

# Terracotta Management and Monitoring

Version 10.1

October 2017

This document applies to Terracotta DB and Terracotta Ehcache Version 10.1 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.

Copyright © 2010-2017 Software AG, Darmstadt, Germany and/or Software AG USA Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://softwareag.com/licenses>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

## Table of Contents

<b>Getting Started with the Terracotta Management Console.....</b>	<b>5</b>
<b>Using the Home Page.....</b>	<b>7</b>
<b>Using the Ehcache Tab.....</b>	<b>17</b>
<b>Using the TCStore Tab.....</b>	<b>25</b>
<b>Using the Monitoring Tab.....</b>	<b>33</b>



# 1 Getting Started with the Terracotta Management Console

---

## About the Terracotta Management Console

This document describes the Terracotta Management Console (TMC), which is a browser-based application served by the Terracotta Management Server (TMS). The TMC provides a complete view of your Terracotta Server Array (TSA) and connected clients.

With the TMC you can:

- observe the cluster topology and categories of connected clients
- view statistics
- clear cache contents
- and more

## Installing the TMS

The TMS is a standard Java web application and bundled web container. It gathers information from throughout the cluster and serves it to the TMC for display in your browser. The TMS is located in the installed Terracotta kit under the `tools/management` directory.

## Configuring the TMS

Certain aspects of the TMS can be customized via the properties file located in the installed Terracotta kit at `tools/management/conf/tmc.properties`. While that file contains many more properties, those that follow are the most likely to be useful to the TMS administrator:

```
server.port=9480
tms.storageFolder=target/tmc/data
tms.allowedOrigins=[*]
tms.offheapSizeMB=1024
tms.offheapMonitoringDetail=false
tms.statisticsMaxAgeMinutes=10
tms.eventLogMaxRecords=5000
tms.diagnosticConnectTimeoutSec=5
tms.diagnosticRequestTimeoutSec=5
tms.stripeMonitoringFrequencySec=5
tms.topologyUpdateFrequencySec=10
```

## Starting and Stopping the TMS

To start the TMS, execute the script `start.bat` (on Windows) or `start.sh` (on UNIX-based systems), located in `tools/management/bin` under the installed Terracotta kit.

To terminate the TMS, use the associated `stop.bat` or `stop.sh` script, or platform-provided process management tools.

### Adding Manageability to your Ehcache CacheManager

In order to get the fullest manageability and monitorability you must configure your CacheManager to make use of a ManagementRegistryService:

```
import org.ehcache.config.builders.CacheManagerBuilder;
import org.ehcache.management.registry.DefaultManagementRegistryConfiguration;
import org.ehcache.management.registry.DefaultManagementRegistryService;
DefaultManagementRegistryConfiguration registryConfig =
    new DefaultManagementRegistryConfiguration()
        .setCacheManagerAlias("MyCacheManager");
DefaultManagementRegistryService managementRegistry =
    new DefaultManagementRegistryService(registryConfig);
CacheManagerBuilder<PersistentCacheManager>
    clusteredCacheManagerBuilder =
    CacheManagerBuilder.newCacheManagerBuilder()
        .using(managementRegistry);
```

Associating your CacheManager with a management registry bestows upon it various capabilities allowing it to be exposed via the TMS REST interface with the alias you provide. It is encouraged to assign the same alias to equivalent instances of your CacheManager, across different clients so as to support statistics aggregation.

If you *do not* make use of a management registry, your CacheManager will still appear in the TMC but will not be uniquely named (UNKNOWN) and most management and monitoring features will not operate.

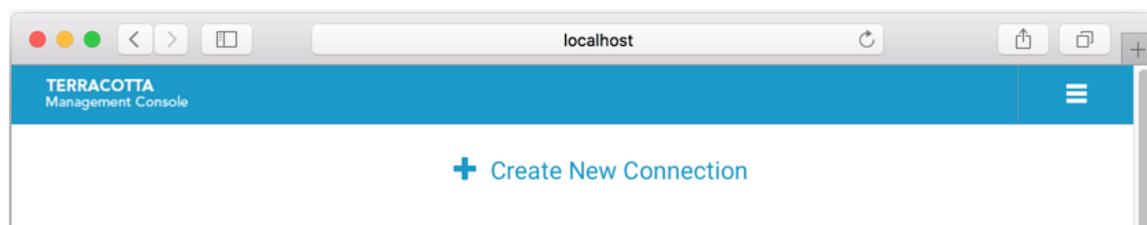
For related information, see the section *Ehcache API Developer Guide > Management and Monitoring with Ehcache*.

### Adding Manageability to your TCStore Dataset

Unlike for an EhcacheCacheManager, there is no need to explicitly add manageability to your TCStoreDataset.

### Connecting to the TMC

After starting the TMS, open a browser and visit `http://localhost:9480`. The TMC will load and present you with the Home Page, where persistent connections to your clusters can be created, viewed and managed.



## 2 Using the Home Page

The TMC home page is where you:

- create/delete persistent connections to your cluster(s)
- optionally modify your connection properties
- view
  - the status of the servers that make up your cluster
  - the various categories of clients making use of your cluster
  - the server entities that are contained by the cluster and to which clients connect
- drill-down/jump to various presentations such as statistics and monitoring relating to those servers, clients, and entities
- take actions, such a clearing the contents of a cache

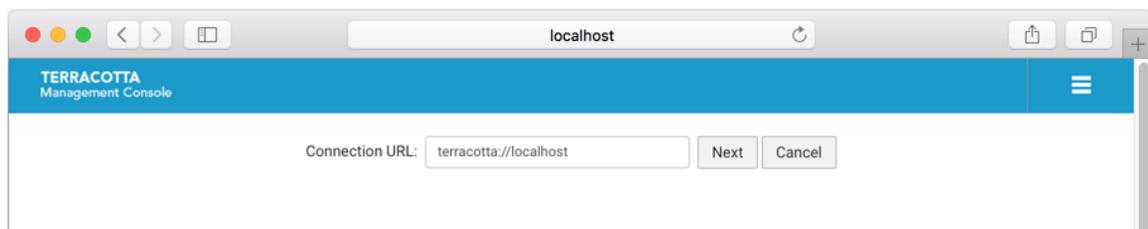
For information about starting and stopping Terracotta servers, refer to the section *Starting and Stopping the Terracotta Server* in the *Terracotta Server Administration Guide*.

For information about creating Terracotta clusters, refer to the section *The Cluster Tool* in the *Terracotta Server Administration Guide*.

### Connections and Global Settings

To create a persistent connection to a particular operational cluster:

1. Click *Create New Connection*
2. In the *Connection URL* input area enter a URL addressing at least a single, running member of the fully configured Terracotta Server Array (TSA):



#### *Terracotta Server URL*

```
terracotta://<server-host>:<listen-port>[,<server-host>:<listen-port>]*
```

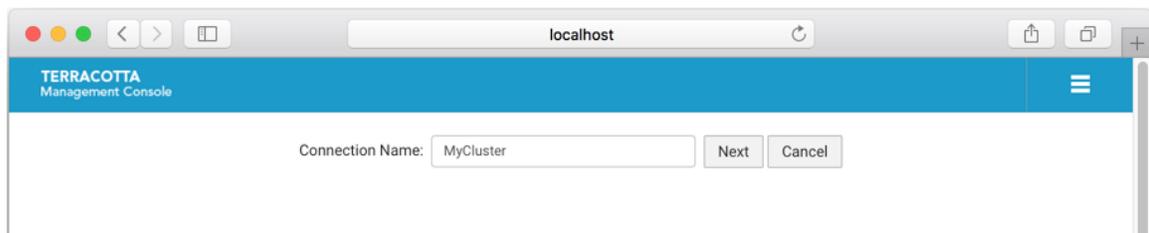
If the TMS is able to connect to the specified server, it will request the complete TSA topology, persisting the addresses of each server in its database. This means that, in the future, the TMS will be able to connect to the TSA, even if the originally specified server should be unreachable, as long as at least one member of the TSA is running.

If you attempt to connect to a running server that is not part of a *configured* cluster, the TMS will return an error indicating that no license was discovered.

**Note:** It is a best-practice to specify the addresses of all the members of a particular TSA stripe in the connection URL to support high-availability. Simply comma-separate each server's address (host:port).

