# TABMON USER GUIDE

## BEFORE YOU BEGIN

### SYSTEM REQUIREMENTS

In order to install TabMon on your system please ensure 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++ Redistributable Packages for VS 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 Tableau Server "readonly" user:**

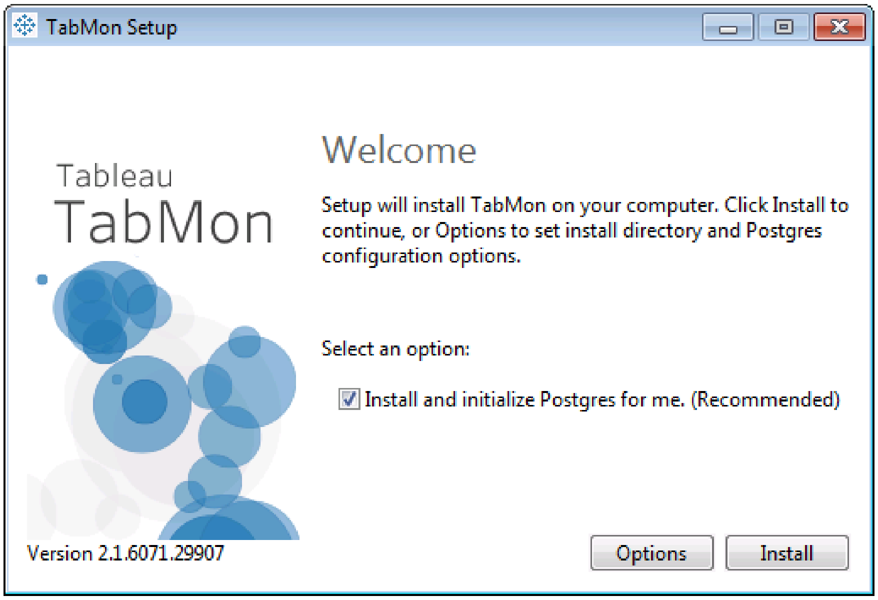
This step is only required if you want to use the "What's Going On?" dashboard in the provided sample workbook. TabMon does not actually utilize this user account in any way.

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

1. Run the installer (**make sure to run as administrator**).



TabMon Setup

Tableau TabMon

Welcome

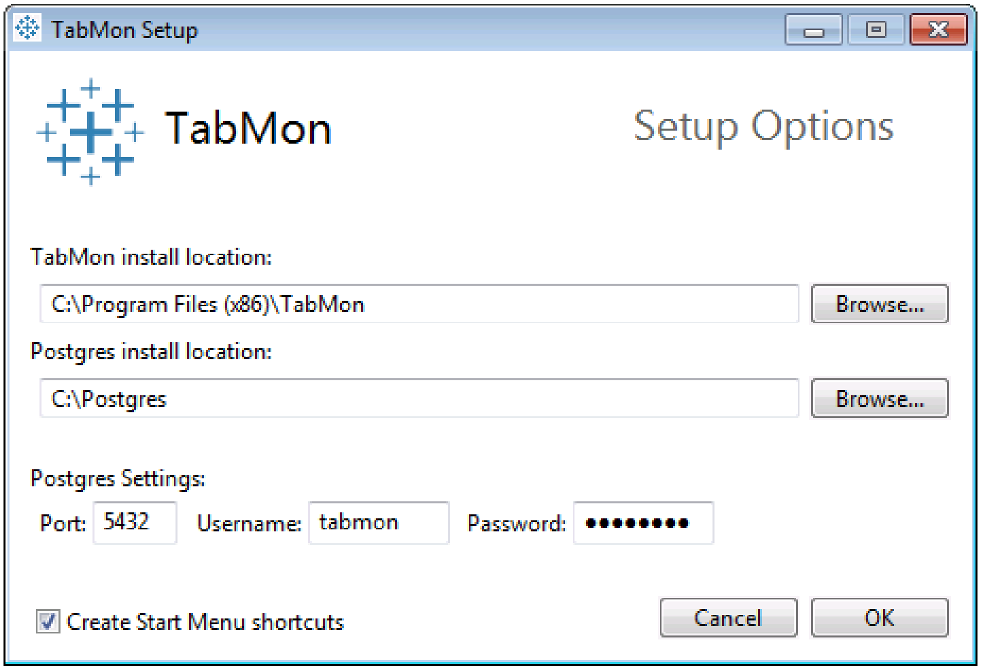
Setup will install TabMon on your computer. Click Install to continue, or Options to set install directory and Postgres configuration options.

