

TIBCO® Messaging Manager

User Guide

Version 4.0.0 | March 2025

Contents

Contents	2
About this Product	4
Getting Started	5
msgmx Command Summary	7
Feature Reference	8
Window Layout	8
Syntax	9
Help	10
Autocompletion	10
MSGMX Commands	11
!	12
/	12
#	12
exit	12
help	13
manage	14
msgmx	14
Error Checking	18
Logging	19
Command History	20
External Shell Commands and Environment Variables	20
Colors	21
Monitoring Dashboards	22
Getting-Started	22

Commands	23
TIBCO Documentation and Support Services	28
Updated Resources on TIBCO Community	28
Legal and Third-Party Notices	30

About this Product

TIBCO® is proud to announce the latest release of this TIBCO® Messaging Manager software component.

This release is the latest in a long history of TIBCO products that leverage the power of Information Bus® technology to enable truly event-driven IT environments. To find out more about how TIBCO Messaging Manager software and other TIBCO products are powered by TIB® technology, please visit us at www.tibco.com.

TIBCO Messaging Manager software is part of TIBCO Messaging.

Product Overview

TIBCO Messaging Manager (MSGMX) provides an intelligent, predictive, and auto-completing command-line interface for messaging/streaming systems such as TIBCO Enterprise Message Service (EMS), Apache Kafka, and Apache Pulsar. MSGMX runs in any standard console window or terminal emulator.

Getting Started

After installation, you can quickly verify your MSGMX installation by performing the following tasks.

Enterprise Message Service (EMS) Considerations

If MSGMX is used to manage EMS, EMS must be installed. See the [TIBCO Enterprise Message Service Installation Guide](#) for details.

Environment Settings

If MSGMX is used to manage Apache Kafka or EMS, you need to set environment variables before starting MSGMX. Refer to the corresponding command reference document, Getting Started, for guidance and examples.

Python Virtual Environment

The Python virtual environment (venv) isolates the Python dependencies of MSGMX from other globally installed Python packages that may exist on your system. The Python virtual environment is automatically created during the installation process and is used when running MSGMX.

Starting MSGMX

Perform the following Procedure steps.

Procedure

1. Confirm that the messaging system is installed.
2. In a local console window, either add the directory containing the MSGMX executable file to your PATH or navigate to the directory containing the executable. See the following examples.

Default Unix location:

```
cd /opt/tibco/msgmx/bin
```

Default Windows location:

```
cd C:\tibco\msgmx\bin
```

3. Launch the MSGMX program using the `msgmx` command.

Linux and macOS:

```
./msgmx
```

Windows:

```
msgmx.bat
```

4. Use the `manage` command to select the component to manage.
For example, to manage Apache Kafka:

```
manage kafka
```

To manage EMS:

```
manage ems
```

5. At the MSGMX prompt, initiate a connection to the system to be managed.
6. To continue, see the appropriate command reference manual.
7. Exit MSGMX by using the `exit` command or by selecting `Ctrl+D`.

msgmx Command Summary

Launch the TIBCO Messaging Manager (MSGMX) program.

Synopsis

```
msgmx [-h] [-i|-c CONFIG] [-v] [-l LOG_FILE] [-f FILE]
```

Optional Arguments

-h, --help

Show this help message and exit.

-i, --ignore-config

Ignore the user-saved default configuration file and run with all default values.

-c CONFIG, --config CONFIG

Start MSGMX with the specified configuration file.

-v, --version

Display the current version of MSGMX.

-f FILE, --file FILE

Execute commands from FILE and exit at EOF.

-l LOG_FILE, --log LOG_FILE

Log all output to specified file. LOG_FILE will be created if it does not exist, and overwritten if it exists.

Additional Option

--create-config CONFIG

Start MSGMX with all default values, and save those values to the file given by CONFIG.

