Terracotta Management Console User Guide

Version 4.3.9

October 2020

TERRACOTTA

This document applies to BigMemory 4.3.9 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: BMM-TMCUG-439-20201015

Table of Contents

About This Documentation

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at https://empower.softwareag.com/.

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Getting Started with Terracotta Management Console

About the Terracotta Management Console

The Terracotta Management Console (TMC) is a web-based administration and monitoring application with the following advantages:

- Multi-level security architecture, with end-to-end SSL secure connections available
- Feature-rich and easy-to-use interface
- Remote management capabilities requiring only a web browser and network connection
- Cross-platform deployment
- Role-based authentication and authorization
- Support for LDAP directories and Microsoft Active Directory
- Aggregate statistics from multiple nodes
- Flexible deployment model, which can plug into both development environments and secure production architectures

The TMC can monitor BigMemory nodes and clusters through the Terracotta Management Server (TMS). The TMS acts as an aggregator and also provides a connection and security context for the TMC. The TMS must be available and accessible for the TMC to provide management services.

The TMS is included with your BigMemory kit under the `tools/management-console` directory.

Installing the Terracotta Management Server

Use the following steps to install the Terracotta Management Server.

To install the Terracotta Management Server

1. Locate the `management-console` directory in the BigMemory kit.
2. Copy this directory to the location where the Terracotta Management Server will run.
3. Copy the license file into the `management-console` directory you just created.

Running the Terracotta Management Server with a Different Container

The Terracotta Management Server can be run directly with the provided Jetty Java Servlet container. To run the server with an application server of your choice, use the file `management-console/webapps/tmc.war`.

Follow the specifications and requirements of your chosen application server for deploying a WAR-based application.

Configuring the Terracotta Management Server

A BigMemory Max cluster can be managed directly by connecting the Terracotta Management Console(TMC) to any one of the servers in the cluster. All other servers and clients become visible to the TMC through that initial connection. Create a new connection and enter the URI to a server in the following form:

```
<scheme>://<host-address>:<management-port>
```

where `<management-port>` is the port number configured in the server's `tc-config.xml` file. The default is 9540.

To manage a client or standalone node (Terracotta Ehcache client or BigMemory Go) using the TMC, enable the REST management service on that node by setting the following element in the ehcache.xml configuration:

```
<ehcache ... >
...
  <managementRESTService enabled="true" bind="<ip_address>:<port>"/>
...
</ehcache>
```

where `<ip_address>` is the local network interface's IP address and `<port>` is the port number used to manage the node. The following defaults are in effect for `<managementRESTService>`:

- enabled="false" (This must be set to "true" for standalone caches.)
- bind="0.0.0.0:9888" (This IP address binds the specified port to all network interfaces on the local node.)

The REST management service can also be enabled programmatically:

```
ManagementRESTServiceConfiguration rest = new ManagementRESTServiceConfiguration();
rest.setBind("0.0.0.0:9888");
rest.setEnabled(true);
config.addManagementRESTService(rest);
```

Note:
If performance becomes an issue when the TMC is in heavy use, see "Number of Clients Impacts Performance" on page 74.

Displaying Update Statistics

By default, caches distributed in BigMemory Max generate put events whenever elements are put or updated. To have the TMC track and display updates separately from puts, set the Terracotta property `ehcache.clusteredStore.checkContainsKeyOnPut` at the top of the Terracotta configuration file (`tc-config.xml` by default) before starting the Terracotta Server Array:

```
<tc-properties>
  <property name="ehcache.clusteredStore.checkContainsKeyOnPut" value="true" />
</tc-properties>
```

Note:*Enabling this property can substantially degrade performance.* Before using in production, test the effect of enabling this property.

Using Multiple Instances of BigMemory Go CacheManagers With the TMC

When loading multiple instances of BigMemory Go CacheManagers with the TMC rest agent enabled in the same JVM, CacheManagers must be loaded by distinct classloaders. Two different web applications (two different WARs), for example, are loaded by two different classloaders.

The TMC Update Checker

The Update Checker automatically checks to see if you have the latest updates, and collects diagnostic information on TMC usage. The Update Checker is on by default. To disable the update checker, use the following system property:

```
-Dcom.terracotta.management.skipUpdateCheck=true
```

Starting and Stopping the Terracotta Management Console

Starting the Terracotta Management Console

Use the following procedure to start the Terracotta Management Console.

To start the Terracotta Management Console

- Run the following command:

  On UNIX: `management-console/bin/start-tmc.sh`

  On Windows: `management-console/bin/start.bat`

  Note:
  The TMC requires that the path have no spaces.

To Stop the Terracotta Management Console

Use the following procedure to stop the Terracotta Management Console.

To stop the Terracotta Management Console

- Run the following command:

  On UNIX: `management-console/bin/stop-tmc.sh`

  On Windows: `management-console/bin/stop.bat`

Connecting to the Terracotta Management Console

To connect to the Terracotta Management Console, do the following:

To connect to the Terracotta Management Console

- Connect to the TMC using the following URI with a standard web browser:
  `http://localhost:9889/tmc`

  If you are connecting remotely, substitute the appropriate hostname. If you have set up secure browser connections, use "https" instead of "http."

  When you first connect to the TMC, the security setup page appears, where you can choose to run the TMC with or without authentication. Authentication can also be enabled/disabled in the TMC Settings panel. For more information, see "Setting up Security" on page 57.

  For more information on using the TMC, you can click the **Help** links available on certain pages within the UI, or access the TMC online help by choosing **Help** from the toolbar.

Updating the Terracotta Management Server

Installing a new version of a Terracotta kit also installs an updated version of the TMS. When this new version is started, it checks for existing configuration files under `<user.home>/.tc/mgmt`, backing up any incompatible files (extension .bak). In this case, previously configured connections will not appear in the TMC and must be re-added.

Uninstalling the Terracotta Management Console

If you want to remove the Terracotta Management Console, do the following.

To uninstall the Terracotta Management Console

- Delete the `~/.tc/mgmt/` directory.

2 Using the Terracotta Management Console

Initial Setup

The Terracotta Management Console (TMC) is a web-based administration and monitoring application for Terracotta products. TMC connections are managed through the Terracotta Management Server (TMS), which must be running for the TMC to function.

Note:
You can confirm the version of the TMC you are running and get other information about the TMC by clicking **About** on the toolbar.

When you first connect to the TMC, the authentication setup page appears, where you can choose to run the TMC with authentication or without. Authentication can also be enabled/disabled in the TMC Settings panel.

If you do not enable authentication, you can connect to the TMC without being prompted for a login or password.

If you enable authentication, the following choices appear:

- .ini file - Simple, built-in, role-based authentication. For information, see "Simple Account-Based Authentication" on page 14.
- LDAP - Use with LDAP server. For more information, see "Setting up LDAP or Active Directory Authorization" on page 65.
- Microsoft Active Directory - Use with an Active Directory server. Instructions for setting up connections to LDAP and Active Directory are available with the form that appears when you select the LDAP or Active Directory. See also, "Setting up LDAP or Active Directory Authorization" on page 65.

Setting up authorization and authentication controls access to the TMC but does not affect connections, which must be secured separately. For more information, see the *BigMemory Max Security Guide*. In addition, an appropriate Terracotta license file is needed to run the TMC with security.

Simple Account-Based Authentication

Authentication using built-in role-based accounts backed by a .ini file is the simplest scheme. When you choose .ini-file authentication, you must restart the TMC using the stop-tmc and start-tmc scripts. A setup page appears for initializing the two accounts that control access to the TMC:

- **Administrator** - This account (username "admin") has read and write control over the TMC.
- **Operator** - This read-only account (username "operator") can view statistics and other information about configured connections. This account cannot add or edit connections.

Create a password for each account, then click **Done** to go to the login screen. The login screen appears each time a connection is made to the TMC.

Inactivity Timeout

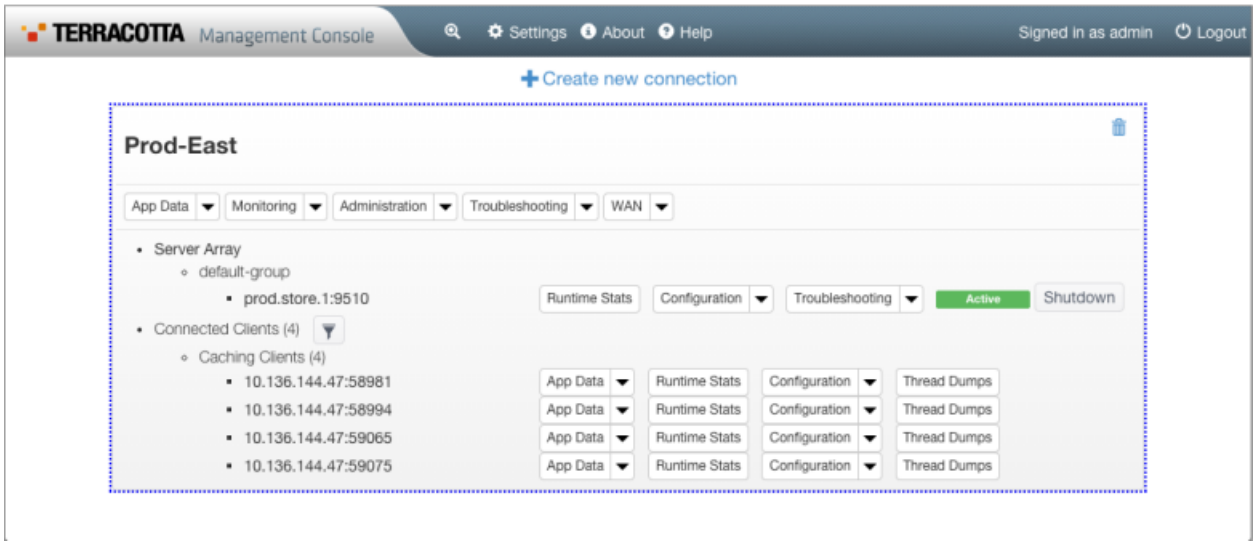
The Terracotta Management Console allows a connected user to remained connected indefinitely, whether or not that user is active. To set a default timeout for inactivity, navigate to the WEB_INF directory, open the `web.xml` file, and uncomment the following block. You can then accept its default value of 30 for `idleTimeoutMinutes` or specify a different value:

```
<context-param>
  <description>
    After this amount of time has passed without user activity, the user will
    be automatically logged out.
  </description>
  <param-name>idleTimeoutMinutes</param-name>
  <param-value>30</param-value>
</context-param>
```

Note that internal to the TSA and TMC, the Apache Shiro session management is configured with an inactivity timeout of 10 minutes, expressed in milliseconds, `securityManager.sessionManager.globalSessionTimeout = 600000`. However, this timeout setting is unrelated to the human end-user activity. For more information about Apache Shiro, see Shiro session management.

The TMC Home Page

The home page provides an overview of all of your Terracotta servers and clients, with buttons and drop-down menus that allow you to navigate directly to any TMC panel or view. The home page is comprised of connection panes for each cluster or connection group that is recognized by the TMC. The connection pane shows the status of the connection's server(s) and lists the connection's client(s), with direct access to TMC functionality for each.



To view connections

When you first view the TMC, only default connections will appear. For additional connections to appear on the home page, you must add them, either by clicking **+Create New Connection** at

the top of the page, or by clicking **Settings** in the top toolbar. For more information, refer to "Connections and Global Settings" on page 16.

To view connection details

On the home page, click on a connection pane to select it, or select the connection name from the drop-down menu in the top toolbar. You can then click the magnifier icon to start viewing details about that connection.

Alternatively, you can navigate to any TMC panel or view using the buttons and drop-down menus within the connection pane.

Active tooltips on the home page provide additional information, for example, CacheManager names pop up when you hover over the App Data drop-down menus.

To filter the client list

To filter the clients appearing in a connection pane, click the filter icon next to **Connected Clients**, and enter the addresses of the clients that you want to view in the connection pane.

To return to the home page

After navigating away from the home page, you can return by clicking the home icon that appears in the top toolbar.

Connections and Global Settings

Click **Settings** on the top toolbar to open a window where connections and global TMC options can be configured. The Settings window includes Connections, Polling, and Security panels.

Working With Connections

Connections allow you to monitor and administer nodes, both clustered and standalone. Clustered connections are for Terracotta Server Arrays, and Connection Groups are for standalone connections to agents that are assigned to groups to simplify management tasks.

Connections from the TMS to agents are made using a location URI in the following form:

```
<scheme>:<host-address>:<port>
```

URIs showing "http:" are for non-secure connections.

If the URI is for a server in a Terracotta Server Array, all other nodes participating in the cluster are automatically found. It is not required to create separate connections for those other nodes. A typical URI for a server is similar to:

```
http://myServer:9540
```

where an IP address or resolvable hostname is followed by the management-port number (9540 by default). This port is configured in `tc-config.xml`.