Select an option:

☑ Install and initialize Postgres for me. (Recommended)

Version 2.1.6071.29907

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



TabMon Setup

Setup Options

TabMon install location: C:\Program Files (x86)\TabMon

Postgres install location: C:\Postgres

Postgres Settings:

Port: 5432 Username: tabmon Password: •••••••••

☑ Create Start Menu shortcuts

4. Click OK and TabMon will install to your selected destination.

## UPGRADING FROM A PREVIOUS VERSION OF TABMON

The TabMon installer will handle most of the upgrade work for you, but during the upgrade your TabMon.config and Counters.config files will be overwritten. Please follow the following steps for a smooth upgrade experience:

- Back up your TabMon.config and Counters.config files.
- Run the TabMon installer. Make sure to deselect “Install and initialize Postgres for Me”.
- Open up the TabMon.config file from the newly installed version and update the database and host options to match your backup copy. Note that the schema of this config can change between versions, so you shouldn’t just overwrite the full contents of TabMon.config with your backup, as there may be new options that are required.
- Ensure you are using the correct Counters.config and Sample Workbook files for your Tableau Server version. (*See “Configuring TabMon” below*)
- Fire up TabMon!

# CONFIGURING TABMON

## BASIC CONFIGURATION OPTIONS

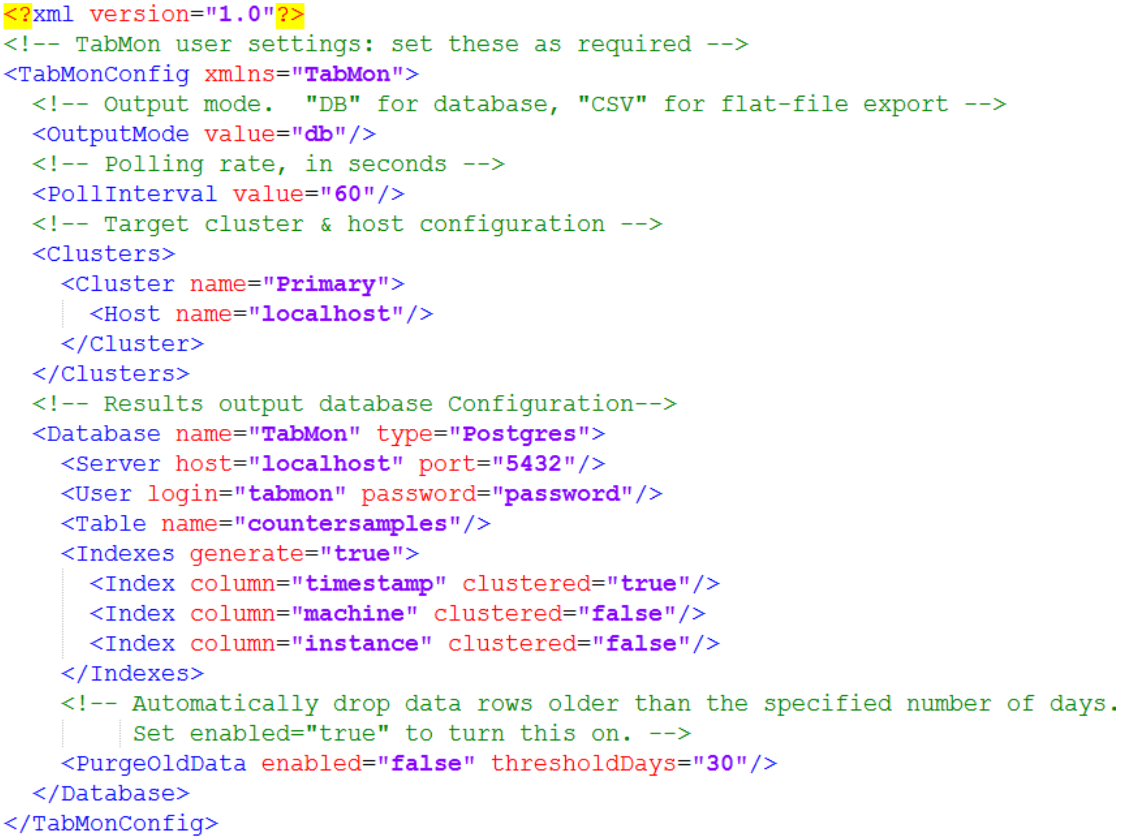
TabMon utilizes two main application config files, located in the “Config” directory in the TabMon install location:

- **TabMon.config** – Governs which machines will be monitored, how results will be written, and how often that performance will be sampled.
- **Counters.config** – Specifies the set of performance counters that will be monitored.

