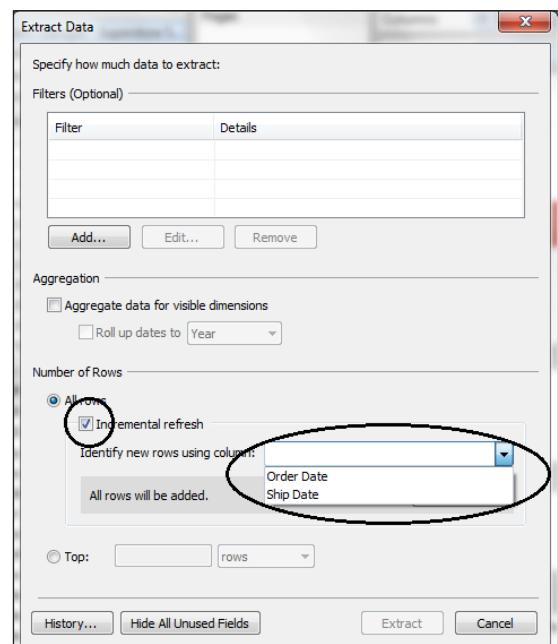


FIGURE 10–13 Enabling incremental updates

to information and encouraging collaboration is of primary importance, and that is the principle value business information systems provide. Tableau Server provides tools for finding information, commenting on reports, sharing discoveries, or customizing views, Tableau Server even allows information consumers to create completely new visualizations from within Tableau Server.

FINDING INFORMATION

Tableau's security structure provides an initial level of categorization, but Tableau Server also allows information consumers to customize access further through tagging, marking favorites, and even altering existing workbooks without the need for a Desktop license.

Tagging

You learned about tagging from the publishing perspective earlier in this chapter. However, users having only interactor access can also tag projects, workbooks, views, and datasources. While interactor users can add and remove their own tags, administrators and publishers have visibility of all tags applied to workbooks. Tags are applied from the content menus for projects, workbooks, views, or datasources. To apply the tag, select the Tag menu option at the top, and either select an existing tag or enter a new tag. Figure 10–14 shows a tag being applied from within the content thumbnail view for workbooks.

The thumbnail workbook views checked in Figure 10–14 will have the Version8 tag applied to them. This tag will provide a means for filtering based on that tag. If no existing tag is appropriate, use the Add option at the bottom of the tag menu and create a new tag. Note, the new tag won't appear in the tag's menu until the view is refreshed in your browser.

Removing Tags

If you want to delete a tag associated with a project, workbook, view, or datasource, navigate to the server view containing the item from which you wish to delete the tag. Scroll to the bottom of the page below the comments area. You will see the tags associated with the item there as displayed in Figure 10–15.

The version tag will be deleted in Figure 10–15 by clicking on the X in the tag area. Note that administrators and publishers may remove any tag, but interactor users can only remove tags they have created.

Workbooks

- Admin
- Content**
- Delete
- Permissions
- Move
- Tag**
- Tabs
- Scheduled Tasks
- Download

Sort By: Name ↑ Project Sheet

Book Writing Plan Version 8

Testing

beta
blending
book
dan
extract
global
maps
play
sales_analysis
Schedule
stuff
test
testing
Testing
va
va-test
version8 
Add...

VA Administration - Veteran Population Map 2013

VA Administration - Veteran Population Map 2013

Veteran's Benefits Claims Processing - By State

World GDP Through Time

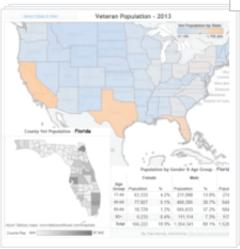
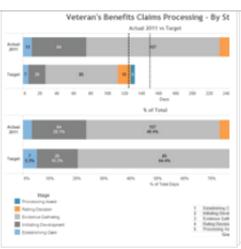
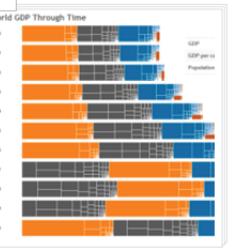





FIGURE 10-14 Tagging workbooks

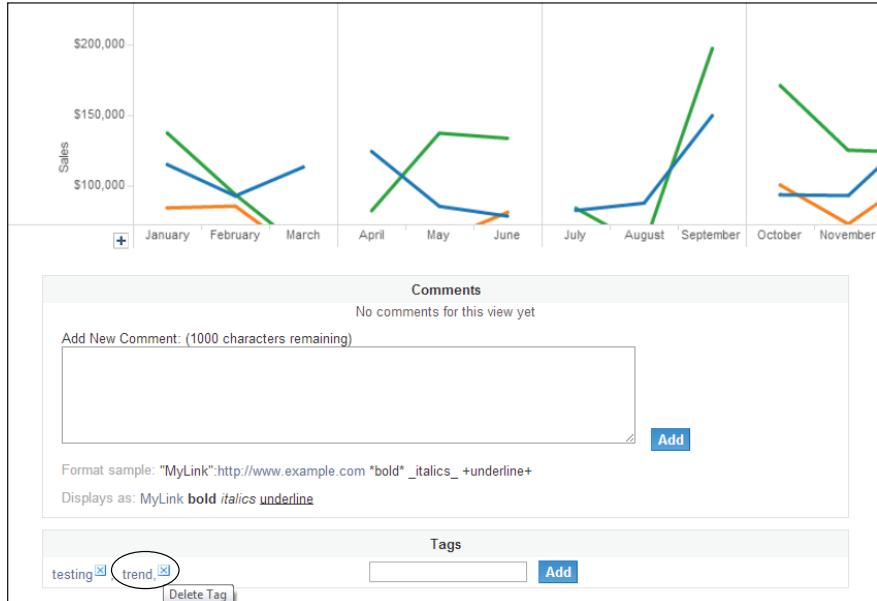


FIGURE 10-15 Deleting a tag

(c) OOBTech 2018

Favorites

Next to every workbook or view listing in Tableau Server is a star icon that allows your users to create a personal favorites list. If the icon is colored yellow, that item is a favorite. Greyed items are not. Clicking on the star will add it to the favorites list. Clicking a second time toggles it off the list. The Favorites list is a bookmarking mechanism that provides fast access to your most frequently used items. Refer to Figure 10–5, presented earlier in the “Creating a Favorite” section of this chapter to review the details regarding how favorites are added.

SHARING COMMENTS AND VIEWS

Comments can be applied in any server view if the user has the proper permission and assuming the comment save option has not been disabled through an embedded view. Comments are found at the bottom of the view in Server, giving users the opportunity to share ideas and ask questions.

Views can be shared with anyone via the Share Link found at the top of the page. If the view is embedded, you will find the Share Link at the bottom of the page. Figure 10–16 shows the dialog box that appears when the share menu option is selected.

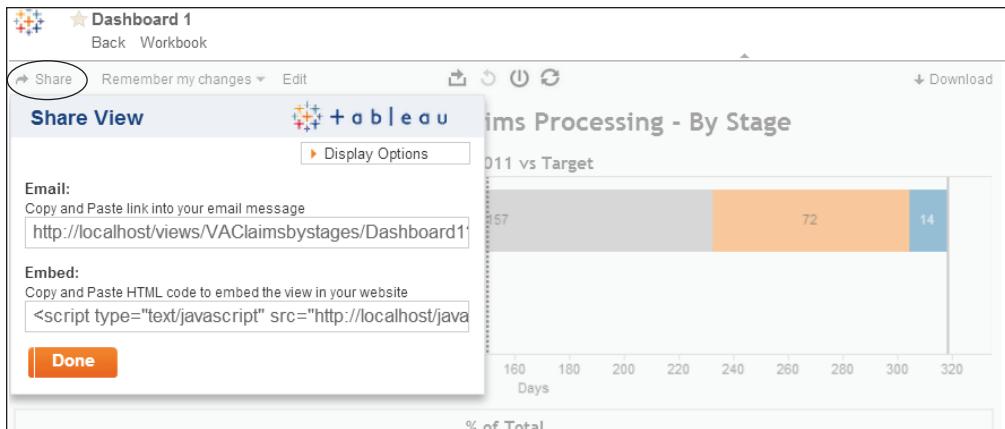


FIGURE 10–16 The share view dialog box

The Share Links option allows you to send a link in an e-mail or to embed the view within a website. Both of the options allow you to optionally set the pixel height and width of the view, and to define whether or not the toolbar or tabs are displayed.

CUSTOMIZED VIEWS

Users can make mark selections, apply filters or highlights, and then save those settings in a customized view. Figure 10–17 shows a visualization in which the color legend for the Americas has been saved as a custom view called Americas Highlighted View.

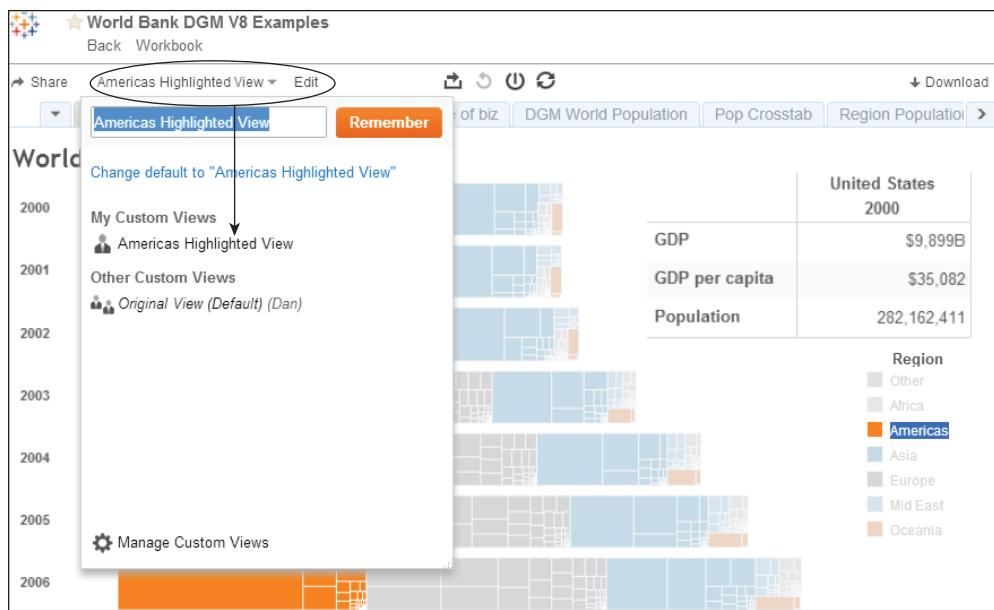


FIGURE 10–17 Saving a custom view

Saving any customized view requires three steps:

1. Click on the Remember my changes link.
2. Provide a name for the customized view.
3. Click the Remember button to save the view.