A typical URI for a Terracotta client or BigMemory Go will appear similar to:

```
http://myHost:9888
```

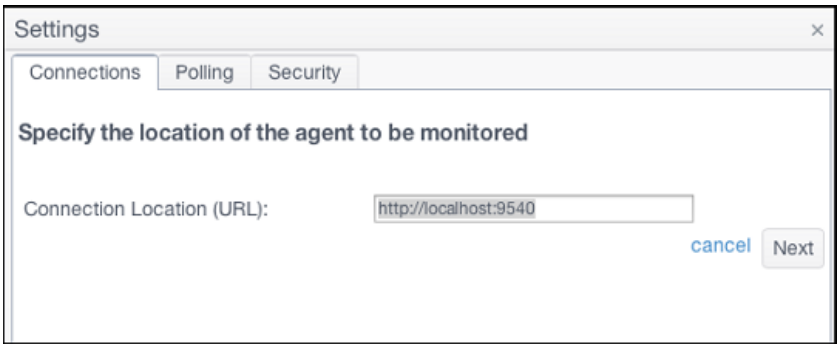
where an IP address or resolvable hostname is followed by the agent's management port (9888 by default), which has been set in the node's configuration file. For BigMemory Go, for example, use the `managementRESTService` element in `ehcache.xml`.

Adding a Connection

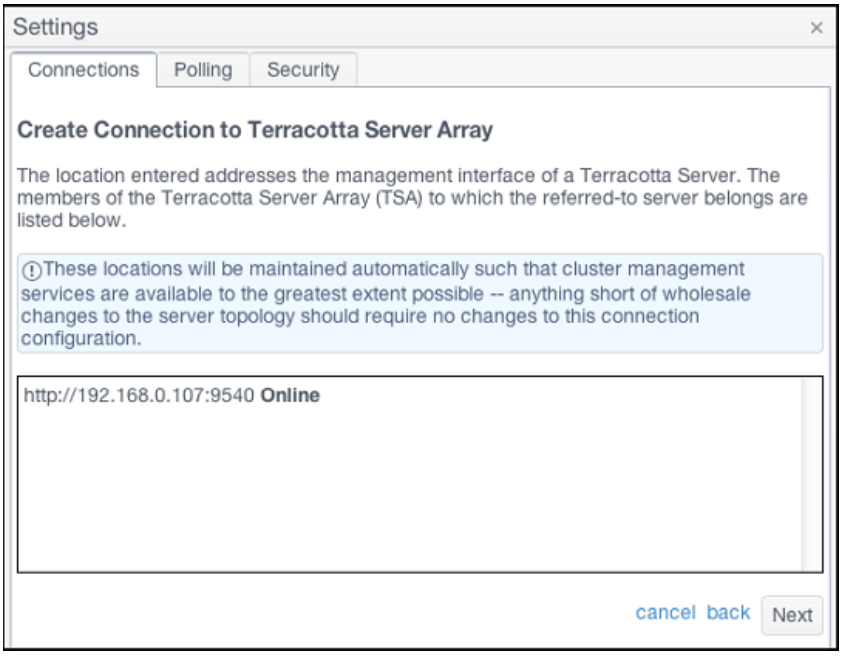
When you initially log on to the TMC, only default connections exist. If a node that can be monitored is running on localhost at the port specified by one of the default connections, that default connection appears as an active connection. Other default connections appear as unavailable (inactive) connections.

To add a new connection:

1. Click **+Create New Connection** button in the Connections panel of the Settings window, or at the top of the home page. The first window of the New Connection wizard appears.

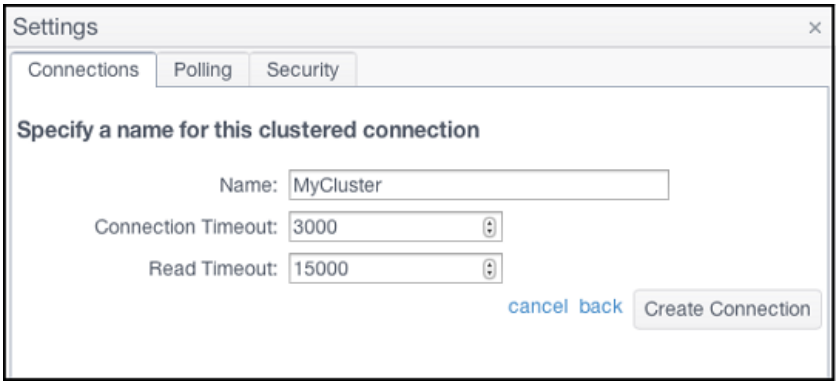


2. Enter the location URI of the node you want to monitor, then click **Next**.



A screen appears confirming the agent found at the given location. If no agent is found, a warning appears and no connection can be set up. The location is relative to the machine running the Terracotta Management Server (TMS). The default location, "localhost", is the machine the TMS is running on, which might not be the machine your browser is running on.

3. Choose an existing connection group for the connection, or create a new one, then click **Next**.



4. Enter a name to identify the connection.
5. Enter a connection timeout or accept the default value.

   The connection timeout ensures that the TMC does not hang waiting for a connection to an unreachable node.

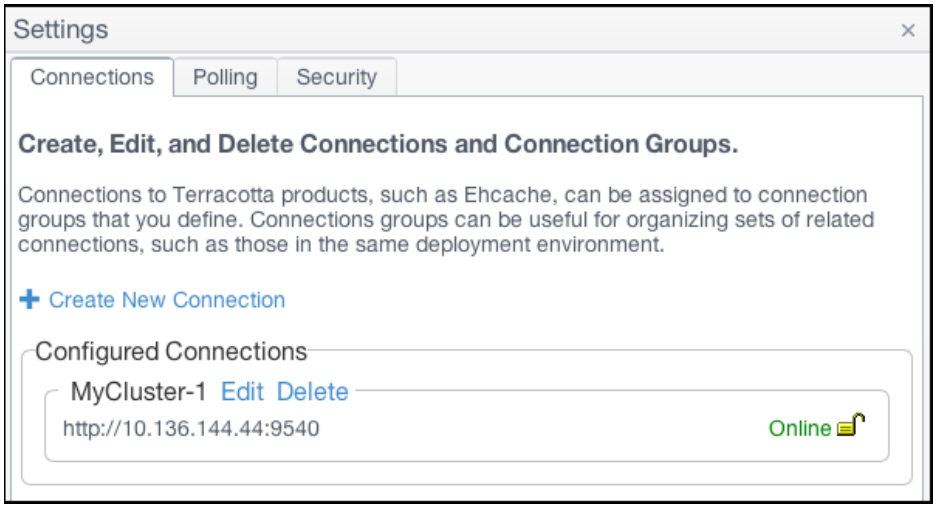
6. Enter a read timeout or accept the default value.

   The read timeout ensures that the TMC does not hang waiting for a connection to an unresponsive node.

7. Click **Create Connection** to save the new connection or **Cancel** to discard the new connection.

Editing and Deleting Connections

Managed connections that appear in the connections list can be edited or deleted.



Delete a Connection

To delete an existing standalone connection, click **Settings** on the toolbar to view the **Connections** panel. Locate the connection under its connection group in the **Configured Connections** list and click the red X next to that connection's name.

To delete an existing cluster connection, click **Settings** on the toolbar to view the **Connections** panel. Locate the connection group in the **Configured Connections** list and click **Delete** next to that group's name.

Alternatively, from the home page, click the trash icon in the connection pane you want to delete.

Edit a Standalone Connection

To edit a standalone connection:

1. Click **Settings** on the toolbar.
2. In the **Connections** panel, click the pencil icon for the connection you want to edit.
3. Edit the connection's location, group, and name.

   You can choose a group for the connection from the menu of existing groups, or create a new connection group. If you create a new group, enter a name for the group.
4. Enter a connection timeout or accept the default value.

   The connection timeout ensures that the TMC does not hang waiting for a connection to an unreachable node.
5. Enter a read timeout or accept the default value.

The read timeout ensures that the TMC does not hang waiting for a connection to an unresponsive node.

6. Click **Save Changes** to save the new values or **Cancel** to revert to the original values.

Edit a Cluster Connection

To edit a cluster connection, click **Edit** for the cluster group, then edit the group name, connection location, and timeouts. Click **Save Changes** to save the new values or **Cancel** to revert to the original values.

Polling Period

In the Settings window, click the **Polling** tab to set the **Polling Interval Seconds**, which controls the granularity of polled statistical data. Note that shorter polling intervals can have a greater effect on the overall performance of the nodes being polled. To reset to default values, click **Reset to Defaults**.

Security Settings

In the Settings window, click the **Security** tab to configure security. If you choose to change the type of security used by the TMS, note the following:

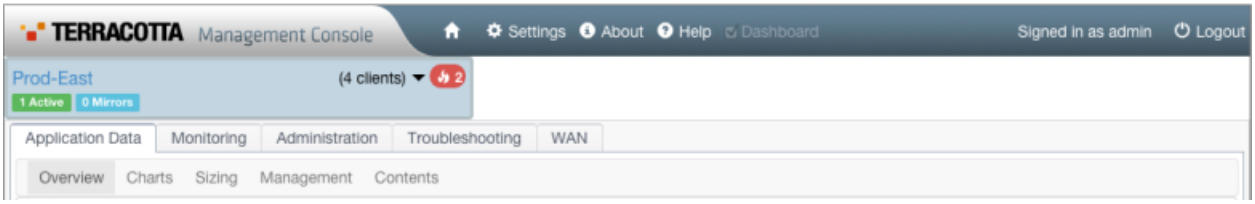
- Changing security settings requires restarting the TMC.
- Changing the type of security might require additional configuration information and infrastructure.
- If you add security, connections to unsecured nodes might be lost.
- If you disable authentication, connections to secured nodes might be lost.

For SSL connections, you can use a custom truststore instead of the default Java cacerts. The custom truststore must be located in the default directory specified in the **Security** panel.

For more information about setting up security, see "Setting up Security" on page 57.

Dashboards, Tabs, and Management Panels

Once you have navigated away from the home page, configured connection groups appear in mini-dashboards across the top of the page. Click on the dashboard to select it and view management panels for that connection group. Alternatively, you can select a connection from the drop-down menu in the top toolbar.



Each TSA connection-group dashboard displays the number of connected active (green) and mirror (blue) servers. It also displays the number of clients connected to that TSA. Certain other server states might also be indicated on the dashboard, including server starting or recovering (yellow) and server unreachable (red).

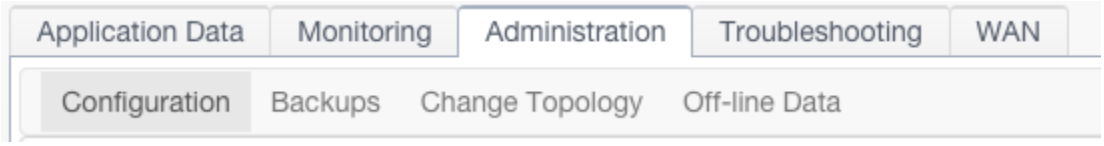
Each standalone connection group dashboard displays its number of configured connections and the number currently connected.

Each dashboard has a control drop-down menu with commands applicable to that dashboard and its associated connection group. For example, to hide a connection group's dashboard, choose **Hide This Connection** from the group's dashboard control menu. The connection group's connections are unaffected by hiding the dashboard. To restore the dashboard to the connections, click **Settings** from the toolbar, then enable **Show in Dashboard** checkbox for that group.

To hide (or view) all of the dashboards, click **Dashboard** in the top toolbar. Note that dashboards will automatically appear whenever there are unread events.

If there are unread operator events, a flame icon will appear in a dashboard. Hover over it to see the number and level of the events, or click it to go directly to the Monitoring Events panel.

Below the dashboards are the TMC tabs: Application Data, Monitoring, Administration, Troubleshooting and WAN. Click on a tab to access to its panels, and click on a panel to view information about the selected connection.



Note that each management panel has active tooltip hints that pop up when you hover over various page elements.

3 Using the Application Data Tab

About the Application Data Tab

To manage the application data of nodes in a connection group, select the group, then click the **Application Data** tab. Each **Application Data** panel has a **CacheManager** and **Scope** menu to select which CacheManagers and nodes supply the data for that panel.

Overview Panel

The **Overview** panel displays health metrics for CacheManagers and their caches, including certain cache statistics to help you track performance and resource usage across all CacheManagers.

Real-time statistics are displayed in a table with the following columns:

- **Hit Ratio**- The ratio of cache hits to get attempts. A ratio of 1.00 means that all requested data was obtained from the cache (every put was a hit). A low ratio (closer to 0.00) implies a higher number of misses that result in more faulting in of data from outside the cache.
- **Hit Rate** - The number of cache hits per second. An effective cache shows a high number of hits relative to misses.
- **Local Disk Hit Rate** - The fault rate (data faulted in from the local disk).
- **Local Heap Hit Rate** - The rate of local (in-heap) hits (no faulting).
- **OffHeap Hit Rate** - The rate of local (off-heap) memory hits. Available only when off-heap memory is configured.
- **Miss Rate** - The number of cache misses per second. An effective cache shows a high number of hits relative to misses.
- **Local Disk Miss Rate** - The fault rate (data faulted in from remote source).
- **Local Heap Miss Rate** - The rate of local (in-heap) misses (causing faulting).
- **OffHeap Miss Rate** - The rate of local (off-heap) memory misses (causing faulting). Available only when off-heap memory is configured.
- **Size** - Overall data size (in entries).
- **Local Heap Size** - Overall data size (in entries) in the local heap.
- **Local Disk Size** - Overall data size (in entries) on the local disk.
- **Local OffHeap Size** - Overall data size (in entries) in local memory (off-heap). Available only when off-heap memory is configured.
- **Local Heap Bytes** - Overall data size (in bytes) in the local heap.
- **Local Disk Bytes** - Overall data size (in bytes) on the local disk.
- **Local OffHeap Bytes** - Overall data size (in bytes) in local memory (off-heap). Available only when off-heap memory is configured.