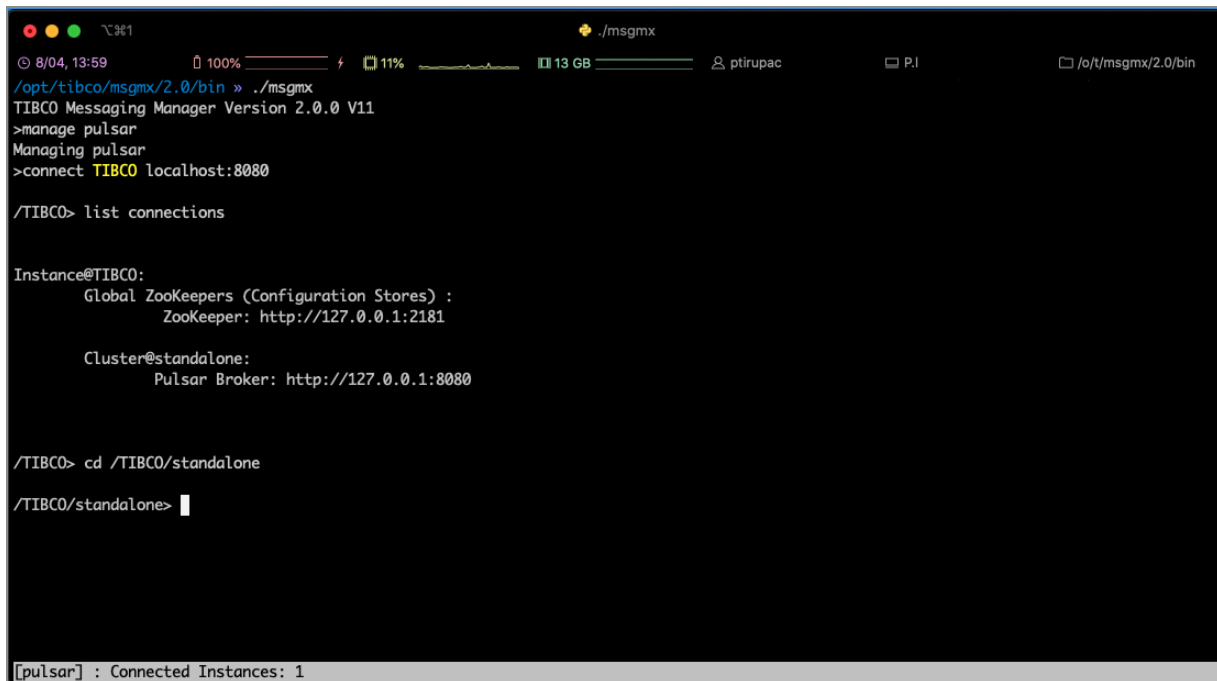
Feature Reference

This chapter provides an overview of the features and functions of TIBCO Messaging Manager. These features apply to any supported product administrative interface. There may be some variation between products, platforms, or command windows, and where applicable, these will be noted or called out as appropriate.

Window Layout

The TIBCO Messaging Manager leverages the advanced features of modern terminal emulators to display specific information in the terminal window.

Pulsar console window example:



```
8/04, 13:59 100% 11% 13 GB ptirupac P.I /o/t/msgmx/2.0/bin
/opt/tibco/msgmx/2.0/bin » ./msgmx
TIBCO Messaging Manager Version 2.0.0 V11
>manage pulsar
Managing pulsar
>connect TIBCO localhost:8080

/TIBCO> list connections

Instance@TIBCO:
  Global ZooKeepers (Configuration Stores) :
    ZooKeeper: http://127.0.0.1:2181

  Cluster@standalone:
    Pulsar Broker: http://127.0.0.1:8080

/TIBCO> cd /TIBCO/standalone
/TIBCO/standalone>

[pulsar] : Connected Instances: 1
```

Command Area

As in a standard console window interface, the line with the cursor is the active line. But unlike a typical console window, MSGMX displays drop-down lists for command and option

auto-completion with optional help text about each potential selection as you build the command line.