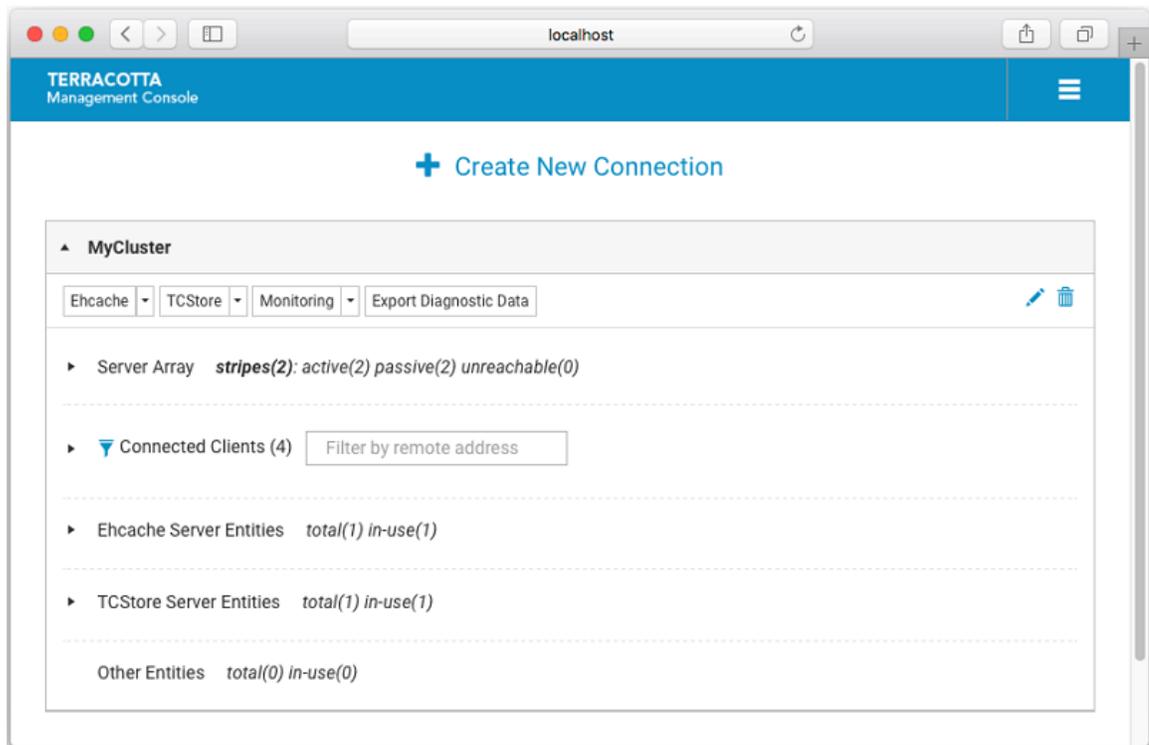
The TMS will attempt to connect to each server, in turn, until a successful connection is established.

3. Enter the name you would like associated with this connection, for example "MyCluster", then click **Next** to create the connection.



This connection name will be used to disambiguate multiple connections in the UI and when communicating with the TMS REST interface.

4. Upon completion of creating the new connection, a new *connection region* is shown on the Home Page, displaying the current state of your cluster.

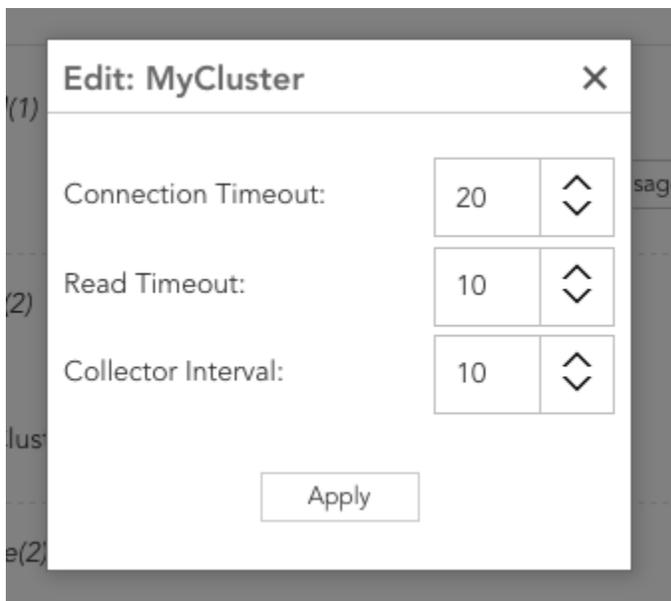


The connection to `MyCluster` in the example above shows a TSA comprised of 2 stripes, each containing a pair of Active-Passive servers, 4 currently connected clients, a single Ehcache server entity and a single TCStore server entity. Further detail can be exposed by drilling down into the display hierarchy.

**Note:** For Terracotta Ehcache product users, the **TCStore** dropdown is deactivated.

### Modifying a Connection's Properties

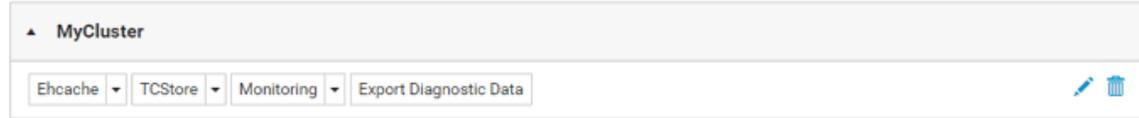
If you wish to modify your new connection's properties, use the *Edit this cluster* (  ) icon on the connection's header area.



Connection Timeout	Timeout value, in seconds, when connecting to a cluster the first time or when reconnecting. Default is 20 seconds.
Read Timeout	Timeout value, in seconds, for all cluster communications. Default is 10 seconds.
Collector Interval	Collector interval for statistics. Default is 10 seconds.

### Deleting a Connection

To delete an existing connection use the *Delete this cluster* (  ) icon on the connection header area.



### Using the Configured Connections

The TMC home page shows each of your configured connections in its own collapsible region of the display that presents the totality of the cluster, allowing you to drill-down to different levels of hierarchy then jump to various detail views, such as for statistics or an entity's server-side resource usage.

Each configured connection region is comprised of several high-level facets which will now be described.

#### Buttons in the connection region header

The header of the connection region for a selected cluster contains the following selectable buttons:

- **Ehcache**
- **TCStore**
- **Monitoring**
- **Export Diagnostic Data**

If you select the **Ehcache**, **TCStore** or **Monitoring** button, the **Detail** page will be displayed, with the appropriate tab (**Ehcache**, **TCStore** or **Monitoring**) already selected.

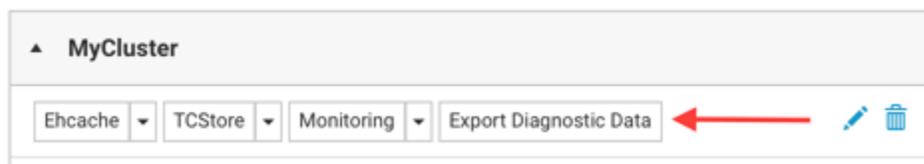
Each of these three buttons contains a dropdown menu. If you select an item from any of these dropdown menus, the **Detail** page will be displayed and the details for the selected dropdown entry will be in focus.

For details of the **Ehcache**, **TCStore** and **Monitoring** tabs of the **Detail** page, see the sections ["Using the Ehcache Tab" on page 17](#), ["Using the TCStore Tab" on page 25](#) and ["Using the Monitoring Tab" on page 33](#).

For information about the button **Export Diagnostic Data**, see the following topic.

#### Export Diagnostic Data (Cluster level)

The **Export Diagnostic Data** button at the cluster level provides diagnostic information for every server in the cluster.



The cluster level **Export Diagnostic Data** button exports the following diagnostic data as a zip file.

- **Event Log** - a csv file named `EventLog.csv`, containing cluster wide events. This can also be exported from the monitoring event log panel.

- For each server it also downloads 6 diagnostic data files, which are explained in the next section, *View Diagnostic Details (Server level)*.

The zip file is named using the format: `diagnostic-connectionName-dateTime.zip`

- e.g. `diagnostic-myClusterConnection-20170705221353.zip`

and the layout of the zip file is:

- one folder for each stripe, which contains one folder for each server within the stripe
- each server folder contains the 6 files referenced in the next section, *View Diagnostic Details (Server level)*
- in the root of the zip file there is the cluster wide event log, `EventLog.csv`

### Server Array

The *Terracotta Server Array (TSA)* is a collection of groups of servers, known as *stripes*. Servers within a stripe work together to provide *High-Availability*. If the *Active* server should fail, any one of the remaining *Passive* servers takes over as active. The *Active* server serves to (1) handle requests from clients to entities it contains and (2) to relay those client requests to each of the *Passive* servers. Servers in different stripes do not interact.

The TMC presents the current state of the server array, indicating the roles of each server. The following shows a server array consisting of two stripes, each containing two members, one *active* and one *passive*.

```

Server Array stripes(2): active(2) passive(2) unreachable(0)
├─ stripe[0] active(1) passive(1) unreachable(0)
│   ├── localhost:19410 (server21) [View Diagnostic Details] [Export Diagnostic Data]
│   └── localhost:19411 (server22) [View Diagnostic Details] [Export Diagnostic Data]
└─ stripe[1] active(1) passive(1) unreachable(0)
    ├── localhost:9410 (server11) [View Diagnostic Details] [Export Diagnostic Data]
    └── localhost:9411 (server12) [View Diagnostic Details] [Export Diagnostic Data]
  
```

The possible server states are:

Server state	Icon	Description
STARTING		Server is beginning to run

Server state	Icon	Description
SYNCHRONIZING		Server is loading data from the active server
PASSIVE		This server has finished synchronizing with the active server and is currently running as a passive server. It is ready to take over as the active server when needed.
ACTIVE		This server is the active server and handles the client requests
ACTIVE_RECONNECTING		Newly started active server that is waiting for existing clients to reconnect
UNREACHABLE		Server is not running or is otherwise inaccessible
UNKNOWN		Unable to determine server status

Diagnostic information for the members of your TSA can be downloaded as an archive file or viewed directly.

