

TIBCO® Messaging Manager

Apache Kafka® Command Reference

Version 4.0.0 | March 2025

Contents

Contents	2
About this Product	5
Getting Started	6
MSGMX Apache Kafka Management and Commands	8
4lw (Apache ZooKeeper-specific)	10
connect	12
create	14
create cluster-acl	14
create delegation-token-acl	14
create group-acl	15
create topic-acl	15
create topic	16
delete	17
delete cluster-acl	17
delete delegation-token-acl	17
delete group-acl	18
delete topic	18
delete topic-acl	18
disconnect	20
exit	21
help	22

list	23
list acls	23
list bootstrap-servers	23
list brokers	23
list connections	24
list topics	24
list zookeepers	25
reset	26
reset broker	26
reset client	29
reset topic	29
reset user	31
set	32
set broker	32
set ips	36
set user	36
set topic	36
set zookeeper	38
show	39
show adk-script-version	39
show all-topics	39
show broker	40
show broker-api-version	40
show client	40
show log-dirs	41
show log-dump	41
show topic	41
show user	42
show zookeeper	42

Kerberos Authentication Notes	43
TIBCO Documentation and Support Services	44
Updated Resources on TIBCO Community	44
Legal and Third-Party Notices	46

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

Before using MSGMX to manage Apache Kafka, review the following information and refer to the MSGMX User Guide.

Environment Settings

Set the `KAFKA_HOME` environment variable so that it contains the directory in which Apache Kafka was installed.

Unix example:

```
export KAFKA_HOME=/opt/tibco/akd/core/<version_number>
```

Starting MSGMX to Manage Apache Kafka

Perform the following steps:

1. Confirm you have Apache Kafka installed and are running at least one accessible Apache Kafka Broker and one Apache ZooKeeper or Metadata Quorum. These services do not need to be running on the system MSGMX is installed on but must be accessible from it.
2. In a local console window, either add the directory containing the MSGMX executable file to your path, or change into the directory containing the executable. For example:

Default Unix location :

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

3. Launch the MSGMX program using the `msgmx` command.
4. Once you've started MSGMX, use the `MSGMX manage` command to select the component to manage. For example, to manage Apache Kafka:

```
manage kafka
```

5. At the MSGMX command prompt, initiate a connection to an Apache Kafka cluster using the `connect` command:

```
connect cluster1 localhost:2181
```



Note: 2181 is the default port for Apache ZooKeeper. Be sure to specify a host and port that is appropriate for the Apache ZooKeeper you are running. MSGMX does not officially support quorum as of this release.

When the connection is established, the MSGMX command prompt changes to include the name of the cluster you just connected to. Verify that the prompt changes, indicating that a connection has been established.

If MSGMX cannot connect using the information you provided, it displays an error message. Ensure that the target Apache Broker is running and is accessible at the specified host and port.

6. As a test command, enter:

```
list topics
```

You see a listing of configured topics (or a blank line if no topics are configured). Confirm that no error messages are displayed.

MSGMX Apache Kafka Management and Commands

TIBCO Messaging Manager provides a comprehensive and efficient command-line interface for administering Apache Kafka.

Bottom Status Bar

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

- The total number of under-replicated partitions at all configured Apache Kafka Brokers for the currently connected cluster. This is the default status bar display.
- Names of the Apache Kafka Brokers available for the currently connected cluster.
- Number of clusters to which you are connected.

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

Command Overview

The Apache Kafka commands supported by MSGMX have an intuitive correlation with native Apache Kafka command combinations and tasks while providing a flexible and assistive interface that frees you from the arcane syntax and limitations of using the native Apache Kafka scripts.

All MSGMX Apache Kafka commands start with a verb that states the action to be taken—create, delete, list, etc. These verbs are listed in this manual in alphabetical order for easy reference.

Commands have options or flags that can be required or optional. Options require an argument while flags do not. A required option or flag is a selection from a list of choices. Optional ones are shown in this document enclosed in square brackets, "[]". Free text entry is designated as <FreeTextEntry>.

i **Note:** For details on Apache Kafka as well as specific options or flags, see the [Apache Kafka documentation](#).

4lw (Apache ZooKeeper-specific)

Issue a ZooKeeper 4-letter-word command.