Your customized view is saved, and you are redirected to a unique URL that is generated for the view. The Remember My Changes link is also changed to the name of your customized view. If you click the link, you'll see a list of all the customized views you have saved along with a link to the original view published. To rename or delete any of your customized views click the Manage Custom Views link at the bottom of the listing.

Other users accessing the view on their own will still be presented with the original view as originally published, unless they use the unique URL for your

customized view. Customized views provide a good way for users to save frequently used filter combinations without the need to rely on the publisher.

A significant new feature that arrived with the release of Tableau Version 8 is web authoring. This goes beyond saving customized views by allowing users that lack a Tableau Desktop license to alter and create totally new visualizations within Tableau Server.

AUTHORING AND EDITING REPORTS VIA SERVER

Tableau server's in-browser editing functionality provides a simplified version of the Desktop experience. It allows users to edit existing workbooks, create new visualizations, and save that work back to Server. This feature now provides for an under-supported component of the Enterprise community—the middle tier of users who do not require the complexity and power of the Desktop tool, but want the ability to probe the data in ways that were not anticipated by the report publisher. Web-Tablet authoring provides the ability to self-serve information from any device capable of accessing the Tableau Server and creating a web-session, without installing any software.

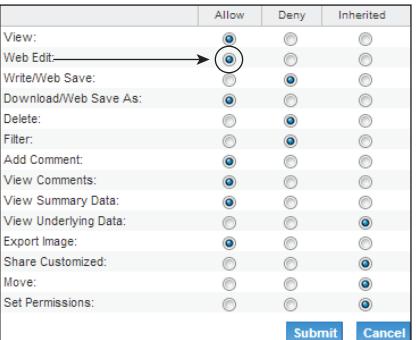
WHAT IS REQUIRED TO AUTHOR REPORTS ON THE WEB?

In Chapter 8 you learned about web and mobile access to Tableau Server reports. You observed that a personal computer's web-based interaction is very similar to the tablet-based interaction. This is also true when considering the Web-Tablet authoring functionality. This functionality, like all web-based interfaces to Tableau, is exclusively a function of the Tableau Server environment and is not possible utilizing only the Desktop or Reader products. To author on the web you need:

- A live web connection
- A Tableau Server Interactor license
- A standard web browser with a live web connection
- A pre-existing workbook must be published to Tableau Server.
- Appropriate permissions to be able to Web Edit

If you want to author via a tablet, you must download Tableau's iPad application from Apple, or the Android application from Google Play if you plan to use an Android tablet. While these elements are essential, they are not sufficient. The server permission for Web Edit must be allowed as you see in Figure 10-18.

As with all permissions on Tableau Server, web editing can be configured at multiple levels—user, group, workbook, project, or Site. Tableau Server users can also access saved datasources via the content tab and create a new workbook using that datasource. One Desktop license is required to publish the original report template to the server but any number of Tableau Server Interactor licensees can edit the report or create new reports from published datasources. This is the first iteration of server-based authoring. Tableau does not impose additional licensing fees to access the Web-Tablet authoring tool.



A screenshot of a Tableau permission configuration dialog. The dialog has three columns: 'Allow' (radio button), 'Deny' (radio button), and 'Inherited' (radio button). A horizontal arrow points from the text 'Web Edit' in the left column to the 'Allow' radio button in the middle column. The 'Web Edit' row is highlighted with a blue selection bar. Other rows include: View, Write/Web Save, Download/Web Save As, Delete, Filter, Add Comment, View Comments, View Summary Data, View Underlying Data, Export Image, Share Customized, Move, and Set Permissions. At the bottom right are 'Submit' and 'Cancel' buttons.

	Allow	Deny	Inherited
View:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Web Edit:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Write/Web Save:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Download/Web Save As:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delete:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Filter:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Add Comment:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
View Comments:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
View Summary Data:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
View Underlying Data:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Export Image:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Share Customized:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Move:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Set Permissions:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

FIGURE 10-18 Permitting web editing

SERVER DESIGN AND USAGE CONSIDERATIONS RELATED TO WEB AND TABLET AUTHORING

Tableau's web-tablet authoring system is largely a client-side functionality provided through an HTML5 layer. This means that the Web-Tablet authoring system will have limited impact on the majority of Tableau Server processes.

The Tableau Server administrator should be aware that users editing views via this method will generate activity on the server's VizQL process. And, if the workbook being edited is based on a data server extract-driven datasource, those processes will also experience increased loads. This impact is identical to the effective impact of adding additional Tableau Desktop interactions—presuming a server-mediated data connection.

Should Web-Tablet authoring result in a high number of workbook saves or creates via the server Save As dialog, Tableau Server will experience additional demand placed on its repository and storage systems in a manner similar to the load that would be expected via Tableau Desktop utilization.

These additional loads are a good thing. They mean that your user-base is engaged and actively using the system.

DIFFERENCES BETWEEN DESKTOP AND WEB OR TABLET AUTHORING

The experienced Tableau Desktop user will immediately notice that the web-tablet editing interface closely mirrors the familiar desktop environment. Editing through the web or tablet is very similar to the Desktop tool, though it is simplified and limited in a few ways. This section will detail the functional differences between the two authoring experiences. While limitations in the web-tablet authoring environment are highlighted, you should not interpret this section as a negative critique. Web authoring is a significant innovation that will provide benefits to the majority of your user base.

The goal is to highlight the differences so that you are aware of what can be done via the web versus what must be done using the Desktop application. The Web-Tablet authoring environment is designed to provide a simplified version of the Desktop experience. It is not intended to replace the Desktop application. Tableau Version 8 is also the first iteration of this functionality, and it will probably evolve and improve in future releases.

Drop Areas for Rows and Columns but No Show Me! Drop Area

Many of the standard Desktop Options and layouts are available within the Web-Tablet authoring interface. The left-hand side re-creates the data window, including any datasource(s), measures, and dimensions. The Column and Row shelves, along with the pages, filters, and marks cards also exist in their standard positions. Users can create visualizations in the same drag/drop manner that is fundamental to Tableau's Desktop authoring experience. One difference is that all fields must be dragged to the shelves and that the in-view drop areas for rows and columns do not exist, nor does the default Show Me! drop-area function in the view's center. See the web-authoring interface in Figure 10-19.

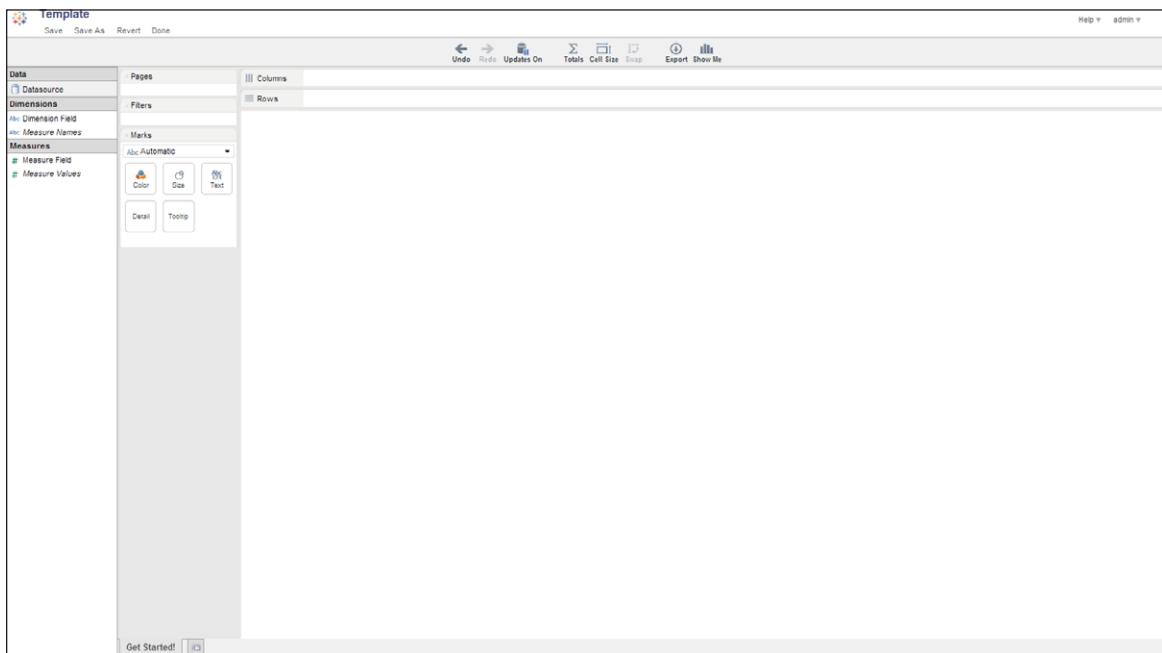


FIGURE 10-19 The Web-tablet authoring view

Notice that some of the tools found in the desktop product icons or main menu appear in the web environment at the top of the authoring space.