#### View Diagnostic Details (Server level)

The **View Diagnostic Details** dropdown at the server level lets you view directly the diagnostic information for that particular server. There is one diagnostic data dropdown per server and it is always positioned to the immediate right of the server name.

Server Array *stripes(2): active(2) passive(2) unreachable(0)*

stripe[0] *active(1) passive(1) unreachable(0)*

 localhost:19410 (server21)

 localhost:19411 (server22)

stripe[1] *active(1) passive(1) unreachable(0)*

 localhost:9410 (server11)

 localhost:9411 (server12)



View Diagnostic Details ▾ Export Diagnostic Data

Environment Export Diagnostic Data

TC Properties Export Diagnostic Data

Process Args  Export Diagnostic Data

TC Config

Cluster State Export Diagnostic Data

Thread Dumps

View Diagnostic Details ▾ Export Diagnostic Data

**Note:** Please don't confuse the stripe count with the stripe name. In this example there are two stripes, indicated by **stripes(2)**, and they are named `stripe[0]` and `stripe[1]`.

### Export Diagnostic Data (Server level)

The **Export Diagnostic Data** button at the server level lets you download diagnostic information for that particular server in an archive file, similar in format to that provided at the cluster level.

The dropdown provides access to the 6 diagnostic data files below. The file contents can be viewed/exported individually by selecting that option from the dropdown or by selecting the top level **Export Diagnostic Data** option which exports a zip file containing all of them.

1. Environment - shows a list of all the environment variables.
2. TC Properties - provides a list of all the TC config properties.
3. Process Args - displays all the command line arguments submitted for the process.
4. TC Config - shows the Terracotta configuration file.
5. Cluster State - displays information on the current cluster state.
6. Thread dump - exports the thread dump as a txt file.

What is a thread dump? A Java thread dump is a way of finding out what every thread in the JVM is doing at a particular point in time. This is especially useful if your Java application sometimes seems to hang when running under load, as an analysis of the dump will show where the threads are stuck.

The zip file is named using the format: `diagnostic-connectionName-stripeName-serverName-dateTime.zip`

- e.g. `diagnostic-myClusterConnection-stripe[0]-testServer0-20170705221356.zip`

and the layout of the zip file is one folder, named after the server name, containing the 6 data files above.

### Connected Clients

In the Terracotta Platform, a *client* is an application end-point. In your application, the clustered `CacheManager` that is configured and initialized **is** a Terracotta client. Each client maintains a connection to the active server in each stripe. In general, anything that connects to a server is considered a client.

For more information, see section *Terracotta Server Administration Guide > Clients in a Cluster*.

Each client has a *Client Identifier* that serves to uniquely identify that client. The form of the identifier is:

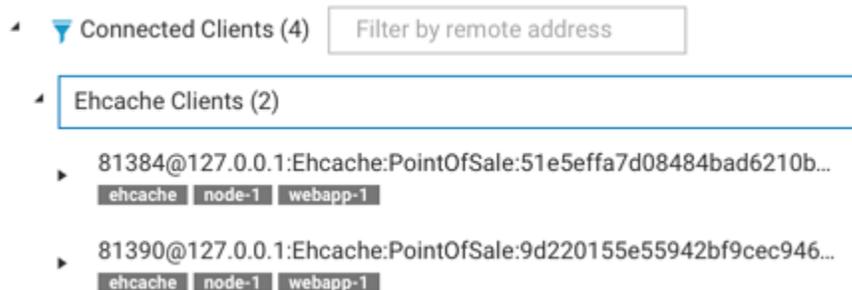
```
<pid>@<ip-addr>:<client-type>:<server-entity-name>:<uuid>
```

where:

<pid>	the <i>process identifier</i> of the Java Virtual Machine hosting this client
<ip-addr>	the <i>IP address</i> of the machine hosting this JVM
<client-type>	[Ehcache   Store   Unknown]
<server-entity-name>	the name of the server-side entity the client is connected to
<uuid>	a unique identifier that serves to disambiguate clients in the same JVM, accessing the same server entity

### ■ Ehcache Clients

The following shows two caching clients.



The input field located next to *Connected Clients* serves to show only those clients whose identifier contains the entered value. In the example above, entering 81390 (the *process identifier* of the 2nd client) would filter out the first client.

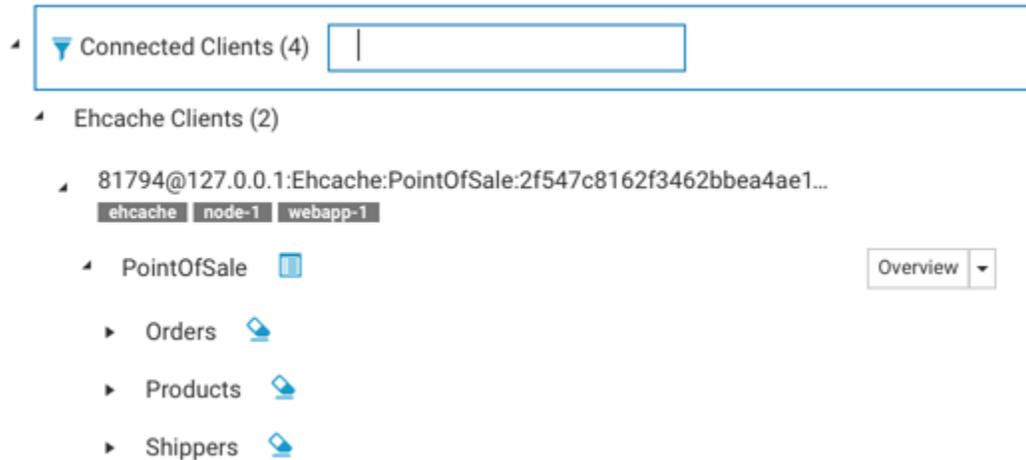
#### *Filtering Rules*

- accepts a space-separated list of terms to match (prefix with ! to negate)
- terms can apply to any components of the *Client Identifier* or any supplied tags
  - negated terms must not match any of the above
- clear the input field content or click the toggle button (  ) to show all clients

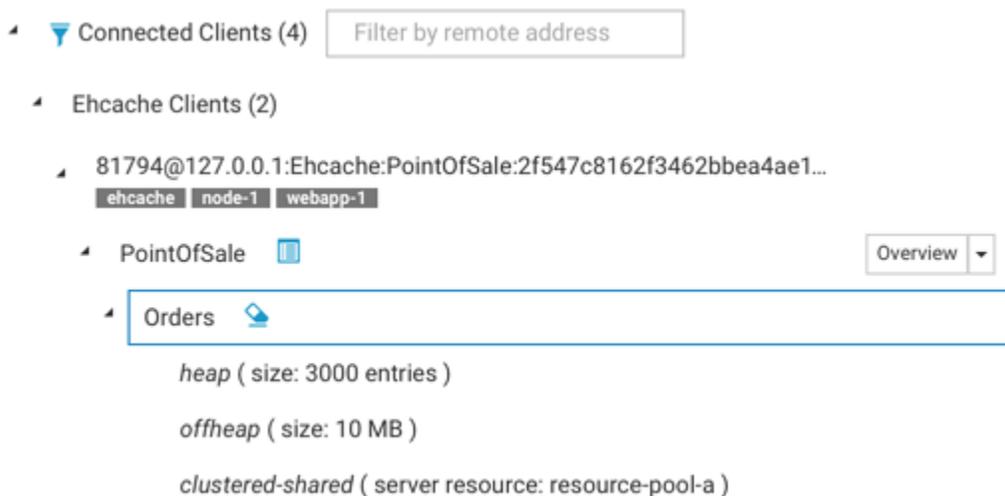
Under each identifier is an optional set of user-defined tags, specified in the CacheManager's configuration.

Expanding a *caching client* exposes the CacheManager's alias, which is defined in the configuration. The dropdown lets you jump to various detail views, pre-selecting this client and CacheManager in those views. View the CacheManager's

configuration by selecting the **configuration** (📄) icon. The cache entries can be cleared by selecting the **clear cache** (🗑️) icon.



Expanding the cache shows important configuration elements related to that cache.



### ■ TCStore Clients

Within the connected clients section there is also a subsection for all the dataset clients.

