

## Tableau Software: Best Practices in the Enterprise

This document describes best practices and recommendations on the usage of Tableau Software – Tableau Desktop, Tableau Server, and associated utilities – **with a focus on technical and enterprise subjects**. Please note:

- The concepts listed herein are not hard and fast requirements, nor are they meant to be a complete list, nor are they meant to replace the online help and knowledge base (In fact we link to official online resources extensively), and
- That there is always more than one way to perform certain steps with Tableau Software, and
- That these are meant to be *possible* guidelines for enterprise deployments.

This is a living document which may undergo frequent changes. We have provided as much detail with screen shots and HTML links wherever possible.

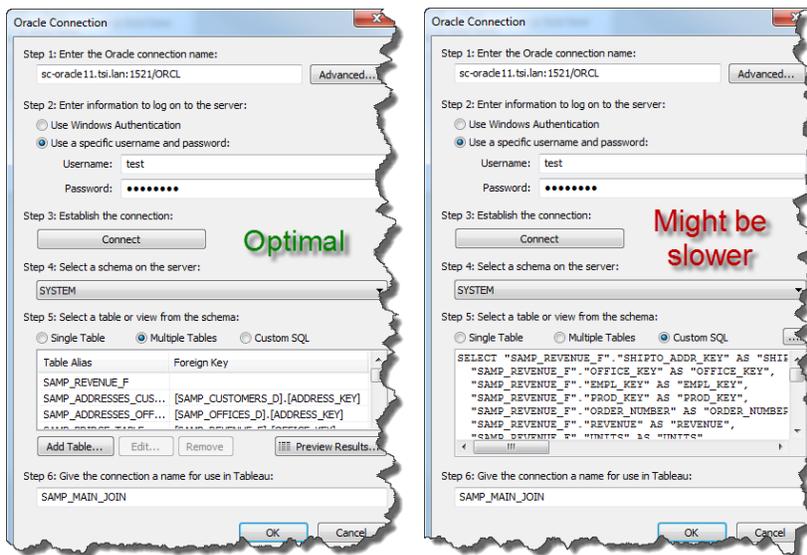
**In summary, the purpose of this document is to provide best practice *opinions* on enterprise usage based upon our collective experiences with existing Tableau customers.**

**Contents (*control-click to navigate*):**

- [Connecting to Data](#)
- [Metadata Management](#)
- [Publishing to and using Tableau Server](#)
- [Data-driven Security](#)
- [Tableau Server Permissions](#)
- [Administration of Tableau Server](#)
- [Tableau Server Usage Auditing](#)
- [Performance Concepts](#)
- [Automation Concepts](#)
- [Authentication Concepts](#)
- [Distributed Installations / Clustering](#)
- [IT Tasks and Personnel Requirements](#)
- [IT Process Control Concepts](#)

## Best Practices for... Connecting to Data

**Custom SQL Statements:** When connecting to multiple tables, choose to build a multi-table join using the regular multi-table Tableau interface, and *avoid* using custom SQL. Custom SQL will take the entire SQL statement in the custom SQL dialogue and “wrap” all of your human drag and drop actions inside of that statement. This might result in an onerous and complex query being sent to the database. Custom SQL can be very useful in specific situations; However, it should only be considered an option when regular multi-table joins are insufficient.

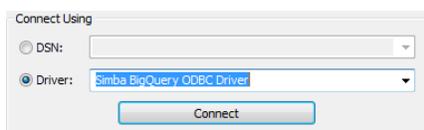


**Driver Support:** Always use the database drivers provided by Tableau Software, located on our support pages.

**General Order of Database Connectivity Support:** The general best practice of support for databases and their drivers is:

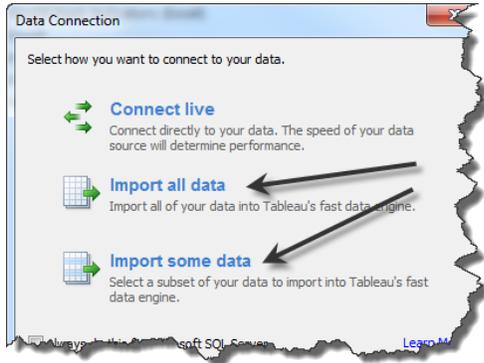
1. Use the “First Class” database connection if it is listed in Tableau’s “connect to data” dialogue
2. Use a generic “ODBC driver” connection if #1 is not available, and
3. Use a generic “ODBC DSN” connection if #2 is not available or not working.

**ODBC Connections:** See above. If possible, try to build your generic ODBC connections directly inside of Tableau Desktop, as opposed to leveraging a DSN entry in Microsoft Windows. By doing so you will be providing maximum portability when publishing from Tableau Desktop to Tableau Server.

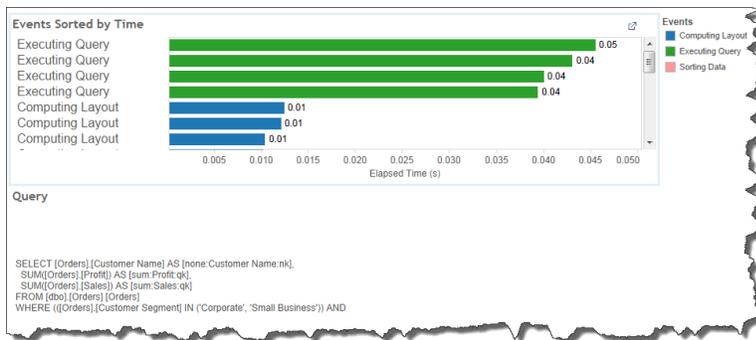


**Customizing ODBC Data Connections:** If you want to improve performance when using ODBC connections, refer to the knowledge base article on [customizing ODBC connections](#). There are many, many possible options here.

Managing Slow Connections: If your live connection to your data source is slower than desired, you may want to consider a partial or full extraction of that data using Tableau’s Data Engine:



Another approach to solving a slow performance data connection is to analyze the queries that Tableau is sending to your data source with an eye for tuning that data source. These queries can be found by enabling Performance recordings directly inside of Tableau Desktop (from the help menu):



You can then use the queries observed to perform various database tunings as needed.