In the following Kafka example, the **List** command auto-completes and "topics" is selected:

```
Cluster@ZooKeeper:> list topics_
bootstrap-servers (deprecated)
brokers           List all currently connected Kafka Brokers.
zookeepers        List all currently connected Apache ZooKeepers.
topics            List all topics.
acls              list all ACLs.
connections       List all connections.
```

Bottom Status Bar

The bottom status bar displays one of several status messages providing key information about the currently managed system.

To cycle through the different status bar displays, select F1.

The status bar display automatically refreshes periodically, but you can refresh the live status display at any time by selecting F5. (The frequency of the status updates can be configured via the [msgmx](#) command.)

Temporary Error Messages

If a command string being entered contains an error, MSGMX displays the error message and correction suggestions below the command-line area above the status bar. Error messages are discussed in [Error Checking](#).

Syntax

An MSGMX command takes the following general form:

```
verb noun name (option option-value|flag)
```

As you type a command, MSGMX offers drop-down selections for verbs, nouns, names, and options to ensure valid entries and reduce keystrokes. For details, see [Autocompletion](#).

Help

TIBCO Messaging Manager offers a concise, context-sensitive help feature. You can view help information in two ways: via the `help` command (below), and within the autocompletion drop-down feature (see [Autocompletion](#)).

help Command

Enter `help` followed by a command name. You see a summary of syntax and usage examples for this command, followed by a brief description.

Kafka help command example:

```
Cluster@ZooKeeper:> help list
Usage:
  list bootstrap-servers
  list brokers
  list zookeepers
  list topics
  list acls
  list connections

Cluster@ZooKeeper:> _
```

Autocompletion

As you enter commands, options, or flags, MSGMX offers drop-down menu text-completion choices pertinent to what you are entering.

When choices appear, use the up/down arrow keys or the tab key to move to a specific choice. Then use the space bar to add it to your command line and advance to the next option or flag. (You can also manually enter items on the command line by just continuing to type.) If a specific choice does not appear in the drop-down, then that choice is not valid for that command or place in the command line and MSGMX will not allow that command, option, or flag to be entered.

Kafka autocomplete drop-down example:

```
Cluster@ZooKeeper:> create
topic          Add a new topic.
topic-acl      Topic ACL to add.
cluster-acl    Cluster ACL to add.
group-acl      Add a group ACL.
delegation-token-acl  Add a delegation token ACL.
```


Command completion is based upon characters you have already typed, and as you continue to enter characters, the options available will limit themselves to match your input. For example, if 10 total options are available but only three of these start with the letter "a", when you type "a" you see only those three matching options as drop-down selections.

MSGMX automatically displays a matching previous command, if any, based on your command history. You can use the right arrow to complete an entire command based on a previous entry. You can use the right and left arrows to move through the command and change it without having to re-type any unchanged portions.

When entering a file path, if you enter a "/" character, MSGMX suggests valid filenames and/or directory names that could follow the "/".


Autocompletion Help

When an autocompletion drop-down appears, by default you see two columns of text: the actual value that would be included as part of the autocompletion, and some help text associated with each potential selection. The left column contains the actual command, option, or flag to be inserted at the command prompt, and the right column (with a darker-shaded background) provides a brief description of the command or option.

 **Note:** Autocompletion can be turned off with `msgmx set prediction off`

MSGMX Commands

The MSGMX command set includes commands related to the product it is supporting (e.g., Apache Kafka, Apache Pulsar, etc.) and its configuration, as well as commands native to MSGMX that control how MSGMX behaves and operates.

 **Note:** Commands are asynchronous and take a brief time to execute. You may not notice a delay if you are working interactively. If the commands are in a script, you can use a sleep command.

!

Execute a shell command

Synopsis

!<ShellCommand>

Property

None.

/

Search the command history.

Synopsis

/<SearchString>

Property

None.