No Dashboard Support

There is no dashboard display in the server authoring environment. In fact, Dashboard's editing is not interpreted in the web-tablet editing experience. Any workbook that contains Dashboards will display those views broken into their component parts (even if hidden) versus being displayed as the combined entity you see in the desktop application.

Datasource Manipulation Is Not Supported

All datasources needed for analysis must be included at the time of publishing from Tableau Desktop. The web authoring system does not allow for any manipulation of the metadata layer. You can't add new datasources, remove unused datasources, create calculated fields or parameters, change default field properties, or edit relationships between datasources. In general, the web authoring environment doesn't support metadata management. These capabilities exist exclusively in the desktop tool.

No Right-Mouse Button Click Functionality

While Tablet users won't be surprised at the lack of a secondary click option, this may surprise PC-based web authors. Any functionality accessed through right-clicking in Tableau's Desktop tool is not supported in the web-tablet authoring system. Some of these functions facilitated by right-button clicks in the Desktop are enabled through simple menu-based controls in the web-table environment. Dimension-specific controls are shown in Figure 10–20.

The dimension specific controls in Figure 10–20 were exposed by selecting the small drop-down arrow in the market dimension pill. Similar measure-specific controls are shown in Figure 10–21.

Quick table calculations are referenced using the drop-down arrow in the sales pill on the column shelf.

Quick Filters Only: No Complex Filtering

Desktop users accustomed to creating "complex" filters like "Top 10," or utilizing specific conditions, will notice that these filters will persist in the web authoring session. However, web editors can't add new versions of these complex filters. Web editors are able to add Quick Filters to views, and the full suite of Quick Filter types are available.

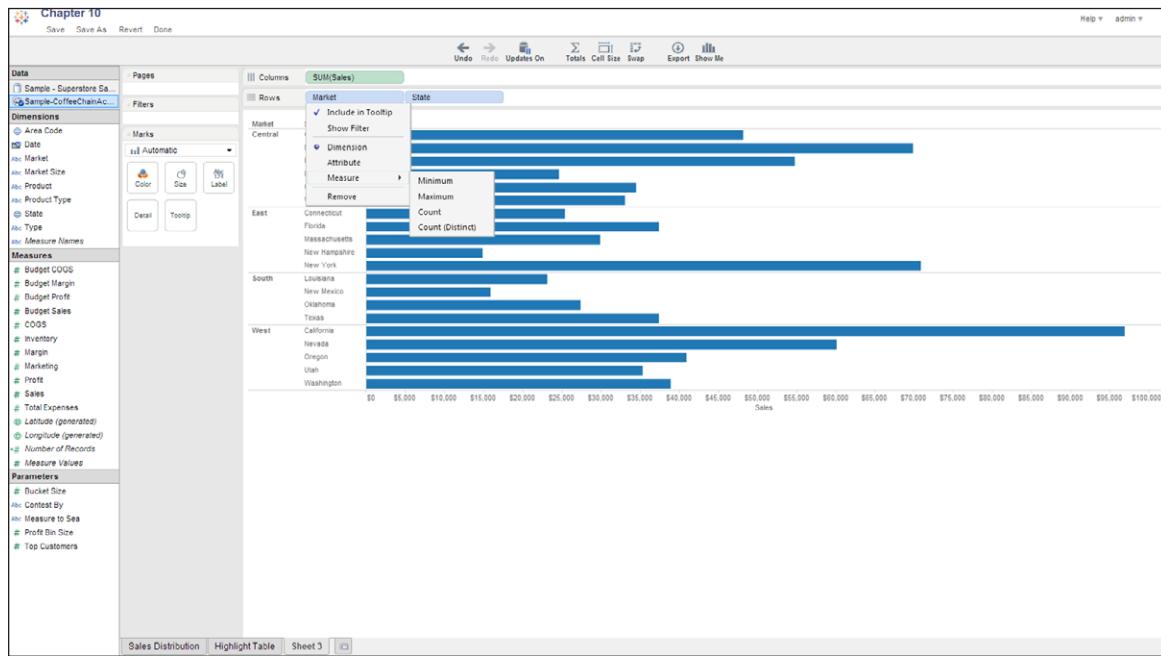


FIGURE 10–20 Dimension-specific controls

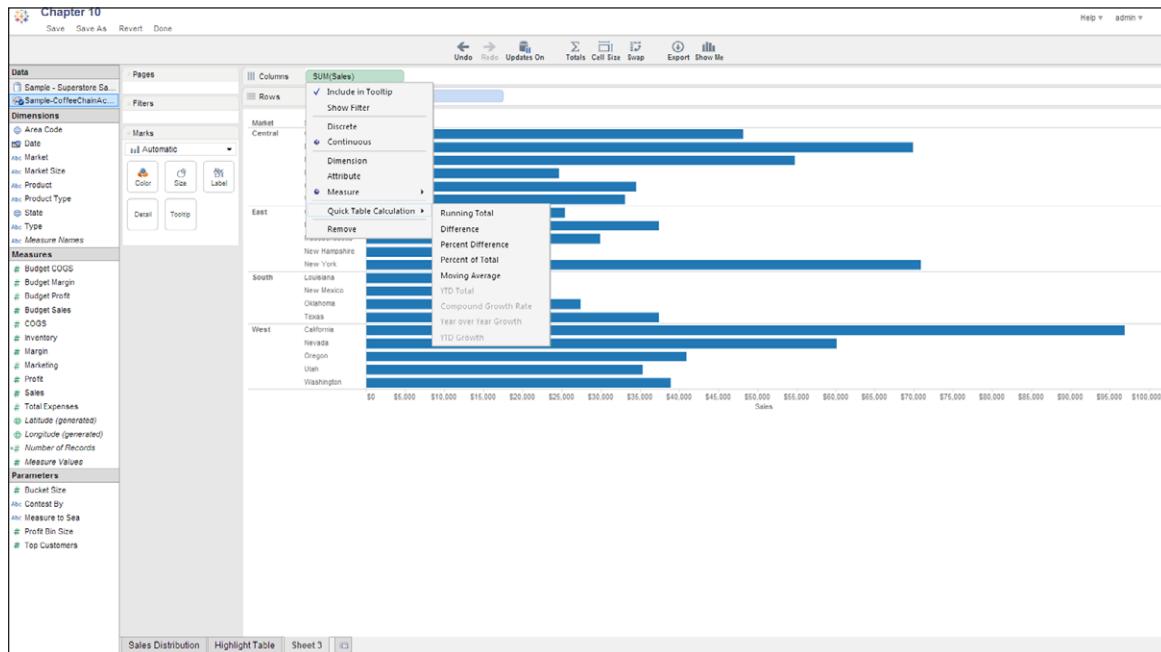


FIGURE 10–21 Measure-specific controls

Cell Sizing Is Exclusively Menu-Based

Cell sizing is controlled exclusively through the cell size menus; users can't drag elements of the visualization to resize those items—nor can they drag to resize the sheet as a whole. And, web editors cannot drag to control the "Fit" of the view within the design space. Figure 10–22 displays the web tools Fit and Sizing controls.

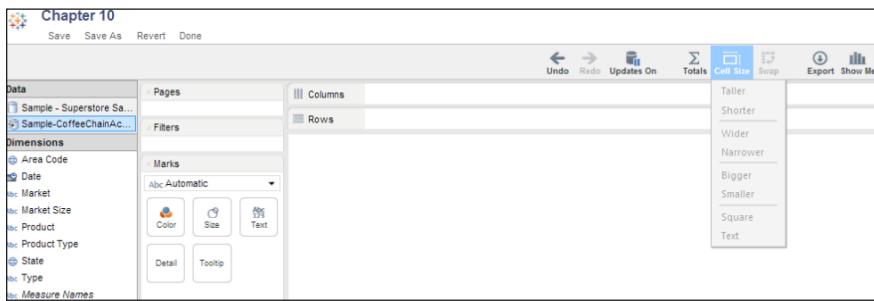


FIGURE 10–22 Web fit and cell-sizing controls

Even though Desktop dragging to resize elements isn't available in the web environment, the cell-size menu provides this facility.

Sheets Cannot Be Renamed

While creating new worksheets is supported, web-authors cannot enter customized names for the worksheets. New sheets created are numbered sequentially.

Sorting Is Only Available through Quick Sorts

Unlike Tableau Desktop, authors cannot set sorts based on specific fields, default sorts, or pre-sort information in a robust manner. Sorting is exclusively allowed through the in-visualization Quick Sorts that are omnipresent on headers in all Tableau visualizations.

Limited Control of Color, Size, Text, and Tooltips

Tableau Desktop allows nearly infinite control of color palettes, size ranges, shapes, and Tooltip content. The web-tablet environment provides none of these fine-grain controls.

Multi-Select/Ctrl-Key Functions Are Not Available

Tableau's Show Me! facility is available and works very well. However, Desktop users fond of using the Ctrl+Select technique to multi-select fields and then apply Show Me! to create visualizations, will notice that this isn't possible

through the Web-Tablet system. All of Tableau's standard visualizations are available in the web authoring system as you see in Figure 10–23.

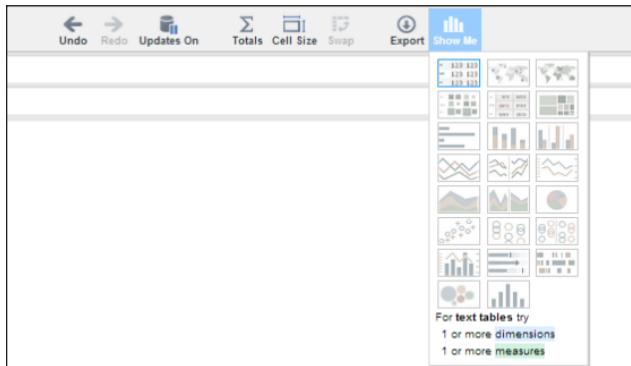


FIGURE 10–23 Web authoring with Show Me!