---

---

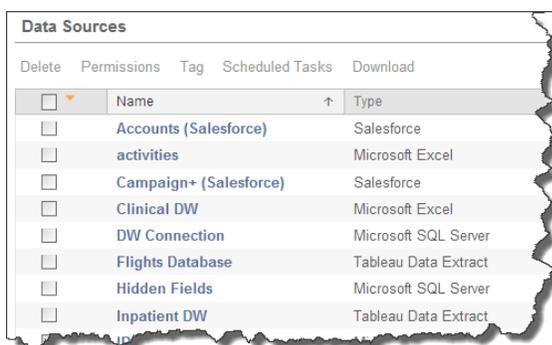
*Best Practices for... Metadata Management*

**Overview:** Tableau Desktop and Tableau Server provide a wealth of features around metadata management. We have consistently been ranked as one of the top analytics companies around this area of functionality. A great place to start learning about these features is our white paper on the [Tableau Metadata Model](#). Some of the specific tools in our toolkit that help manage metadata include:

- Complete separation of the data connection from the meta data management and experience.
- Built-in ability to recognize data types such as dates, strings, numbers, and geographies, as well as the ability to easily cast or override these choices.
- Built-in metadata features such as Tableau Groups, Sets, Bins, Hierarchies and similar.
- Built-in non-aggregate or aggregate calculation engine.
- Built-in post-database-operative aggregation calculations called “Tableau Table Calculations” which allow you to perform 2<sup>nd</sup>, 3<sup>rd</sup>, or Nth pass calculations on your answer sets.
- Field Obfuscation (hide fields), field renaming (edit column names locally), and field value renaming (edit aliases)
- Single Version of the Truth controls using Tableau Server and the Tableau Server “Data Server”

**The Number One Best Practice Suggestion for Metadata Management is...** whenever possible, use Tableau Server and its “Data Server” component to centralize all metadata management. Learn more about this feature by reading the [white paper](#), watching a [training video](#), reading our [blog entry](#) on the subject, or reading the [built-in help](#).

*Example of Several Managed Tableau Server Data Sources:*



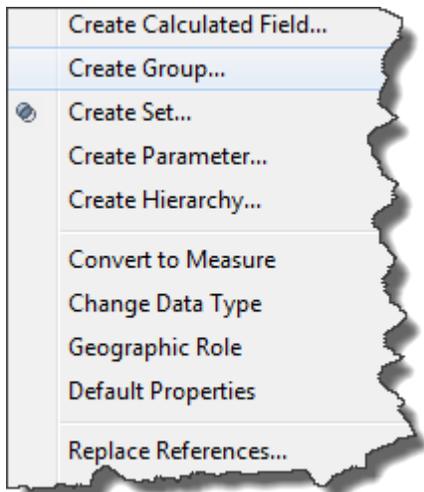
Data Sources				
Delete	Permissions	Tag	Scheduled Tasks	Download
<input type="checkbox"/>	Name			Type
<input type="checkbox"/>	Accounts (Salesforce)			Salesforce
<input type="checkbox"/>	activities			Microsoft Excel
<input type="checkbox"/>	Campaign+ (Salesforce)			Salesforce
<input type="checkbox"/>	Clinical DW			Microsoft Excel
<input type="checkbox"/>	DW Connection			Microsoft SQL Server
<input type="checkbox"/>	Flights Database			Tableau Data Extract
<input type="checkbox"/>	Hidden Fields			Microsoft SQL Server
<input type="checkbox"/>	Inpatient DW			Tableau Data Extract

**PHI, HIPAA or Field Security:** If there are fields which should not be exposed to the analytic experience, hide them and then perform a Tableau extraction. These fields will not be pulled into the extraction file.

- City
- # Customer ID
- Customer Name
- Customer Segment
- Order Date
- Order Priority
- Postal Code
- Product Category
- Product Container
- Product Name
- Product Sub-Category
- Region
- Ship Date
- Ship Mode
- State or Province
- Measure Names

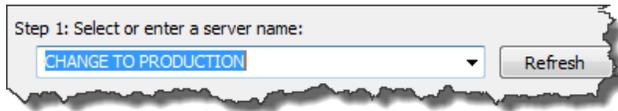
Fields in grey  
will not be  
extracted

**Metadata Feature Support:** If desired, create common metadata objects like sets, groups, calculated fields and hierarchies directly inside the Tableau experience, as opposed to in the database directly. Two important results of doing so are: 1. Improved database maintenance and performance, 2. Centralization of the metadata experience in the Tableau layer. This is a subject which generates many opinions. Some folks think that all data definitions should live in the database, and other folks think that they should live in Tableau. This is a choice you get to make – not a hard fast rule.

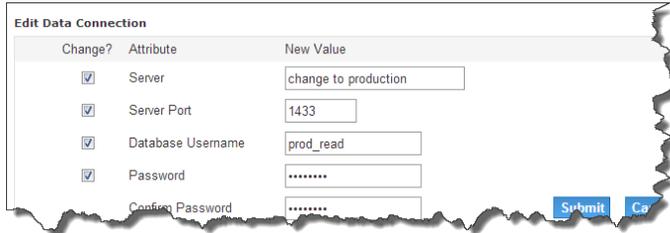


**Development vs. Production Environments:** You can have your analysts perform ad-hoc discovery or full-blown dashboard design using a test or development database, and at a later point you can flip the switch to production. You can do this in either Tableau Desktop or Tableau Server. Benefits include allowing your analysts to design against a non-production environment, as well as the ability to hide the production data from the designers (PHI or HIPAA compliance). All work created is maintained with 100% integrity – this means that any calculations, groups, sets, or similar will continue to work correctly when you change the data connection.

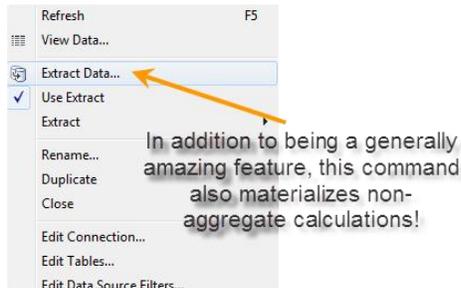
You can swap out the connection information in either Tableau Desktop:



...or on Tableau Server:



**Materialize non-aggregate calculations (including groups, sets and bins) using the Tableau Data Engine:** When connected to databases live, any non-aggregate calculations you have created in Tableau Desktop will be constructed and cached as you use the software. Running a data extraction on the source data will materialize these as true data fields for future use which will improve performance.



---

*Best Practices for...* **Publishing to and using Tableau Server**

**Overview**

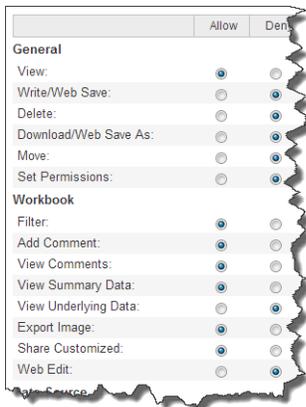
Tableau Server has extensive tools and features designed to make publishing Visualizations easy and powerful. Once published, you can manage your web content by following a few simple best practice suggestions.

**Authentication vs. Authorization**

Take some time to understand the difference between how you get into Tableau Server, and what happens once you are in (authentication vs. authorization). The Tableau Server [built-in help](#) is a great place to start. [How to work with permissions](#) is also a good read.

