TabMon User Guide

TABLE OF CONTENTS

BEFORE YOU BEGIN

SYSTEM REQUIREMENTS

In order to install TabMon on your system please en sure that you have the following:

- OS: Windows
- Environment:
    - Tableau Server (9.0 or above)
        - http://www.tableau.com/products/server
    - Microsoft .NET framework 4.5 and above (installer will install this for you)
        - http://www.microsoft.com/en-us/download/details.aspx?id=30653
    - Visual C++ Runtime 2013 (installer will install this for you)
        - http://www.microsoft.com/en-us/download/details.aspx?id=40784
    - Windows Management Framework 3.0
        - http://www.microsoft.com/en-us/download/details.aspx?id=34595

CONFIGURE TABLEAU SERVER

Open Tableau Server JMX ports:

1. Open command line as administrator
2. Navigate to Tableau Server Bin Folder
3. Run the following tabadmin commands:
    ```
    1. tabadmin set service.jmx_enabled true
    2. tabadmin stop
    3. tabadmin configure
    4. tabadmin start
    ```
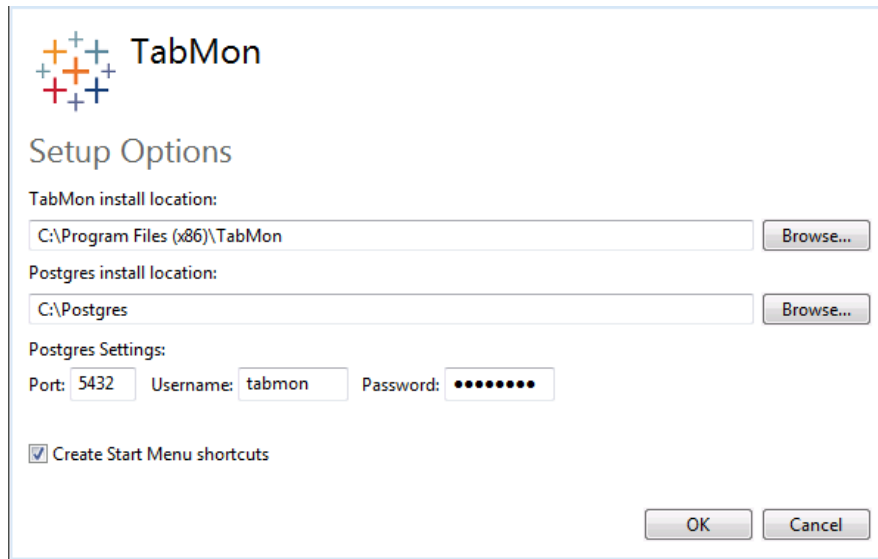
Configure the readonly user:

1. Open command line as administrator
2. Navigate to Tableau Server Bin Folder
3. Run the following tabadmin commands
   a. `tabadmin dbpass --username readonly [Password here]`
   b. `tabadmin restart`

INSTALLING TABMON

Run the installer (**make sure to run as administrator**).
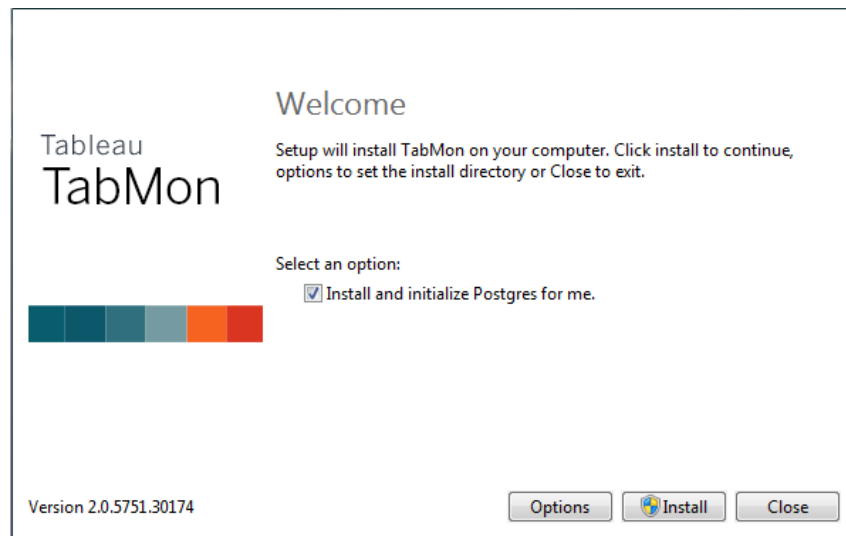
To install PostgreSQL along with your TabMon installation check "Install and initialize Postgres for me (recommended)". Select "Options" if you wish to modify your installation destination or modify your PostgreSQL database configuration parameters. **If you choose to install PostgreSQL separately from your TabMon installation, see appendix A.**