### TABMON.CONFIG

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

The default TabMon.config file looks like the following:



```
<?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="60"/>
  <!-- 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"/>
    <Indexes generate="true">
      <Index column="timestamp" clustered="true"/>
      <Index column="machine" clustered="false"/>
      <Index column="instance" clustered="false"/>
    </Indexes>
    <!-- Automatically drop data rows older than the specified number of days.
         Set enabled="true" to turn this on. -->
    <PurgeOldData enabled="false" thresholdDays="30"/>
  </Database>
</TabMonConfig>
```

By default, this configuration will poll the performance counters on localhost every 60 seconds and write out the results to the specified database.

**NOTE:** If you changed your database parameters during setup, you will need to update them here.

### COUNTERS.CONFIG

The default Counters.config file comes with a good starter set of counters for monitoring Tableau Server 9.3 & 10.0. If you are using TabMon to monitor an older version of Tableau, you will need to overwrite this with the Counters.config file specific to your version, these are kept in the "Resources\Configurations" folder in the TabMon install directory. Note that if you switch your Counters.config file in this way, you should also switch to the appropriate version of the Sample Workbook (found in \Resources\Sample Workbooks).

Feel free to add or remove counters from Counters.config to suit your monitoring needs!

## ADVANCED CONFIGURATION OPTIONS

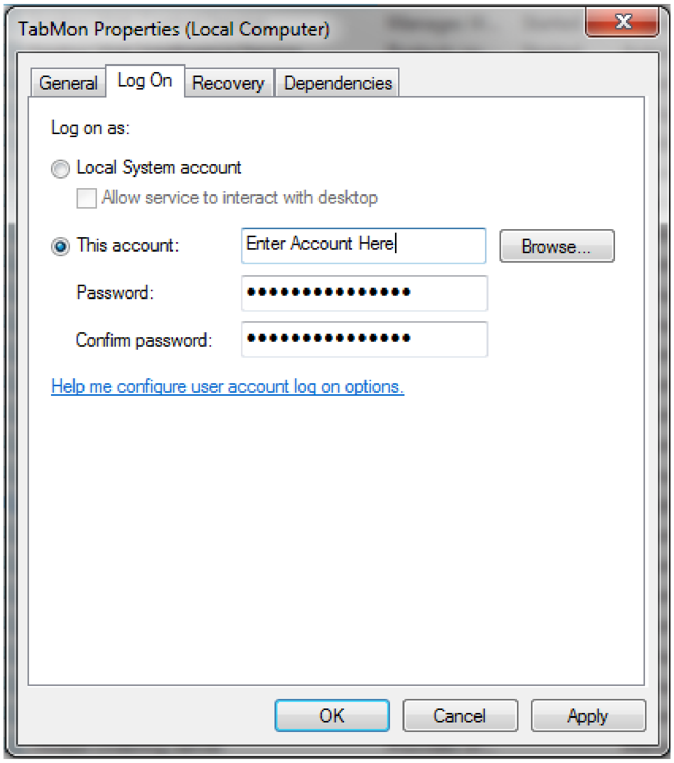
There are a few optional advanced options you can configure TabMon with:

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**

    To poll data from a machine/machines other than your local host:

    a. Add a new <Host name=""> entry, with the name set to the network name or IP address of the target machine. You can nest multiple hosts within a "Cluster" element to logically group them.
    b. Ensure that the "Remote Registry" and "Performance Logs & Alerts" Windows services are running and set to Start Mode: Automatic on all remote machines.
    c. On the TabMon machine, go to Services and right click and select properties for "TabMon". Make sure that the account for logon has privileges on the machine you would like to remotely connect to. Specifically, the service account must:
        i. On each server you wish to remotely monitor: have admin permissions or be a member of the "Performance Monitor Users" and "Performance Log Users" domain group
        ii. On the machine where TabMon is installed: have permissions to run as a service and write to the TabMon install directory.



TabMon Properties (Local Computer)

Log on as:

○ Local System account

□ Allow service to interact with desktop

● This account: Enter Account Here

Password: •••••••••••••••

Confirm password: •••••••••••••••

Help me configure user account log on options.

2. **Set a Data Retention Threshold:**

TabMon generates a lot of data, especially when using an aggressive polling rate. To help manage this, when using TabMon to output to database, you can set a data retention threshold so that old data is automatically dropped.