**Security Projects and Security Groups**

The Tableau Server authorization scheme (who can do what to which content) is designed to be straightforward and elegant. The list of actions you can take upon content is deliberately small:

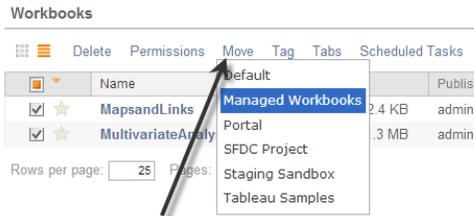


	Allow	Deny
<b>General</b>		
View:	<input checked="" type="radio"/>	<input type="radio"/>
Write/Web Save:	<input type="radio"/>	<input checked="" type="radio"/>
Delete:	<input type="radio"/>	<input checked="" type="radio"/>
Download/Web Save As:	<input type="radio"/>	<input checked="" type="radio"/>
Move: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

Tableau recommends using “groups” as opposed to individual users when assigning permissions to projects, workbooks or views. In this manner, you will not have to add or remove individuals to projects over time. Instead, you will only have to add or remove users from groups. This works whether you have integrated with Active Directory or are using Tableau’s built-in authentication.

**Make a staging area for content**

Tableau customers routinely use staging projects to park content which is still under development. Once “approved” by whatever human process is in place, it’s trivial to move content:

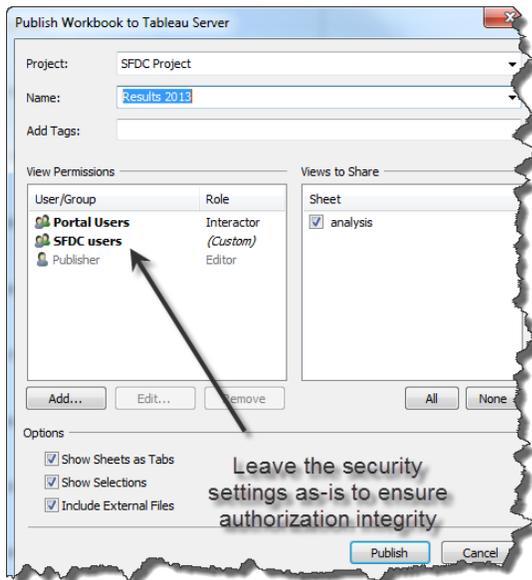


The "Move" command is an administrators' best friend...

Once you move content from Project A to Project B, you will want to reassign Project B's permissions for that workbook. By design, this step is not performed automatically by Tableau Server.

**Leave default security settings alone during publishing**

By design, any Tableau Desktop publisher has full control over "their content". They can decide unilaterally who can and cannot interact with their content. Meanwhile, IT organizations will have set up convenient default authorization schemas inside of Tableau Server. A best-practice step is to educate desktop publishers to leave the security settings along and then *make no changes here*:



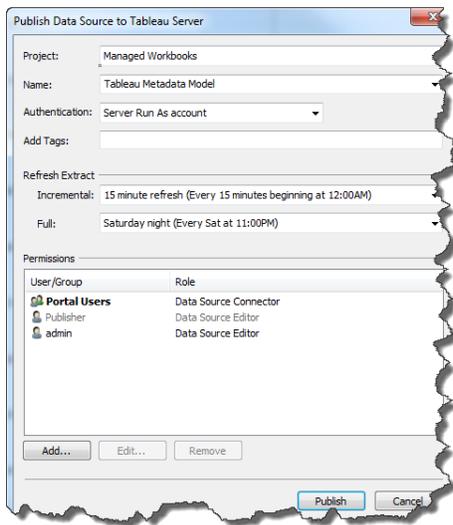
The Tableau Desktop publisher simply chooses the security "project" for publishing and Tableau Server takes care of all the correct permission assignments automatically.

**In strict environments where an organization cannot allow desktop publishing due to security concerns, use Tableau Server's "tabcmd" utility to publish**

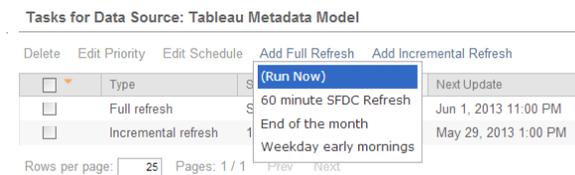
Each copy of Tableau Server ships with a command line utility called “tabcmd”. This Microsoft Windows utility can be scripted and automated using Windows Task manager or similar to publish regular or periodic workbook updates, removing the desktop user from the publishing experience entirely. This can solve complex security situations where no Tableau Desktop user should be allowed to publish whatsoever. Read up on the tabcmd utility [here](#).

**Leverage Tableau Server’s Scheduling Engine for data extractions**

Earlier, we discussed using Tableau Server’s Data Server as well as Data Engine to make Tableau Server the single version of the truth and central store for metadata. You can then enable a refresh and automation of this result for maximum benefit. The settings can be controlled during the publishing event:



As well as after the model has been published in the Tableau Server user interface:



## Best Practices for... Data-driven Security

### Overview

Tableau supports row-level and column-level security with either a manually managed process directly inside of the Tableau platform, or by leveraging data table -driven models. A common scenario is for regional managers to only see data for their regions or for doctors to only see data for their patients.

#### ***If you have no data-driven security tables, use Tableau’s built-in row-level security***

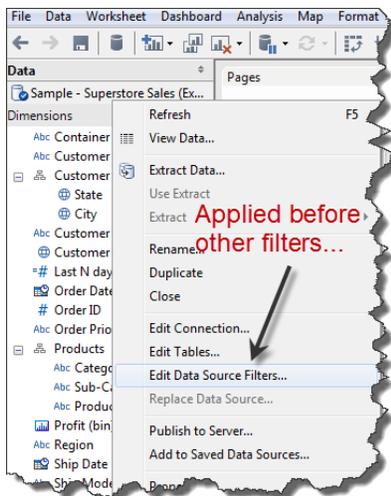
This easy to configure feature is documented in the top half of the knowledge base article [here](#).

#### ***If you do have data-driven security tables, use Tableau’s built-in automatic security via calculated fields***

This method derives the access to rows of data because you have defined this access information in a security table and have then joined this table into your main fact and dimension tables. This approach is documented in the bottom half of the knowledge base article [here](#).

### Tableau Server Data Source Filters

Starting with version 8, Tableau Desktop and Server include a powerful new feature called “Data Source Filters”. These are hidden filters which will be applied prior to any other filter on the filter shelf:



These data source filters can allow an enterprise organization to restrict rows based upon security, or simply restrict data for performance reasons (e.g. show last ninety days only). An excellent [community forum thread](#) discusses this in detail on Tableau’s website. The [knowledge base article on data source filters](#) is also a must read.

#### ***Data source filters versus “context” filters:***

Here is a brief overview of data source filters versus context filters. Starting with version 8, data source filters have had a tendency to replace context filters entirely, due to their broader usage applicability.