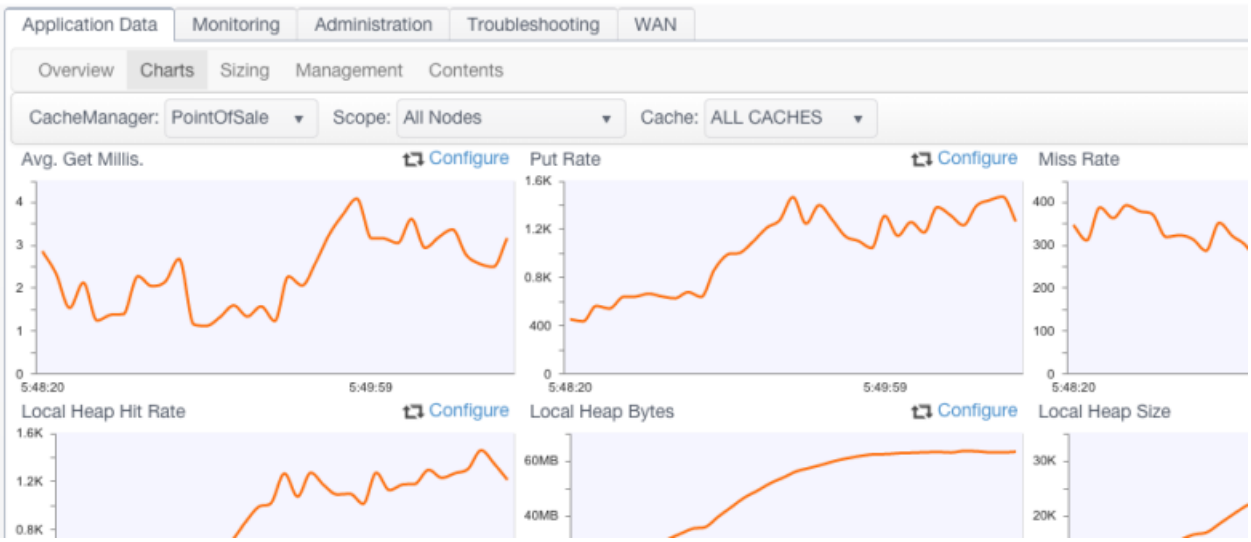
- **Average Get Time** - The average time for executing a get operation.
- **Average Search Time** - The Average Search Time graph displays how long each search operation takes (as well as the current values for that search time). This graph indicates how quickly cache searches are being performed. You might notice a correlation between how long searches are taking (Average Search Time) and how many searches are executed (Search Rate).
- **Search Rate** - The search-rate graph displays how many searches per second are being executed (as well as the current values for that rate). This graph provides a view into how many cache searches are being performed. You might notice a correlation between how long searches are taking (Average Search Time) and how many searches are executed (Search Rate).
- **Put Rate** - The number of cache puts executed per second. The number of puts always equals or exceeds the number of misses. This is because every miss leads to a put and updates are counted as puts. Efficient caches have a low overall put rate.
- **Remove Rate** - The rate of element eviction.
- **Update Rate** - The number of updates per second to elements in the cache. A high number of updates implies a high eviction rate or rapidly changing data.
- **Expiration Rate** - The number of elements per second reaching expiration in the cache. Expired elements are not automatically evicted.
- **Eviction Rate** - The number of elements being evicted per second from the cache. Evicted elements are expired or evicted according to a usage algorithm when size limits are exceeded.
- **Transaction Rollback Rate**- A Java Transaction API (JTA) graph that displays the rollback rate (as well as the current values for that rate) for transactional caches.
- **Transaction Commit Rate**- A JTA graph that displays the transaction commit rate (as well as the current values for that rate) for transactional caches.
- **Writer Queue Length** - The Write-Behind graph displays the total number of writes in the write-behind queue or queues (blue line), as well as the current value.

To choose the types of statistics displayed in the table, click **Configure Columns** to open a list of available statistics. Choose statistics (or set the option to display all statistics), then click **OK** to accept the change. The table immediately begins to display the chosen statistics.

To sort the table by a specific statistic, click the column head for that statistic.

Charts Panel

The **Charts** panel graphs the same statistics available in the **Overview** panel. This is useful for tracking performance trends and discovering potential issues. For more information about the statistics on the **Overview** panel, see "Overview Panel" on page 24.



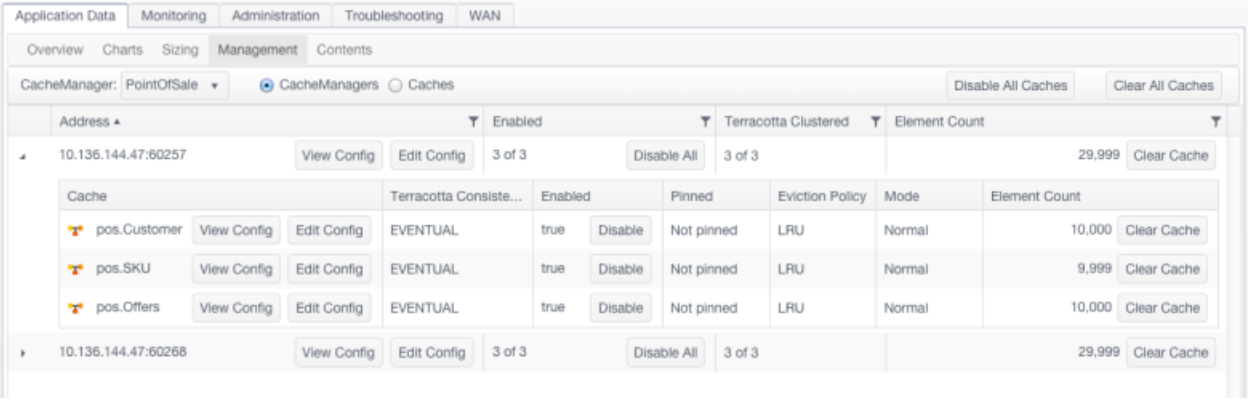
In addition to being able to select a CacheManager and scope for the displayed data, you can also select a specific cache (or all caches) for the selected CacheManager.

Each graph plots the appropriate metrics along the Y axis against system time (X axis). To view the value along a single point on a graph, float the mouse pointer over that point. This also displays the units used for the statistic being graphed.

To choose the type of statistic graphed by a particular chart, click the chart's corresponding **Configure** link to open a list of available statistics. Choose a statistic, then click **OK** to accept the change. The chart immediately begins to graph the chosen statistic.

Sizing Panel

The **Sizing** panel provides information on the usage of the heap, off-heap, and disk tiers by the caches of the selected CacheManager.



To view tier usage by any active CacheManager, select that CacheManager from the **CacheManager** drop-down menu.

Usage by Tier

The **Relative Cache Sizes by Tier** - table displays usage of the tier selected from the **Tier** drop-down menu. The table has the following columns:

- **Cache** - The name of the cache. An icon indicates whether the cache is distributed ( ) or standalone ( ).
- **Size (MB)** - The size of the cache's data in megabytes. This value is a snapshot and might not be accurate until the server has fully processed the data.
- **% of Used** - Percent of the total storage allotted to the cache that is currently used for cache data.
- **Entries** - The total number of cache entries.
- **Mean Entry Size (bytes)** - An estimate of the average size of each cache entry.

Click a row in the table to set the cache-related tier graphs to display values for the named cache.

Usage Graphs

The panel shows the following bar graphs:

- **Usage by Tier** - Overall usage of each tier. Each bar shows the total resource allocated, the amount in use, and the amount available.
- **Cache Usage by Tier** - Usage of each tier by the selected cache. Choose the cache from the **Selected Cache** drop-down menu. Each bar shows the total resource allocated, the amount in use, and the amount available.
- **Cache Miss Rate by Tier** - The rate of cache misses at each tier of the cache specified in the **Selected Cache** drop-down menu. The number of misses is displayed in each bar.

To display an exact usage value, float the mouse pointer over a bar. To display values for that tier in the **Relative Cache Sizes by Tier** table, click a tier's bar. The selected tier's bar is lighter in color than the other bars.

The Selected Cache Menu

The **Selected Cache** drop-down menu determines which cache is shown in the cache-related tier graphs and highlighted in the **Relative Cache Sizes by Tier**. The menu also indicates if the cache uses size-based (automatic resource control, that is, ARC) or entry-based sizing.

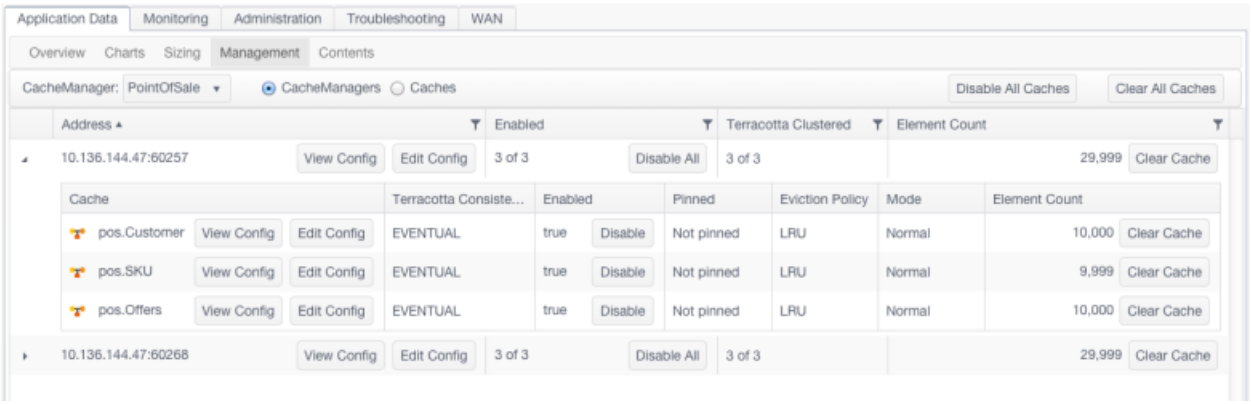
Management Panel

The **Management** panel displays a table listing information about the selected CacheManager by node (where the CacheManager exists) or by its caches. Choose the **CacheManagers** radio button to show a table with a node list, or the **Caches** radio button to show a table with a cache list. These tables (and any sublist tables) can be sorted and ordered by any column by clicking the column head.

Global cache disable/enable controls at the top of the panel.

List by Cache

The cache list is a table of caches under the selected cache manager.



The table has the following columns:

- **Cache** - The name of the cache.
- **Enabled** - Shows how many instances of the cache are enabled out of the total number of instances in the cluster. Clicking **Disable All** disables (stops) all instances of the cache in the cluster. If caches are disabled, the control becomes **Enable All**, which can enable the operations of all of the cache instances at once.
- **Terracotta Clustered** - Shows how many of the instances of the cache are distributed.
- **Element Count**- Shows the total number of elements in all instances of the cache. Click **Clear Cache** to wipe the data from all instances of the cache in the cluster.

  Note:
  Be sure to disable a cache using the **Disable** button before clearing it with the **Clear Cache** button.

If a cache listing is expanded using the arrow to the left of the cache name, a sublist appears with a table of all of the nodes that contain the cache. The table has the following columns:

- **Address** - The connection name for node. To view the cache's configuration on the node, click **View Config**. Click **Edit Config** to open a dialog where you can edit the values of the following parameters (depending upon your settings, a subset of these parameters will be present):
  - MaxEntriesInCache
  - MaxEntriesLocalHeap
  - MaxBytesLocalHeap
  - TimeToIdleSeconds
  - TimeToLiveSeconds

- **Terracotta Consistency** - For clustered caches, indicates whether consistency is EVENTUAL (default) or STRONG. Eventual consistency uses no cache-level locks for superior performance while allowing a short window when stale values might be read. Strong consistency uses locks to prevent any stale reads, but at a high cost to performance. This setting is not dynamic.
- **Enabled** - Indicates whether the cache is enabled on the node. Clicking **Disable** disables (stops) the cache on the node. If a cache is disabled, the control becomes **Enable**, which can enable the operations of the cache.
- **Pinned** - Indicates that the cache data is pinned to local memory (LOCALMEMORY), anywhere the cache's data is stored (INCACHE), or is not pinned (na).
- **Eviction Policy** - Indicates the eviction policy used for evicting entries from the cache. For example, LRU indicates that the Least Recently Used policy is in effect.
- **Mode** - Indicates whether the cache is in bulk-load or normal operating mode. Applications set the cache in bulk-load mode temporarily while warming the cache.
- **Element Count**- The total number of elements in the cache on the node. To wipe the data of the cache on the node, click **Clear Cache**.