Expanding the **TCStore Clients** section provides a list of all the connected clients, ordered by client identifiers.

Further expanding a client identifier shows all datasets associated with that particular client and for each dataset there is a dropdown to navigate quickly to its overview or chart statistics.

Lastly when expanding each dataset there is also a list of all its dataset instances



- **Other Clients** This section contains clients that are unknown to the TMS at this time; in other words, anything that is **not** a caching or store client.

### Ehcache Entities

The Terracotta Platform consists of client-side programmatic (API) artifacts and server-side entities that work together to provide highly-available, performant, distributed data access. In your application you configure a CacheManager, both on the client- and server-side. The client-side configuration relates to such things as the maximum OnHeap size-in-bytes for a particular cache. The server-side configuration relates to the remote storage tier that is used to store your cache entries.

Each of your *Ehcache Clients* communicates with its remote storage tier when executing normal cache operations, such as putting an entry into the cache.

Ehcache clients configured to use the same remote storage tier are effectively sharing access to the same cache data.

### TCStore Entities

A TCStore client is essentially a connection to a clustered or embedded Dataset.

A single client instance can *fetch* an arbitrary number of *handles* to the underlying Dataset, referred to as a Dataset instance.

Operations statistics are maintained on a per-instance basis.

### Other Entities

Server entities created by 3rd parties, and therefore unknown to the TMC, appear here.

Currently there is not much that can be done with these entities, but a future release may allow for the declarative exposure of management and monitoring features from entities.

## 3 Using the Ehcache Tab

The **Ehcache** tab of the **Detail** page contains detailed presentations relating to your caching clients. The **Detail** page is selectable via buttons in the home page (see the section "[Using the Configured Connections](#)" on page 10 for details).

You can view:

- Overview statistics concerning this moment in time, in grid form. Select this view by clicking the icon .
- Historical statistics, in chart form. Select this view by clicking the icon .
- Cache size information. Select this view by clicking the icon .

Your caching client automatically gathers and periodically sends to the TMS low-level statistics concerning counters and sizing. From those low-level statistics the TMS can synthesize a variety of derived statistics, such as rates (HitRate) and ratios (HitRatio).

The following categories of statistics are available:

Category	Description
Cache	encompasses all the following caching tiers
OnHeap	the JVM heap tier
OffHeap	the JVM OffHeap (direct memory) tier
Disk	the disk tier
Clustered	the cluster tier

The following raw counter values, as well as associated rate of change, are available for each category listed above:

Counter	Description
<Category>:PutCount	number of times your application has put a new entry into the cache
<Category>:UpdateCount	number of times your application has put a new entry into the cache that replaced an existing entry (same key)

Counter	Description
<Category>:RemovalCount	number of times your application explicitly removed an entry from the cache
<Category>:ExpirationCount	number of entries removed from the cache due to an expiration policy (time-to-live, etc.)
<Category>:EvictionCount	number of entries removed from the cache due to space constraints
<Category>:HitCount	number of gets that returned an existing entry
<Category>:HitRatio	ratio of hits to gets
<Category>:MissCount	number of gets that did not return an entry
<Category>:MissRatio	ratio of misses to gets

The following category-specific raw counters are also available:

Counter	Description
Disk:MappingCount	number of entries stored on disk
Disk:OccupiedByteSize	total size of all entries stored on disk
OffHeap:MappingCount	number of entries stored in direct memory
OffHeap:OccupiedByteSize	total size of all entries stored in direct memory
OnHeap:MappingCount	number of entries that store in regular heap memory
OnHeap:OccupiedByteSize	total size of all entries store in regular heap memory

**Note:** Ratios and Rates

The TMS gathers raw counters from your application and the servers. Rates of change (rates) and ratios are synthesized based on windowed sample averaging. For example, the TMC requests 30 samples over the last 5 minutes for display in the Charts panels. The median raw counter values are calculated for each window, from which rates and ratios can be derived.

The units for rates are always operations per second. Ratios are unitless.

## Overview Panel

The Overview Panel lets you view statistics concerning what is happening right now (or just a while ago) with your caching client, broken out by cache. By default the statistics are aggregated across all clients using the same CacheManager. Any particular caching client can also be selected for viewing.

CacheName	Cache:HitCount	Cache:HitRate	Cache:HitRatio	Cache:MissCount	Cache:MissRate	Cache:MissRatio
Orders --> Orders	0	0.00	NaN	0	0.00	NaN
Shippers --> Shippers	6.1K	2.10	53 %	6.1K	1.90	48 %

Use the **CacheManagers** dropdown to select the aliased CacheManager for which to show statistics. The entries in this dropdown are of the form *CacheManager Alias > Storage Handle* due to the fact that different client-side CacheManagers can be configured to use the same storage on the TSA. Furthermore, CacheManagers that are otherwise identical can be configured with different aliases. We show the mapping here so you have a chance to disambiguate CacheManagers should you choose to configure them in this way.

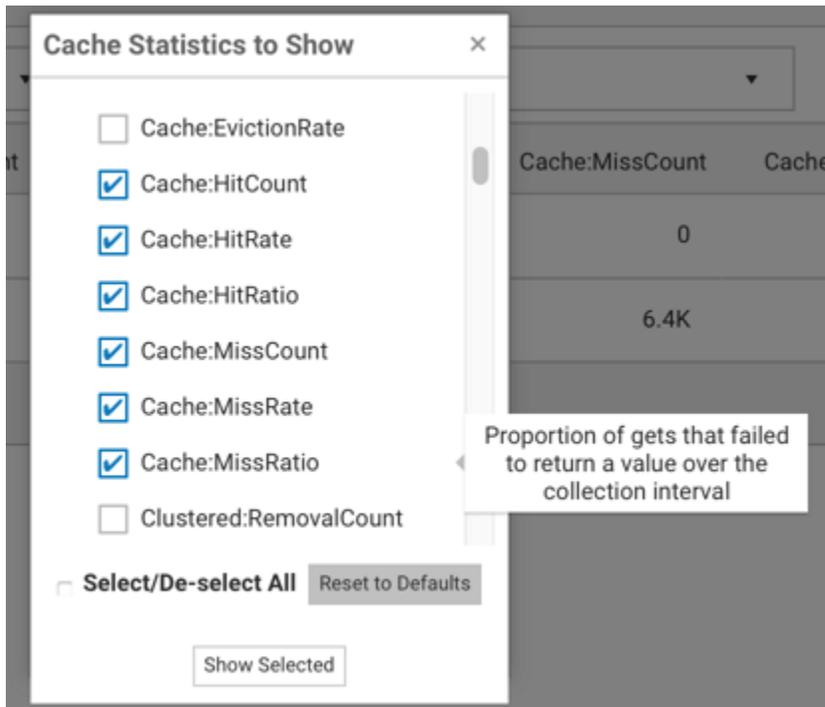
Use the **Clients** dropdown to select a particular caching client for viewing. By default, statistics are aggregated across all clients of the selected CacheManager.

### Additional Grid Features

- Temporarily re-arrange grid columns via drag-and-drop
- Sort on columns or filter on CacheName
- Scroll horizontally if the grid columns overflow the available space

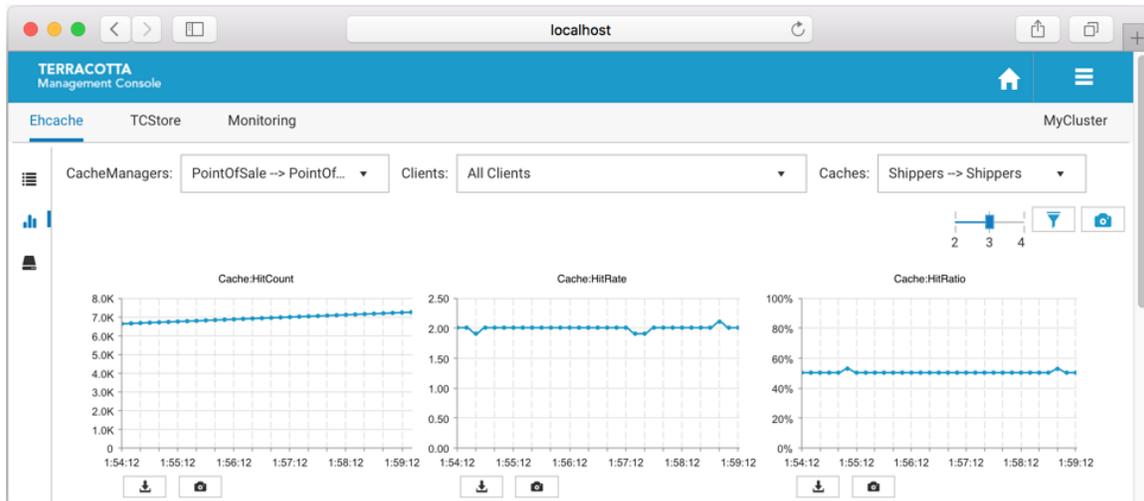
Use the **Export** (  ) icon to download a spreadsheet of the currently displayed values.