As fields are placed on shelves in Figure 10–23, relevant chart types will be highlighted just as they are in the Desktop. Even though this doesn't quite match the desktop authoring experience, the web-authoring environment provides robust visualization capabilities.

SAVING AND EXPORTING VIA THE WEB-TABLET ENVIRONMENT

The web-tablet environment provides a number of options for sharing work and insights.

Export

Similar to the Desktop tool, web-tablet authors have a full suite of export functions in addition to the standard server-based export functions. Figure 10–24 displays the available web options.

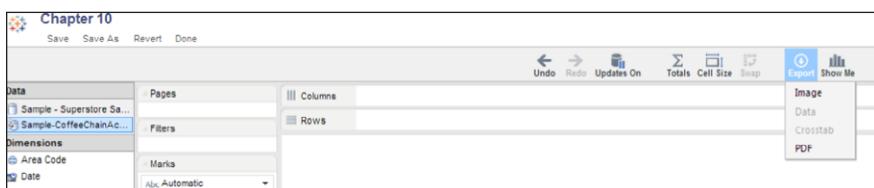


FIGURE 10–24 Web-tablet export functions

Exporting images, data, cross tabs, and pdf documents are all supported.

Save and Save As

Recall in Figure 10–3 the web/edit permission must be allowed to enter the web edit system. Similarly, the ability to overwrite the existing workbook is also permission based. Should a user have sufficient permission to save their work from the Web-Tablet editing system, they will be given the option to save the workbook, which will overwrite the original Desktop version. Note that Tableau Server does not save a copy of the original document by default, so saving in the Web-Tablet system is equivalent to a republishing through the Desktop tool. The Save As dialog does provide the user with the ability to republish the altered workbook under a new name, or into another project.

RECOMMENDATIONS FOR IMPLEMENTING WEB-TABLET AUTHORING

Web-Tablet based design has not been designed to entirely replace the Desktop tool. Enterprises should view Web-Tablet authoring as a supplementary tool that enables a previously under-served cohort group access to Tableau's ad-hoc analysis and reporting capabilities.

Paired with relevant datasource access and training, the Web-Tablet authoring tool facilitates self-service business analysis in a controlled environment—providing users with the ability to ask questions that were not anticipated within the original design of the report. Key points to remember when designing reports that will be open to Web-Tablet authoring:

- Give component worksheets logical name(s) that will not be obscured by the standard Desktop practice of hiding sheets that have been added to Dashboards.
- Design template workbooks and template datasources that can be readily approached by non-technical users that may not have data analysis expertise or experience with the Tableau Desktop tool.
- Provide transparent information about datasources—including refresh rates, sources, assumptions, and contact information for the original publisher.
- Do not presume users will understand how to use the Web-Tablet authoring environment, provide training, and help motivated individuals use the system effectively.
- Create a specific Sandbox project/area where new users can save work and gain confidence.

SHARING CONNECTIONS, DATA MODELS, AND DATA EXTRACTS

Tableau's data server can lower the data access barrier while still providing data governance controls. Database administrators can define data connections once, publish them to the data server, and manage access by applying Tableau's object level. Consequently, data analysts using Tableau require no knowledge of the underlying tables, joins, or related criteria driving the connections.

OFFERING A COMMON DATA LIBRARY

It's common for organizations to manage and use a variety of datasources. Transactional data may live in one database while historical data is maintained in another completely different database. Business users may maintain their own spreadsheets of forecasts and budgets. While Tableau can easily connect to each of these disparate datasources, Data Server offers the capability to host these connections in a central place. This reduces the potential for data misuse because Tableau users can simply connect to Tableau Server and the data they need access to, regardless of source system. Data Server simply acts as a proxy to databases while also serving as a host for items such as Excel and Access files, or even data extracts.

SHARING DATA MODELS

Earlier in the Chapter in the section related to the Tableau Data Server you learned how datasources can be published and used by many different users. The related metadata that comes with the shared datasources allows data administrators to manage inconsistencies through:

- Consistent field name aliasing
- Consistent field grouping
- Consistent application of field hierarchies

This permits the organization to tap into the best subject matter experts to create and validate calculations and publish the resulting data models for your entire organization to benefit from. These capabilities can reduce variations in how business rules are interpreted and applied, while giving analysts the ability to do their own ad-hoc analysis by adding their own customizations on a per-workbook basis.

INHERITANCE OF UPDATES

Once a datasource has been published to Data Server, workbooks using the connection automatically inherit any future updates to the datasource. This greatly simplifies the process of datasource maintenance while reducing the risk of outdated business rules persisting in production when an underlying change has been made.

EMBEDDING TABLEAU REPORTS SECURELY ON THE WEB

If your organization is accustomed to consuming information via a specific web portal, Tableau provides a variety of ways to embed your reports as interactive dashboards or static images—all the while persisting the same licensing and security framework available to you on Server.

WHEN TO EMBED A DASHBOARD

When does it make sense to embed a dashboard rather than simply having your users access it directly from Tableau Server? If your user base is already familiar with a particular web portal, it makes sense to use that website as a repository for interactive Tableau visualizations and Dashboards. In addition, there may be advantages to leveraging the existing security options that the web portal may already have defined.

Going to Your Users

It can be frustrating for some people when they are given yet another website and login to access for information that they need to effectively perform their job and make critical decisions. Many people don't enjoy the accelerating change fostered by advances in technology. At work, people are constantly presented with new software and initiatives designed to make them more efficient, but that (at least during the transition) make them less efficient.

By embedding Tableau into an existing web portal that is familiar to users, you can provide them with the benefits that Tableau has to offer while keeping them in a familiar environment that they are already used to accessing for their reporting needs.

When Your Reports Are a Piece of a Larger SaaS Offering

Anytime you are offering a service to your clients you'll want to control the overall branding of the product you create. When including Dashboards the process is no different. Rather than providing what appears to be a third party

solution by redirecting your users from your product to Tableau Server, you can embed the Tableau reports directly into your product instead. The result can often be a seamless integration where your users are unaware—and need not know—the underlying technology driving the reports you provide them with. Instead, the details blend into the background while you offer a single cohesive product to your users.

Providing a More Robust Environment

We've already discussed situations where your users are already concentrated in another environment, and how it can be advantageous to embed Tableau reports in that environment. Now, let's explore the opposite situation. What happens when, over the years, multiple reporting environments have been created as the result of various initiatives spread over multiple departments? One part of the company may rely on Business Object reports created a decade earlier. Other teams depend on SSRS reports to make their day-to-day decisions. Now, you look to open a new world of analytical discovery by implementing Tableau.

So, do you go about re-creating the existing Business Objects and SSRS reports in Tableau and redirect all users to Tableau Server? Does this always make sense when specific reports work perfectly well and already directly impact a critical business need? Why not create a single environment that combines these disparate systems into a single environment that you can further enhance through documentation and an interactive user community?

What is worse than bad business intelligence? Good business intelligence that nobody can find. By creating a single seamless environment for your users, they no longer have to track down and find the reports that may be out there. You mitigate the risk of users reinventing the wheel by re-creating existing reports. Most important of all, you remain agile because you can blend in future technologies while servicing your users in the same location you do now.

HOW TO EMBED A DASHBOARD

Embedding Dashboards usually boils down to one of these methods:

- Using Tableau's JavaScript code
- Using the Dashboard's URL in an iFrame or Image tag
- Writing your own code using Tableau's JavaScript API

No matter which method you choose to use, you can control your embedded view through the use of Passed Parameters. We'll explore all three methods below while also diving into the details of all the parameters you can use.

Note that all the embedded solutions in this section require the user to log in via the embedded view as they normally would when accessing Tableau Server directly. For information relating to providing a single sign on experience for your users, see [Tips and Tricks for Embedding Dashboards](#).

Using Tableau's JavaScript Code

The easiest way to embed a Dashboard in another web page is to use the JavaScript code provided by Tableau in its Share button. With the options for setting the width and height of your Dashboard and the ability to turn on/off the toolbar and tabs, this provides a mechanism for quickly embedding your Dashboards into another web page. Below is a quick example of the resulting code:

```
<script type="text/javascript" src="https://yourtableauserver.com/javascripts/api/viz_v1.js"></script><div class="tableauPlaceholder" style="width:979px; height:662px;"><object class="tableauViz" width="979" height="662" style="display:none;"><param name="host_url" value="https%3A%2F%2Fyourtableauserver.com%2F" /><param name="site_root" value="#47;t#47;YourSite" /><param name="name" value="YourWorkbook#47;YourView" /><param name="tabs" value="yes" /><param name="toolbar" value="yes" /></object></div>
```

Notice the use of `<param>` tags to pass specific values to Tableau Server. Through the use of these tags, it is possible to pass additional parameters such as an initial filter. For example, the following entry will initially filter the embedded view by restricting the Region dimension to West only:

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

The name and site root parameters are the only ones required when embedding a view.

Using an iFrame or Image Tag

Another option is to use the URL for a dashboard or a view in an iFrame or image tag. Additional parameters can still be passed but must be included at the end of the URL. The embed parameter is required, but all others are optional. An example of an embedded view using an iFrame is displayed below this paragraph. The Dashboard is once again filtered to the West Region only, while also restricting to the date June 1st, 2012.

```
<iframe src="https://yourtableauserver.com/t/views/MyWorkbook/MyDashboard?:embed=yes&Region=West&Date=2012-06-01" width="800" height="600"></iframe>
```

Note the required embed parameter is set first. A value of Yes hides Tableau's default navigation options and comments section below the view, while also moving the toolbar and share options below the view.