List by CacheManager

The CacheManager list is a table of nodes under the selected cache manager. The table has the following columns:

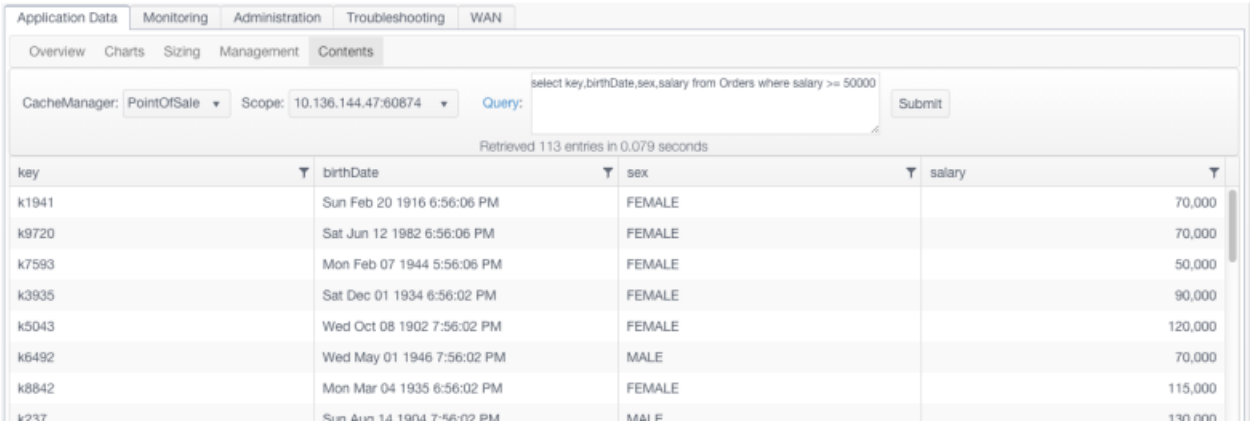
- **Address** - The connection name for node. To view the CacheManager's configuration on the node, click **View Config**. Click **Edit Config** to open a dialog where you can edit the values of the following parameters. Depending upon your settings, a subset of these parameters might be present.
  - MaxEntriesInCache
  - MaxEntriesLocalHeap
  - MaxBytesLocalHeap
  - TimeToIdleSeconds
  - TimeToLiveSeconds
- **Enabled** - Shows how many instances of the cache are enabled out of the total number of instances in the cluster. Clicking **Disable All** disables (stops) all instances of the cache in the cluster. If caches are disabled, the control becomes **Enable All**, which can enable the operations of all of the cache instances at once.
- **Terracotta Clustered** - Shows how many of the instances of the cache are distributed.
- **Element Count** - The total number of elements in the cache on the node. To wipe the data of the cache on the node, click **Clear Cache**.

If a node listing is expanded using the arrow to the left of the connection name, a sublist appears with a table of all of the nodes that contain the cache:

- **Cache** - The name of the cache. To view the cache's configuration, click **View Config**. Click **Edit Config** to open a dialog where you can edit the values of the following parameters (depending upon your settings, a subset of these parameters will be present):
  - MaxEntriesInCache
  - MaxEntriesLocalHeap
  - MaxBytesLocalHeap
  - TimeToIdleSeconds
  - TimeToLiveSeconds
- **Terracotta Consistency** - For clustered caches, indicates whether consistency is EVENTUAL (default) or STRONG. Eventual consistency uses no cache-level locks for superior performance while allowing a short window when stale values might be read. Strong consistency uses locks to prevent any stale reads, but at a high cost to performance. This setting is not dynamic.
- **Enabled** - Indicates whether the cache is enabled on the node. Clicking **Disable** disables (stops) the cache on the node. If a cache is disabled, the control becomes **Enable**, which can enable the operations of the cache.
- **Pinned** - Indicates that the cache data is pinned to local memory (LOCALMEMORY), anywhere the cache's data is stored (INCACHE), or is not pinned (na).
- **Eviction Policy** - Indicates the eviction policy used for evicting entries from the cache. For example, LRU indicates that the Least Recently Used policy is in effect.
- **Mode** - Indicates whether the cache is in bulk-load or normal operating mode. Applications set the cache in bulk-load mode temporarily while warming the cache.
- **Element Count** - The total number of elements in the cache on the node. To wipe the data of the cache on the node, click **Clear Cache**.

Contents Panel

The **Contents** panel allows you to issue BigMemory SQL queries against your caches.



For BigMemory SQL help, click the blue **Query** link next to the text box.

4 Using the Monitoring Tab

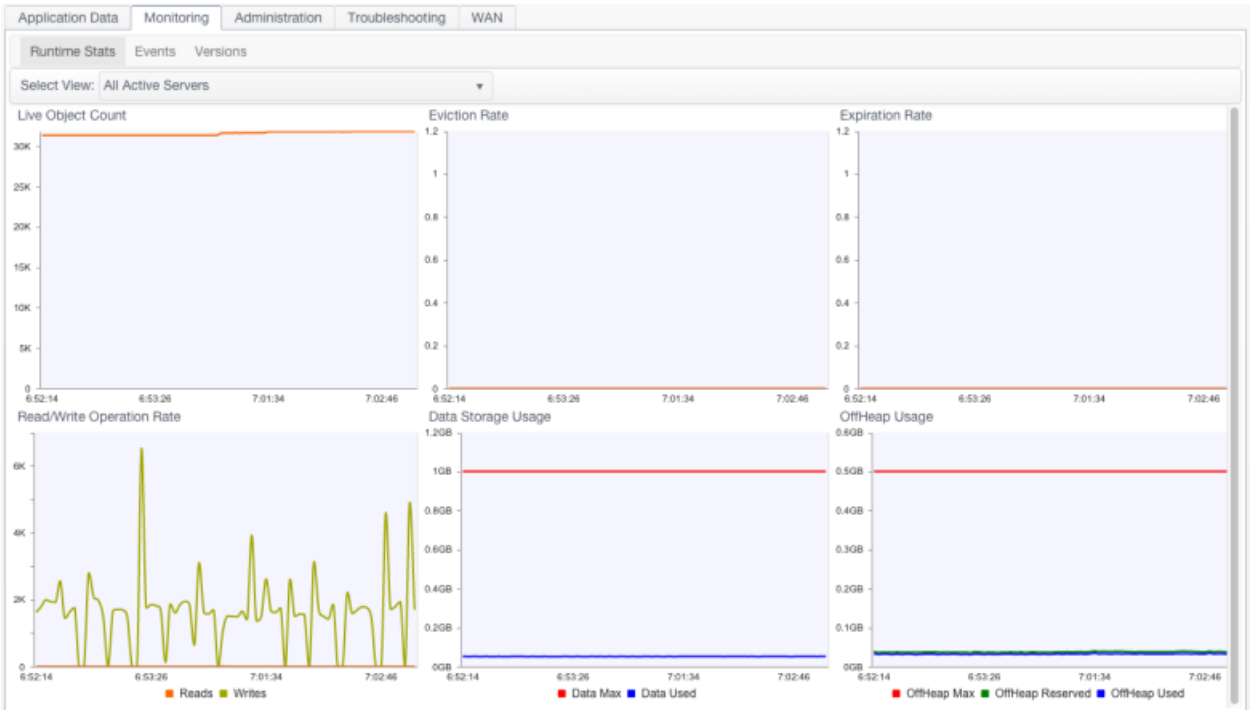
About the Monitoring Tab

The **Monitoring** tab is available only for cluster connection groups.

You use the features on this tab to monitor the functioning of the cluster, as well as the functioning of individual cluster components.

Runtime Statistics

The Runtime statistics graphs provide a continuous feed of server and client metrics. Sampling begins automatically when a runtime statistic panel is first viewed, but historical data is not saved.



Use the **Select View** menu to set the runtime statistics view to one of the following:

- **All Servers** - Display aggregated statistics for the TSA.
- **Specified mirror group** - Display aggregated statistics for the selected mirror group.
- **Specified server** - Display runtime statistics for the selected server.
- **All Clients** - Display aggregated statistics for all of the cluster's clients.
- **Specified client**- Display runtime statistics for the selected client.

Specific runtime statistics are defined in the following sections. The cluster components for which the statistic is available are indicated in the text.

Live Object Count

Shows the total number of live objects in the cluster, mirror group, server, or clients.

If the trend for the total number of live objects goes up continuously, clients in the cluster will eventually run out of memory and applications might fail. Upward trends indicate a problem with application logic, garbage collection, or the tuning of one or more clients.

Eviction Rate

Shows the number of entries being evicted from the cluster, mirror group, or server.

Expiration Rate

Shows the number of expired entries found (and being evicted) on the TSA, mirror group, or server.

Read/Write Operation Rate

Shows the number of completed writes (or mutations) in the TSA or selected server. Operations can include evictions and expirations. Large-scale eviction or expiration operations can cause spikes in the operations rate (see the corresponding evictions and expirations statistical graphs). This rate is low in read-mostly situations, indicating that there are few writes and little data to evict. If this number drops or deviates regularly from an established baseline, it might indicate issues with network connections or overloaded servers.

When clients are selected, this statistic is reported as the **Write Transaction Rate**, tracking client-to-server write transactions.

A measure of how many objects (per second) are being faulted in from the TSA in response to application requests. Faults from off-heap or disk occur when an object is not available in a server's on-heap cache. Flushes occur when the heap or off-heap cache must clear data due to memory constraints. Objects being requested for the first time, or objects that have been flushed from off-heap memory before a request arrives, must be faulted in from disk. High rates could indicate inadequate memory allocation at the server.

Data Storage Usage

BigMemory Max provides support for a "hybrid" mix of solid-state device (SSD) "flash drives" along with the standard DRAM-based off-heap storage. This Data Storage Usage graph, when compared to the OffHeap Usage graph, shows that the hybrid maximum data storage, which includes both off-heap memory and any "flash drives," can be on an entirely larger scale than off-heap alone.

Data Max shows the total amount of data storage. This is the configured amount (dataStorage size in the tc-config.xml) that can be stored in BigMemory, both in off-heap DRAM and in any SSD flash drive, if BigMemory Hybrid is configured. **Data Used** shows the amount of the data storage that is currently in use.

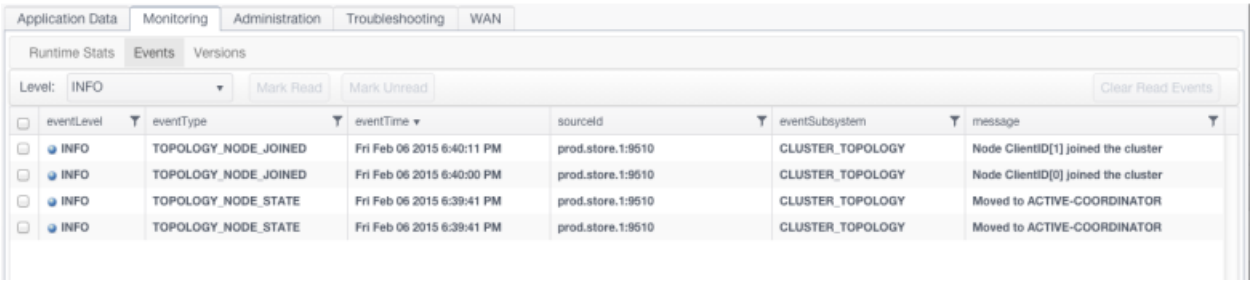
OffHeap Usage

Shows the amount, in megabytes or gigabytes, of maximum available off-heap memory (configured limit), the "OffHeap Reserved" (made available), and used off-heap memory (containing data). These statistics appear only if off-heap memory is configured.

OffHeap Max shows the configured maximum amount of off-heap memory (offheap size in the tc-config.xml). **OffHeap Reserved** shows the amount of off-heap memory that is currently available. **OffHeap Used** shows the amount of off-heap memory currently in use (containing data).

Events

The **Events** panel displays cluster events received by the Terracotta server array. You can use this panel to quickly view these events in one location in an easy-to-read format, without having to search the Terracotta logs.



The number of unread events is shown in a badge on each clustered connection's mini dashboard. The badge color indicates the severity of unread events: red for warnings and above, or gray if all unread events are of lower severity.

From Monitoring > Events, the **Level** dropdown list allows you to select DEBUG, INFO, WARN, ERROR, or CRITICAL. Events will display that are equal to or higher than the level you select. For example:

- If you select INFO, only DEBUG events are filtered out.
- If you select WARN, you will see events at the WARN, ERROR, and CRITICAL levels.

Click any column head to order the list. Use the buttons to mark events as read or unread, and to clear read events.

For more information about event types, see "Monitoring Cluster Events" in the *BigMemory Max Administrator Guide*.