Use the **Filter cache statistics** (  ) icon to select which statistics to display. The set of statistics to view defaults to **Cache**-level statistics that are applicable to all use-cases.



## Charts Panel

The **Charts** Panel lets you view statistics over the past 5 minutes. The TMC requests 30 samples be returned. Unlike the **Overview** Panel, which shows all the caches contained by the selected CacheManager, the **Charts** Panel shows statistics for particular caches.



Use the **CacheManagers** dropdown to select the aliased CacheManager for which to show statistics. The entries in this dropdown are of the form **CacheManager Alias > Storage Handle** due to the fact that different client-side CacheManagers can be configured to use the same storage on the TSA. Furthermore, CacheManagers that are otherwise identical can be configured with different aliases. We show the mapping here so you have a chance to disambiguate CacheManagers should you choose to configure them in this way.

Use the **Clients** dropdown to select a particular caching client for viewing. By default, statistics are aggregated across all clients of the selected CacheManager.

Use the **Caches** dropdown to select a particular cache for viewing. By default, the first cache listed is selected.

Use the slider (  ) to set how many columns of charts you would like displayed.

Use the **Filter cache statistics** (  ) icon to select which statistics to display. The set of statistics to view defaults to **Cache**-level statistics that are applicable to all use-cases.

Use the **Take a snapshot of all charts** (  ) icon to download a single PNG file containing the current values of all displayed charts.

Directly under each individual chart, use the **Export** (  ) icon or the **Export to PNG image** (  ) icon to download the chart in the selected format.

#### *Additional Chart Features*

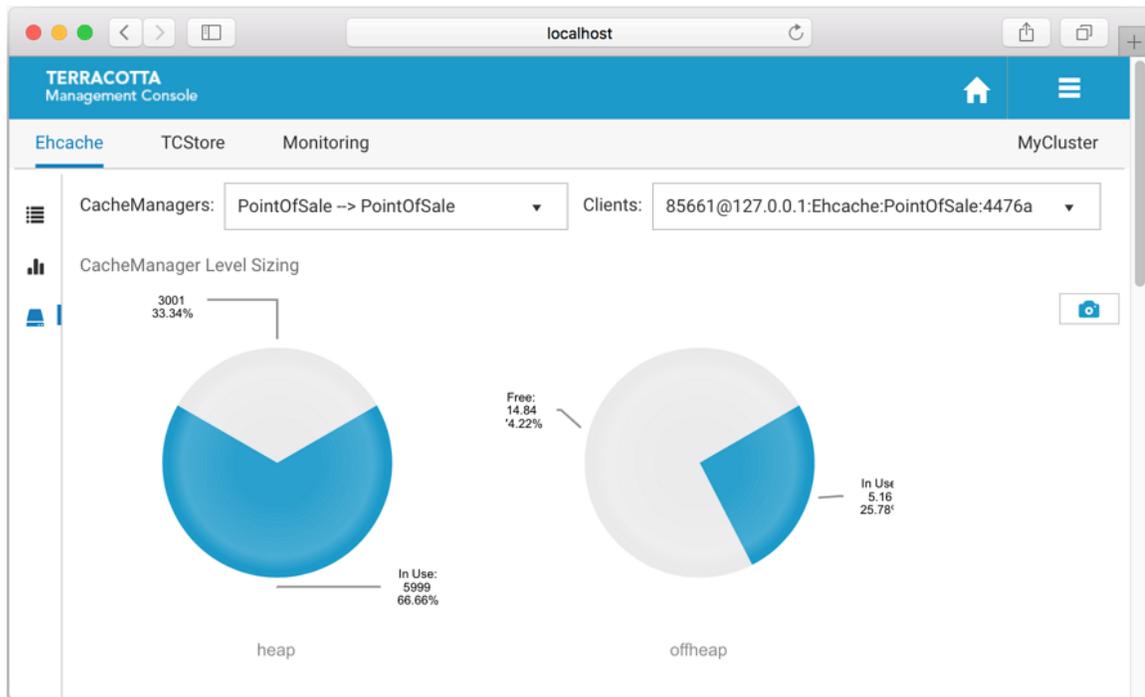
- Temporarily re-arrange charts via drag-and-drop
- Double-click a chart to get an enlarged snapshot

### **Sizing Panel**

The Sizing Panel shows how much space is being used by your CacheManagers and caches across the different local storage tiers you've configured. Sizing information related to the *Clustered Storage Tier* can be accessed via the **Remote** buttons.

The **Sizing** Panel is composed of two sections: *CacheManager Level Sizing* and *Cache Level Sizing*.

- **CacheManager Level Sizing**



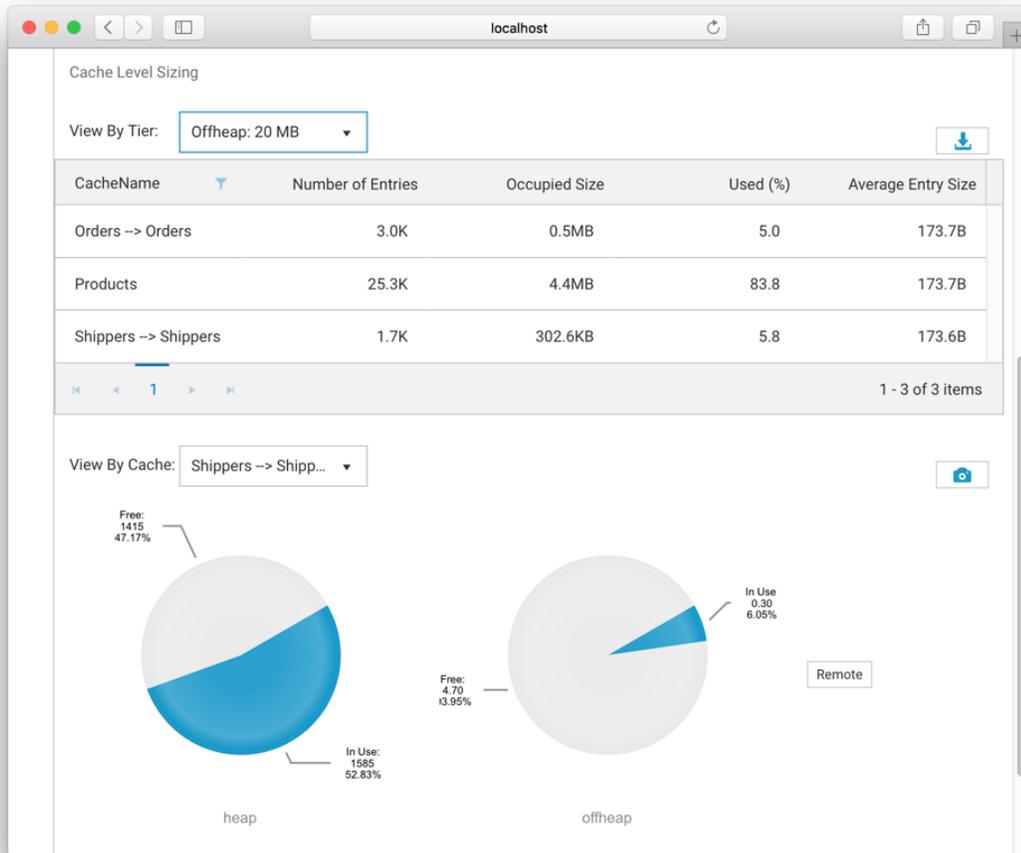
Shown is a pie chart for each local storage tier you've configured for use by your caches, displaying the amount of the available storage that is being used.

Use the **CacheManagers** dropdown to select the aliased CacheManager for which to display sizing information.

Use the **Clients** dropdown to select a particular caching client for which to display sizing information. By default the first client listed is selected.

Use the **Remote** button to navigate to the Resource Usage panel to view how the server-side caching entity is making use of the server's configured resources, such as OffHeap storage. See the section "[Resource Usage Panel](#)" on page 33 for related information.

#### ■ Cache Level Sizing



Use the **View By Tier** dropdown to view tier-specific sizing information for all caches contained by the selected CacheManager in grid form.

**Note:** *Byte-sizing Limitations*

Cache tiers that are sized by *entries* cannot provide *Occupied Size* or *Average Entry Size* and are displayed as N/A for *Not Applicable*.

Use the **View By Cache** dropdown to view the tier usage breakdown for a particular cache, in pie chart form.



## 4 Using the TCStore Tab

The **TCStore** tab of the **Detail** page opens a view containing detailed statistics for TCStore dataset operations. The **Detail** page is selectable via buttons in the home page (see the section "[Using the Configured Connections](#)" on page 10 for details).

There are two TMC panels that show visualizations of the dataset operation statistics:

1. **Overview** panel: displays the latest statistics for datasets and dataset instances in a tabular layout. Select this view by clicking the icon .
2. **Charts** panel: presents a historical view of dataset statistics over a period of time via a graphical layout. Select this view by clicking the icon .