TabMon will install to your selected destination.

CONFIGURING TABMON

Basic configuration: Go to the install directory and modify TabMon.config (Located in Program Files (x86)\TabMon\Config\TabMon.config) as necessary.

The default TabMon.config file looks like the following:

```xml
<?xml version="1.0"?>
<!-- TabMon user settings: set these as required -->
<TabMonConfig xmlns="TabMon">
  <!-- Output mode.  "DB" for database, "CSV" for flat-file export -->
  <OutputMode value="db"/>
  <!-- Polling rate, in seconds -->
  <PollInterval value="30"/>
  <!-- Target cluster & host configuration -->
  <Clusters>
    <Cluster name="Primary">
      <Host name="localhost"/>
    </Cluster>
  </Clusters>
  <!-- Results output database Configuration-->
  <Database name="TabMon" type="Postgres">
    <Server host="localhost" port="5432"/>
    <User login="tabmon" password="password"/>
    <Table name="countersamples"/>
  </Database>
</TabMonConfig>
```

By default, this configuration will poll the performance counters on localhost every 30 seconds and write out the results to the specified database.

If you changed your database parameters during setup, you will need to update them here.
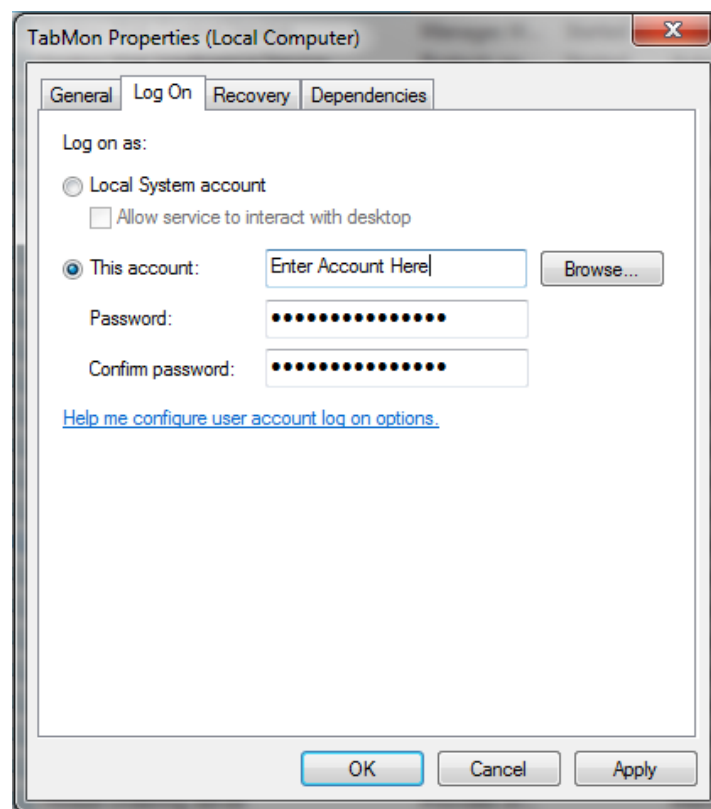
There are a few advanced options you can configure TabMon with:

1. **Export to CSV**
    a. To export to CSV file, replace "db" with "csv" in the line <OutputMode value="db"/>.
       This causes counter data to be stored in a CSV file in a folder called "Results" in the main
       TabMon directory.
2. **Remote Polling Setup**
    a. To poll data from a machine/machines other than your local host:
        i. Add a new <Host name=""> entry, with the name set to the network name or IP
           address of the target machine.
            1. You can nest hosts within a "Cluster" element to logically group them.
        ii. Go to Services and right click and select properties for "TabMon".
       iii. Make sure that the account for logon has admin privileges on the machine you
            would like to remotely connect to.

(Optional) Configure custom performance counters:

1. Configure Counters.config as desired. Add/remove counters as you wish.

USING TABMON

START & STOP THE TABMON SERVICE

1. Go to the TabMon group in the Start menu and run "Start TabMon" (**make sure to run as administrator**).
   a. TabMon can be instead be run as a console application by directly executing the file "TabMon.exe" from your installation directory.
2. To stop TabMon, go to the TabMon group in the Start menu and select "Stop TabMon".

CONNECT TO TABMON VIA TABLEAU

1. Open attached workbook.
2. Enter postgres database password set at installation.
   a. If you chose to output to CSV, connect to the csv files found in the \Results folder in your install directory.
3. Enter readonly credentials for your Tableau Server.

FAQ/TROUBLESHOOTING

1. **How to I connect to the TabMon workbook?**

To connect to your own server, please enter the following:

   a. Your readonly password for the data source: Server Auditing
   b. Your postgres DB name and password for you locally installed or remotely installed database (for the data source: TabMon)

2. **What do all these counters represent?**

The counters collected by the default Counters.config file represent a common set of performance metrics that would be of interest to an average server admin. Information on the Perfmon counters can be found at Microsoft's Perfmon counter reference at https://technet.microsoft.com/en-us/library/cc768048.aspx.

3. **My "%Processor Time" is over 100%; how is this possible?**

TabMon uses the Windows Perfmon counter "Process\% Processor Time" to calculate the amount of CPU used by individual processes. By the nature of this counter, for a single CPU (single core) the maximum value is 100%. For a multi-core system the maximum percentage is #cores * 100% (for example a 4 core system will have a max of 400%).

4. **TabMon is not collecting several of my JMX counters.**

Between version 9.0 and 9.1 of Tableau Server, we changed the internal names of many of our JMX counters. The Counters.config file that ships with TabMon supports the new 9.1+ naming conventions. To adjust for the old naming system, simply open Counters.config and do a find and replace all, substituting the term "webclientappservice" with "iwebclientappservice."

5. TabMon is dropping JMX counters after I restart Tableau Server.

This is normal. Restarting the server will cycle the open JMX ports and you will experience a temporary loss in connectivity for JMX counters. TabMon includes reconnection logic which will eventually automatically re-establish all of the connections.

6. What Processes/Services does TabMon monitor?

Tabmon tracks all Tableau Server related processes below (taken from http://onlinehelp.tableau.com/current/server/en-us/processes.htm):

Process	File Name	Purpose	Performance Characteristics
API Server	wgserver.exe	Handles REST API calls	Unless you are using REST APIs for critical business processes, this service can be down without impacting the overall health of Tableau Server.
Application Server	vizportal.exe	Handles the web application, supports browsing and searching	Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.
Backgrounder	backgrounder.exe	Executes server tasks, including extract refreshes, 'Run Now' tasks, and tasks initiated from tabcmd	A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.
Cache Server	redis-server.exe	Query cache	A query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and API server and application server to a lesser extent) make cache requests to the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.
Cluster Controller	clustercontroller.exe	Responsible for monitoring various components, detecting failures, and executing failover when needed	Included in the base install on every node.
Coordination Service	zookeeper.exe	In distributed installations, responsible for ensuring there is a quorum for making decisions during failover	Included in the base install on every node.
Data Engine	tdeserver64.exe tdeserver.exe (32-bit)	Stores data extracts and answers queries	The data engine's workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them. Memory consumption is primarily based on the size of the data extracts being loaded. The 64-bit binary is used as the default on 64-bit operating systems, even if 32-bit Tableau Server is installed. The data engine is multi-threaded to handle multiple requests at a time. Under high load it can consume CPU, I/O, and network resources, all of which can be a performance bottleneck under load. At high load, a single instance of the data engine can consume all CPU resources to process requests.