Note:
In respect of considerations concerning EU General Data Protection Regulation (GDPR), be aware that in cases of incorrect login procedure or other error scenarios, LDAP username and IP address may be logged in the tmc-security log file. Amongst other events, such data may be collected for reasons of configuring messages based on such logs. LOG4J offers possibilities for log-purging and log-retention, which may offer useful strategies to avoid unwanted loss or exposure of sensitive data possibly conflicting with regulations.

Versions

The **Versions** panel displays version information for all members of the cluster. Note that all nodes must run the same major version, however, from BigMemory Max 4.2.0 on, nodes running different minor versions, such as 4.2.1 or 4.2.2, are allowed.

5 Using the Administration Tab

About the Administration Tab

The **Administration** panels provide information about the Terracotta cluster as well as tools for operations, including backing up cluster data.

Configuration

Using subpanels, the **Configuration** panel displays the status, environment, and configuration information for the servers and clients selected in the **Cluster Node** menu. This information is useful for debugging and when reporting problems.

The **Main** subpanel displays the server status and a list of properties, including IP address, version, license (capabilities), and restartability and failover modes. A specific server must be selected to view this subpanel. Administrators can shut down servers from this panel.

The following additional subpanels are available:

- **Environment** - The server's JVM system properties.
- **TCProperties** - The Terracotta properties that the server is using.
- **Process Args** - The JVM arguments that the server was started with.
- **TCConfig** - The Terracotta configuration file that the server was started with.

Backing Up Cluster Data

The **Backups** panel provides a control for creating a backup of cluster data. The following server configuration elements control backup execution:

- `<restartable enabled="true"/>` - Global setting required to be "true" for backups (for all servers) to be enabled. False by default.
- `<data-backup>terracotta/backups` - server-level element setting the path for storing the backup files. The default path is shown.

For more information on restoring from backups, see "Restoring Data from a Backup" in the *Terracotta Server Array Administer Guide*.

Changing Cluster Topology

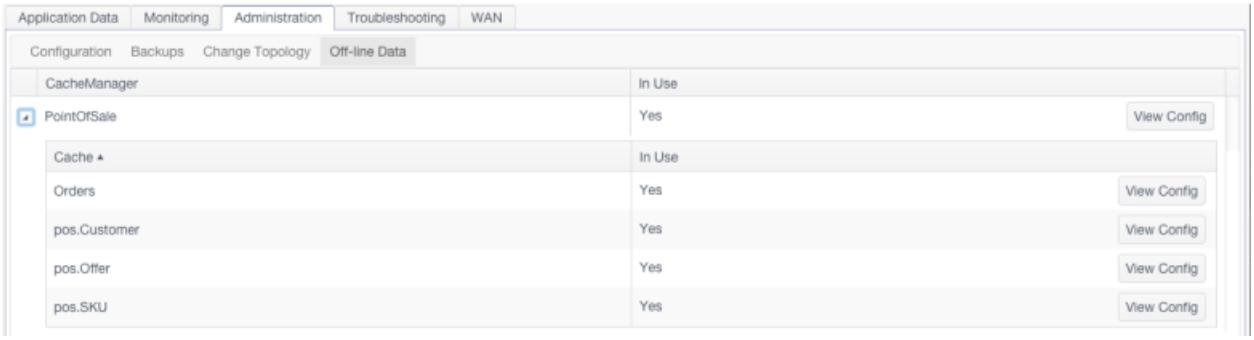
You can reload the Terracotta configuration to add or remove servers. The configuration file must be edited and made available to every server and client before it can be reloaded successfully.

For more information about the Terracotta configuration and editing the servers section, see "Changing Topology of a Live Cluster" in the *BigMemory Max Administrator Guide*.

Off-line Data

Data lifecycle operations have been added to the TMC for more control and visibility of clustered data. The **Off-line Data** panel provides the following capabilities:

- To enumerate caches and cache managers on the server side even when no clients are connected to it.
- To destroy clustered cache when no clients are connected to it.
- To know if clients are connected to the cache.



Only the administrator can see the "Destroy" feature. Use of this feature appears only in the TMC/TMS logs and not in server logs.

6 Using the Troubleshooting Tab

About the Troubleshooting Tab

Troubleshooting Terracotta clusters with the TMC includes passive monitoring through viewing events and statistical trends using the monitoring panels as well as proactively investigating logs and thread dumps. For information about viewing events and monitoring system trends, see "Using the Monitoring Tab" on page 33. For more information about examining logs and generating thread dumps, see "Viewing Server Logs" on page 45 and "Generating Thread Dumps" on page 44.

If a cluster crosses certain resource thresholds, it might enter a mode of limited functionality to prevent a crash. For more information about this behavior, see "Understanding Special TSA Modes" on page 44.

Understanding Special TSA Modes

The Terracotta Management Console flashes warnings if the TSA enters throttled or restricted mode. These modes are initiated if memory resources drop below a certain threshold and endanger the operations of the cluster.

The TSA can automatically recover from throttled mode after sufficient expired data is evicted. Under certain conditions recovery might fail and restricted mode is entered. You can provide temporary relief by clearing or disabling caches. For information about clearing or disabling caches, see "Management Panel" on page 27.

If the TSA enters a throttled or restricted mode, it is an indication that memory resources have been under-allocated. The cluster might need to be stopped and additional steps taken to ensure that enough memory is available to cover cluster operations.

Generating Thread Dumps

You can get a snapshot of the state of each server and client in the Terracotta cluster using thread dumps. To display the thread-dumps feature, click **Troubleshooting**.

The thread-dump navigation pane lists completed thread dumps by date-time stamp. The contents of selected thread dumps are displayed in the right-side pane. To delete all shown thread dumps, click **Clear All**.

To generate a thread dump

1. Choose the target of the thread dump from the **Scope** menu.
2. Click **Take Thread Dump**.
3. Expand the entry created in the thread-dumps navigation pane to see the thread-dump entry or entries.
4. Click a server or client entry to display the thread dump for that server or client.

5. To archive listed thread dumps, click **Download All**.

6. To remove all thread dumps from the list, click **Clear All**.

Servers that appear in the **Scope** menu, but are not connected, produce empty thread dumps.

Viewing Server Logs

Use the following procedure to view the log of each server in the Terracotta cluster.

To view server logs

1. Click **Troubleshooting**, then click **Logs**.

2. Select the server from the **Cluster Node** menu.

3. Click **Pause** (or scroll up) to pause the logs for easier viewing.

4. To dump the cluster state to the log, click the **Dump Cluster State** button.

5. To archive the logs, click **Download Logs**.

7 Using the WAN Tab

About the WAN Tab

The WAN tab enables you to monitor information about the following aspects of the WAN Replication Service:

- Your WAN's topology and configuration.
- Each WAN-enabled cache, including performance statistics and details of their deployment, such as the orchestrator topology, configuration, and status. Terracotta gathers statistics from all Orchestrators and displays them for each cache.

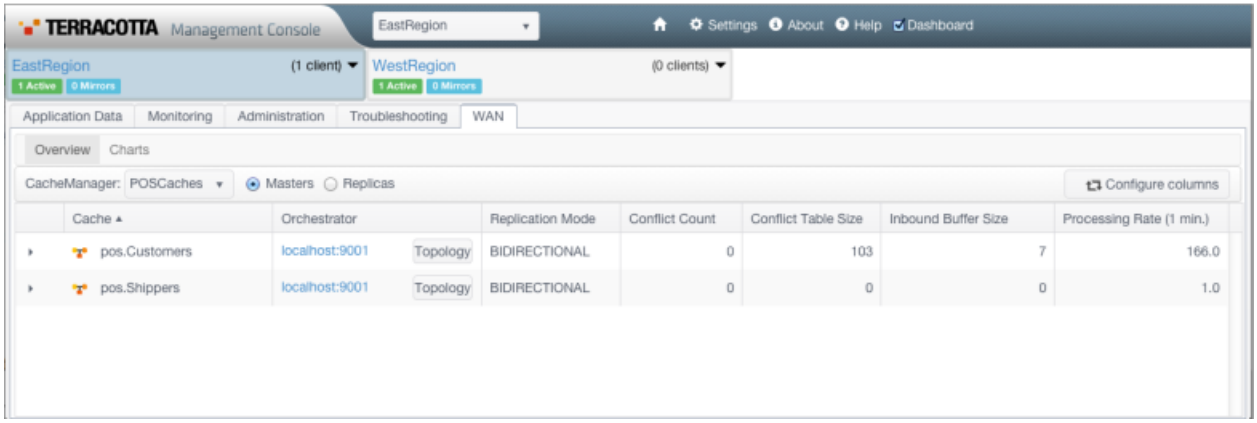
You can display this information in "Overview" format (a table) or in "Charts" format.

Note:
This monitoring capability is enabled by default for each Orchestrator. If you want to disable the monitoring capability, set the `monitoringEnabled` parameter to false in each Orchestrator's wan-config.xml file. For details, see "Orchestrator Configuration Parameters" in the *WAN Replication User Guide*.

Note:
If no caches are currently being replicated, this will be indicated in the WAN tab.

Overview Panel for Master Caches

The Overview panel displays information and performance statistics for caches that are being replicated to remote data centers over a Wide-Area-Network (WAN). This information is provided by the orchestrators performing that replication.



To view master cache information in the Overview panel

1. At the top of the Overview panel select the **CacheManager** of the master caches you want to view.
2. Click **Masters**.

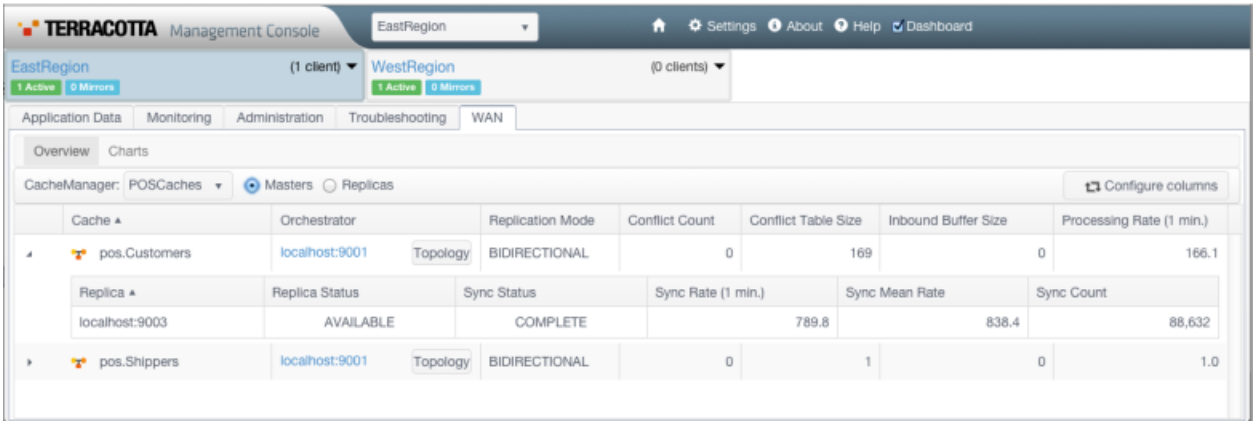
The following columns are displayed by default for each master cache:

Column	Description
Cache	The master cache name.

Orchestrator	The address (the hostname and communication port) of the master cache's Orchestrator process.
	
	To display the Orchestrator's XML configuration, click the address link.
	
	To view information about your WAN's Orchestrator topology, click **Topology**. For example:
	`linux-001.net:9001`
	`linux-002.net:9001`

Replication Mode	The master cache's replication mode (unidirectional or bidirectional).
Conflict Count	The number of conflicts that were resolved during replication.
Conflict Table Size	The number of element modification entries (puts/updates/deletes) that are currently outstanding. These entries may or may not have had a conflict; it simply means that they have not yet been replicated to all the replicas in the WAN. This number could be higher than **Conflict Count**.
Inbound Buffer Size	The size of the cluster listener buffer.
Processing Rate (1 Min.)	The average processed transactions per second (tps) over a one-minute period. A processed transaction is one that has been recorded by the master orchestrator for immediate or eventual transmission to the replica orchestrators.

3. To view synchronization information about the master cache's replica caches, expand the master cache name.



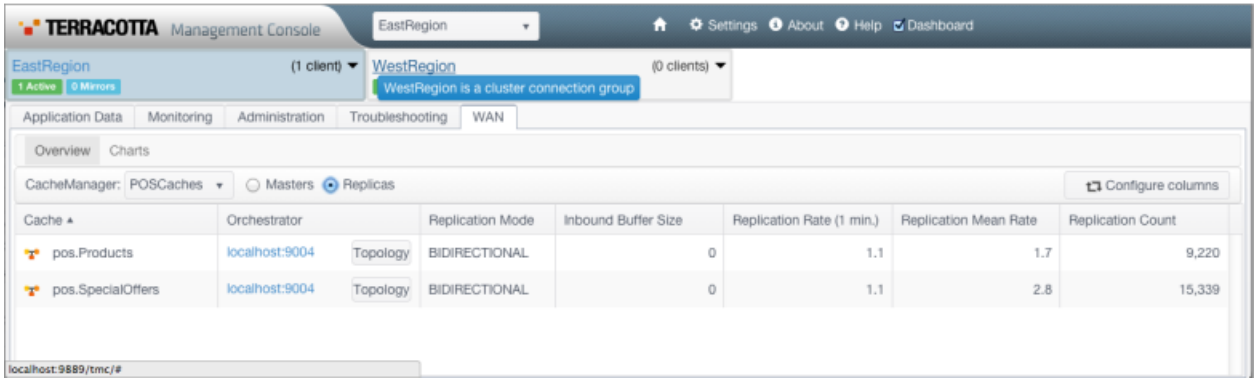
The following columns are displayed by default for each replica cache:

Column	Description
Replica	The replica Orchestrator address (hostname, communication port).
Replica Status	The connection status of the replica cache (AVAILABLE or UNAVAILABLE).
Sync Status	Indicates whether the replica cache's synchronization with the master cache is COMPLETE or RUNNING.
Sync Rate (1 Min.)	The average synchronization transactions per second over the last minute.
Sync Mean Rate	The average transactions per second (tps) over the entire synchronization interval.
Sync Count	The total number of transactions over the entire synchronization interval.