**Note:** A dataset (e.g. `dataset1`) can have multiple instances and you can distinguish each dataset instance by its name, which will end in a dash '-' followed by a number. Thus an instance of `dataset1` could be named: `dataset1-1`.

The following raw counter values, as well as associated rate of change, are available:

Counter	Description
Add:AlreadyExists	count of adds that failed due to a record with the specified key already existing in the dataset
Add:Failure	count of unsuccessful additions to a dataset
Add:Success	count of successful additions to a dataset
Delete:Failure	count of unsuccessful deletions from a dataset
Delete:NotFound	count of deletes that failed due to a record with the specified key not existing in the dataset
Delete:Success	count of successful deletions from a dataset
Get:Failure	count of unsuccessful gets from a dataset
Get:NotFound	count of failed gets that were due to no record with the specified key existing in the dataset
Get:Success	count of successful gets from a dataset

Counter	Description
Update:Failure	count of unsuccessful attempts to update a record in the dataset
Update:NotFound	count of update failures due to no record with the specified key existing in the dataset
Update:Success	count of successful record updates
Stream:Request	count of record streams obtained from the dataset
Stream:Failure	count of unsuccessful stream requests

**Note:** Each of the statistics listed above is a simple counter but each has an associated rate, for instance, `Get:Success:Rate`, whose units are gets per second.

These dataset operation statistics are sent to the TMS periodically by each dataset client and then are available to view in the TMC.

### The TCStore Overview Panel

The TCStore overview panel allows you to see the latest real time statistics for all of your datasets. In addition, the datasets can be filtered by:

- Dataset name
- Clients (Dataset instances in your application)

**Note:** Note that the top highlighted dataset row is an aggregation of all the dataset instance statistics.

### Filter by Dataset Name

The **Datasets** dropdown option allows you to view a selected dataset. The resulting view will show all dataset instances for the selected dataset.

The screenshot shows the Terracotta Management Console interface. The 'TCStore' tab is active. The 'Datasets' dropdown is set to 'RegionalSalesStore' and the 'Clients' dropdown is set to '81797@127.0.0.1:Store:RUxAZHeYcomxRs70...'. The table below shows the following data:

Instance Name	Add:Failure	Add:Success	Delete:Failure	Delete:Success	Get:Failure	Get:Success
RegionalSalesStore	0	12.9K	0	0	0	12.9K
RegionalSalesStore-1	0	0	0	0	0	0
RegionalSalesStore-2	0	10.5K	0	0	0	10.5K
RegionalSalesStore-43	0	463.0	0	0	0	463.0

### Filter by a single client

The **Clients** dropdown option allows you to filter by a particular client or all clients. When selecting a single client the result will only include the dataset and dataset instances of the chosen client.

The screenshot shows the Terracotta Management Console interface. The 'TCStore' tab is active. The 'Datasets' dropdown is set to 'RegionalSalesStore' and the 'Clients' dropdown is set to 'All Clients'. The table below shows the following data:

Instance Name	Add:Failure	Add:Success	Delete:Failure	Delete:Success	Get:Failure	Get:Success
RegionalSalesStore	0	13.0K	0	0	0	13.0K
RegionalSalesStore-1	0	0	0	0	0	0
RegionalSalesStore-2	0	11.0K	0	0	0	11.0K
RegionalSalesStore-45	0	42.0	0	0	0	42.0

### Filter by All Clients

When selecting all clients from the dropdown list the view will slightly change to show all clients for a particular dataset. In this layout the highlighted top row shows the aggregated statistics from all clients for the selected dataset instance. The row directly below the aggregated dataset statistics displays the client identifier, which can be expanded to show statistics for every dataset instance on that client.

TERRACOTTA Management Console

Ehcache TCStore Monitoring MyCluster

Datasets: RegionalSalesStore Clients: All Clients

Instance Name	Add:Failure	Add:Success	Delete:Failure	Delete:Success	Get:Failure	Get:Success
RegionalSalesStore	0	27.8K	0	0	0	27.8K

81797@127.0.0.1:Store:RUx...

Instance Name	Add:Failure	Add:Success	Delete:Failure	Delete:Success	Get:Failure	Get:Success
RegionalSalesStore-1	0	0	0	0	0	0
RegionalSalesStore-2	0	11.5K	0	0	0	11.5K
RegionalSalesStore-47	---	---	---	---	---	---

1 - 3 of 3 items

81802@127.0.0.1:Store:7Uq7...

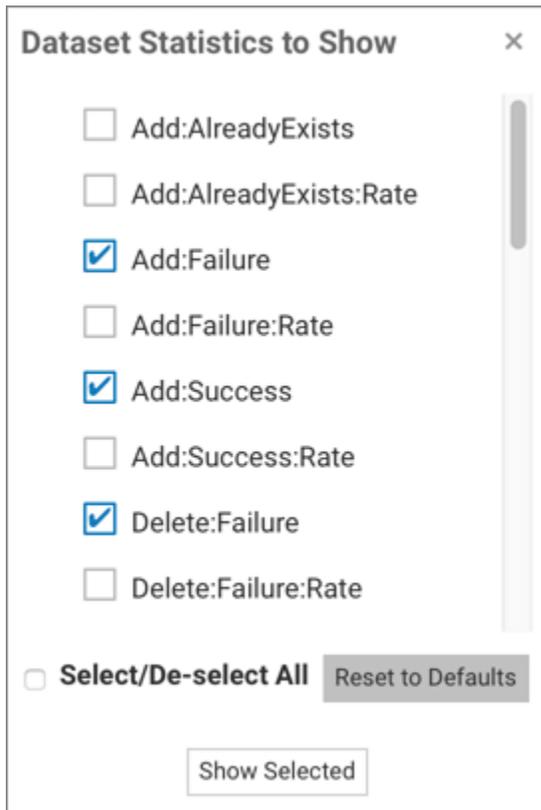
1 - 3 of 3 items

**Note:** Additional Grid Features

- Temporarily re-arrange grid columns using drag-and-drop
- Sort on columns
- Scroll horizontally if the grid columns overflow the available space

Use the Export to Excel (  ) icon to download a spreadsheet of the currently displayed values.

Use the Filter dataset statistics (  ) icon to select which statistics to display.



**Dataset Statistics to Show** ×

- Add:AlreadyExists
- Add:AlreadyExists:Rate
- Add:Failure
- Add:Failure:Rate
- Add:Success
- Add:Success:Rate
- Delete:Failure
- Delete:Failure:Rate

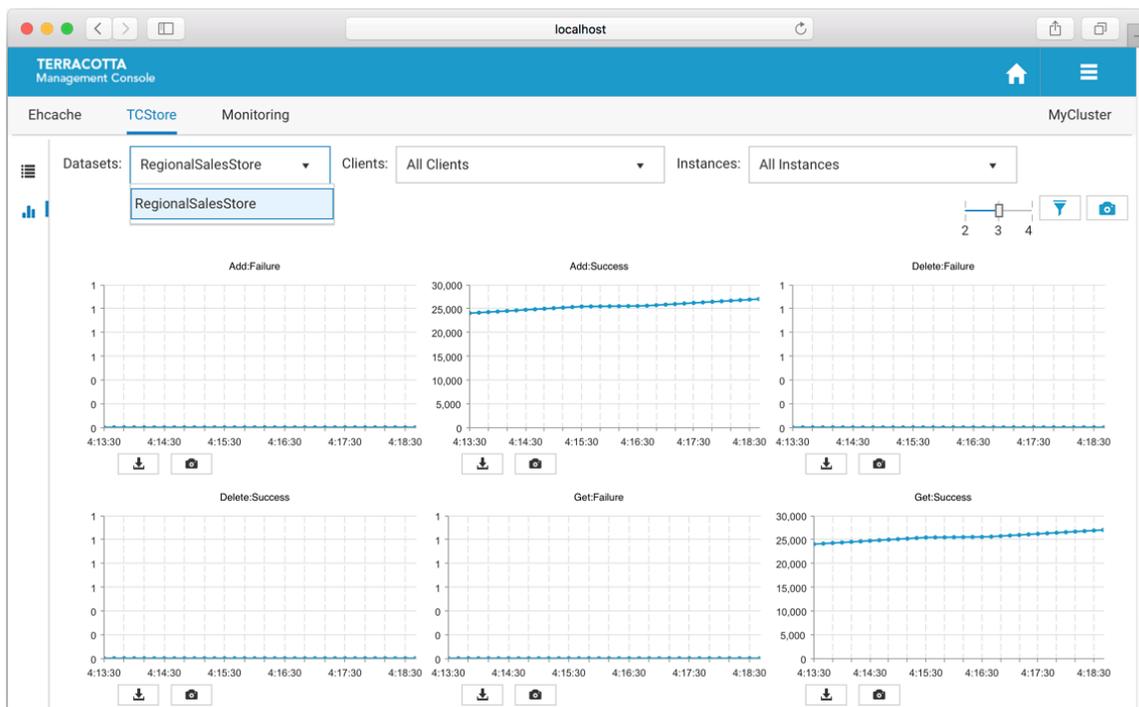
**Select/De-select All**

### The TCStore Charts Panel

The charts panel allows you to view dataset statistics over a period of time. Each statistic is represented in its own chart which shows its values over the last 5 minutes. This 5 minute window also constantly updates to ensure always seeing the most recent historical statistics.