Writing Your Own JavaScript API

You can also write your own code by leveraging Tableau's JavaScript API. This is often the preferred choice by web developers looking to embed Tableau views into their existing web applications as it offers a deeper level of control.

Tableau provides developers with the ability to interact with embedded views in real time. By listening for events generated by Tableau views, developers can capture actions performed by a user and respond to them in rich, interactive ways. For instance, developers can respond to a user selecting marks on an embedded view and trigger a response in their web application. Developers can also interactively set filters and select marks within an embedded view in real time—no longer limited to simply setting initial values prior to a view loading. The best part is that each of these API functions are enacted as they would be if the user had performed the action in the view itself, meaning no page refreshes occur. The result is a completely seamless experience between your application and the embedded Tableau reports. Check this book's companion website for an example of how you might use the JavaScript API to create interaction between a Tableau dashboard and a website.

Further Control Using Passed Parameters

Whether you choose to use an iFrame or JavaScript, you can pass additional parameters to the view. Search Tableau's website to find a complete list of supported parameters.

TIPS AND TRICKS FOR EMBEDDING DASHBOARDS

Tableau has streamlined the embedding process to a great extent. You may get some additional benefit from the following tips and tricks.

Filter Formats for Dimensions, Measures, and Dates/Times

When passing dimension filter parameters, simply list each value in a comma-separated list. To filter on multiple dimensions, separate each with an ampersand. The general form is:

```
Field=Value1,Value2,Value3&Field2=Value1,Value2
```

You can filter on measures in the same manner by passing explicit values. However, Tableau Server does not support filtering by a range of values or using greater than or less than logic.

To filter on a Date or Date/Time field, use the following form:

DateField=yyyy-mm-dd hh:mm:ss

When filtering a Date/Time field, the time component is considered optional.

Know Your Character Limits

Theoretically there is no limit to the number of parameter values you may pass to an embedded view. However, you may ultimately run into a URL length restriction imposed by the end user's browser.

While HTTP protocol does not impose a cap on URL length and many modern browsers can handle URLs with up to 80,000 characters, Internet Explorer 8 and 9 have a maximum character limit of only 2,083 characters.

As a result, you should strive to keep your URLs under this limit to ensure compatibility. Keep in mind that the complete URL—not just the parameters and corresponding values—are included in this length.

Use Trusted Ticket Authentication as an Alternative Single Sign-On Method

When an embedded view is accessed, the same authentication mode enabled on Tableau Server is used to verify the user's identity. For instance, if your server is configured to use local authentication, your users will be required to log in via a form provided by the embedded view. This can be cumbersome if the user has already authenticated in the web application. To work around this, Tableau provides a couple of options for single sign-on authentication—a process where your user is authenticated by your web application and is not required to further authenticate themselves by any embedded Tableau views.

If your server is configured to use Active Directory and SSPI, you can enable SSPI on your web server as a single sign-on solution given the user is in your Active Directory and is a licensed Tableau Server user. In all other situations, you will need to use Trusted Ticket Authentication as an alternative single sign-on method.

When using Trusted Ticket Authentication, the web server assumes all responsibility for authenticating users. Before embedding the view, the web server passes two POST parameters to Tableau Server:

- Username (must match a licensed Tableau server user)
- Client_ip

The web server will receive a response in the form of a unique_id, which is used in the embedded view's URL, as shown in this form:

```
https://yourtableauserver.com/  
trusted/unique_id/t/views/MyWorkbook/MyDashboard?:embed=yes
```

If you are using JavaScript, the ticket parameter can be used:

```
<param name="ticket" value="unique_id"/>
```

Once a unique_id has been issued, it must be redeemed within 15 seconds from a machine matching the client _ip specified or it is considered no longer valid. When Tableau Server receives the request, the user is logged in as they would be if using forms authentication, and the trusted ticket URL is resolved to that of a standard request.

Before a web server can make a Trusted Ticket Authentication request, it must first be "White Listed" on Tableau Server. This can be accomplished using the following TabAdmin command, where xxx.xxxx.xxxx.xxxx represents IP addresses for any trusted web servers:

```
tabadmin set wgserver.trusted_hosts "xxx.xxxx.xxxx.xxxx, xxx.xxxx.xxxx.xxxx"
```

USING SUBSCRIPTIONS TO DELIVER REPORTS VIA E-MAIL

Busy managers sometimes like to have their reports delivered. Tableau supports this through subscriptions. By subscribing to a workbook or a view, users can have an image of the report delivered directly to their e-mail. The user will be notified via e-mail when the report is updated. These schedules are defined by a server administrator, allowing administrators to ensure that any additional server load is balanced appropriately. Subscription e-mails also contain links to their live and interactive counterparts located on Server.

To subscribe to a report, a user follows these steps:

1. Log in to Tableau Server.
2. Open the view or dashboard to which you wish to subscribe, or subscribe to an entire workbook by opening a single view or dashboard belonging to that workbook.
3. Click the Subscribe icon in the top right corner of the dashboard. This is represented by the small e-mail icon on the left side of the download link.
4. This causes the subscribe dialog box to appear.

Select or enter the following values:

1. To—Verify the e-mail address associated with your account.
2. Subject—Enter the desired subject for the subscription e-mail.
3. Schedule—Select the schedule you'd like your snapshots sent on. These schedules are created and maintained by Tableau Server administrators.
4. Content—Choose whether to subscribe to the current sheet only or all sheets in the Workbook.
5. Click on subscribe to finish.

ADMINISTRATORS SUBSCRIPTION MANAGEMENT

Allowing subscriptions on Tableau Server requires a few additional steps to be performed by an administrator. You can change these in the Tableau Server configurations to Enable Subscriptions using the following steps:

1. Log in to the Tableau Server machine, stop the server, and open the Tableau Configuration utility.
2. Click the E-mail Alerts/Subscriptions tab.
3. Check Enable E-mail Subscriptions.
4. In the SMTP Server section, ensure that a SMTP Server and Port is specified. Enter a Username and Password if required by your SMTP Server.
5. Enter a Send E-mail From an Address. This e-mail will send out all subscription e-mails.
6. Enter your Tableau Server URL.
7. Click OK to finish.
8. Start the server.

CREATING SUBSCRIPTION SCHEDULES

Administrators have to enable the subscription notification system in Tableau Server in order for the feature to be available to users. To create subscription notification schedules, the administrator should follow these steps:

1. Log in to Tableau Server using a System Admin account.
2. From the Admin tab click Schedules > New.
3. The Create New Schedule page appears.

Define the desired schedule ensuring Schedule Scope is set to Subscriptions and click on the Create Schedule button to finish the definition.

Your users will now have the option to subscribe to workbooks and views using the schedule(s) you've defined. Like all scheduled tasks you have the option to initiate them manually by using the Run Now option on the Schedules page.

In the last two chapters you've learned how to install Tableau Server and were introduced to the features available to Tableau Server users.

The next chapter is geared toward administrators charged with the responsibility of keeping information up to date and maintaining the environment. You will learn how Tableau's command line tools (tabadmin and tabcmd) can help you automate repetitive administrative tasks.

NOTES

1. Collins, James C. *Good to Great: Why Some Companies Make the Leap—and Others Don't*. New York, NY: HarperBusiness, 2001. Print. Page 88.

Automating Server with Tableau's Command Line Tools

As your Tableau Server deployment expands, the number of users and amount of data you have to manage will grow. Tableau provides two command line tools that will help you automate routine tasks. Most of the functions the command line tools provide are available directly within Tableau Server's user interface.

Using Windows Notepad (or your favorite text editor) you can automate `tabcmd` to run via a batch file. Then by using Windows Task Scheduler you can trigger the batch file to run at a specific time or based on a specific triggering event. Of course many popular scripting or programming tools can call Tableau's command line functions to automate tasks. How you use these tools is only limited by your desire and creativity.

If you are a system administrator and accustomed to writing script and using the Windows Command Processor and Windows Task Scheduler, you will not have difficulty incorporating `tabcmd` or `tabadmin` into your existing toolset. Many people don't use the command line utilities because their full functionality is not clearly understood, or they have not seen specific use case examples. Tableau Software provides some good introductory videos on their website. You can find those by searching for On Demand Training and looking in the server section for the `tabcmd` and `tabadmin` videos.

WHAT DO TABCMD AND TABADMIN DO?

Tableau's two command line tools are `tabcmd` and `tabadmin`. `Tabcmd` provides functions for performing workflow tasks like publishing workbooks, adding users, or exporting workbooks as image or data files. `tabadmin` is designed for server administration—configuring server options, activating users, resetting

passwords, and other tasks associated with managing the deployment and usage of Server within the enterprise.

A person with publishing rights might want to use `tabcmd` to automate repetitive tasks associated with updating and publishing datasources. A server administrator can leverage `tabadmin` to set up a new site, grant or revoke user rights, backup data, alter default session time-out settings (get input from Tableau Support or a qualified Tableau Partner before changing these settings), or reset user passwords. Think of `tabcmd` as a tool for helping those who publish and share. `tabadmin` is an automation tool for staff with administration responsibilities—helping them control access, tweak settings, or observe system status.