4. To change the default set of statistics displayed, click **Configure columns**. In the pop-up that displays, click either **Master** or **Replica** and select one or more (or all) of the statistics. For descriptions of the statistics, see "The WAN Tab Statistics" on page 53.

Overview Panel for Replica Caches

The Overview panel can display information and performance statistics for CacheManagers and their replica caches, enabling you to you track performance and resource usage across all CacheManagers. This information is collected by the master Orchestrators.



To view replica cache information in the Overview panel

1. At the top of the Overview panel select the **CacheManager** of the replica caches you want to view.

2. Click **Replicas**. If your WAN contains no replica caches, the **Replicas** button is greyed out.

   The following columns are displayed by default for each replica cache:

Column	Description
Cache	The replica Orchestrator address (hostname, communication port).

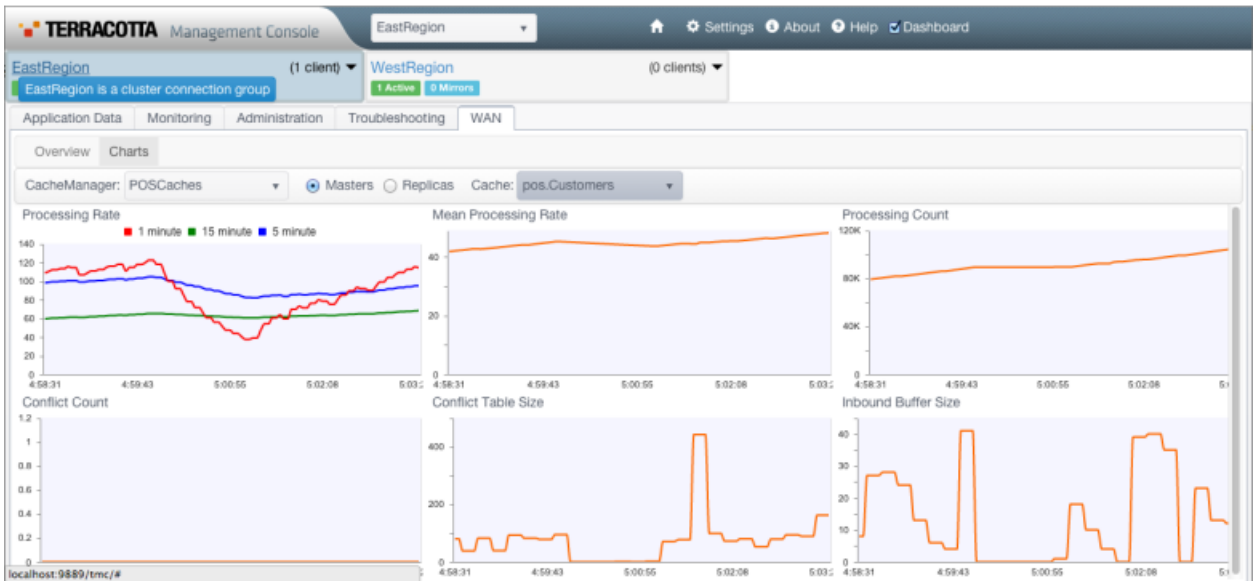
Orchestrator	The address (the hostname and communication port) of the master cache's Orchestrator process.
	
	To display the Orchestrator's XML configuration, click the address link.
	
	To view information about your WAN's Orchestrator topology, click **Topology**. For example:
	`linux-001.net:9001`
	`linux-002.net:9001`

Replication Mode	The replica cache's replication mode (unidirectional or bidirectional).
Inbound Buffer Size	The size of the cluster listener buffer.
Replication Rate (1 Min.)	The average transactions per second (tps) during replication over the last minute.
Replication Mean Rate	The average transactions per second (tps) over the entire synchronization interval.
Replication Count	The total number of transactions over the entire synchronization interval.

3. To change the default set of statistics displayed, click **Configure columns**. In the pop-up that displays, click **Replica** and select one or more (or all) of the statistics. For descriptions of the statistics, see "The WAN Tab Statistics" on page 53.

Charts Panel for Master Caches

Use the Charts panel as an alternative way to view some of the master cache statistics available on the Overview panel.



To view master cache information in the Charts panel

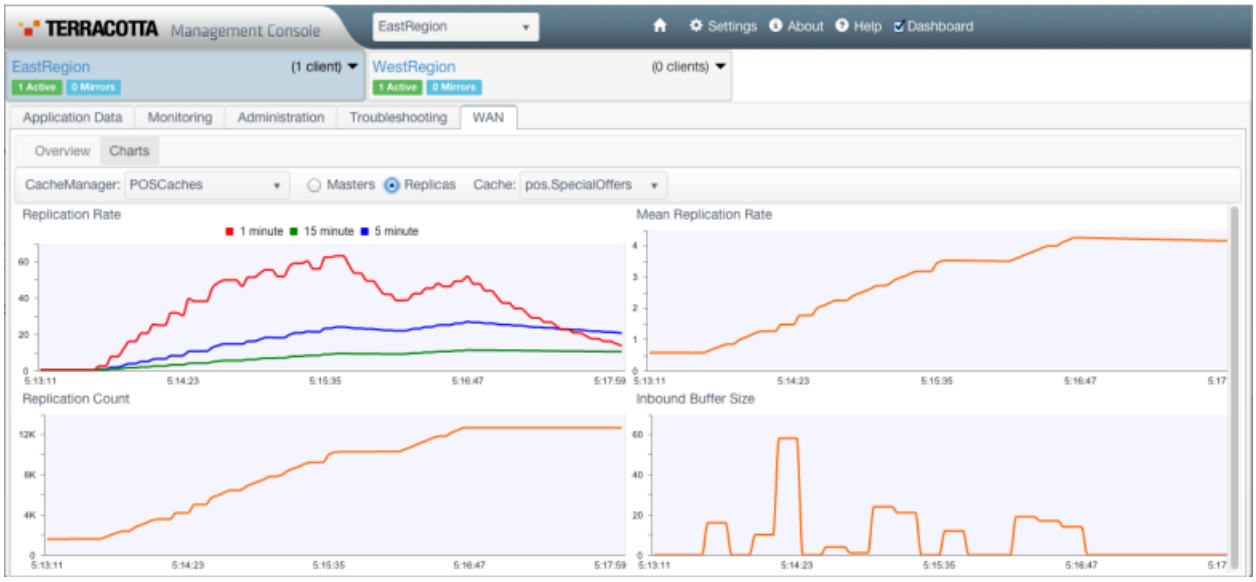
1. On the WAN tab, click **Charts** to display the Charts panel.
2. Select the **CacheManager** of the master cache you want to view.
3. Click **Masters**.
4. In the **Cache** field, select the cache you want to view.

   The following charts are displayed:

   - **Processing Rate:** Charts the average processed transactions per second (tps) during replication. This chart plots the replication rates over one-minute, five-minute, and fifteen-minute periods. A processed transaction is one that has been recorded by the master orchestrator for immediate or eventual transmission to the replica orchestrators.
   - **Mean Processing Rate:** Charts the average processed transactions per second (tps) rate for your application's entire lifetime.
   - **Processing Count:** Charts the total number of processed transactions during your application's entire lifetime.
   - **Conflict Count:** Charts the number of conflicts that were resolved during replication.
   - **Conflict Table Size:**: Charts the number of element modification entries (puts/updates/deletes) that are currently outstanding. These entries may or may not have had a conflict; it simply means that they have not yet been replicated to all the replicas in the WAN. This number could be higher than **Conflict Count**.
   - **Inbound Buffer Size:** Charts the size of the cluster listener buffer.

Charts Panel for Replica Caches

Use the Charts panel as an alternative way to view some of the replica cache statistics available on the Overview panel.



To view replica cache information in the Charts panel

1. On the WAN tab, click **Charts** to display the Charts panel.
2. Select the **CacheManager** of the replica cache you want to view.
3. Click **Replicas**.
4. In the **Cache** field, select the cache you want to view.

   The following charts are displayed:

   - **Replication Rate:** Charts the average transactions per second (tps) during replication. This chart plots the replication rates over one-minute, five-minute, and fifteen-minute periods.
   - **Mean Replication Rate:** Charts the average transactions per second (tps) rate for your application's entire lifetime.
   - **Replication Count:** Charts the total number of transactions during your application's entire lifetime.
   - **Inbound Buffer Size:** Charts the size of the cluster listener buffer.

The WAN Tab Statistics

In the WAN tab you can monitor statistics for the master orchestrator and the replica orchestrator.

To change the default set of statistics displayed on the WAN tab's Overview panels, click **Configure columns** and select one or more of the following statistics for the master orchestrator or the replica orchestrator.

Master Orchestrator Statistics

Statistic	Description
Conflict Count	Number of conflicts resolved. Applicable only in bidirectional mode.
Conflict Table Size	The number of outstanding conflicts. These entries may or may not have had a conflict; it simply means that they have not yet been replicated to all the replicas in the WAN. This number could be higher than Conflict Count. Applicable only in bidirectional mode.
Inbound Buffer Size	The size of the cluster listener buffer.
Processing Rate (1 Min.)	The average processed transactions per second (tps) over the last minute. *A processed transaction is one that has been recorded by the master orchestrator for immediate or eventual transmission to the replica orchestrators.*
Processing Rate (5 Min.)	The average processed transactions per second (tps) over the last five minutes.
Processing Rate (15 Min.)	The average processed transactions per second (tps) over the last fifteen minutes.
Processing Mean Rate	The average processed transactions per second (tps) over the entire application lifespan.
Processing Count	The total number of processed transactions over the entire application lifespan.
Sync Rate (1 Min.)	The average synchronization transactions per second over the last minute.
Sync Rate (5 Min.)	The average synchronization transactions per second over the last five minutes.
Sync Rate (15 Min.)	The average synchronization transactions per second over the last fifteen minutes.
Sync Mean Rate	The average transactions per second (tps) over the entire synchronization interval.
Sync Count	The total number of transactions over the entire synchronization interval.

Replica Orchestrator Statistics

Statistic	Description
Inbound Buffer Size	The size of the cluster listener buffer.
Replication Rate (1 Min.)	The average transactions per second (tps) during replication over the last minute.
Replication Rate (5 Min.)	The average transactions per second (tps) during replication over the last five minutes.
Replication Rate (15 Min.)	The average transactions per second (tps) during replication over the last fifteen minutes.
Replication Mean Rate	The average transactions per second (tps) over the entire application lifespan.
Replication Count	The total number of transactions over the entire application lifespan.

8 Setting up Security

Available Security Levels

The Terracotta Management Server (TMS) includes a flexible, multi-level security architecture to easily integrate into a variety of environments.

The following levels of security are available:

- **No Security:** No authentication, and no or limited secured connections. For additional details, see "No Security" on page 58.
- **Default Security:** Default role-based user authentication only. This is built in and setup when you first connect to the TMS, and is intended to control access to the TMS. For additional details, see "Default Security" on page 59.

  Standard LDAP and Microsoft Active Directory integration is also available. For information about using LDAP or Active Directory, see "Setting up LDAP or Active Directory Authorization" on page 65.
- **Basic Connection Security:** Authentication and authorization of BigMemory Go and BigMemory Max nodes (referred to as *agents* or *managed agents* in this context), as well as message hashing and other protective measures. For additional details, see "Basic Connection Security" on page 59.

  Secured connections based on Secure Sockets Layer (SSL) technology can be used in conjunction with basic security. For information about using SSL, see "Adding SSL" on page 62.
- **Certificate-Based Client Authentication:** Enhances SSL-based security. In this case, basic security is disabled. For additional details, see "Certificate-Based Client Authentication" on page 63.

With the noted exceptions, these security layers can be used together to provide the overall level of security required by your environment.

This document discusses security from the perspective of the TMS. However, the TMS and the Terracotta Management Console (TMC) function in the same security context.

No Security

Upon initial connection to a properly licensed TMC, the authentication setup page appears, where you can choose to run the TMC with or without authentication.

Authentication can also be enabled or disabled in the TMC Settings panel. If you enable authentication, all of the security features described in this document are available.

If you do not enable authentication, you will be directly connected to the TMC without being prompted for a username and password.