### View by dataset

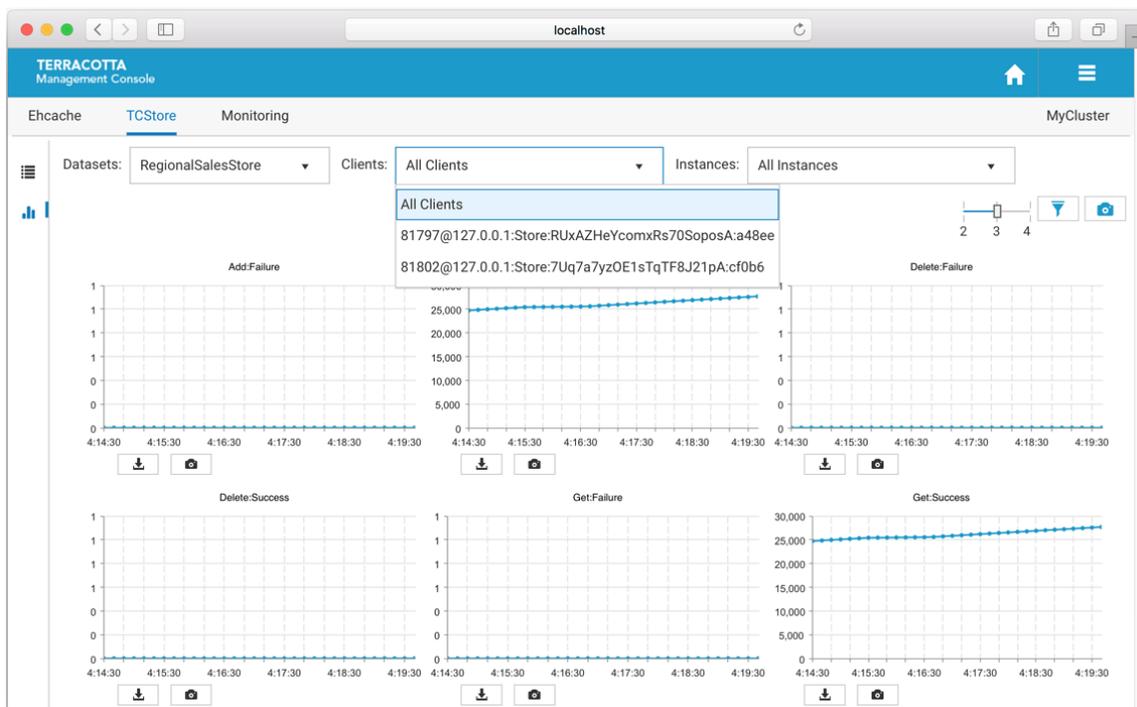
You can select a dataset to monitor.



### Filter by client

There is also an option to filter the history by a specific client or all clients.

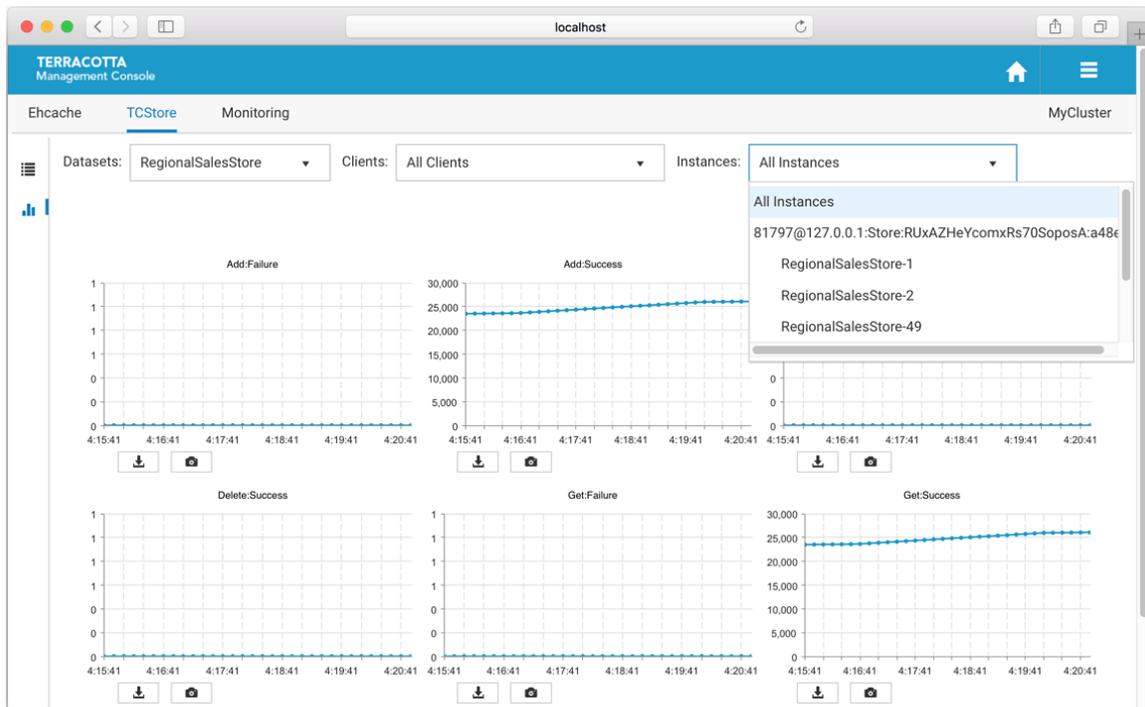
- one client - you only see dataset statistics for the selected client, which is represented as a client identifier in the dropdown
- all clients - the dataset instance statistics are aggregated across all the clients



### Filter by instances

You can filter by dataset instances, either by a specific instance or all instances.

- one instance - this only shows statistics for the selected dataset instance
- all instances - this selection aggregates all dataset instance values for the selected dataset



Use the slider (  ) to set how many columns of charts you would like displayed.

Use the **Filter dataset statistics** (  ) icon to select which statistics to display.

Use the **Take a snapshot of all charts** (  ) icon to download a single PNG file containing the current values of all displayed charts.

Directly under each individual chart, use the **Export to PDF** (  ) icon or the **Export to PNG image** (  ) icon to download the chart in the selected format.

**Note:** Additional Chart Features

- Temporarily re-arrange charts using drag-and-drop
- Double-click a chart to get an enlarged snapshot

## 5 Using the Monitoring Tab

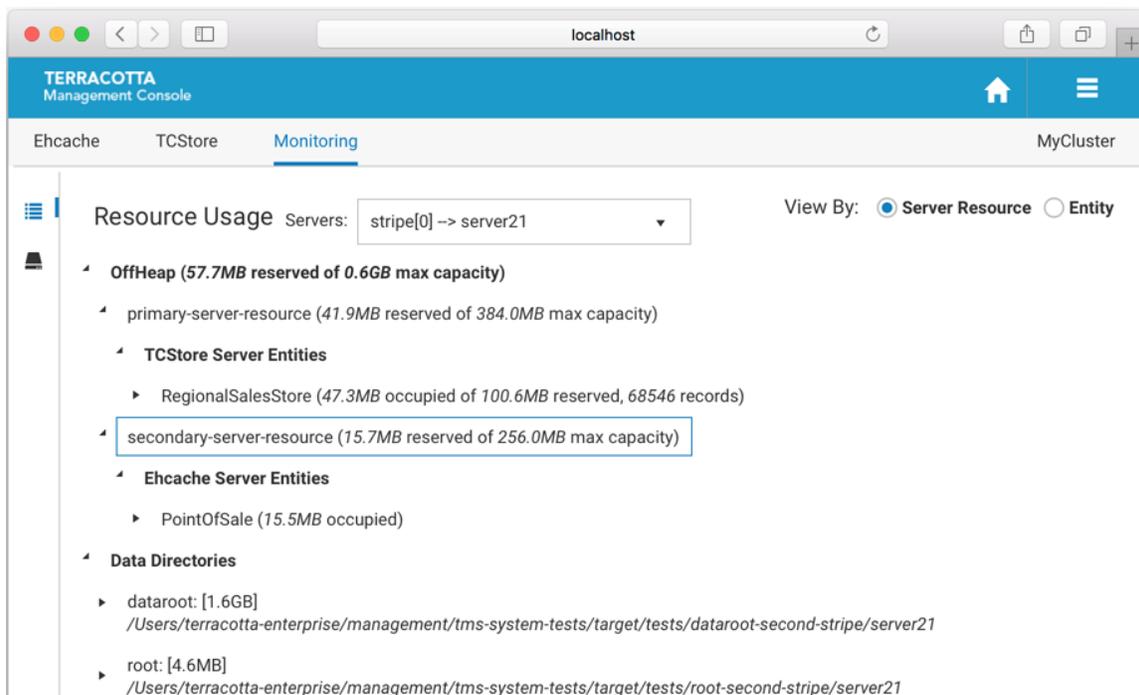
The **Monitoring** tab of the **Detail** page is where you can find information relating to the server-side of your application. The **Detail** page is selectable via buttons in the home page (see the section "Using the Configured Connections" on page 10 for details).