INSTALLING THE COMMAND LINE TOOLS

When Tableau Server is installed `tabcmd` and `tabadmin` are automatically installed in Tableau Server's bin folder. Depending on the operating system being used (Window 32-bit or 64-bit) the program will be installed in one of these locations:

- 32-bit—C:\Program Files\Tableau\Tableau Server\8.0\bin
- 64-bit—C:\Program Files (x86)\Tableau\Tableau Server\8.0\bin

If you are using an older version of Tableau Server the portion of the address that says 8.0 would be replaced with the specific version number that you are using. If you are running a distributed environment—with multiple worker machines—and you want to utilize `tabcmd` on one or more of the worker boxes, you must install `tabcmd` on those other machines. Tableau provides an installer program for doing that. Those programs are:

- 32-bit—C:\Program Files\Tableau\Tableau Server\8.0\extras\Tabcmdinstaller.exe
- 64-bit—C:\Program Files\Tableau\Tableau Server\8.0\extras\Tabcmdinstaller.exe

Copy the `Tabcmdinstaller.exe` program to the computer that you want to install it on and double-click on the file to run the program. The program provides prompts as it installs. Tableau Software recommends installing the `tabcmd` program on the root drive (C:\`tabcmd`).

Since the setup program doesn't automatically add the bin folder containing `tabcmd` or `tabadmin` to the Windows PATH system variable, you have to manually navigate to the bin folder subdirectory to use the programs. This can be avoided if you modify your computer's PATH system variable to include the

path to the bin folder. Doing this allows you to run the executable commands without needing to manually enter the directory location of the bin folder. To start using tabcmd, open the Windows Command Prompt. Figure 11-1 shows you how to do that in a Windows 7 environment.

If you are using a different version of Windows, find the accessories folder by searching your computer's hard disk. Once you've entered the accessories folder, click on Command Prompt to open the command prompt window. In order for you to have access to the tabcmd program files you must first navigate to one of the bin folders listed in the first section. If you are using a 64-bit version of Windows, type in the following command and press enter:

```
cd "C:\Program Files(x86)\Tableau\Tableau Server\8.0\bin"
```

This will change the active directory to the bin folder that holds the tabcmd program. Assuming that your Tableau Server address is `http://mytableauserver.com`, and that your Tableau Server uses port 80, start a tabcmd session by typing the following into the command prompt window:

```
tabcmd login -s http://mytableauserver.com -u USER -p PASSWORD
```

The end of the string immediately following the ".com" is case sensitive.

After entering the `tabcmd login` command, and the `-s` site URL, substitute the URL location of your Tableau Server installation. Then enter your username and password after the `-u` and `-p` global option variables.

The instance of Tableau server used in this example is a local installation on a laptop. The username is Admin and the password is Admin. The command line entry to log into this server can be seen in Figure 11-2.

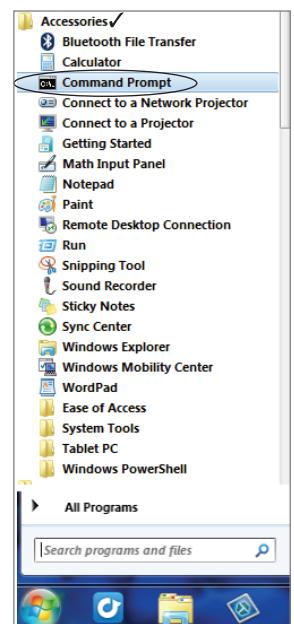


FIGURE 11-1 Opening Window Command Prompt

A screenshot of a Windows Command Processor window titled 'Administrator: Windows Command Processor'. The window shows the command `tabcmd login -s http://localhost/:8000 -u admin -p admin` being run. The output shows the process of creating a new session, connecting to the server, and logging in successfully. The text is as follows:

```
C:\Administrator: Windows Command Processor
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>tabcmd login -s http://localhost/:8000 -u admin -p admin
==== Creating new session
====   Server: http://localhost/:8000/
====   Username: admin
==== Connecting to server...
==== Logging in...
==== Login Succeeded.
```

FIGURE 11-2 Tabcmd login example

Notice that the portion of the script that includes the server address also includes an additional element (`:8000`). This defines the TCP/IP port for the local server instance and is required because the port assigned to the local server isn't the default value that Tableau Server normally uses. You can find more details regarding the default port settings in the Tableau Server online

manual by searching for TCP/IP Ports. After completing this step you can now issue other commands to Tableau Server.

SETTING THE WINDOWS PATH

If you want to avoid having to manually change your current directory to the Tableau Server bin folder every time you want to run an executable file, add the bin folder to your Windows PATH system variable. Edit PATH by going to the Window Control Panel, click on System, then Advanced System Settings, and selecting the Environmental Variables button to expose the dialog box you see in Figure 11–3.

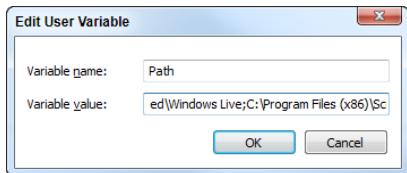


FIGURE 11–3 Editing the PATH system variable through control panel

If this seems intimidating there are free utilities available on the web that make this process easier, and provide a larger editing window. Figure 11–4 shows a free utility called Eveditor in which the PATH has been edited to include the bin folder.

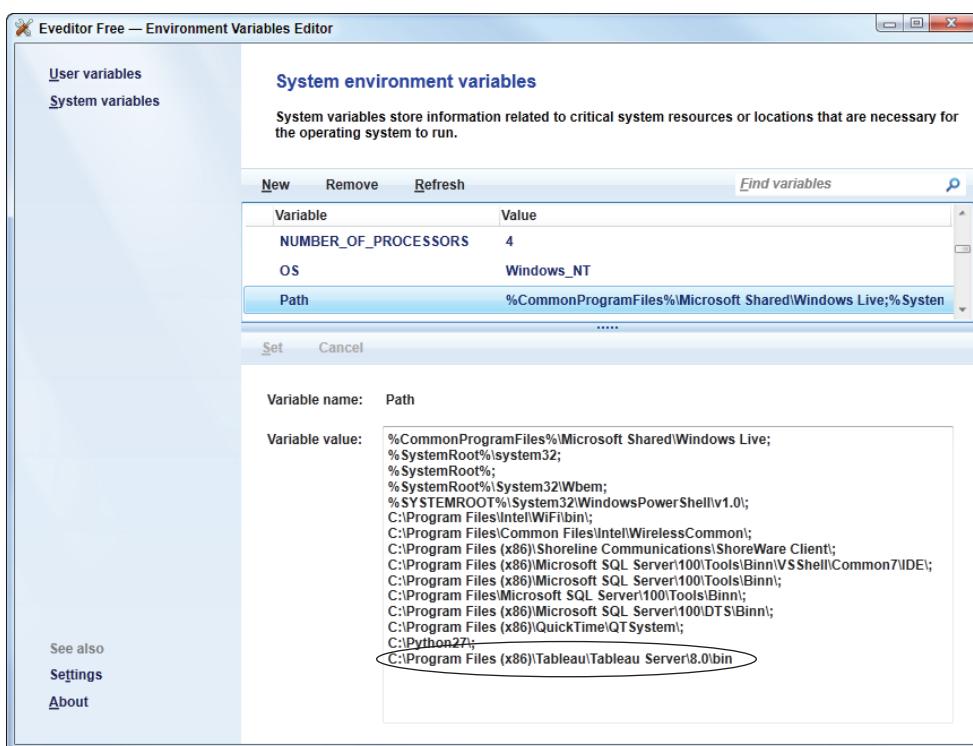


FIGURE 11–4 Editing PATH using a free editing tool

Adding the path for the Tableau Server bin folder eliminates needing to manually type in the path every time you want to start tabcmd or tabadmin in a batch file. Later you'll see how to dynamically set PATH commands inside executable batch files—enabling dynamic setting of the file path so that tabcmd can always find the script it needs to execute.

Keep in mind that any third-party tools (like Eveditor) are not supported by Tableau Software. You may be successful with Eveditor or other tools that you enjoy using, or you may experience problems. This is outside of Tableau Software's control.

WHAT KINDS OF TASKS CAN BE DONE WITH TABCMD?

The tabcmd utility provides the ability to automate routine tasks concerning workflow management activities related to:

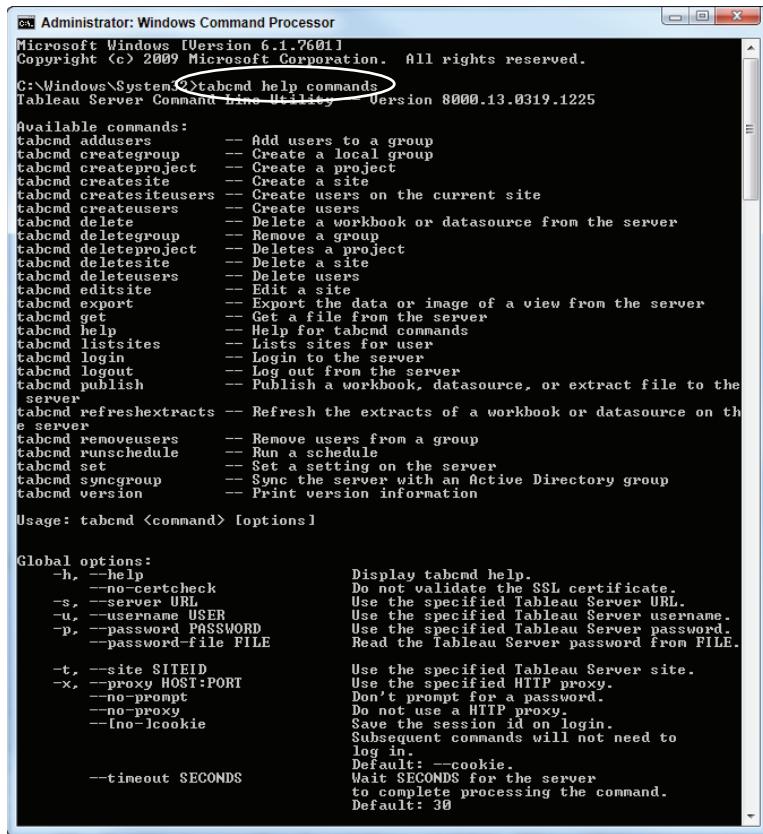
- Users, groups, projects, and sites
- Data management, publishing, updating
- Session management
- Security, site listings
- Server version information

The level of access and control is dependent on the type of administration rights assigned to the person using tabcmd. System Administrators can manage data connections, groups, projects and workbooks. They can add users to groups and projects. But, they are not able to alter user licensing levels. Systems administrators have full rights—including the assignment of licensing levels for users and managing the server itself. System Administrators can assign some administrative roles to Site Administrators. That role determines how much control will be given to a Site Administrator. Site Administrators can manage groups, projects, workbooks, and data connections. If the System Administrator permits it, they can also add or remove site users.