<b>Data Source Filters</b>	<b>Context Filters</b>
Data source filters behave similarly to placing a WHERE clause in customer SQL. (WHERE Dept = 'Furniture')	Context filters also behave similarly to placing a WHERE clause in customer SQL. (WHERE Dept = 'Furniture')
Tableau (Desktop and Server) applies data source filters ahead of any other filters in the workbook.	Tableau (Desktop and Server) applies context filters after data source filters, but ahead of any other filter on the filter shelf.
Valuable when you want to publish the data connection and not have users changing the pre-filtering criteria (e.g., in the custom SQL), as the data source filter is "hidden" from the end user. If you create a data source filter and publish the data connection, the end user will not see your filter if they try to Edit User Filters. (they can add additional DSFs but not see or change existing ones)	Valuable when you want to leverage the database ability to create a "temp" table. The context filter will create the temp table with the filtered criteria of any context filter on the sheet. Context filters can also be marked global to the entire data source. Note: Tableau may or may not actually create a temp table, based upon the ability of the database to do so. In some cases Tableau will perform a sub-select or even create a local, temporary extraction instead of creating a temp table as this will run faster.

---

*Best Practices for...* **Tableau Server Permissions**

***Permissions and authorization in the abstract***

Tableau Server authorization security (e.g. “What permissions are required to perform actions once logged into Tableau Server”) is designed to be both robust and easy to maintain.

However, it is imperative to understand the flow of security “allowances” and “denials” prior to implementing a solution. This document describes at a high level how Tableau Server authorization works.

There are some immutable “truths” to Tableau Server authorization worth getting out of the way right at the beginning of the discussion. These are:

- “System Administrator” - user accounts assigned this right (at the global level) have absolute control over the entire Tableau Server installation, regardless of any further content permissions assigned to that user.
- “Content Administrator” - user accounts assigned this right (on a site by site basis) have absolute control over the entire Tableau “site” to which they have been assigned, regardless of any further content permissions assigned to that user.
- “Project Leader” – user accounts assigned this right (at the project level) have absolute control over the content in that project.
- “Publisher” – a user who is allowed to publish workbooks will always have absolute control over those workbooks.

Anything else listed in this document will be over-ridden by the above statements.

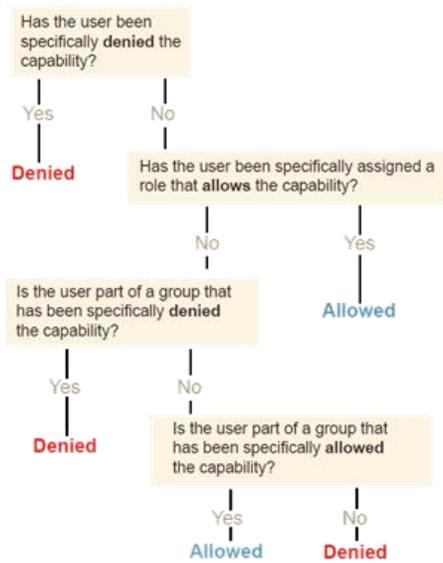
A second set of “truths” that permeate the Tableau Server authorization scheme are

- User specific permissions always outweigh/trump group permissions, and
- Denial assignment always outweigh/trump allowance assignments.

Taken in order, this means there are levels at work here: 1 = Grant Group, 2 = Deny Group, 3 = Grant User, 4 = Deny User, and 5 (fully denied because nothing was ever specified in the first place, i.e., Tableau’s default state is to “deny” a permission in the absence of any other information)

Thus, there is a very specific order of permissions, outlined in the graphic shown below, which is stated textually as follows:

*“Given a set of possible permissions and authorizations for an object, Tableau Server will look for the **largest** integer/level possible in determining if a user is allowed to perform a particular action: 1 = Grant Group, 2 = Deny Group, 3 = Grant User, 4 = Deny User, and 5 = fully denied.”*

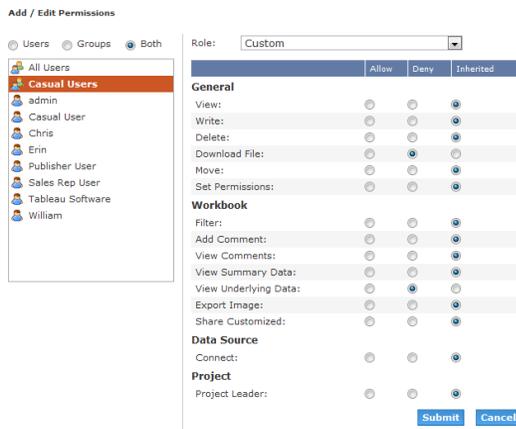


### Real world examples

Let’s look at a few simple business examples using screenshots from an actual Tableau Server installation.

- A) **Data Restriction:** A group called “Casual Users” that can never download the underlying data or download the workbook

What the permission assignment looks like:



What the end result looks like. A user “Erin” who is part of the “Casual Users” group will *not* see a “download workbook” link, and also will *not* see the underlying data tab, only the summary tab. They can download the “summary” data but not the row-by-row raw data.

Summary

Showing 198 out of 198 rows...  
[Download all rows as a text file](#)

Zip Code	Latitude (generated)	Longitude (generated)	SUM(Sales)	AGG(Profit Ratio)
84663	40.1329	-111.3618	\$21,649	6.82%
27006	35.9438	-80.4365	\$26,288	21.80%
80001	39.8028	-105.0861	\$22,886	19.19%
98103	47.6745	-122.3429	\$18,303	5.15%
06703	41.5541	-73.0461	\$33,450	21.23%
32122	29.2381	-81.036	\$2,996	-2.20%
29808	33.4518	-81.8508	\$24,966	9.02%
77037	29.8919	-95.3937	\$10,234	0.02%
68001	41.3331	-96.9504	\$69,774	16.46%
60406	41.655	-87.6815	\$16,042	5.23%
11769	40.7377	-73.1301	\$18,029	10.46%
62444	39.2445	-88.7619	\$37,697	-1.53%
81221	38.3577	-105.1415	\$12,413	22.19%
05544	42.6489	-71.1827	\$61,430	8.85%

B) **Content Management Restriction:** A user “William” who can never delete content, move content, or set permissions on content, but can still publish.