#

Add a comment to the command history.

Synopsis

#<CommentText>

Property

None.

exit

Exit MSGMX.

An EOF (End Of File), typically Ctrl+D when used interactively, will also exit MSGMX .

Synopsis

exit

Property

None.

help

Display help information for any command.

Enter the help command or "?" to display information about a specific command.

Synopsis

help <command-name>

? <command-name>

? is an alias for help.

Required Property

Command name to provide help for.

Examples

```
Cluster@ZooKeeper:> help msgmx
Usage:
  msgmx source <File>
  msgmx load
  msgmx save
  msgmx set prediction < on|off >
  msgmx set summary < on|off >
  msgmx set timeout <Int>
  msgmx set status < on|off >
  msgmx set history-depth <Int>
  msgmx set echo-command < on|off >
  msgmx set color <ColorMode>
  msgmx set source-error <String>
  msgmx set cache-update-time <Int>
  msgmx set status-refresh <Int>
  msgmx set max-threads <Int>
  msgmx show all
  msgmx show prediction
  msgmx show summary

...Content truncated...
```

manage

Select the component for MSGMX to manage.

Synopsis

```
manage <component>
```

Required Parameters

Component name:

- **cli** : Manage aspects of, and limits available commands to, the MSGMX component.
- **kafka** : Manage an Apache Kafka installation. This mode also includes all commands available in the `cli` option.
- **pulsar**: Manage an Apache Pulsar installation. This mode also includes all commands available in the `cli` option.
- **ems**: Manage an EMS installation. This mode also includes all commands available in the `cli` option.

Examples

```
>manage kafka  
>manage cli  
>manage pulsar  
>manage ems
```

msgmx

Display or change MSGMX settings.

You can execute any of the MSGMX commands at any time, even if not connected to any particular deployment.

Synopsis

```
msgmx history flush|export [<Filename>]  
msgmx set <set option>
```

```
msgmx show <show option>
msgmx save|load [<Filename>]
msgmx source <Filename>
msgmx log file <Filename> [ on|off|show [ contents ] ]
msgmx clear> [clear the screen]
```

Properties for msgmx

Assign values to MSGMX setting properties.

history

Clear or export the command history.

log

Activate/deactivate logging and/or change the log file destination (see [Properties for msgmx log](#) which follows).

clear

Clear the screen.

set

Set specified operational properties (see [Properties for msgmx set](#) which follows).

show

Display the value of the specified setting (see [Properties for msgmx show](#) which follows).

save

Save the current configurations to either the current configuration file or the specified file.

load

Load the last-saved configuration settings from either the current configuration file or the specified file.

source

Executes the commands in the specified file sequentially and then return to the prompt.

Properties for msgmx log

Assign values to MSGMX log-setting properties with the `msgmx log` command.

<on|off>

Start or stop sending information to the log file.

file <filename|path/filename>

Change the file that collects logging information. Changing the log file name closes the previous file and opens a new file. If the file does not exist, MSGMX creates it. If the new file cannot be created or opened, the `msgmx` command fails.

show

Display the log file path.

show contents

Display the log file contents.

Properties for msgmx set

Assign values to MSGMX setting properties with the `msgmx set` command.

cache-update-time <Int>

MSGMX caches system status and command history for the status bar and for predictive autocompletion at the command prompt. MSGMX updates the cache at this configurable interval in seconds (default 3).

color < LightMode|DarkMode|LightMonochromeMode|DarkMonochromeMode >

Set MSGMX to use dark text on a light-colored background or vice-versa. You can also choose between monochrome and color palettes. For `LightMode` and `DarkMode`, MSGMX uses color to differentiate commands, operands, and options.

echo-command <on|off>

Display the command to be executed with a preceding "+" prior to execution.

history-depth <Int>

Set the number of commands saved as history.

max-threads <Int>

MSGMX can execute different commands in different threads when called in a script. This property sets the maximum number of threads MSGMX will utilize.

prediction <on|off>

Show command, option, and flag drop-down prediction menu based on characters already entered as you type.

source-error <abort|warn|ignore>

If an error is encountered while running an `msgmx source` command, or when or using the `-f / --file` option while starting up MSGMX, the type of response is determined by this setting.