Even with no security enabled, however, you can still force SSL connection between browsers and the Terracotta Management Console. For details, see "Forcing SSL Connections For TMC Clients" on page 65.

Default Security

Default security consists of the built-in role-based accounts that are used to log into the TMC. This level of security controls access to the TMC only, and is appropriate for environments where all components, including the TMC, managed agents, and any custom Rich Internet Applications (RIAs), are on protected networks. An internal network behind a firewall, where all access is trusted, is one example of such an environment. Note that connections between the TMC and managed agents remain unsecured.

Optionally, integration with an LDAP or Microsoft Active Directory is also available. For more information, see "Initial Setup" on page 14.

When TMS/TMC authentication is configured (whether with the .ini file, or LDAP or Active Directory), if a non-Administrator user logs into the TMS/TMC, that user is unable to see the Administration panel in the TMC or perform administrative tasks, such as shutting down a server. However, if a cluster is not secured, a non-Administrator user can use the TMS Rest API to perform administrative tasks on the cluster.

In other words, if you secure the TMS/TMC but do not secure your TSA cluster, any user can perform administrative tasks on the cluster through the Rest API. To prevent this, you must secure both the TMS/TMC and your cluster.

If you are unsure whether your cluster is secured, go to the Connections tab in the Settings window, and look for the locked padlock icon next to your connection.

For more information about TSA security, see the *BigMemory Max Security Guide*.

Basic Connection Security

You can secure the connections between the TMS and managed agents using a built-in hash-based message authentication scheme and digital certificates, also known as "identity assertion" (IA). Use this level of security in environments where the TMS might be exposed to unwanted connection attempts from rogue agents, or where managed agents might come under attack from a rogue TMS.

Note:
To fully secure connections between the TMS and managed agents, it is recommended that SSL be used for encryption. For information about adding SSL to a connection, see "Adding SSL" on page 62.

To set up identity assertion, complete the following steps:

- Set up a truststore as described in "Setting Up a Truststore" on page 60.
- Configure identity assertion as described in "Configuring Identity Assertion" on page 60.
- Create a shared secret for the TMS and the managed agents as described in "Creating a Shared Secret" on page 61.

Setting Up a Truststore

The TMS must have a truststore containing the public-key certificate of every agent that connects to it. If you are not using a Certificate Authority (which provides the public keys), export public keys from the self-signed certificates in the keystore of each agent using a command similar to the following:

```
keytool -export -alias myAgent -keystore keystore-file.jks \
    -file myAgentCert.cert
```

Then import the keys into the TMS truststore, creating it as shown (if it does not already exist):

```
keytool -import -alias myAgent -file myAgentCert.cert \
    -keystore truststore.jks
```

When you use the `keytool` utility, you can maintain additional certificates for the chain of trust in a file `cacerts`. If you wish to use these additional certificates for the import, refer to the use of the option `-trustcacerts` in the documentation of the `keytool` utility.

Tip:
As an alternative to using the command line tool *keytool*, you might want to try the open source graphical tool *KeyStore Explorer*, available at http://www.keystore-explorer.org/index.html.

If a managed agent does not have a keystore, set one up. For examples, see the *BigMemory Max Security Guide*.

Make your truststore available to the TMS in one of the following ways:

- `${user.home}/.tc/mgmt/tms-truststore`
- a location configured with the system property `javax.net.ssl.trustStore`

Alternatively, you can import these public keys into the default truststore for the JVM (typically the cacerts file).

Note:
If a different default location for TMS-related files is required, set it using the system property `com.tc.management.config.directory`.

Configuring Identity Assertion

To configure identity assertion (IA) for the Terracotta Server Array, see the *BigMemory Max Security Guide*.

To configure IA on a Terracotta client, enable security (authentication by IA) on the REST service by adding the `securityServiceLocation` attribute to the managementRESTService element in the managed agent's configuration. The following example is for Ehcache:

```
    <ehcache ...>
    ...
      <managementRESTService enabled="true"
      securityServiceLocation="http://localhost:9889/tmc/api/assertIdentity"  />
    ...
```

```
    </ehcache>
```

If `securityServiceLocation` is not set, the authentication feature is disabled. To enable it, set its value to the URI used to connect to the TMC, with `/tmc/api/assertIdentity` appended. In the example above, "http://localhost:9889" is the TMC URI.

For BigMemory Go, use the same procedure as for a Terracotta client.

Creating a Shared Secret

You must create a password (or secret) that is shared between the TMS and managed agents, storing it in a Terracotta keychain file.

The scripts required in the following procedures are found in `${BIGMEMORY_GO_HOME}/management-console/bin` or `${BIGMEMORY_MAX_HOME}/tools/management-console/bin`. Use the equivalent .bat scripts for Microsoft Windows.

Shared Secret on the TMS

1. Create a shared secret for the assertion of trust between the TMS and managed agents by running the following script:

   ```
   ./add-tc-agent.sh <agent-url>
   ```

   where `<agent-url>` is the URI of the agent. This value should correspond exactly to the URI you use in the TMC to connect to the given agent. For example:

   ```
   ./add-tc-agent.sh http://localhost:9888
   ```

   Use `add-tc-agent.bat` with Microsoft Windows.

   The script automatically creates the Terracotta keychain file `<user_home>/.tc/mgmt/keychain` if it does not already exist. Do not move or delete this keychain file because it must remain accessible to the TMS at that location.

2. When prompted, enter a shared secret of your choice. Be sure to remember the secret that you enter because you might need to enter it again in a later step.

3. Run the add-tc-agent script once for each agent, using that agent's URI. The script saves these entries to the same keychain file.

Shared Secret on Managed Agents

1. Each agent with a keychain entry must also have access to the same shared secret through a Terracotta keychain file:

   ```
   ./keychain.sh -c <user_home>/.tc/mgmt/agentKeychainFile \
       http://myHost:9889/tc-management-api
   ```

   where `<tmc-url>` is the URI used to connect to the TMC, with `/tc-management-api` appended. If the named keychain file already exists on the node, omit the `-c` flag. Agents running on the same node can share a keychain file.

2. Enter the master key for the keychain file:

```
Terracotta Management Console - Keychain Client
KeyChain file successfully created in /path/to/agentKeychainFile
Open the keychain by entering its master key:
```

3. Enter the shared secret associated with the TMS:

```
Enter the password you wish to associate with this URL:
Password for http://myHost:9889/ successfully stored
```

   The secret you enter must match the one entered for the TMS. Note that the script's success acknowledgment does *not* confirm that the secret matches the one stored on the TMS.

Adding SSL

In an environment where connections might be intercepted, or a higher level of authentication is required, adding SSL provides encryption. SSL should be used to enhance basic security.

To add SSL to BigMemory Max, see the *BigMemory Max Security Guide*.

To add SSL to BigMemory Go

1. Enable SSL on the REST service by setting the managementRESTService element's sslEnabled attribute to "true" in the managed agent's configuration:

```
<ehcache ...>
...
  <managementRESTService enabled="true"
  securityServiceLocation="https://localhost:9889/tmc/api/assertIdentity"
  sslEnabled="true" />
...
</ehcache>
```

2. Provide an identity store for the managed agent either at the default location, `${user.home}/.tc/mgmt/keystore`, or by setting the store's location with the system property `javax.net.ssl.keyStore`.

   The identity store is where the server-authentication certificate is stored. If the identity store cannot be found, the managed agent fails at startup.

3. Add a password for the managed agent's identity store to its keychain.

   The password must be keyed with the identity-store file's URI. Alternatively, set the password with the system property `javax.net.ssl.keyStorePassword`. If no password is found, the managed agent fails at startup.

4. The JVM running the TMS must have the same server-authentication certificate in one of the following locations:

   - The default truststore for the JVM (typically the cacerts file)

- `${user.home}/.tc/mgmt/tms-truststore`
- A location configured with the system property `javax.net.ssl.trustStore`

If a truststore was already set up for the TMS and it contains the required public key, skip this step. For information about setting up the truststore, see "Setting Up a Truststore" on page 60.

5. If a custom truststore (not cacerts) is designated for the TMS, the truststore password must be included in the TMS keychain.

   The password must be keyed with the truststore file's URI. Alternatively, set the password with the system property `javax.net.ssl.trustStorePassword`.

Certificate-Based Client Authentication

As an alternative to the hash-based message authentication scheme of basic security (as described in "Basic Connection Security" on page 59), you can use certificate-based client authentication with BigMemory Go nodes. This form of authentication is not available for use with the Terracotta Server Array.

Setting up client authentication automatically turns off hash-based authentication. Note that you must configure SSL to use this security option. For procedures, see "Adding SSL" on page 62.

You must set up keystores for all managed agents and a truststore for the TMS as described in "Basic Connection Security" on page 59 and "Adding SSL" on page 62. In addition, you must also set up truststores for all managed agents and a keystore for the TMS, as described in the following procedure.

To enable certificate-based client authentication:

1. Enable client authentication on the REST service by setting the `managementRESTService` element's `needClientAuth` attribute to "true" in the managed agent's configuration:

```
<ehcache ...>
...
  <managementRESTService enabled="true"
  securityServiceLocation="http://localhost:9889/tmc/api/assertIdentity"
  sslEnabled="true" needClientAuth="true" />
...
</ehcache>
```

2. Provide a truststore for the managed agent at the default location, `${user.home}/.tc/mgmt/truststore`, or by setting the truststore location with the system property `javax.net.ssl.trustStore`.

3. The password for the truststore must be included in the managed agent's keychain.

   The password must be keyed with the truststore file's URI. Or set the password with the system property `javax.net.ssl.trustStorePassword`.

4. Provide an identity store for the TMS at the default location, `${user.home}/.tc/mgmt/tms-keystore`, or by setting the identity-store location with the system property `javax.net.ssl.keyStore`.

   The managed agent is rejected by the TMS unless a valid certificate is found.

5. The password for the TMS identity store must be included in the TM keychain, as described in "Creating a Shared Secret" on page 61.

   The password must be keyed with the identity-store file's URI. Alternatively, set the password with the system property `javax.net.ssl.keyStorePassword`.

6. To allow an SSL connection from the managed agent, an SSL connector must be configured. If the TMS is deployed with the provided Jetty web server, add the following to `/management-console/etc/jetty.xml` (in the BigMemory kit) as shown:

```
<Configure id="sslContextFactory"
class="org.eclipse.jetty.util.ssl.SslContextFactory">
    <Set name="KeyStorePath">etc/dev-keystore.jks</Set>
    <Set name="TrustStorePath">etc/dev-keystore.jks</Set>
    <Set name="KeyStorePassword">terracotta</Set>
    <Set name="TrustStorePassword">terracotta</Set>
    <Set name="KeyManagerPassword">terracotta</Set>
    <Set name="NeedClientAuth">true</Set>
</Configure>
```

   Note the following about the configuration shown:

   - If the TMS WAR is deployed with a different container, make the equivalent changes appropriate to that container.
   - The SSL port must be free (unused by any another process) to avoid collisions.
   - maxIdletime can be set to a value that suits your environment.
   - If the default keystore or truststore are not being used, enter the correct paths to the keystore and truststore being used.
   - Passwords have been obfuscated using a built-in Jetty tool:

```
java -cp lib/jetty-util-9.4.11.v20180605.jar org.eclipse.jetty.util.security.Password
 myPassword
```

   The `lib` folder is the `/management-console/jetty-distribution/lib/` folder in your product installation. The jar file name consists of "jetty-util" followed by additional characters. If the name of your `jetty-util` jar file does not match the one shown above, adapt the `java` command accordingly.

   This command returns an obfuscated version of `myPassword`.

Forcing SSL Connections For TMC Clients

If the TMC is deployed with the provided Jetty web server, web browsers connecting to the TMC can use an unsecured connection (via HTTP port 9889). A secure SSL-based connection is also available using HTTPS port 9443.

To force all web browsers to connect using SSL, disable the non-secure connector by commenting it out in `/management-console/start.d/http.ini` (located in the BigMemory kit):

```
#### Connector port to listen on
# jetty.http.port=9889
```

If the TMC WAR is deployed with a different container, make the equivalent changes appropriate to that container.

About the Default Keystore

By default, the built-in Jetty container's configuration file (`management-console/etc/custom-jetty-ssl.xml`) uses a JKS identity store, located in the same directory. This keystore contains a self-signed certificate (not signed by a certificate authority). If you intend to use this "untrusted" certificate, all SSL browser connections must recognize this certificate and register it as an exception for future connections. This is usually done at the time the browser first connects to the TMS.

Setting up LDAP or Active Directory Authorization

When you select either the LDAP or Active Directory authorization method from the TMC authentication page, a setup page opens. Filling out the form on the setup page allows the TMC to use your enterprise directory for authentication and authorization. The information below clarifies what is required in the fields on the setup form.

Enter your directory URL: The complete URL of the LDAP server. The URL has the format `protocol://hostname:portnumber` where the protocol is LDAP for standard connections or LDAPS for secure connections.

- The host is the host name or IP address of the LDAP server.
- The port is the port on which the server is running. The port is optional. If omitted, the port defaults to 389 for LDAP, or 636 for LDAPS.