Use this command to access 4-letter-word ZooKeeper commands in a manner similar to issuing these commands via telnet to a ZooKeeper server.

By default 4-letter-word commands are disabled by ZooKeeper—you must enable them when configuring the ZooKeeper.

Example lines you can add to the `zookeeper.properties` file:

```
// Enable all 4lw commands  
4lw.commands.whitelist=*
```

```
// Enable only ruok, stat, conf, isro  
4lw.commands.whitelist=stat, ruok, conf, isro
```

Synopsis

4lw <command> [<host:port>]

Options

The following 4-letter-word ZooKeeper commands are supported by MSGMX:

- conf
- cons
- crst
- dump
- envi
- mntr
- ruok
- srst

- `srvr`
- `stat`
- `wchc`
- `wchp`
- `wchs`

For information on available Apache ZooKeeper 4-letter-word commands, see the [ZooKeeper Administrator's Guide, The Four Letter Words](#).

Examples

```
>4lw conf localhost:2181
```

```
Cluster@sample> 4lw srvr
```

connect

Connect to an Apache Kafka cluster.

Use connect to associate your MSGMX session with an Apache Kafka cluster containing one or more Apache ZooKeeper and/or Quorum nodes and Apache Kafka Brokers. To be able to do this, the target service must be running, and an accessible IP address must be supplied.

Synopsis

```
connect <ClusterName> <host:port> [ command-config <File> ]
```

Required Arguments

ClusterName

A name you assign while issuing the connect command. It exists only during the connect session and provides a quick named way to identify and access a specific cluster if you are managing multiple clusters in a single MSGMX session.

host

Specify the host with either a valid hostname or a network address in the form address:port.



Caution: Apache Kafka version 2.8.0 offers an early access Metadata Quorum, however, it is not complete and should not be used in production. For details, see [KIP-500: Replace ZooKeeper with a Self-Managed Metadata Quorum](#).

For Apache Kafka versions 2.7.0 and later, only one Apache ZooKeeper host argument is required, provided the Apache Zookeepers supporting the Kafka cluster are configured per Apache ZooKeeper Dynamic Reconfiguration practices. For more information, see the [Zookeeper Dynamic Reconfiguration](#) documentation. However, you can optionally provide additional host arguments for the other Apache ZooKeepers in the cluster, and MSGMX will try to connect to them in the order specified until it achieves a successful connection. In either case, once a connection has been made to a single Apache ZooKeeper, MSGMX can obtain the connection information for all of the other ZooKeeper nodes in that ZooKeeper cluster when using Apache Kafka 2.7.0 or later.

i Note: You cannot connect to Apache ZooKeepers from different clusters using the same connect command. Also, you cannot mix ZooKeeper and Broker host/IP addresses in the same connect command.

Optional Arguments

command-config <ConfigFilename>

For a bootstrap-server connection, you can specify a credential file to provide the credentials for access to a secure bootstrap-server. These configuration properties are automatically applied to all commands run on this bootstrap-server, and are unset when disconnected from the bootstrap-server or by connecting to another bootstrap-server. If you reconnect to this server later, you must respecify the credential file.

Examples

```
> connect myCluster1 localhost:2181
```

```
> connect myCluster2 localhost:2181 localhost:2182 localhost:2183  
command-config my_cred_file
```

See Also

[disconnect](#)

create

Create an object.

Use this command to create an object on the connected cluster. For Apache Kafka, you can create topics, ACLs for topics, clusters, groups, and delegation tokens.

Synopsis

```
create <CreateCommandArg> [CreateCommandArg Options]
```

create cluster-acl

```
create cluster-acl allow-principal|deny-principal <PrincipalType:name> [force|allow-host <String>|deny-host <String>|operation <String>]
```

Create a cluster ACL.

Example:

```
cluster@myCluster> create cluster-acl allow-principal type:Name
```

create delegation-token-acl

```
create delegation-token-acl <String> allow-principal|deny-principal <PrincipalType:name> [force|allow-host <String>|deny-host <String>|operation <String>]
```

Create a delegation token ACL.

Example:

```
cluster@myCluster> create delegation-token-acl aclName allow-principal
```

create group-acl

create group-acl <String> allow-principal|deny-principal <PrincipalType:name>
[force|allow-host <String>|deny-host <String>|operation <String>]

Create a group ACL.