Add / Edit Permissions

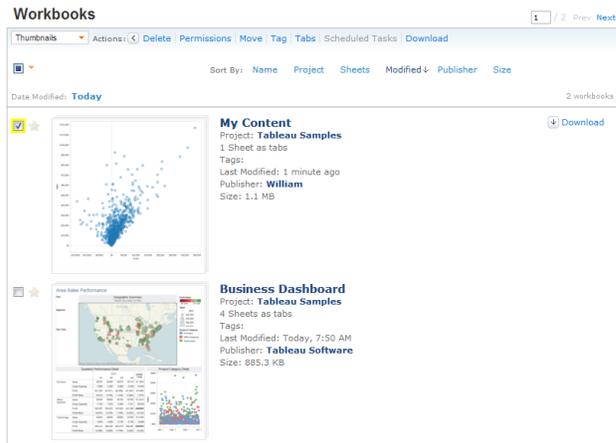
Users Groups Both

Role: Custom

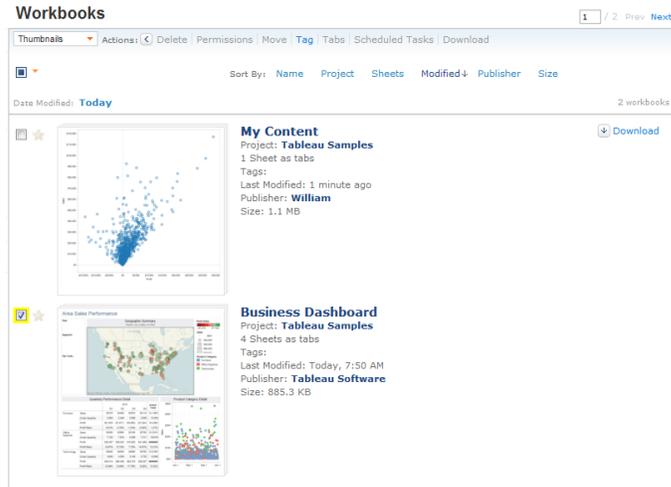
	Allow	Deny	Inherited
<b>General</b>			
View:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Write:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Delete:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Download File:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Move:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Set Permissions:	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
<b>Workbook</b>			
Filter:	<input checked="" type="radio"/>	<input 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 checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Export Image:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Share Customized:	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<b>Data Source</b>			
Connect:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Project</b>			
Project Leader:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Submit Cancel

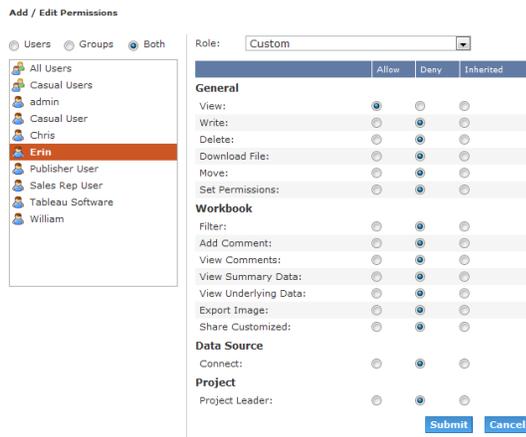
Notice that “William” can still set permissions, delete, or move his *own* content! This is one of the immutable rules (look for the blue links for “Delete”, “Permissions” and “Move”):



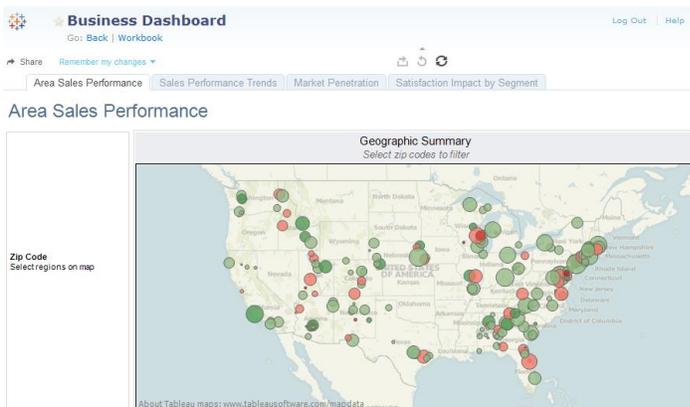
However, “William” cannot do so on other content within that same project (notice that the links are now greyed out since William selected a book that he did not publish):



- C) **Usage Restriction:** A user “Erin” who can interact with a visual in the most minimal manner. This user can use Tableau “actions” if they have been designed, but no other filters will show up. Nor can this user download the content, export to PDF, or similar. “Erin” can still save a Tableau “customized view” and also add Tableau “tags”.



Erin cannot interact with the view shown below – a shipping Tableau sample with quick filters. These filters should appear to the left of the map. Notice that the filters are gone.



### ***Best-practices from an implementation and management perspective***

Listed below are some elegant best-practices that Tableau recommends. These are not hard and fast rules, but general guidelines to implementing a successful installation of Tableau Server.

#### ***Create and use a “testing” project***

Nothing hurts more than making live changes to a production system, only to have your user community cry out in anguish because you have removed a critical permission. Tableau Server allows you to create unlimited numbers of projects, so why not use one of them for testing purposes? Here is a screenshot of a project called “Sandbox” with only one Tableau group called “designers” assigned to it:

Added new permission for *designers* to the selected project.

**Permissions: Project**

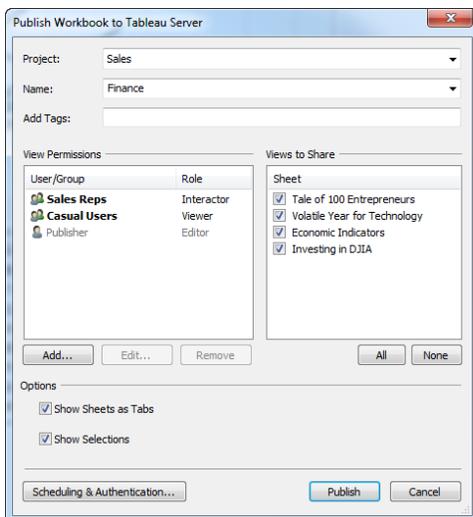
1 Selected project: Sandbox

User/Group	Role	Project	
<b>designers</b>	Project Leader	<i>all selected</i>	<a href="#">edit</a>

[Add / Edit Permissions](#) | [Assign Permissions to Contents](#)  
 Permissions are not automatically inherited by contained objects. To assign the above permissions to the contained 0 workbooks and 0 views click the link above.

**Create and use projects as security “containers” that have a specific security profile.**

All default security settings that you add to a Tableau project are automatically included when you publish workbooks or data sources to that project. Your workbooks inherit the overall “profile” of the security whenever you publish. Here is a screenshot of the Tableau Desktop publishing experience. The two groups listed (“Sales Reps” and “Casual Users”) were *not* data-entered by the publisher. They were there to begin with.



**Always remember to ‘Assign to content’ whenever you make a change to project security**

While this is not a required step, it is highly recommended. Any security changes you make to a project will *not* automatically cascade down to workbooks inside of that project. Here is a screenshot of the “Assign To Contents” link:

[Assign Permissions to Contents](#)  
 Permissions are not automatically inherited by contained objects. To assign the above permissions to the contained 7 workbooks and 39 views click the link above.