The tabcmd utility currently provides 24 functions with an additional 12 global option settings.

You can access a complete function reference in the Tableau Server's online manual in the **tabcmd Commands** section. Located at: <http://onlinehelp.tableausoftware.com/current/server/en-us/tabcmd.htm>.

Tabcmd also has a built-in help function for listing the available commands by entering Tabcmd Help Commands. Figure 11-5 shows the help command display.



The screenshot shows a Windows Command Processor window titled "Administrator: Windows Command Processor". The title bar includes the text "Microsoft Windows [Version 6.1.7601]", "Copyright (c) 2009 Microsoft Corporation. All rights reserved.", and "C:\Windows\System32\tabcmd help commands". The main content area displays the help output for the "tabcmd help commands" command. It lists various available commands with their descriptions:

- tabcmd addusers -- Add users to a group
- tabcmd creategroup -- Create a local group
- tabcmd createproject -- Create a project
- tabcmd createsite -- Create a site
- tabcmd createsiteusers -- Create users on the current site
- tabcmd createusers -- Create users
- tabcmd delete -- Delete a workbook or datasource from the server
- tabcmd deletegroup -- Remove a group
- tabcmd deleteproject -- Deletes a project
- tabcmd deletesite -- Delete a site
- tabcmd deleteusers -- Delete users
- tabcmd editsite -- Edit a site
- tabcmd export -- Export the data or image of a view from the server
- tabcmd get -- Get a file from the server
- tabcmd help -- Help for tabcmd commands
- tabcmd listsites -- Lists sites for user
- tabcmd login -- Login to the server
- tabcmd logout -- Log out from the server
- tabcmd publish -- Publish a workbook, datasource, or extract file to the server
- tabcmd refreshextracts -- Refresh the extracts of a workbook or datasource on the server
- tabcmd removeusers -- Remove users from a group
- tabcmd runschedule -- Run a schedule
- tabcmd set -- Set a setting on the server
- tabcmd syncgroup -- Sync the server with an Active Directory group
- tabcmd version -- Print version information

Below the command list, there is a usage section:

```
Usage: tabcmd <command> [options]
```

Then, there is a section for global options with detailed descriptions:

Global options:	
-h, --help	Display tabcmd help.
--no-certcheck	Do not validate the SSL certificate.
-s, --server URL	Use the specified Tableau Server URL.
-u, --username USER	Use the specified Tableau Server username.
-p, --password PASSWORD	Use the specified Tableau Server password.
--password-file FILE	Read the Tableau Server password from FILE.
-t, --site SITEID	Use the specified Tableau Server site.
-x, --proxy HOST:PORT	Use the specified HTTP proxy.
--no-prompt	Don't prompt for a password.
--no-proxy	Do not use a HTTP proxy.
--lno-jeckie	Save the session id on login.
--timeout SECONDS	Subsequent commands will not need to log in. Default: --cookie. Wait SECONDS for the server to complete processing the command. Default: 30

FIGURE 11-5 The tabcmd help function display

Entering Tabcmd Help and then a specific command name causes more complete options for that single command to be displayed.

LEARNING TO LEVERAGE TABCMD

In the following examples you'll see progressively more advanced ways to use tabcmd including:

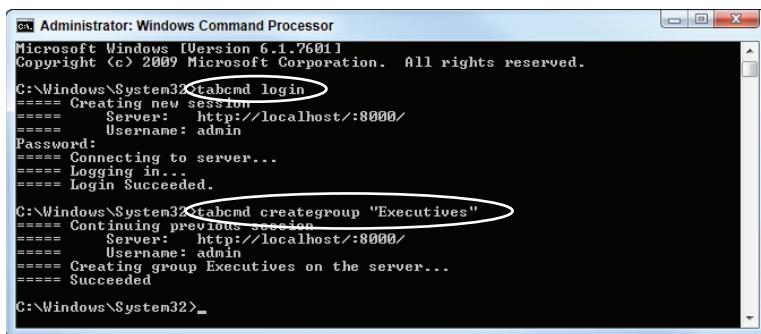
- Manually creating and running a tabcmd script

- Creating a Windows batch (.bat) file to run a saved script
- Using Windows Task Scheduler to automatically run a saved script

MANUALLY ENTERING AND RUNNING A SCRIPT IN TABCMD

The most basic way of using tabcmd is to manually enter commands that can also be accessed from the Tableau Server manual. This is also a good way to test tabcmd before you attempt to create script that automatically runs tabcmd.

A common task required of a content administrator is to create groups on Server and assign users to those groups. Figure 11–6 displays the script used to create a new group called Executives.



```

Administrator: Windows Command Processor
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>tabcmd login
==== Creating new session
===== Server: http://localhost/:8000/
===== Username: admin
Password:
===== Connecting to server...
===== Logging in...
===== Login Succeeded.

C:\Windows\System32>tabcmd creategroup "Executives"
==== Continuing previous session
===== Server: http://localhost/:8000/
===== Username: admin
===== Creating group Executives on the server...
===== Succeeded

C:\Windows\System32>

```

FIGURE 11–6 Adding a new group to server

The first command in Figure 11–6, `tabcmd login`, initiates a new session and prompts the user to enter a password. It is also possible to append the password to the end of the login command by adding `-p` or `--password` followed by your actual password. The script `Tabcmd Creategroup "Executives"` triggers the addition of the new group to server. At the bottom of the script you can see that tabcmd provides a status while processing and then confirms that the operation succeeded.

The next step is to assign users to the group. By creating a list of valid usernames (`egroupadd.csv`) and saving it in the Tableau Server bin folder, tabcmd can assign the specified users to the executive group. Figure 11–7 shows a list of server users on the left (Allen, Bill, Cal, Dave, Eric). On the right you see the executed script.

```

Administrator: Windows Command Processor
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\System32>tabcmd addusers "Executives" --users "egroupadd.csv"
==== Continuing previous session
==== Server: http://localhost:8000/
==== Username: admin
*** No such file or directory - C:/Windows/System32/egroupadd.csv

C:\Windows\System32>c "C:\Program Files <x86>\Tableau\Tableau Server\8.0\Bin
C:\Program Files <x86>\Tableau\Tableau Server\8.0\bin>tabcmd addusers "Executive
" --users "egroupadd.csv"
==== Continuing previous session
==== Server: http://localhost:8000/
==== Username: admin
==== Adding users listed in C:\Program Files <x86>/Tableau/Tableau Server/8.0/b
in/egroupadd.csv to group Executives...
==== Succeeded
==== Added 5 users to group Executives

C:\Program Files <x86>\Tableau\Tableau Server\8.0\bin>

```

FIGURE 11-7 Adding users to the new group

This is the script used to add the users:

```
tabcmd addusers "Executives" --users "egroupadd.csv".
```

These activities can be done directly in the Tableau Server GUI environment, but tabcmd may be a more efficient way to make group assignments if they change frequently or you have a large number of users to assign.

RUNNING TABCMD SCRIPTS VIA BATCH FILES

If you find yourself using the same script repeatedly, you can use a text editor to create and save the script for reuse later. Windows includes a text editor program called Notepad that can be used to enter and save tabcmd script. Notepad is normally located in the Windows accessories folder. Another Windows application—Task Scheduler—can be used to launch the script saved using Notepad. There are many other programming tools you can use for this purpose, but these are part of the Windows toolset.

The Steps Required to Create Batch Processing Scripts

Regardless of whether you prefer to use Windows Notepad or some other text editing software, the basic steps to create a batch process are the same.

1. Create the tabcmd script in Notepad or another text editor.
2. Save the script as a (.bat) executable file.
3. Double-click the batch file to execute the script.

In this scenario the script is still run manually but you no longer have to type all of the instructions every time you want to make changes, export data, or update files. These may be activities you repeat periodically—often enough to warrant saving a script—but not so often that you need to fully automate processing.

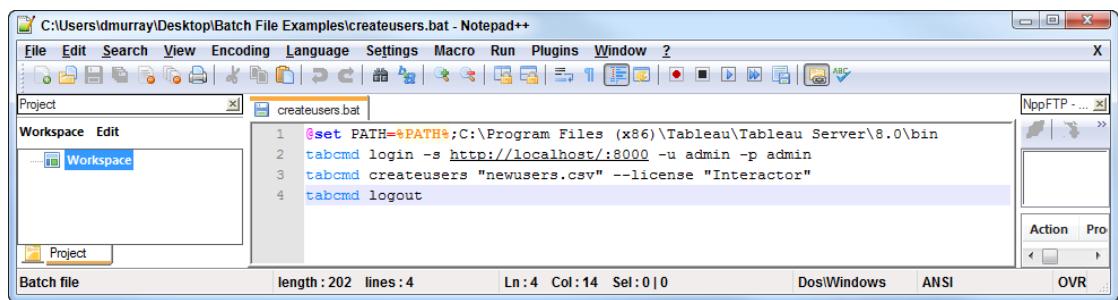
In the next example you'll see how to create a script in a text editor, save the script as a batch file, and then run the script using a CSV source file that provides the usernames and permissions needed to update Tableau Server.