- `abort` (default) displays the line causing the error and exits the command.
- `warn` displays the line causing the error and continues to execute commands.
- `ignore` silently continues to execute commands.

status <on|off>

Toggle display of the status bar.

status-refresh <Int>

Status bar refresh rate.



Caution: The `status-refresh` setting is deprecated. It will be removed in a future release.

summary <on|off>

Show brief help text for each drop-down prediction menu item. Turning the summary off does not affect the display or use of the drop-down prediction menu selections.

timeout <Int>

Set command timeout in seconds. This is the time MSGMX waits for a response after executing a command against the connected system being managed.

Properties for `msgmx show`

Determine what settings or information to display with the `msgmx show` command.

- `all`

- cache-update-time
- color
- echo-command
- history
- history-depth
- live-status
- max-threads
- prediction
- source-error
- status
- status-refresh (deprecated)
- summary
- timeout
- version

Error Checking

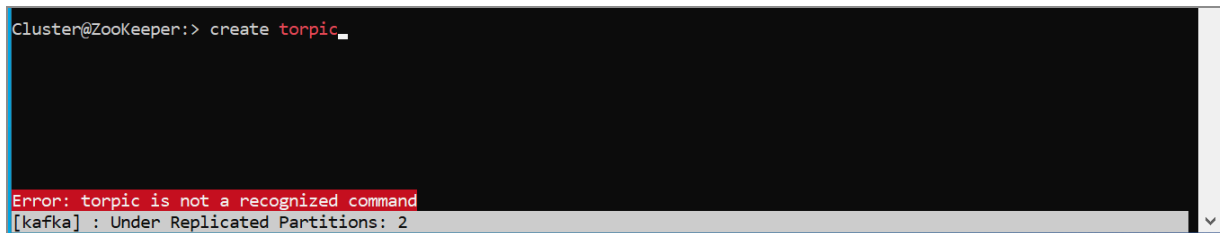
MSGMX tracks command prompt input and alerts you at the first invalid term entered.

A command or argument is considered an error or invalid if:

- The command, option, or flag is unknown.
- An entered option, argument, or flag cannot be used with the command or option it follows.
- The value specified is out of range or not of the appropriate type (e.g., numeric or alphabetic) for the option with which it is associated.

If the syntax of the entered command-line entry is invalid, MSGMX prevents you from executing the command or option and displays an error message at the bottom of the console window above the status bar explaining why the entry is in error. The error will need to be corrected before you can press <ENTER> and execute the command.

Syntax error example



```
Cluster@ZooKeeper:> create torpic_  
  
Error: torpic is not a recognized command  
[kafka] : Under Replicated Partitions: 2
```

Logging

MSGMX provides a logging option to capture all commands and their associated output.

In addition to using the [msgmx](#) command, logging can also be enabled at startup by specifying the log file location path.

```
msgmx --log logfile
```

The log file follows a specific format:

- Timestamp for each executed command (starts with '^ ' and ends with a newline).
- User-executed-command for each executed command (starts with '>' and ends with a newline).
- All other MSGMX command and output (all command output is also displayed in the MSGMX command window).

i Note: Interactive window output (status bar, dropdown menus, predictions, etc.) is not logged.

Example

```
^ 09/12/2019 15:18:30  
>list topics  
topic1  
topic2  
topic3  
topicname
```

```
^ 09/12/2019 15:20:10  
>create topic topic6 partitions 1 replication-factor 1  
Created topic topic6.
```

Command History

MSGMX remembers previous commands entered and you can use this capability to recall previous commands and rerun them as-is or edit them before re-execution.