Example:

```
cluster@myCluster> create group-acl 0 allow-principal type:Name
```

create topic-acl

An ACL is an access control list.

create topic-acl <TopicNames> allow-principal|deny-principal <PrincipalType:name>
[force|allow-host <String>|deny-host <String>|operation <String>]

Create an ACL for a specified topic or topics.

Example:

```
cluster@myCluster> create topic-acl topic2 allow-principal type:Name
```

create topic-acl <TopicNames> consumer allow-principal
<PrincipalType:principalName> [<options>]

Create a consumer-role ACL for a specified topic or topics.

Options include:

- allow-host <String> force group <String>
- allow-host <String> group <String> force [force]
- force allow-host <String> group <String>
- force group <String> allow-host <String>
- group <String> allow-host <String> [force]
- group <String> force [force|allow-host <String>]

Example:

```
cluster@myCluster> create topic-acl topic1 consumer allow-principal
type:Name group 0
```

**create topic-acl <TopicNames> producer allow-principal
<PrincipalType:principalName> [<options>]**

Create a producer-role ACL for a specified topic or topics.

Example:

```
cluster@myCluster> create topic-acl topic1 producer allow-principal
type:Name
```

create topic

create topic <TopicNames> partitions <Int> replication-factor <Int> [if-not-exists|disable-rack-aware]

Create a topic or topics with specified characteristics.

Example:

```
cluster@myCluster> create topic topic1 partitions 1 replication-
factor2
```

create topic <TopicNames> replica-assignment <String> [if-not-exists|disable-rack-aware]

Create a topic or topics based on a specified replica assignment.

Example:

```
cluster@myCluster> create topic topic1, topic2, topic3 replica-
assignment 0:2
```

See Also

[delete](#)

delete

Delete an object.

Use this command to delete an object on the connected cluster.

Synopsis

`delete <DeleteCommandArg> [DeleteCommandArg Options]`

delete cluster-acl

`delete cluster-acl allow-principal|deny-principal <PrincipalType:name> [force|allow-host <String>|deny-host <String>|operation <String>]`

Delete a cluster ACL.

Example:

```
cluster@myCluster> delete cluster-acl allow-principal type:Name
```

delete delegation-token-acl

`delete delegation-token-acl <String> allow-principal|deny-principal <PrincipalType:name> [force|allow-host <String>|deny-host <String>|operation <String>]`

Delete a delegation token ACL.

Example:

```
cluster@myCluster> delete delegation-token-acl aclName
```

delete group-acl

delete group-acl <String> allow-principal|deny-principal <PrincipalType:name>
[force|allow-host <String>|deny-host <String>|operation <String>]

Delete a group ACL.

Example:

```
cluster@myCluster> delete group-acl 0 allow-principal type:Name
```

delete topic

delete topic <TopicNames>[if-exists|disable-rack-aware]

Delete a topic or topics, with options.

Example:

```
cluster@myCluster> delete topic topic1, topic2, topic3
```

delete topic-acl

delete topic-acl <TopicNames> allow-principal|deny-principal
<principalType:principalName> [force|allow-host <String>|deny-host
<String>|operation <String>]

Delete an ACL for a specified topic or topics.

delete topic-acl <Topic Names> consumer allow-principal
<principalType:principalName> [group <String>|force|allow-host <String>]

Delete a consumer-role ACL for a specified topic or topics.

Example:

```
cluster@myCluster> delete topic-acl topic1 consumer allow-principal  
type:Name group 0
```

```
delete topic-acl <TopicNames> producer allow-principal <PrincipalType:name>  
[force|allow-host <String>|cluster|transaction-id <String>]
```

Delete a producer-role ACL for a specified topic or topics.



Note: For explanations of options for the above command forms, see the [Apache Kafka documentation](#).

See Also

[create](#)

disconnect

Disconnect MSGMX from a specific Apache Kafka cluster.

Use the disconnect command to disassociate your MSGMX session from a specific Apache Kafka cluster. MSGMX falls back to the most recent previous connection (if any) and you see an updated prompt reflecting the now-active connection.