Process	File Name	Purpose	Performance Characteristics
Data Server	dataserver.exe	Manages connections to Tableau Server data sources	Because it's a proxy, it's normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.
File Store	filestore.exe	Automatically replicates extracts across data engine nodes	Installed with data engine (cannot be installed separately). A file store process will always be present if there are one or more data engine processes installed.
Repository	postgres.exe	Tableau Server database, stores workbook and user metadata	Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository.
Search & Browse	searchserver.exe	Handles fast search, filter, retrieval , and display of content metadata on the server	The process is memory bound first, and IO bound second. The amount of memory used scales with the amount of content (number of sites/projects/workbooks/datasources/views/users) on the server.
VizQL Server	vizqlserver.exe	Loads and renders views, computes and executes queries	Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.

7. How do I determine if I have system bottlenecks on my Server Instance?

Microsoft has some general guidelines for server health available in help documents and online articles. Refer to: https://technet.microsoft.com/en-us/magazine/2008.08.pulse.aspx

In general here are some common bottlenecks/general guidelines **(these are NOT an authoritative nor is this a definitive guide to the server health; actual results will vary)**:

    a. Free Disk Space: Keep above 15% for critical systems
    b. Memory (RAM): Keep under 80% for extended periods of time
    c. Processor %: Keep under 85% for extended periods of time
    d. Network: Keep under 70% of the interface's throughput (depending on network card)

8. What are cache hits/misses?

- A Cache hit is a successful lookup in a cache server.
- A Cache miss is an unsuccessful lookup in a cache server
- Generally speaking, an effective cache server has more hits than misses (aka a high Cache Hit Ratio)

9. What is an active session?

An active session in Tableau server is recorded as from the time a user logs on to the time they log out or time out.

10. What is Request Latency and what does it mean for me?

Request Latency is the time it takes for the host server to receive and process a request. Latency can be reduced by tweaking and upgrading computer hardware.

11. Where can I get additional support/feedback?

As an open source product, TabMon is not officially supported by Tableau and is a completely community supported effort. Please refer to the GitHub page for TabMon here: http://github.com/tableau/TabMon

MANUALLY INSTALLING & CONFIGURING POSTGRES

Follow the steps below to manually install PostgreSQL on your system. If you already have a PostgreSQL server and just need to initialize the TabMon results database, skip to step 3.

1. Install Postgres
   a. http://www.enterprisedb.com/products-services-training/pgdownload#windows
2. Setup superuser during setup
   a. Username: tabmon
   b. Password: your choice
   c. Port: 5432
3. Open PGadminIII
   a. Right click your postgres server and go to add databases
   b. Input the following parameters
      i. Owner = tabmon
      ii. Encoding = UTF8
      iii. Tablespace = pg_default
      iv. Collation = English_United States.1252
      v. Character type = English_United States.1252
      vi. Connection Limit = -1
   c. Under privileges
      i. Give all privileges to tabmon
      ii. Revoke all privileges from public

UNINSTALLING TABMON & POSTGRES

TabMon can be removed by simply uninstalling it through the Windows "Add/Remove Programs" interface. Application log files and CSV results are left behind and must be manually deleted.

However, in order to preserve your data, this uninstall does not automatically remove Postgres.

To remove Postgres:

1. Open up a console window as administrator.
2. Enter the following commands to remove the service:
   ```
   a. sc stop TabMon-Postgres
   b. sc delete TabMon-Postgres
   ```
3. Delete the folder where you installed Postgres (default is C:\Postgres).