Assume you have five new users to add and will be provisioning Interactor licenses for all of them. Figure 11–8 shows the CSV file with the names of the users.

Creating a robust script that will work flexibly is the goal. To do that requires a little knowledge of Windows commands and tabcmd. Figure 11–9 shows one way to accomplish adding the users.

A	B
1	Username
2	Brenden
3	Phil
4	Joe
5	Paul
6	Darren

FIGURE 11–8 CSV file containing new user list



```
C:\Users\dmurray\Desktop\Batch File Examples\createusers.bat - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
Project createusers.bat NppFTP - ...
Workspace Edit
... Workspace
Project
Batch file length : 202 lines : 4 Ln : 4 Col : 14 Sel : 0 | 0 Dos\Windows ANSI OVR ...
@set PATH=%PATH%;C:\Program Files (x86)\Tableau\Tableau Server\8.0\bin
tabcmd login -s http://localhost:8000 -u admin -p admin
tabcmd createusers "newusers.csv" --license "Interactor"
tabcmd logout
```

FIGURE 11–9 Creating and saving script in a .bat file

For the batch file to run properly, place it in the same directory as the CSV file that contains the users that need to be added to server. The first line of the code in Figure 11–9—`@set PATH=%PATH%`—defines the path for the file to search if any elements are not located there. These Windows commands allow you to define the path for the batch session only. This is a better practice than blending data files in with Windows system files (not a good practice). It also has the virtue of persisting only while the batch file is being executed—rendering the earlier example of permanently editing the PATH system variable unnecessary.

The rest of the script in Figure 11–9 includes tabcmd commands that are located in the bin folder specified by the set path command. In fact, you can define many different paths using this method for files that you want to keep separated.

The bulleted list below may be easier for to read than Figure 11–9. Alter the specific code where applicable to match your system's setup and the name of the CSV file that you created to load new users.

- Line 2—Login to Tableau Server
- Line 3—Create the users from the newusers.csv file
- Line 4—Logout of Tableau Server

When the program starts you'll see each command run and when it is finished the command window will close automatically. Figure 11–10 shows a screenshot of the Windows Command Processor window—running the script.

FIGURE 11-10 The executed script

As the program finishes, the screen automatically clears. If you want to keep it in view, add a fifth line to the script using the Pause command. With just a few lines of code you can update many records this way. If you aren't using Active Directory to secure Tableau Server, this method provides a quick way to mass load hundreds of users from a file.

Using Windows Scheduler to Fully Automate Scripts

By adding the batch file from the previous example to Windows Task Scheduler, the file can be updated based on a trigger event or a specific time schedule. For example, if you have an actively expanding user base it might be necessary to create new users in the system every day. The system administrator

could add new users to the Createuser.csv and schedule updates every day at a specific time. Figure 11–11 shows the Task Scheduler application. A new task was created to ADD USERS to Tableau Server (daily). The following steps were used to define the schedule:

1. General tab—Name and describe the task and set security options.
2. Triggers tab—Define what causes action (daily at 7:00AM).
3. Actions tab—Select the batch file to run (point to the Createuser.bat file).
4. Conditions tab—Set desired limitations for the run to occur.
5. Settings tab—Specify additional settings affecting the task behavior.

This will cause the file to be updated on a regular basis without the need for the batch file to be manually selected. Figure 11–11 shows the task scheduled for automatic update of new user additions.

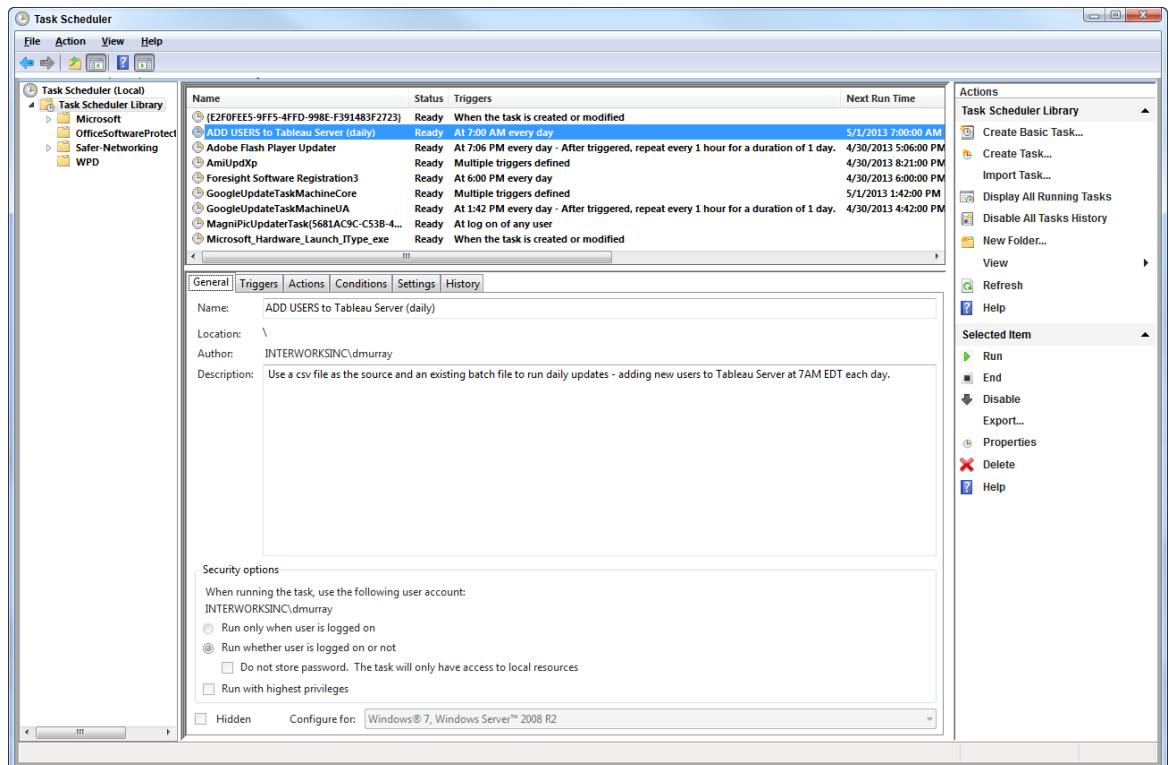


FIGURE 11–11 Window task scheduler

Even if you're on vacation, updates can continue if you delegate the task of adding the usernames and license level to the Newusers.csv.

COMMON USE CASES FOR TABCMD

There are many different ways to utilize tabcmd to automate repetitive or intensive production issues. If you find yourself doing repetitive tasks consistently, you should consider using tabcmd to automate the process to Save time, improve accuracy, and enhance the way you can share and update files.

The examples presented next are intended as a sampling of the ways you could use tabcmd. You will undoubtedly think of many more ways to automate processes that repeatedly require your attention.

RETRIEVING FORECAST DATA FROM WORKBOOKS

Tableau's forecasting ability can be used to create initial projections based on historical patterns. The tabcmd Export function can be used to publish forecasted data points from a workbook view. Exporting data in CSV format can then be used to update a source database or a spreadsheet. This first-pass view of the forecast can then be tweaked and returned to a database and stored.

Even more commonly, historical data can be published as well. Even though it may be easier to analyze data using Tableau, some users may lack license access. You may wish to share exported PDF, PNG, or CSV files with vendors that don't have access to Tableau. Alternatively, you might publish packaged workbooks specifically for partners and allow them to access specific groups on your server.

MANAGE DATA GOVERNANCE VIA TABCMD

You may want to create a quality control directory that you publish raw files to for review, then after auditing and approval use the Publish command to move the preliminary file into a production group or project. This is an interesting alternative to heavy-handed quality control. Instead of focusing on the end report, IT can focus on ensuring the quality of the data extract file and provide information to consumers with a vetted preliminary view that can be modified to suit specific needs.

USING TABADMIN FOR ADMINISTRATIVE TASK AUTOMATION

The Tabadmin Toolset is intended for use by the designed server administrators responsible for configuring and maintaining Tableau Server's data and

metadata. Tabadmin has its own set of commands that are exclusively used for these purposes. You can find a categorized list of these in Appendix A.

Normally, a very limited number of technical staff members are tasked with the responsibility of developing, maintaining, and monitoring system performance. The tasks performed using Tabadmin include:

- Tabadmin help
- Conducting system backups and restores
- Displaying information on system status
- Cleaning service log files
- Resetting the password for the Tableau Server account
- Enabling or disabling access to Tableau Server's Postgres database
- Creating zipped log files
- Stopping Tableau Server

STARTING TABLEAU SERVER

The tabadmin command tool may also be useful for additional tasks. The recommendations in the list below should only be attempted under the direction of Tableau Software Support or a qualified Tableau Server Partner.

- Altering default time-out provisions for queries.
- Changing default time-out limits for idle users.
- Creating a server log file.
- Configuring Tableau Server processes.
- Printing Tableau Server license information.
- Printing information on active users.
- Setting primary and secondary gateway hosts.
- Executing system changes via the configure command.

The Tabadmin command line utility is the primary tool maintaining the safety and performance of the server. Tableau software provides extensive documentation on their website.

Finally, Tableau provides online reference material that covers all of the details well. You can find this by navigating to the Admin view and selecting the help menu in the upper right of the screen, then Get Help and Support.

That will take you to the online help screen where you will find the online administrator guide. Specific sections related to Tabadmin activities include Tabadmin, Database Maintenance, and Troubleshooting.

The first eleven chapters of this text have been about introducing you to Tableau Desktop, Tableau's Server products (Server, Online, and Public), and Tableau Server's command line toolset. In the final chapter, short case studies will be presented that show how others have used Tableau to address their business needs.