**Whenever possible, avoid setting permissions at the “view” level.**

Tableau authorization settings can be configured at the Project, Workbook, and individual View level. While it is possible to have “exceptions to the rule” down to the view level, this may quickly spiral out of control from a management perspective. It is reasonable from time to time to assign permissions for a workbook here or there, but it should be avoided whenever possible. If you feel like your security has become too complex, you can refer to the previous suggestion “Assign Permissions to Contents” – this would overwrite and wipe out any one-off assignments you have implemented.

**Create a project leader for each project**

When you create a new Tableau Server “project”, we recommend assigning a user to be in charge of that project. You can even edit the project description to include their name and email. That syntax looks like this:

Name:

Description: (867 characters remaining)

The end result looks like this:

<b>staging</b> <i>No description</i>	# Workbooks: 0
<b>Tableau Samples</b> A set of sample workbooks provided by Tableau Software. This project is maintained by <a href="#">Debby North</a>	# Workbooks: 7
<b>WebOPS</b> <i>No description</i>	# Workbooks: 0

**Use the “Default” project as a permissions template for all other projects.**

This is because all new projects will inherit permissions from this “default” project.

This is perhaps one of the most important best-practice tips available. Tableau ships with a built-in project called “Default” that cannot be deleted. The shipping permission settings for this “default” project are to allow “all users” (a built-in group) some basic abilities to view content. You have options here.

One option is to *make the default project restrictive*, and to explicitly *deny all permissions*. In this scenario, all new projects created will inherit this concept of “deny” from the default project – this could be useful for organizations that desire a locked down environment with “allowance” being the exception and not the rule.

Another option is to *make the default project interactive but read-only with no access to underlying data*. By disabling “View Underlying Data” and “Download Workbook” for the default project, all new projects (and thus their content) will inherit these permissions – this could be useful for organizations that want to standardize on visualization access, while still restricting data and workbook access.

A third option is simply to *remove “all users” from the “default” group*. This means any new projects created are truly a blank canvass from a security perspective – this option could be useful for organizations that do not have a standard in place, or those organizations where the standards change from project to project.

These are choices, and your actual deployment can differ and be fully customized.

**Use the built-in permission checker to troubleshoot permission assignments**

Tableau Server has a built-in permission checker for projects, workbooks, and even individual views. This is great way to spot check permissions for users. You can even highlight an individual permission, and Tableau will highlight for you which group or user has that permission:

**Permissions: View**

1 Selected view: Sales Performance Trends

User/Group	Role	View	Workbook	
All Users	Interactor	all selected	Business Dashboard	<a href="#">edit</a>
Casual Users	Custom	all selected	Business Dashboard	<a href="#">edit</a>
Chris	Custom	all selected	Business Dashboard	<a href="#">edit</a>
Erin	Custom	all selected	Business Dashboard	<a href="#">edit</a>
William	Custom	all selected	Business Dashboard	<a href="#">edit</a>

[Add / Edit Permissions](#)

Check user permissions:

Capability	Allow	Deny	Reason
View	✓		
Write		✓	Not granted by any permission for Sales Performance Trends.
Delete		✓	Not granted by any permission for Sales Performance Trends.
Filter	✓		
Add Comment	✓		
View Comments	✓		
View Summary Data	✓		
View Underlying Data	✓		
Export Image	✓		
Share Customized	✓		
Move		✓	Not granted by any permission for Sales Performance Trends.
Set Permissions		✓	Not granted by any permission for Sales Performance Trends.
Connect		✓	Not granted by any permission for Sales Performance Trends.

Click on a capability above to highlight the associated permission.

**Allow non-admin users to see all other users in the system**

If, as an administrator, you enable the setting for “Public User List” as shown on the maintenance page:

Settings	
Setting	Description
<input checked="" type="checkbox"/> Embedded Credentials	Allow publishers to attach passwords to published workbooks that will automatically authenticate web users.
<input checked="" type="checkbox"/> Scheduling	Allow publishers to assign workbooks to schedules. Embedded Credentials must also be enabled.
<input checked="" type="checkbox"/> Public User List	Allow web users to see a list of all users on the system.
<input type="checkbox"/> Saved Passwords	Allow users to save data source passwords across multiple visits and browsers.

Then all users will be able to see other users in the system, as well as their content (permissions still apply, however!) This is a great way to enable collaboration and sharing across the entire system!

The screenshot shows the Tableau interface with the 'Users' section active. On the left, there are navigation menus for PROJECTS, WORKBOOKS, VIEWS, RECENTLY USED, FAVORITES, and USERS. The main content area shows 'Recently Viewed' (Tableau Software, admin, Sales Rep User) and a 'List of All Users' section with a search box and a list of users: admin, Casual User, Chris, Erin, Publisher User, Sales Rep User, Tableau Software, and William. The top right corner says 'Welcome, Erin | Log Out'.

**Glossary of Individual Permissions (shown for a project. Workbook level permissions will not include the last two items):**

<b>General:</b>	
View	The ability to view the visual
Write	The ability to publish the visual, or, overwrite a published visual
Delete	The ability to delete a workbook or view
Download File	The ability to download the workbook in TWBX format
Move	The ability to move the workbook from one project to another project
Set Permissions	The ability to set permissions on the workbook or view
<b>Workbook</b>	
Filter	The ability to use quick filters, parameters or other interactive elements (actions and tooltips excluded)
Add Comment	The ability to add comments underneath the visual
View Comments	The ability to read comments underneath the visual
View Summary Data	The ability to pull up a window with the summary data in text form for the items selected
View Underlying Data	The ability to pull up a window with the underlying row by row data in text form for the items selected
Export Image	The ability to export the visual as an image or a PDF
Share Customized	The ability to share your customized views with other users in the system
<b>Data Source</b>	
Connect	The ability to connect to the data source with your identifying information (applies to published data sources only)
<b>Project</b>	
Project Leader	This setting overrides virtually all of the above settings, for a given project

**Add / Edit Permissions**

Users
  Groups
  Both

- All Users**
- Casual Users
- designers
- Sales Reps
- admin
- Casual User
- Chris
- Erin
- Publisher User
- Sales Rep User
- Tableau Software
- William

Role:

	Allow	Deny	Inherited
<b>General</b>			
View:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Write:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Delete:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Download File:	<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"/>
<b>Workbook</b>			
Filter:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Add Comment:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
View Comments:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
View Summary Data:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
View Underlying Data:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Export Image:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Share Customized:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Data Source</b>			
Connect:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
<b>Project</b>			
Project Leader:	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

## Best Practices for... Administration of Tableau Server

### Overview

A great place to start learning about Tableau Server administration is to watch the training video on [installation and configuration](#) of Tableau Server.

### Monitor the health of Tableau Server

Tableau Server includes a built-in xml output path with details about the [status of Tableau Server](#) itself. This can be scripted or leveraged by any number of third party solutions.