You can access your command history in the following ways:

- Navigate through the history using up / down arrow keys.
- Type ' / ' to search through the history. Characters typed after the '/' are used as search patterns, and you see a drop-down menu of matching terms. The search is done in an unanchored manner, so any commands in your history containing the search string in any location within the command will be displayed.
- Display all retained history commands (e.g., `msgmx show history`).
- Use regular expressions to narrow down selections from the history drop-down.

You can control your history-related settings with the following commands:

- Set the number of saved commands (e.g., `msgmx set history-depth 200`).
- Export stored history commands (e.g., `msgmx history export <FilePath>`).
- Clear the history contents (e.g., `msgmx history flush`).

External Shell Commands and Environment Variables

While running MSGMX you can execute shell commands without exiting MSGMX. You can also reference environment variables from within MSGMX.

To execute a shell command from the MSGMX prompt, use a '!' character in front of any string. The entire string following the '!' (including all spaces as entered) will be passed to the shell for execution.

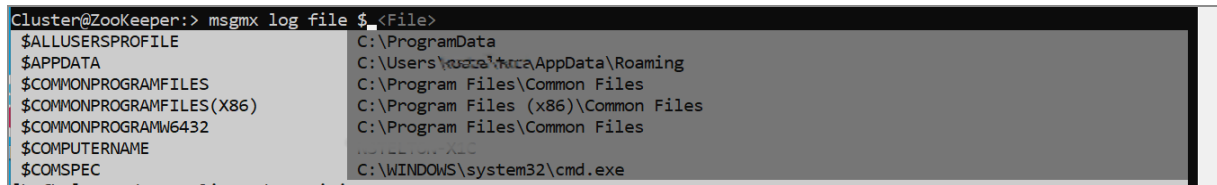
For example:

```
cluster@myCluster> !more logfile
```

To access a shell environment variable from the MSGMX prompt, use a '\$' character in front of the variable name.

Typing \$ alone produces a drop-down of all the environment variables available with their respective values in the summary.

For example:



Colors

Text and background may appear in different colors depending on context. Using the `msgmx set color` command, you can select from four different modes.

DarkMode

Default. Light text with color variations on a dark background.

DarkMonochromeMode

Black text on a white background.

LightMode

Dark text with color variations on a white background.

LightMonochromeMode

White text on a black background.

Monitoring Dashboards

TIBCO provides a set of monitoring dashboards in a public Github repository, for various products. These dashboards are based on Grafana, using a Prometheus data source.

Using MSGMX, these product dashboards can be

- Fetched from Github into a local storage directory
- Pushed from a local storage directory into Grafana.

Pushing a product's dashboards involves:

- Creating or updating the required Grafana data source (named TIBCO Messaging Monitoring) with the specified Prometheus data source options.
- Creating a Grafana dashboard folder, if it does not already exist.
- Creating or updating any dashboards within the product.

The normal workflow involves:

- Ensure any Prometheus options are set properly for the environment.
- Ensure any Grafana options are set properly for the environment.
- Ensure the local storage directory exists and is writeable.
- Fetch one or more products from Github into local-storage.
- Push one or more products from local-storage into Grafana.

Getting-Started

Follow the normal steps to start msgmx

manage msgmon

Commands

Github Options

These options may be set via the `set github` command, or reset via the `reset github` command.

Parameter	Description	Default Value
<code>jwt-token</code>	JSON Web Token to authenticate to Github	-
<code>name</code>	Github repository name	- (subject to final repo location)
<code>org</code>	Github repository organization	- (subject to final repo location)
<code>password</code>	Password used to authenticate to Github	-
<code>path</code>	Path within Github repository	- (subject to final repo location)
<code>token</code>	Token used to authenticate to Github	-
<code>username</code>	Username used to authenticate to Github	-

Prometheus options

These options may be set via the `set prometheus` command, or reset via the `reset prometheus` command.

