Tableau Server Utilities: Tabcmd

The Tableau™ Server Command Tool is one of the two command line tools that comes with Tableau Server. The Command Tool (tabcmd.exe) helps you automate common tasks including batch publishing workbooks and user/group administration. Refer to the Tableau Server Administrator Guide to learn more about the Tabadmin command line utility.

The Tableau Command Tool takes a command, an argument, and options as shown in the format below:

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

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

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

When the command is successful, tabcmd will return a status code of zero. A full error message for non-zero status codes is printed to stderr. In addition, informative or progress messages may be printed to stdout. A full log (including debugging, progress, and error messages) is written to C:\Documents and Settings\<username>\Application Data\Tableau\tabcmd.log.

What's in this Guide

Global Options 2

Commands 3

- addusers *group-name* 3
- creategroup *group-name* 3
- createproject *project-name* 3
- createusers *filename.csv* 4
- delete *workbook-name* 5
- deletegroup *group-name* 5
- deleteusers *filename.csv* 5
- get *url* 6
- login 6
- logout 6
- publish *filename.twb(x)* 7
- refreshextracts *workbook-name* 7
- removeusers *group-name* 8
- runschedule *schedule-name* 8
- set *setting* 8
- syncgroup *group-name* 9
- version 9

Global Options

Some options are common to all commands. The table below shows the options that are used by all commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

Option (short)	Option (long)	Argument	Description
`-h`	`--help`		Displays the help for the command.
`-s`	`--server`	Tableau Server URL	Required at least once to begin session.
`-u`	`--user`	Tableau Server username	Required at least once to begin session.
`-p`	`--password`	Tableau Server password	Required at least once to begin session. You can alternatively use the -P option.
`-P`	`--password-file`	filename.txt	Allows the password to be stored in the given file rather than the command line for increased security.
`-x`	`--proxy`	Host:Port	Uses the specified HTTP proxy.
	`--no-prompt`		When specified the command will not prompt for a password. If no valid password is provided the command will fail.
	`--no-proxy`		When specified an HTTP proxy will not be used.
	`--[no-]cookie`		When specified the session id is saved on login so subsequent commands will not need to log in. Use the “no-” prefix to not save the session id. By default the session is saved.
	`--timeout`	Seconds	Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

Commands

Each command that can be used with the Command Tool is described below.

help *command-name*

Shows detailed usage for the given command.

Example: `tabcmd help publish`

addusers *group-name*

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

Example: `tabcmd addusers "Development" --users "users.csv"`

Option (short)	Option (long)	Argument	Description
	`--users`	filename.csv	Add the users in the given file to the specified group. The file should be a simple list with one username per line. The users should already be created on Tableau Server.
	`--[no-]complete`		When set to "complete" this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.

creategroup *group-name*

Creates a group with the given group name. Use addusers (for local groups) and syncgroup (for Active Directory groups) commands to add users after the group has been created.

Example: `tabcmd creategroup "Development"`

createproject *project-name*

Creates a project with the given project name.

Example: `tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing quarterly sales reports."`

Option (short)	Option (long)	Argument	Description
`-n`	`--name`	name	Specify the name of the project that you want to create.
`-d`	`--description`	description	Specify a description for the project.

createusers *filename.csv*

Creates the users listed in the given comma separated values (csv) file. The file may have the columns in the order shown below.

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

The file can have fewer columns. For example it can be a simple list with one username per line. When the Server is using Active Directory authentication, the password column should be left blank. Quotes may be used if a value contains commas.

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

Option (short)	Option (long)	Argument	Description
	`--nowait`		Do not wait for asynchronous jobs to complete
	`--silent-progress`		Do not display progress messages for asynchronous jobs.
	`--license`	interactor, viewer, or unlicensed	Sets the default license level for all users. This setting may be overridden by the value in the CSV file.
	`--[no-]administrators`		DEPRECATED: do not use.
	`--admin-type`	system, content, or none	Assigns or removes the Admin right to all users in the CSV file by default. This setting may be overridden by the value in the CSV file. The default is none for new users and unchanged for existing users.
	`--[no-]publishers`		Assigns the Publish right to all users in the CSV file by default. This setting may be overridden by the value in the CSV file. The default is no for new users and unchanged fro existing users.
	`--[no-]complete`		Requires that all rows be valid for any change to succeed. By default, --complete option is used.

delete *workbook-name*

Deletes the given workbook from the server. This command takes the workbook name as it is on the server, not the file name when it was published.