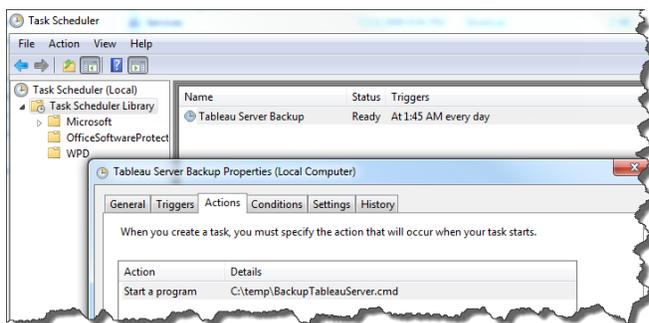
```

<systeminfo>
  <machines>
    <machine name="Primary">
      <repository worker="Primary:8060" status="OK"/>
      <dataengine worker="Primary:27042" status="OK"/>
      <serverwebapplication worker="Primary:8000" status="OK"/>
      <vizqlserver worker="Primary:9100" status="OK"/>
      <dataserver worker="Primary:9700" status="OK"/>
      <backgroundtasks worker="Primary:0" status="OK"/>
      <webserver worker="Primary:80" status="OK"/>
    </machine>
  </machines>
  <service status="OK"/>
</systeminfo>

```

### Schedule frequent and periodic backups

Tableau Server includes an easy monolithic [backup process](#) which can be configured and scheduled using Windows Task Manager. There is also a [knowledge base article on automating this process](#).



### Schedule frequent and periodic cleanup of log files

Tableau Server performs extensive logging of all events for support and debug purposes. You will want to periodically clean this up by reviewing this [knowledge base article](#).

```

C:\Program Files (x86)\Tableau\Tableau Server\Kraken\bin>tabadmin cleanup
==== The service is running. Files in use will not be cleaned up. Use the --
restart command to have cleanup stop and restart Tableau Server
==== Cleaning entries from http_requests log older than 7 days
-- Deleted 8 rows
==== Removing service logs.
==== Removing tempdir contents.
C:\Program Files (x86)\Tableau\Tableau Server\Kraken\bin>

```

***If you are planning on heavily using data extraction files, ensure that you have sufficiently powerful disk space***

When you first install Tableau Server, you may want to install it to your larger, more powerful drive, potentially including a strong SAN drive. Two knowledge articles which will help are [Location of the Tableau Server Application Data](#) and [Moving Tableau to a Different Drive](#).

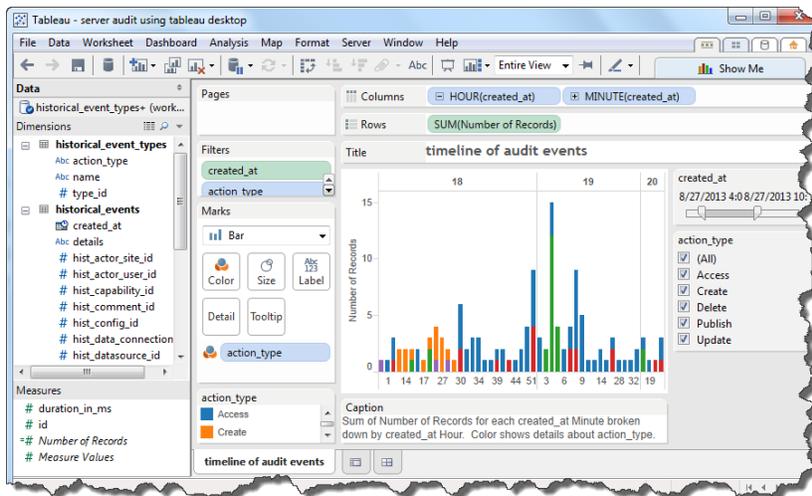
***Also note the following extra errata:***

- 1) Tableau Server can be deployed to a D drive or another drive which may have more disk space than a C drive, and
- 2) Tableau Server can be installed in a distributed manner, and the data engine files can be dedicated to a worker node, isolating the disk usage to that node. Read more on [Tableau Server distributed environments](#), and
- 3) In a distributed environment, make sure the Primary (Gateway) node has the same amount of disk space as the other nodes (although # of cores and RAM can be significantly less powerful). The reason is that all backups are created and restored through the primary node, therefore it needs sufficient hard drive space to gather/distribute all data before/after it is compressed/decompressed.

## Best Practices for... Tableau Server Usage Auditing

### Overview

Tableau Server ships with several built-in tables and views for reporting. Customers typically use Tableau Desktop to connect to the database which is *inside of Tableau Server*. They then build custom views and dashboards of this information. Here is an example:



### Details

In the image shown above we have connected to the Tableau Server database using the [built-in help on this subject](#).

The two main tables to join are “historical\_events” and “historical\_event\_types” – the key is HISTORICAL\_EVENT\_TYPE\_ID.

From there, you can start to join in other Tableau views which begin “hist” e.g. Join the view hist\_users” to the above tables.

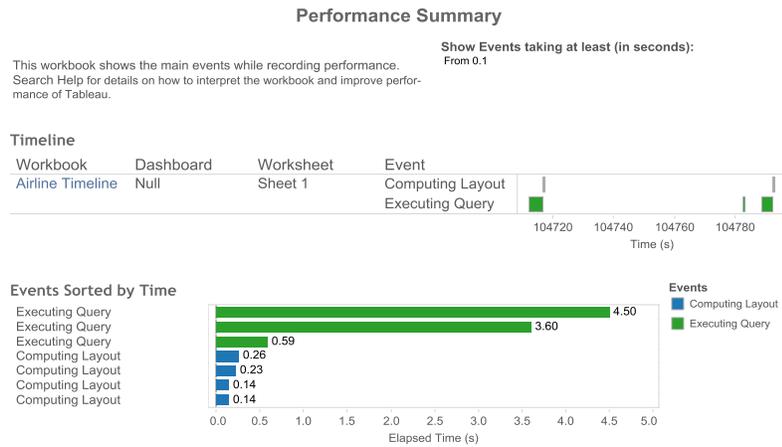
Best Practices for... Performance Concepts

**Overview**

A great place to start learning about Tableau Server performance concepts is by reading the [built-in help performance help](#).

**Record Performance of Dashboards and Interpret them**

Tableau Server includes a built-in performance recorder. If you have content, dashboards or reports that are suspiciously slower than desired, this performance recorder can help you pinpoint the bottleneck, whether in the data layer, the data model, the design, or the web layer. Detailed information is found in our built-in help on [Performance](#).



**Query**

```
<?xml version="1.0" encoding="utf-8" ?>
<sqlproxy>
  <query-function class="restrict">
    <restrictions>
      <field>[none:UNIQUE_CARRIER:nk]</field>..
```

---

*Best Practices for...* **Automation Concepts**

**Overview**

Tableau Server ships with a command line utility called tabcmd.exe. This utility can be installed on any “client” and can connect to Tableau Server to perform two basic types of action: security changes, and content retrieval.