Parameter	Description	Default Value
<code>authentication</code>	Authentication type. Possible values are none, basic, and oauth.	none
<code>client-authentication</code>	Controls whether Grafana must authenticate to Prometheus.	false
<code>client-</code>	Client certificate file, when client authentication is	-

Parameter	Description	Default Value
certificate	enabled.	
client-key	Client key file, when client authentication is enabled.	-
cookies	A list of cookie names to be forwarded from Grafana to Prometheus.	Empty
http-method	HTTP method used to issue queries to Prometheus (POST or GET).	POST
password	Password to authenticate to Prometheus when using basic authentication.	-
query-timeout	Maximum timeout in seconds for Prometheus queries to complete.	60
scrape-interval	Advisory scrape interval in seconds.	15
selfsigned-ca-certificate	Self-signed CA certificate when TLS communication to Prometheus is used.	-
server-name	Server name to be validated when TLS communication to Prometheus is used.	-
skip-validation	Skip validation of server certificate when TLS communication to Prometheus is used.	false
timeout	HTTP timeout in seconds.	60
type	Prometheus server type. Valid values are Prometheus, Cortex, Mimir, and Thanos.	Prometheus
url	Prometheus server URL.	-
username	Username to authenticate to Prometheus when using basic authentication.	-

Parameter	Description	Default Value
version	Prometheus version. The allowed values depend on the type value.	2.50.0

Grafana options

These options may be set via the `set grafana` command, or reset via the `reset grafana` command.

Parameter	Description	Default Value
org-id	Organization ID	-
password	Password used to authenticate to Grafana	-
timeout	Grafana request timeout	0 (no timeout)
token	Grafana Service Account Token to authenticate to Grafana	-
update-datasource	Controls whether the datasource is updated when dashboards are pushed to Grafana.	true
URL	Grafana URL.	-
username	Username used to authenticate to Grafana	-
verify	When using HTTPS, controls whether the host certificate is verified	true

fetch <product>

Fetch monitoring files for the named product (ems, ftl, kafka, pulsar) from Github and store them in the local storage directory.

`list available`

List the products available in Github.

`list fetched`

List the products fetched into the local storage directory.

`push <product>`

Push the files for the named product (ems, ftl, kafka, pulsar) from local storage into Grafana.

`reset github`

Reset one or more of the Github parameters to its default value.

`reset grafana`

Reset one or more of the Grafana parameters to its default value.

`reset local-storage`

Resets the local storage setting to blank.

`reset prometheus`

Reset one or more of the Prometheus parameters to its default value.

`set autosave`

Controls whether changes made to the configuration by the `set grafana`, `reset grafana`, `set local-storage`, `reset local-storage`, `set prometheus`, `reset prometheus`, `set github`, and `reset github` commands are automatically saved to the configuration file. `set autosave true` enables this, while `set autosave false` disables this action. The default is `false`.

`set github`

Sets one or more of the Github parameters.

`set grafana`

Sets one or more of the Grafana parameters.

`set local-storage <Directory>`

Sets the local storage directory parameter. The directory must exist and be writeable.

`set prometheus`

Sets one or more of the Prometheus parameters.

`show autosave`

Show the current state of autosave.

`show github`

Show the current Github parameter values.

`show grafana`

Show the current Grafana parameter values.

`show local-storage`

Show the current local storage directory.

`show prometheus`

Show the current Prometheus parameter values.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for this product is available on the [TIBCO® Messaging Manager Product Documentation](#) page:

Updated Resources on TIBCO Community

Supplemental resources are now distributed at the [TIBCO Messaging Community Wiki](#) in the Reference Info tab. You can always find the latest versions of these resources in that location.

Those resources include quick start guides, tutorials, and usage examples.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request

one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, TIB, Information Bus, TIB, FTL, eFTL, and TIBCO Enterprise Message Service are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file

for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.tibco.com/patents>.

Copyright © 2019-2025. Cloud Software Group, Inc. All Rights Reserved.