Synopsis

```
disconnect <ClusterName> [<Host:Port>]
```

Required Arguments

ClusterName

The name assigned in the connect command.

Example

```
> disconnect myCluster
```

See Also

[connect](#)

exit

Exit MSGMX.

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

Synopsis

`exit`

Options

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>

Required Options

Command name to provide help.

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...
```

list

Display a list of objects.

Use this command to display a list of specified objects from the connected cluster.

Synopsis

```
list <ListCommandArg> [ListCommandArg Options]
```

list acls

list acls [force|principal <String>|cluster|transaction-id <String>|operation <String>]

List specified ACLs.

Example:

```
cluster@myCluster> list acls
Current ACLs for resource `Topic:LITERAL:topic2`:
    type5:fortopic2 has Allow permission for operations: All from
hosts: *
```

list bootstrap-servers

list bootstrap-servers

List bootstrap-servers in the connected cluster.

list brokers

list brokers

List all Apache Brokers in the connected cluster.

list connections

list connections

List all current connections. This command also displays the current health status of the listed servers.

Example:

```
Cluster@my-cluster> list connections

Cluster@my-cluster:
  Zookeeper: localhost:2181

  Zookeeper: localhost:2182

  Zookeeper: localhost:2183

  Broker: localhost:9092
  Broker: localhost:9093
```

list topics

list topics disable-rack-aware [force|exclude-internal]

List specified topics.

exclude-internal: Exclude internal topics when running list or describe command. The internal topics will be listed by default.

list topics exclude-internal [force|disable-rack-aware]

List specified topics.

disable-rack-aware: Disable rack aware replica assignment.

Example:

```
cluster@myCluster> list topics
topic1
topic2
```



```
topic3  
topic6
```

list zookeepers

list zookeepers

List all Apache ZooKeepers in the connected cluster.

See Also

[create](#), [delete](#)

reset

Use this command to reset the value of a specific object back to its default on the connected cluster.

Synopsis

```
reset <ResetCommandArg> [ResetCommandArg Options]
```

reset broker

reset broker <Broker ID> <Broker Option>

Reset specified values for a specified Broker. The Broker Options follow.

Example:

```
cluster@myCluster> reset broker 0 log.message.timestamp.type
```

Broker Options

For details on the following options, refer to the [Apache Kafka](#) documentation.

- advertised.listeners
- background.threads
- compression.type
- follower.replication.throttled.rate
- leader.replication.throttled.rate
- listener.security.protocol.map
- listeners
- log.cleaner.backoff.ms

- log.cleaner.dedupe.buffer.size
- log.cleaner.delete.retention.ms
- log.cleaner.io.buffer.load.factor
- log.cleaner.io.buffer.size
- log.cleaner.io.max.bytes.per.second
- log.cleaner.min.cleanable.ratio
- log.cleaner.min.compaction.lag.ms
- log.cleaner.threads
- log.cleanup.policy
- log.flush.interval.messages
- log.flush.interval.ms
- log.index.interval.bytes
- log.index.size.max.bytes
- log.message.downconversion.enable
- log.message.timestamp.difference.max.ms
- log.message.timestamp.type
- log.preallocate
- log.retention.bytes
- log.retention.ms
- log.roll.jitter.ms
- log.roll.ms
- log.segment.bytes
- log.segment.delete.delay.ms
- max.connections.per.ip
- max.connections.per.ip.overrides
- message.max.bytes

- metric.reporters
- min.insync.replicas
- num.io.threads
- num.network.threads
- num.recovery.threads.per.data.dir
- num.replica.fetchers
- principal.builder.class
- replica.alter.log.dirs.io.max.bytes.per.second
- sasl.enabled.mechanisms
- sasl.jaas.config
- sasl.kerberos.kinit.cmd
- sasl.kerberos.min.time.before.relogin
- sasl.kerberos.principal.to.local.rules
- sasl.kerberos.service.name
- sasl.kerberos.ticket.renew.jitter
- sasl.kerberos.ticket.renew.window.factor
- sasl.login.refresh.buffer.seconds
- sasl.login.refresh.min.period.seconds
- sasl.login.refresh.window.factor
- sasl.login.refresh.window.jitter
- sasl.mechanism.inter.broker.protocol
- ssl.cipher.suites
- ssl.client.auth
- ssl.enabled.protocols
- ssl.endpoint.identification.algorithm
- ssl.key.password

- `ssl.keymanager.algorithm`
- `ssl.keystore.location`
- `ssl.keystore.password`
- `ssl.keystore.type`
- `ssl.protocol`
- `ssl.provider`
- `ssl.secure.random.implementation`
- `ssl.trustmanager.algorithm`
- `ssl.truststore.location`
- `ssl.truststore.password`
- `ssl.truststore.type`
- `unclean.leader.election.enable`

reset client

reset client <Client> <Config>

Reset values for a specified client component.

Example:

```
cluster@myCluster> reset client 0 consumer_byte_rate
```

reset topic

reset topic <Topic Name> <Topic Option>

Reset values for a specified topic. The Topic Options follow.

Example:

```
cluster@myCluster> reset topic topic2 cleanup.policy
```

Topic Options

- cleanup.policy
- compression.type
- delete.retention.ms
- file.delete.delay.ms
- flush.messages
- flush.ms
- follower.replication.throttled.replicas
- index.interval.bytes
- leader.replication.throttled.replicas
- max.message.bytes
- message.downconversion.enable
- message.format.version
- message.timestamp.difference.max.ms
- message.timestamp.type
- min.cleanable.dirty.ratio
- min.compaction.lag.ms
- min.insync.replicas
- preallocate
- retention.bytes
- retention.ms
- segment.bytes
- segment.index.bytes
- segment.jitter.ms
- segment.ms

- `unclean.leader.election.enable`

reset user

reset user <UserName> <Config>

Reset values for a specified user.

Example:

```
cluster@myCluster> reset user User.* request_percentage
```

See Also

[set](#)

set

Use this command to set the configuration value of an object on the connected cluster.

Synopsis

```
set <SetCommandArg> [SetCommandArg Options]
```

set broker

set broker <Broker ID> <Broker Option> See [Broker Options](#) which follows.

Set specified values for a specified Broker.

Example:

```
cluster@myCluster> set broker 0 log.message.timestamp.type CreateTime  
Completed updating config for broker: 0.
```

Broker Options

For details on the following options, refer to the [Apache Kafka](#) documentation.

- advertised.listeners <String>
- background.threads <Int>
- compression.type <Value>
- follower.replication.throttled.rate <Int>
- leader.replication.throttled.rate <Int>
- listener.security.protocol.map <String>
- listener.ssl.truststore.certificates <String>
- listeners <String>

- log.cleaner.backoff.ms <Int>
- log.cleaner.dedupe.buffer.size <Int>
- log.cleaner.delete.retention.ms <Int>
- log.cleaner.io.buffer.load.factor <Double>
- log.cleaner.io.buffer.size <Int>
- log.cleaner.io.max.bytes.per.second <Double>
- log.cleaner.max.compaction.lag.ms <Int>
- log.cleaner.min.cleanable.ratio <Double>
- log.cleaner.min.compaction.lag.ms <Int>
- log.cleaner.threads <Int>
- log.cleanup.policy <String>
- log.flush.interval.messages <Int>
- log.flush.interval.ms <Int>
- log.index.interval.bytes <Int>
- log.index.size.max.bytes <Int>
- log.message.downconversion.enable <Boolean>
- log.message.timestamp.difference.max.ms <Int>
- log.message.timestamp.type <String>
- log.preallocate <Boolean>
- log.retention.bytes <Int>
- log.retention.ms <Int>
- log.roll.jitter.ms <Int>
- log.roll.ms <Int>
- log.segment.bytes <Int>
- log.segment.delete.delay.ms <Int>
- max.connection.creation.rate <Int>

- max.connections <Int>
- max.connections.per.ip <Int>
- max.connections.per.ip.overrides <String>
- message.max.bytes <Int>
- metric.reporters <String>
- min.insync.replicas <Int>
- num.io.threads <Int>
- num.network.threads <Int>
- num.recovery.threads.per.data.dir <Int>
- num.replica.fetchers <Int>
- principal.builder.class <String>
- replica.alter.log.dirs.io.max.bytes.per.second <Int>
- sasl.enabled.mechanisms <String>
- sasl.jaas.config <String>
- sasl.kerberos.kinit.cmd <String>
- sasl.kerberos.min.time.before.relogin <Int>
- sasl.kerberos.principal.to.local.rules <String>
- sasl.kerberos.service.name <String>
- sasl.kerberos.ticket.renew.jitter <String>
- sasl.kerberos.ticket.renew.window.factor <Double>
- sasl.login.refresh.buffer.seconds <Int>
- sasl.login.refresh.min.period.seconds <Int>
- sasl.login.refresh.window.factor <Double>
- sasl.login.refresh.window.jitter <Double>
- sasl.mechanism.inter.broker.protocol <String>
- ssl.cipher.suites <String>

- `ssl.client.auth` <String>
- `ssl.enabled.protocols` <String>
- `ssl.endpoint.identification.algorithm` <String>
- `ssl.engine.factory.class` <String>
- `ssl.key.password` <String>
- `ssl.keymanager.algorithm` <String>
- `ssl.keystore.certificate.chain` <String>
- `ssl.keystore.key` <String>
- `ssl.keystore.location` <String>
- `ssl.keystore.password` <String>
- `ssl.keystore.type` <String>
- `ssl.protocol` <String>
- `ssl.provider` <String>
- `ssl.secure.random.implementation` <String>
- `ssl.trustmanager.algorithm` <String>
- `ssl.truststore.location` <String>
- `ssl.truststore.password` <String>
- `ssl.truststore.type` <String>
- `unclean.leader.election.enable` <Boolean>
- `with-file` <Filepath>

set client

set client <ClientName> **controller_mutation_rate** <Double>|**consumer_byte_rate** <Int>|**producer_byte_rate** <Int>|**request_percentage** <Double>|**with-file** <String>

Set values for a specified client component.

Example:

```
cluster@myCluster> set client 0 consumer_byte_rate 10000
Completed Updating config for entity: client-id '0'.
```

set ips

set ips <IP> connection_creation_rate <Int>|with-file <String>

Set IP.

set user

set user <UserName> consumer_byte_rate <Int>|SCRAM-SHA-512 <String>|SCRAM-SHA-256 <String>|producer_byte_rate <Int>|request_percentage <String>|with-file <String>

Set values for a specified user.

Example:

```
cluster@myCluster> set user tibcoUser.* request_percentage 10
Completed Updating config for entity: user-principal 'tibcoUser'.
```

set topic

set topic <Topic Name> <Topic Option> See [Topic Options](#) which follows.

Set values for a specified topic. Options follow.

Example:

```
cluster@myCluster> set topic topic2 cleanup.policy delete
Completed Updating config for entity: topic 'topic2'.
```

Topic Options

- cleanup.policy <Value>

- `compression.type` <Value>
- `delete.retention.ms` <Int>
- `file.delete.delay.ms` <Int>
- `flush.messages` <Int>
- `flush.ms` <Int>
- `follower.replication.throttled.replicas` <Replicas>
- `index.interval.bytes` <Int>
- `leader.replication.throttled.replicas` <Replica List>
- `max.compaction.lag.ms` <Int>
- `max.message.bytes` <Int>
- `message.downconversion.enable` <Boolean>
- `message.format.version` <Version>
- `message.timestamp.difference.max.ms` <Int>
- `message.timestamp.type` <String>
- `min.cleanable.dirty.ratio` <Double>
- `min.compaction.lag.ms` <Int>
- `min.insync.replicas` <Int>
- `partitions` <Int> [forceif-existsdisable-rack-aware]
- `preallocate` <Boolean>
- `rreplica-assignment` <String> [forceif-existsdisable-rack-aware]
- `retention.bytes` <Int>
- `retention.ms` <Int>
- `segment.bytes` <Int>
- `segment.index.bytes` <Int>
- `segment.jitter.ms` <Int>
- `segment.ms` <Int>

- `unclean.leader.election.enable` <Boolean>
- `with-file` <Filepath>

set zookeeper

set zookeeper <Host>:<Port> zk-tls-config-file <File>

Set Zookeeper.

See Also

[reset](#)

show

Use this command to display the configuration values of an object on the connected cluster.

Synopsis

`show <ShowCommandArg> [ShowCommandArg Options]`

show adk-script-version

show akd-script-version

(UNIX only) Show the version of the underlying Apache Kafka scripts used for command execution.

show all-topics

show all-topics [under-min-isr-partitions|at-min-isr-partitions|exclude-internal|topics-with-overrides|unavailable-partitions|under-replicated-partitions|disable-rack-aware]

Show information about all topics in the cluster.

Example:

```
cluster@myCluster> show all-topics
Topic:topic1    PartitionCount:1    ReplicationFactor:1    Configs:
      Topic: topic1    Partition: 0    Leader: 0    Replicas: 0
Isr: 0

Topic:topic2    PartitionCount:1    ReplicationFactor:1    Configs:
      Topic: topic2    Partition: 0    Leader: 0    Replicas: 0
Isr: 0

Topic:topic3    PartitionCount:1    ReplicationFactor:1    Configs:
      Topic: topic3    Partition: 0    Leader: 0    Replicas: 0
```

```

Isr: 0

Topic:topic6    PartitionCount:1    ReplicationFactor:1    Configs:
      Topic: topic6    Partition: 0    Leader: 0    Replicas: 0
Isr: 0
Topic:topicname PartitionCount:1    ReplicationFactor:1    Configs:
      Topic: topicname    Partition: 0    Leader: 0
Replicas: 0    Isr: 0

```

show broker

show broker <String> [force|all]

Show information about all topics in the cluster.

Example:

```

cluster@myCluster> show broker 0
Configs for brokers '0' are log.message.timestamp.type=CreateTime

```

show broker-api-version

show broker-api-version

Show the Apache Kafka broker API version.

show client

show client <String> [force|all]

Show information about a specified client component in the cluster.

Example:


```
cluster@myCluster> show client 0
Configs for client-id '0' are request_percentage=50
```

show log-dirs

show log-dirs [topic <Topic Name>|no-formatting]

Show information about log directories for all topics, or a specified topic.

show log-dump

show log-dump <File> [deep-iteration|index-sanity-check|key-decoder-class <String>|max-message-size <Int>|offsets-decoder|print-data-log|transaction-log-decoder|value-decoder-class <String>|verify-index-only|cluster-metadata-decoder|skip-record-metadata]

Show the contents of a specified log file.

show topic

show topic <TopicNames> [at-min-isr-partitions|under-min-isr-partitions|exclude-internal|if-exists|topics-with-overrides|unavailable-partitions|under-replicated-partitions|disable-rack-aware]

Show information about a specified topic or topics in the cluster. <TopicNames> is a string that specifies one or multiple topic names. For multiple topics, separate the topics with a comma.

Example:

```
cluster@myCluster> show topic topic2
Topic:topic2    PartitionCount:1    ReplicationFactor:1    Configs:
                Topic: topic2    Partition: 0    Leader: 0    Replicas: 0
Isr: 0
```

show user

show user <String> [force|all]

Show information about a specified client component in the cluster.

Example:

```
cluster@myCluster> show user tibcoUser  
Configs for user-principal 'tibcoUser' are request_percentage=99
```

show zookeeper

show zookeeper

Show information about the cluster's ZooKeeper(s).



Note: For explanations of options for the above command forms, see the [Apache Kafka documentation](#).

Kerberos Authentication Notes

If the Apache Kafka Cluster is secured using Kerberos protocol, note the following information.

Kerberos Authentication and MSGMX

If the Apache Kafka-Cluster is secured with Kerberos, you can still manage the cluster with MSGMX. MSGMX uses the Apache Kafka native administrative scripts (`kafka-topics`, `kafka-acls`, `kafka-configs`, etc.), which in turn uses the Kafka Admin API.

MSGMX, via the administrative scripts, uses the `connect` command's `--command-config` option to read a specified properties file that contains credentials that you configure.

For more details, see the following:

- Apache Kafka with Kerberos: kafka.apache.org/documentation/#security_sasl_kerberos
- Kerberos credentials settings: kafka.apache.org/documentation/#security_sasl_kerberos_clientconfig
- Apache Kafka Admin client configuration: kafka.apache.org/documentation/#adminclientconfigs
- Apache Kafka Admin API: kafka.apache.org/documentation/#adminapi
- MSGMX connect: [connect](#)

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.