[Download and installation instructions can be found here.](#)

---

*Best Practices for...* **Authentication Concepts**

Tableau Server has three modes for authentication:

1. Tableau's built in username and password management.
  - a. Pros: easy to configure; great for smaller deployments
  - b. Cons: no integration with AD or third party SSO.
2. Integration with Active Directory
  - a. Pros: great for corporate networks (behind firewalls); supports multiple domains which have two way trusts with the primary domain. [More information here.](#)
  - b. Cons: no integration with Non-AD SSO providers; unlikely to work outside your network (e.g. a DMZ network or other public facing network).
3. Integration via programmatic web exchange. This is known as "Tableau Trusted Authentication" or "Trusted Tickets" and is the third mode of authentication supported. Information on Trusted Authentication is [located here.](#)
  - a. Pros: able to integrate with **any** third party SSO provider. Great for external facing, DMZ installations.
  - b. Cons: requires a web developer to code up the solution. Example code in various programming languages ships with each copy of Tableau Server.

Additional notes: authentication does not imply "provisioning" of Tableau Server user accounts. In other words, how the user gets added to the system is independent of how that same user actually logs in. For adding users to the system, Tableau provides

[manual data entry.](#)

[CSV mass upload via web UI](#)

[Tabcmd command line automation](#) for adding AD or local user accounts to the system

8.1 note – Tableau is introducing SAML support starting with v8.1 – this is again for the actual authentication, and does not provide any direct support for the "provisioning" of the user accounts themselves.

---

*Best Practices for...* **Distributed Installations / Clustering**

Tableau Server can be distributed across multiple hardware. We provide all the mechanisms for this built-in to our software platform. In addition to working with Tableau Support or Tableau Professional Services to get detailed help with your topology or installation, here are some no-nonsense best practices that we have learned over the years:

**Don't** over-architect the solution!

We can't emphasize this point enough. Please do not try to overcomplicate the network and hardware topology. You will find that Tableau Server performs great across multiple pieces of hardware (or VM's) *without any need for third party software.*

**Don't** split Tableau Server components across different subnets or across a WAN.

This never works. Even if you think that your WAN or hardware is the most amazing situation ever, this is still likely to affect Tableau Performance. Always place all nodes in a distributed environment in the same subnet and domain. *Having all components in the same windows domain is a requirement for all Tableau Server distributed environments!*

**DO** use Tableau's built-in distributed mechanisms.

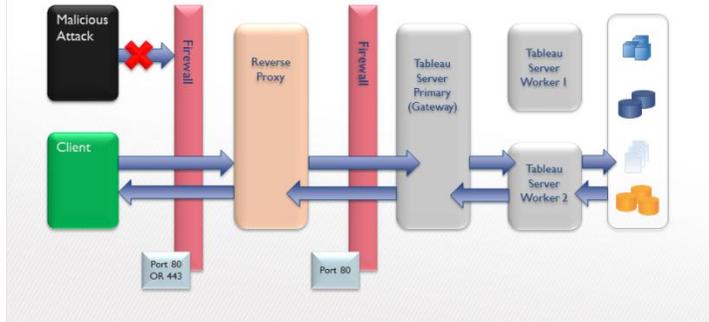
If you follow the instructions, the installation will work great and you will have a very low overhead in terms of configuration. Read up on [Tableau Distributed Environments](#) for more info.

**DO** use a reverse proxy if you are spanning a DMZ environment.

This means you place all Tableau components inside your firewall, and your reverse proxy in your DMZ. You can build a reverse proxy using Apache in 10-20 minutes. Tableau Services can also assist with this. Here's a picture:

## Reverse Proxy – Distributed Environment

- + Reverse Proxy in DMZ
- + Entire Tableau Server cluster in LAN



---

*Best Practices for...* **IT Tasks and Personnel Requirements**

Here is a quick overview of the types of IT tasks involved with Tableau Server. This can vary from installation to installation.

***Rare tasks:***

Disaster recovery, crashes, etc

***Occasional tasks:***

Troubleshooting a particular Tableau workbook, manual security configurations

***Ongoing tasks:***

Performing Tableau hotfix upgrades once every 4-8 weeks, major upgrades once or twice a year

---

### *Best Practices for... IT Process Control Concepts*

Some great ideas for process controls, education, and evangelism of a Tableau Software roll-out:

#### ***Create internal user group***

- Video (@14min) - <http://www.ustream.tv/recorded/26807227>
- Example Article - <http://data-informed.com/facebook-adopts-tableau-for-analytics-dashboards>

#### ***Hold Training Sessions***

- Video (@16:25) - <http://www.ustream.tv/recorded/26807227>
- Example Article - <http://data-informed.com/facebook-adopts-tableau-for-analytics-dashboards>
- Intro is how to connect to database and build simple views
- Advanced is more customized to your organization's data challenges
- Your level of training determines what projects you are allowed to publish into

#### ***Appoint Data Stewards***

- These are leaders in each BU that know their data very well about would be able to QA a published connection and 'bless' it into a productionized project

#### ***Multi-tenancy for 'Certified' and 'Uncertified' instead of different departments or different BU's***

- Facebook – (@17:20) <http://www.ustream.tv/recorded/26807227>
- Create an open environment where users can publish anything
- Have process (Data Stewards do QA) to move things into 'Certified'
  - o Check extracts, sql is correct, performance is good, etc

#### ***Democratize your data and embrace the concept of a SSE (Self Service Environment)***

- eBay - <http://www.tableausoftware.com/learn/stories/ebay-drives-business-analytics>
- Use data server to govern data connections
- Data Server in combination with clear communication of context of report (hidden filters, blended sources, etc) is essential to alleviate 'single source of truth' arguments. There are only truths within context – a single source of truth is a red herring.

#### ***There are generally 4 types of user communities:***

- IT MANAGED DEPLOYMENT
- IT owns standardized, published data sources
- IT owns review process for publishing any content to Prod

-Business owns all content in SSE (Self Service Environment)

Type	Business or IT	Server	Desktop	Create New Content	User Split
Web Consumer	Bus	Y			
Web Consumer /Casual Editor	Bus	Y		Y (web editing)	
Builder (aka Analyst, Power User)	Bus or IT (probably both)	Y	Y	Y (can publish to SSE)	
Publisher	IT	Y	Y	Y (can publish to Prod)	

**IT SUPPORTED (BUT MINIMALLY MANAGED) DEPOLOYMENT**

- IT owns standardized, published data sources
- Business owns all content and validity of report results

Type	Business or IT	Server	Desktop	Create New Content	User Split
Web Consumer	Bus	Y			
Web Consumer /Casual Editor	Bus	Y		Y (web editing)	
Builder (aka Analyst, Power User)	Bus	Y	Y	Y	
Publisher	Bus and IT	Y	Y	Y	