To set this, open up TabMon.config and find the following line:

```
<PurgeOldData enabled="false" thresholdDays="30"/>
```

Change the *enabled* attribute from **false** to **true** and set the *thresholdDays* attribute to the number of days beyond which you want data to be dropped. These changes will take effect on restart.

## 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 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” (**make sure to run as administrator**).

## CONNECT TO TABMON VIA TABLEAU

1. Open the sample workbook.
2. Enter the Postgres database password you 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 the read-only credentials for your Tableau Server when prompted.

## FAQ/TROUBLESHOOTING

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

The TabMon workbook is included in the "Resources\Sample Workbooks" folder in the TabMon install directory. Note that you should make sure to use the version of the Sample Workbook that matches the Tableau Server version. The SampleWorkbook is meant to fit on top of the schema set in the Counters.config file, so make sure you are using the correct Counters.config file for your version as well!

To connect the workbook to your own server, open the workbook and enter the following:

a. Your read-only 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. The TabMon service won't start.

If TabMon is failing to start, it is likely due to a permissions issue. The most common cause of this is that the user trying to start TabMon does not have admin privileges, or else the account the TabMon service is running as does not have sufficient permissions. Remember that if you are using Remote Polling, then TabMon must be running as a user with appropriate permissions on all of the remote machines.

If you've double-checked all relevant permissions and TabMon still isn't starting, a good way to get a clue is to check the most recent event from the application logs – see C:\Program Files (x86)\TabMon\Logs\.

### 5. I'm seeing tons of counter polling failures!

Make sure you are using the proper Counters.config file for the version of Tableau Server that you are monitoring. See the "Configuring TabMon" section for more information on this. If you're using the proper Counters.config file and still several failures, checking the logs can provide insight.

It is worth noting that systems can be very different and the machine(s) you are monitoring may simply not have certain counters.

### 6. 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.

### 7. Help! TabMon is generating way too much data!

By default, TabMon samples all performance counters every minute and keeps that data around forever. Over the long term, this can amount to a lot of data!

There are a couple of ways to address this:

1) If you are using TabMon to write to a database, you can set a data retention threshold so that old/expired data is automatically dropped. See the "Advanced Configuration Options – Set a Data Retention Threshold" section in this guide for details on how to do this.
2) You can set the poll interval to a lower frequency so that less data is generated. You can do this by updating *PollInterval* in TabMon.config.
3) You can update Counters.config to remove performance counters that you don't care about, in order to reduce data volume.

### 8. 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<br>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 request 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. |
| **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 | Normally consumes few resources. It can become a |

| | | | |
|---|---|---|---|
| | | workbook and user metadata | 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. |

### 9. 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)

### 10. 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)

### 11. 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.

### 12. Can I use TabMon to monitor hosts that aren’t running Tableau?

Absolutely! Just edit your Counters.config file to remove all of the MBean counters as well as the PerfMon counters that specifically relate to Tableau processes.

### 13. Where can I get additional support/feedback?

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

# APPENDIX A: ADDITIONAL INSTALLATION INSTRUCTIONS

## 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. Download Postgres
    a. http://www.enterprisedb.com/products-services-training/pgdownload#windows
2. Run the installer & input the following options into the setup wizard:
    a. Superuser password: your choice
    b. Port: 5432
    c. Locale: English, United States
    d. If prompted to launch StackBuilder, choose no.
3. Open PGadminIII
    a. Connect to your Postgres server using the postgres login account.
    b. Right click on Login Roles and choose New Login Role
    c. Enter the following parameters:
        i. Role name: tabmon
        ii. Definition: Enter a password
        iii. Role privileges: Select "Can create databases"
    d. Right click on Databases and choose New Database
    e. Enter the following parameters:
        i. Name = TabMon
        ii. Owner = tabmon
        iii. Encoding = UTF8
        iv. Tablespace = pg_default
        v. Collation = English_United States.1252
        vi. Character type = English_United States.1252
        vii. Connection Limit = -1
        viii. Under privileges:
            1. Give all privileges to tabmon
            2. Revoke all privileges from public
4. Update TabMon.config
    a. Open up C:\Program Files (x86)\TabMon\Config\TabMon.config and change the configured "tabmon" database user password to whatever you set in step 3.

**Note that by default, Postgres does not allow any remote connections**. If you want to connect to Postgres from a remote machine, you will need to add a pg_hba.conf entry to enable it. See instructions for this at: https://www.postgresql.org/docs/current/static/auth-pg-hba-conf.html.

## 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 fully 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).