For example, specifying the URL `ldaps://ldapserv1:700` would create a secure connection to the LDAP server running on the nonstandard port 700 on the host called ldapserv1.

Enter your directory system username: The user ID the TMC should supply to connect to the LDAP server, for example, "Directory manager". This user must have permission to query groups and group membership.

Note:
If your LDAP server allows anonymous access, leave this field blank. If your LDAP does not allow anonymous access, the username must map to a password in the TMC keychain, which

can be configured such as: `bin/keychain.sh -O ~/.tc/mgmt/keychain ldap://admin@localhost:1389`

Static Groups for LDAP

If you want to use a LDAP URL to define a set of rules for explicit group names, consider the following configuration.

Prompt	Example Value
Enter your directory URL	ldap://vminrwa04:1389
Directory System user name	tmcoperatoruser3
Search base	dc=localdomain,dc=com
UserDN Template	uid={0}, ou=Users, dc=localdomain,dc=com
Group DN Template	cn={0}, ou=Groups, dc=localdomain,dc=com
Is your LDAP instance working against dynamic groups?	No
Enter the attribute matching the user with the group	uniqueMember
Admin Groups, Operator group	tmcadminstgroup1,tmcadminstgroup2, tmcadminstgroup3,tmcopstgroup1, tmcopstgroup2 tmcopstgroup3, tmcadminstgroup1,tmcadminstgroup2, tmcadminstgroup3
Operator group	OP,AD
Keychain Formation command	keychain -O -c .tc\mgmt\keychain ldap://tmcoperatoruser3@vminrwa04:1389
Keychain password	manageAD12

Dynamic Groups for LDAP

If you want to use a LDAP URL to define a set of rules that match only for group members, use the dynamic group feature. This alternative to explicit group names works with the filter values you provide. All the members of a dynamic group share a common attribute or set of attributes that are defined in the memberURL filter.

For example, suppose that your organization has two departments Admin and Operator. If the ldap attribute 'departmentNumber' for members of Admin department is AD, and the equivalent for the Operator department is 'OP', configure as follows.

Prompt	Example Value
Enter your directory URL	ldap://vminrwa04:1389
Directory System user name	tmcoperatoruser3
Search base	dc=localdomain,dc=com
UserDN Template	uid={0}, ou=Users, dc=localdomain,dc=com
Group DN Template	cn={0}, ou=Groups, dc=localdomain,dc=com
Is your LDAP instance working against dynamic groups?	Yes
Enter the attribute matching the user with the group	departmentNumber
Admin Groups	AD
Operator group	OP,AD
Keychain Formation command	keychain -O -c .tc\mgmt\keychain ldap://tmcoperatoruser3@vminrwa04:1389
Keychain password	manageAD12

Static Groups for Active Directory

If you want to use a Active Directory URL to define a set of rules for explicit group names, consider the following configuration example.

Prompt	Example Value
Enter your directory URL	ldap://10.60.29.212:389
Directory System user name	tmcoperatoruser3
Search base	DC=igomega,DC=com
Admin Groups	tmcadminstgroup1,tmcadminstgroup2,tmcadminstgroup3
Operator group	tmcopstgroup1,tmcopstgroup2,tmcopstgroup3, tmcadminstgroup1,tmcadminstgroup2, tmcadminstgroup3
Keychain Formation	keychain -O -c .tc\mgmt\keychain ldap://tmcoperatoruser3@10.60.29.212:389
Keychain password	manageAD12

9 Integrating with Nagios

About Integrating with Nagios XI

You can monitor Terracotta nodes using the Nagios XI monitoring solution - see http://www.nagios.com/. To do so, create a Nagios plugin. A Nagios plugin can query the Terracotta Management Server (TMS) for information through the TMS REST interface or directly through a node's REST interface. For information about the REST interface, see the *Terracotta Management REST Developer Guide*.

Plugins can be written in a variety of languages, and should follow the developer guidelines published at http://nagiosplug.sourceforge.net/developer-guidelines.html.

Example of a Shell Script Plugin

The following is an example of a plugin using a shell script. This particular plugin reports an event in Nagios XI when a `node.left` event occurs. A `node.left` event occurs whenever a node leaves the cluster.

For a list of other kinds of events that occur in a Terracotta cluster, see "Monitoring Cluster Events" in the *BigMemory Max Administrator Guide*.

```
#!/bin/bash
# Parameters
# ----------
SERVER=$1    # The IP address or resolvable hostname of a Terracotta server.
PORT=$2      # The Terracotta server's management-port (9540 by default).
INTERVAL=$3  # How far back in time, in minutes, to search for the event.
RESTURL="http://${SERVER}:${PORT}/tc-management-api/agents/
operatorEvents?sinceWhen=${INTERVAL}m"
GET_INFO=`curl "$RESTURL" -s | grep left`
NB_LINES=`echo $GET_INFO | wc -l`
if [[ $NB_LINES -gt 0 ]]; then
       SERVER_LIST=''
       for i in `echo $GET_INFO | sed 's/.*Node\(.*\)left the cluster.*/\1/g'`;
         do SERVER_LIST="$SERVER_LIST $i"; done
       echo $SERVER_LIST
       CHECK="NODE_LEFT"
else
       CHECK="NO_EVENT"
fi
if [[ "$CHECK" == "NODE_LEFT" ]]; then
   echo "NODE LEFT EVENT: $SERVER_LIST"
   exit 2
elif [[ "$CHECK" == "NO_EVENT" ]]; then
   echo "No NODE LEFT Event: ${SERVER}"
   exit 0
else
   echo "Check failed"
   exit 3
fi
```

Note that the script's exit codes follow the standard required for Nagios plugins:

Value	Status	Description
0	OK	The plugin was able to check the service and it appeared to be functioning properly.
1	Warning	The plugin was able to check the service, but it appeared to be above some "warning" threshold or did not working properly.
2	Critical	The plugin detected that either the service was not running or it was above some "critical" threshold.
3	Unknown	Invalid command line arguments were supplied to the plugin or low-level failures internal to the plugin occurred (such as unable to fork or open a tcp socket) that prevent it from performing the specified operation. Higher-level errors (such as name resolution errors or socket timeouts) are outside of the control of plugins and should generally NOT be reported as UNKNOWN states.

After you create the script, install it in Nagios. A number of tutorials on installing Nagios XI plugins are available on the Internet, like the one here: http://www.youtube.com/watch?v=jG1lVnire4E.

You can generalize the script to find other events by editing the REGEX pattern. Or edit the RESTURL to return other types of information.

10 Troubleshooting

Setup Errors

500 Problem Accessing the Keychain File

Problem: After configuring and attempting to use the LDAP or AD URL, you see a message similar to the following:

```
Problem accessing /tmc/setupAuth. Reason:
impossible to initialize the keychain
...
~/.tc/mgmt/keychain doesn't point to a valid file
```

Cause: The keychain file does not exist in the expected location.

Solution: Create the file `keychain` in `$(user.home)/.tc/mgmt` while adding the first entry:

```
bin/keychain.sh -c -O -S ~/.tc/mgmt/keychain ldap://admin@localhost:1389
```

Cannot Retrieve Entry for LDAP or Active Directory User

Problem: After configuring and attempting to use the LDAP or AD URL, you see a message similar to "Impossible to retrieve systemUsername password from the keychain : ldap://admin@localhost:1389".

Cause: The keychain does not contain an entry matching the system user specified.

Solution: Create a correct entry for the specified user. For the example above, the password in the keychain file should be keyed with "ldap://admin@localhost:1389".

Number of Clients Impacts Performance

Problem: You might observe a performance degradation when the number of clients is 200 or more.

Cause: It is possible to have an insufficient number of Java Management Extensions (JMX) threads because the default value of 64 JMX threads is designed for the common scenario in which there are no more than 256 client connections.

Solution: Stop the cluster, adjust the value of `l2.remotejmx.maxthreads` in `tc-config.xml` for your particular environment to ensure there is a JMX thread for every four L1 nodes, then restart the cluster. The following example sets the number of threads high enough to support 1024 clients.

```
<tc:tc-config xmlns:tc="http://www.terracotta.org/config"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.terracotta.org/schema/terracotta-8.xsd">
  <tc-properties>
    <property name="l2.remotejmx.maxthreads" value="256"/>
  </tc-properties>
  <servers>
    ...
  </servers>
</tc:tc-config>
```

Connections Errors

Connection Refused

Problem: An attempt to add a connection to a managed agent is rejected.

Cause: The agent is unreachable or not running.

Solution: Check the following:

- The URI in the connection setup is correct.
- The network connection to the node running the agent is working.
- The agent process is running on the expected node.

404 Connection Not Found

Problem: An attempt to add a connection returns a 404 status code.

Cause: The URI used in the connection setup is incorrect or malformed.

Solution: Check the following:

- The URI in the connection setup is correct.
- The port used in the URI is correct (by default: 9888 for BigMemory Go, 9540 for BigMemory Max).

"A message body reader for Java class... was not found"

Problem: An attempt to add a connection causes the exception "A message body reader for Java class java.util.Collection, and Java type java.util.Collection, and MIME media type unknown/unknown was not found"

Cause: The URI used in the connection setup is incorrect or malformed.

Solution: Check the following:

- The URI in the connection setup is correct.
- The port used in the URI is correct (by default: 9888 for BigMemory Go, 9540 for BigMemory Max).

Connection Timed Out

Problem: An attempt to add a connection fails because the TMS has failed to reach the agent within the timeout limit.

Cause: The agent is unreachable or not running.

Solution: Check the following:

- The URI in the connection setup is correct.
- The network connection to the node running the agent is working.
- The agent process is running on the expected node.

Unexpected End of File From Server

Problem: An attempt to add a connection fails with an EOF error.

Cause: An unsecure connection is being attempted but the agent is set up to use SSL.

Solution: Ensure that the URI is using "https://" not "http://."

"Unrecognized SSL Message, plaintext connection?"

Problem: An attempt to add a connection fails with the error "Unrecognized SSL Message, plaintext connection?".

Cause: A secure connection is being attempted but the agent is not set up to use SSL.

Solution: Ensure that the URI is using "http://" not "https://", or set up SSL on the agent.

Missing Keychain Entry

Problem: An attempt to add a connection fails with the error "Missing keychain entry for URL <agent-url>".

Cause: A connection is being attempted to an agent with identity assertion, but the TMS keychain cannot find that agent's entry.

Solution: Add an entry for the agent using the add-tc-agent script (see “Setting up Security” on page 57).

401 Unauthorized

Problem 1: An attempt to add a connection to an agent configured with identity assertion returns a 401 status code.

Cause: The agent's public key cannot be found in the TMS truststore.

Solution: Import the agent's public key into the TMS truststore (see “Setting up Security” on page 57).

Problem 2: An attempt to add a connection to an agent configured with identity assertion over SSL returns a 401 status code. Errors containing `unknown_certificate` or

```
PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException:

unable to find valid certification path to requested target
```

appear in the agent log and TMS log (see “Logged SSL Connection Errors” on page 77).

Cause: The TMS public key cannot be found in the agent's truststore.

Solution: Import the TMS public key into the agent's truststore (see "Setting up Security" on page 57).

Logged SSL Connection Errors

Certain issues can cause exceptions to appear in the logs when an SSL-enabled connection is attempted. The following list shows parts of log messages that indicate specific exceptions:

- `keyMaterial=null` - The connection URI has not been added to the keychain (see "Setting up Security" on page 57).
- `unknown_certificate` (in the agent log) and `PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: - unable to find valid certification path to requested target` (in the TMS log) - The agent is not using (or cannot find) its keystore (see "Setting up Security" on page 57).
- `unknown_certificate` (in the agent log) and `the counterpart is not ssl compliant` (in the tms log) - The agent is not configured to use SSL (or not configured correctly). Confirm that SSL is set up as shown above.
- `unknown_certificate` (in the TMS log) - Identity assertion (basic TMS security, or IA) is being used over SSL, but the IA URI has not been added to the keychain file. For example:

  ```
  bin/keychain.sh ~/.tc/mgmt/keychain https://localhost:9443/tmc/api/assertIdentity
  ```

  In addition, ensure that the TMS container is configured to use tms-keystore and tms-truststore (see "Setting up Security" on page 57).

Runtime Errors

If CacheException is being thrown as a result of an attempt to perform certain operations in the TMC, see "Bad Cache or CacheManager Names" on page 77.

Display Errors

Bad Cache or CacheManager Names

Using the following characters in the names of caches or CacheManagers causes display and runtime errors: `% | ; , / # & * " < ?`

Issues caused can include statistics not appearing correctly in the **Overview** panel, pop-up TMC error messages in response to an attempt to view cache configuration, and runtime CacheException errors.

Sizing Errors

The TMC displays cache sizing information on certain panels. If there appear to be errors in the way sizing information is displayed for nonstop caches, the sizing operation might be timing out

(it uses the nonstop timeout value). You can tune the sizing operation's timeout value by setting the `bulkOpsTimeoutMultiplyFactor` (see "Tuning for Nonstop Timeouts and Behaviors" in the *BigMemory Max Configuration Guide*).