Example: `tabcmd delete "Sales_Analysis"`

Option (short)	Option (long)	Argument	Description
`-r`	`--project`	Project name	The name of the workbook you want to delete.

deletegroup *group-name*

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

Example: `tabcmd deletegroup "Development"`

deleteusers *filename.csv*

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

Example: `tabcmd deleteusers "users.csv"`

Option (short)	Option (long)	Argument	Description
	`--[no-]complete`		When set to “complete” this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.

get *url*

Returns the file at the given URL using an http "get" command. Instead of rendering html, like a web browser would do, this command saves to a file. The URL should be as it is shown after the view is published. Views can be returned in either .pdf or .png format. Workbooks can be returned in either .twb or .twbx format.

Examples:

```
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename "Weekly-Report.png"

tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Workbooks\Weekly-Reports.twb"

tabcmd get "/user.xml" --filename "UserList.xml"
```

Option (short)	Option (long)	Argument	Description
`-f`	`--filename`	Name to save the file as.	Saves the file with the given filename.

login

Logs into the server. Use the `--server`, `--username`, `--password` global options to create a session. If you want to log in using the same information you've already used to create a session just specify the `--password` option. The server and username stored in the cookie will be used.

If you do not provide a password you will be prompted for one. If the `--no-prompt` option is specified and no password is provided the command will fail.

Once you login, the session will continue until it expires on the server or the "logout" command is run.

Example: `tabcmd login -s sales-server -u administrator -p p@ssw0rd!`

logout

Logs out of the server.

Example: `tabcmd logout`

publish *filename.twb(x)*

Publishes the given workbook to Tableau Server. By default all sheets in the workbook are published without database usernames or passwords.

Example: `tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-user "jsmith" --db-password "p@ssw0rd"`

Option (short)	Option (long)	Argument	Description
`-n`	`--name`	Name of the workbook on the server	If omitted, the workbook will be named after filename, without the twb or twbx extension.
`-o`	`--overwrite`		Overwrites the workbook if it already exists on the server.
`-r`	`--project`	Name of a project	Publishes the workbook into the specified project. Publishes to the "Default" project if not specified.
	`--db-username`	Database username	Use this option to publish a database username with the workbook.
	`--db-password`	Database password	Use this option to publish a database password with the workbook.
	`--save-db-password`		Stores the provided database password on the server.
	`--thumbnail-username`	Username	If the workbook contains users filters, the thumbnails will be generated based on what the specified user can see. Cannot be specified when --thumbnail-group option is set.
	`--thumbnail-group`	Name of group	If the workbook contains users filters the thumbnails will be generated based on what the specified group can see. Cannot be specified when --thumbnail-username otpion is set.
	`--tabbed`		When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.

If the workbook contains user filters, one of the thumbnail options must be specified.

refreshextracts *workbook-name*

Performs a full refresh of the specified workbook's extracts. Any incremental refreshes defined for the workbook will not be run. This command takes the workbook name as it is on the server, not the file name when it was published.

Example: `tabcmd refreshextracts "Sales_Analysis"`

Option (short)	Option (long)	Argument	Description
	`--synchronous`		Runs the full refresh operation immediately in the foreground.

removeusers *group-name*

Removes the users listed in the --users argument from the group with the given group-name.

Example: `tabcmd removeusers "Development" --users "users.csv"`

Option (short)	Option (long)	Argument	Description
	`--users`	filename.csv	Remove the users in the given file from the specified group. The file should be a simple list with one username per line.
	`--[no-]complete`		Requires that all rows be valid for any change to succeed. If not specified --complete is used.

runschedule *schedule-name*

Runs the specified schedule. This command takes the name of the schedule as it is on the server.

Example: `tabcmd runschedule "5AM Sales Refresh"`

set *setting*

Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server. Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- embedded_credentials
- public_users_list
- remember_passwords_forever

Example: `tabcmd set embedded_credentials`

syncgroup *group-name*

Synchronizes the group with the given group-name with Active Directory. This command can also be used to create a new group on the server that is based on an existing Active Directory group.

Example: `tabcmd syncgroup "Development"`

Option (short)	Option (long)	Argument	Description
	`--license`	viewer interactor unlicensed	Sets the license level for all users in the group.
	`--[no-]administrator`		Assigns or removes the System Admin right for all users in the group. If unspecified, new users are not administrators and existing users are unchanged.
	`--[no-]publisher`		Assigns or removes the Publish right for all users in the gorup. If unspecified, new users are not assigned this right and existing users are unchanged.
	`--[no-]complete`		Requires that all rows be valid for any change to succeed. If unspecified, --complete is used.
	`--silent-progress`		Suppresses progress messages.

version

Prints the version information for the current installation of Tableau Server.

Example: `tabcmd version`