You can view:

- Resource usage relating to your cache's or dataset's use of the Terracotta Server's configured resources. Select this view by clicking the icon .
- Event log details. Select this view by clicking the icon .

### Resource Usage Panel

The **Resource Usage** Panel displays information relating to your cache's or dataset's use of the Terracotta Server's configured resources, including *OffHeap* memory and Fast Restartable Store (*FRS*) data directories.

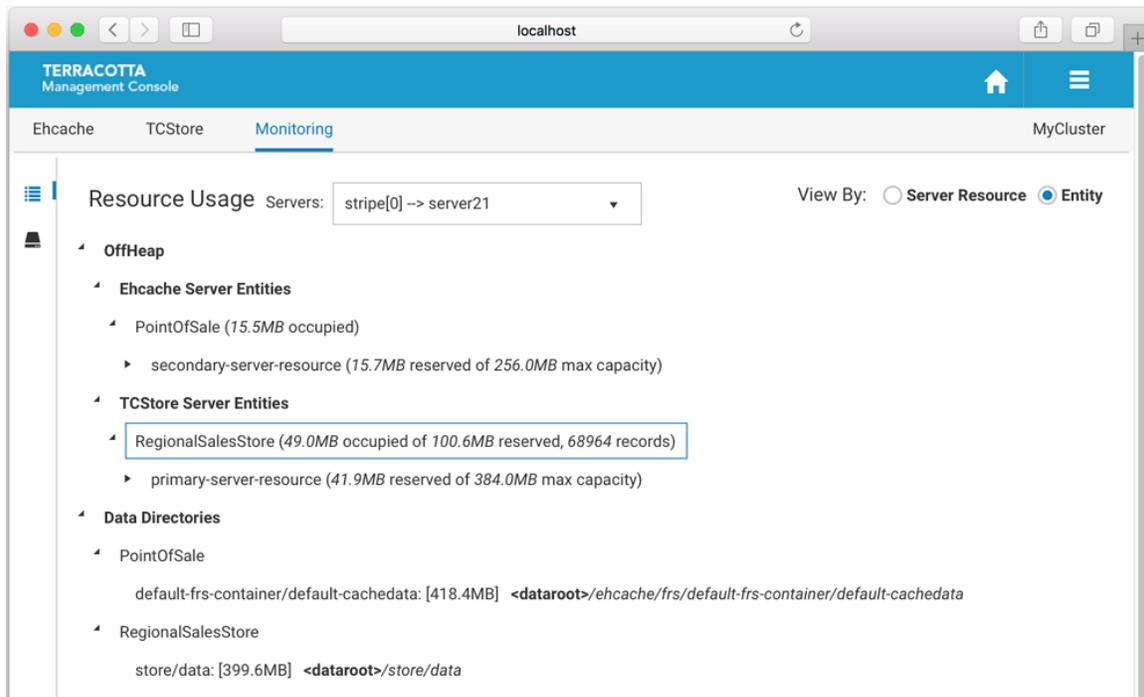


The screenshot shows the Terracotta Management Console interface. The top navigation bar includes "Ehcache", "TCStore", and "Monitoring" (selected). The "Servers" dropdown is set to "stripe[0] -> server21". The "View By" radio buttons are set to "Server Resource". The main content area displays a tree view of resource usage:

- OffHeap (57.7MB reserved of 0.6GB max capacity)
  - primary-server-resource (41.9MB reserved of 384.0MB max capacity)
    - TCStore Server Entities
      - RegionalSalesStore (47.3MB occupied of 100.6MB reserved, 68546 records)
      - secondary-server-resource (15.7MB reserved of 256.0MB max capacity)
    - Ehcache Server Entities
      - PointOfSale (15.5MB occupied)
  - Data Directories
    - dataroot: [1.6GB]
      - /Users/terracotta-enterprise/management/tms-system-tests/target/tests/dataroot-second-stripe/server21
    - root: [4.6MB]
      - /Users/terracotta-enterprise/management/tms-system-tests/target/tests/root-second-stripe/server21

Use the **Servers** dropdown to view the resource usage of a particular Terracotta Server.

Use the **View By** radio buttons to toggle between a server resource or a caching entity-focused presentation, as shown below.



## OffHeap

The **Server Resource** view shows how caching entities are making use of the allocated OffHeap resources configured for the server. Any number of server entities can use a particular server resource simultaneously.

The **Entity** view shows how caching entities are using the OffHeap resources configured for the server.

A caching entity can be configured to make use of one or more server resources via the pools (both shared and dedicated) they carve out of those server resources. Caches can be configured to store their entries in the pools configured by their containing CacheManager. The amount of space occupied by a caching entity is displayed, along with the total amount of space reserved for the pool.

While caching entities are associated with offheap resource pools, dataset entities make use of server offheap resources directly. The amount of space occupied by the dataset entity, the total amount of space reserved for the offheap server resource, as well as the count of records stored are displayed. Further, the cell names of the indexes that have been defined on your dataset, the type of index, the amount of offheap space occupied, the count of records indexed, and the number of times the index has been used are all displayed.

## Data Directories

Data directories are configured in the server configuration and your application points to these named disk areas for persistent storage.

For each defined data directory, the filesystem location and occupied space are displayed, as well as the names of the server entities utilizing the directory and amount of filesystem space each is occupying.

## Event Log

The screenshot shows the Terracotta Management Console interface. The 'Monitoring' tab is active, and the 'Event Log' sub-tab is selected. The log shows a list of events with columns for Alert, Date, Type, Name, and Detail. The events are sorted in reverse chronological order. The first event is a client connection to stripe[0] on server21, followed by a client connection to stripe[1] on server11, and then several other client connections to various stripes on different servers.

Alert	Date +	Type	Name	Detail
	Tue Oct 17 2017 16:53:01 GMT-0400 (EDT)	Client	Client joined	70908@127.0.0.1:Store:FkcXR5FnmogxYe8unwaktA:2686eaa80... connected to stripe[0]:server21
	Tue Oct 17 2017 16:53:01 GMT-0400 (EDT)	Client	Client joined	70908@127.0.0.1:Store:FkcXR5FnmogxYe8unwaktA:2686eaa80... connected to stripe[1]:server11
	Tue Oct 17 2017 16:52:57 GMT-0400 (EDT)	Client	Client joined	70902@127.0.0.1:Store:1k0F5CdYj_8sDVeXeY8hmg:6f37093b3... connected to stripe[0]:server21
	Tue Oct 17 2017 16:52:57 GMT-0400 (EDT)	Client	Client joined	70902@127.0.0.1:Store:1k0F5CdYj_8sDVeXeY8hmg:6f37093b3... connected to stripe[1]:server11
	Tue Oct 17 2017 16:52:54 GMT-0400 (EDT)	Client	Client joined	70901@127.0.0.1:Ehcache:ModelsBork:70393d8ff3814447acff... connected to stripe[0]:server21
				70901@127.0.0.1:Ehcache:ModelsBork:70393d8ff3814447acff...

The **Event Log** tab of the **Monitoring** panel displays, in reverse chronological order, important activities occurring in your cluster. The TMS listens for and persists these events, meaning that if the TMS is not running no events will be stored.

The types of activities that generate events include:

- server state transitions
- clients connecting/disconnecting to TSA stripes
- server entities being created/destroyed
- resource constraint alerts
- and more

The event log is included in the set of diagnostics artifacts that can be downloaded from the cluster's connection panel on the Home Page, but in addition it can be downloaded separately using the **Export** button.

Events that are deemed to be critical are noted as alerts, displayed with an attention-focusing icon in the alert grid. Further, alert events are displayed in a temporary popup and the count of un-read alerts is shown in the cluster's connection panel on the Home Page.

Newly added events are shown in the grid with bold text, and reset to normal after navigating away.

An event is comprised of the following fields:

---

alert	Is this event deemed critical?
Timestamp	Time at which the TMS recorded the event
Type	Categorization of the event
Name	Sub-categorization
Detail	Description of the event

The maximum number of events that are saved in the TMS is controlled by the configuration property `tms.eventLogMaxRecords`, whose default value is 5000. The event log store in the TMS, as well as the event log grid, can be cleared using the **Clear** button.