

Terracotta Operations Center



About Terracotta Documentation

This documentation is about Terracotta DSO, an advanced distributed-computing technology aimed at meeting special clustering requirements.

Terracotta products without the overhead and complexity of DSO meet the needs of almost all use cases and clustering requirements. To learn how to migrate from Terracotta DSO to standard Terracotta products, see [Migrating From Terracotta DSO](#). To find documentation on non-DSO (standard) Terracotta products, see [Terracotta Documentation](#). Terracotta release information, such as release notes and platform compatibility, is found in [Product Information](#).

- **Introduction**
- [How DSO Clustering Works](#)
- [Platform Concepts](#)
- [Hello Clustered World](#)
- **Setup and Configuration**
- [Planning for a Clustered App](#)
- [Configuring Terracotta DSO](#)
- [Configuration Reference](#)
- [Installation](#)
- **APIs**
- [Using Annotations](#)
- [Cluster Events](#)
- [Data Locality Methods](#)
- [Distributed Cache](#)
- [Clustered Async Data Processing](#)
- **Tool Guides**
- [Developer Console](#)
- [Operations Center](#)
- [tim-get \(TIM Management Tool\)](#)
- [Platform Statistics Recorder](#)
- [Eclipse Plugin](#)
- [Sessions Configurator](#)
- [Clustering Spring Webapp with Sessions Configurator](#)
- [Maven](#)
- [JMX](#)
- **Testing, Tuning, and Deployment**
- [Top 5 Tuning Tips](#)
- [Testing a Clustered App](#)
- [Tuning a Clustered App](#)
- [Deployment Guide](#)
- [Operations Guide](#)
- **FAQs and Troubleshooting**
- [General FAQ](#)
- [DSO Technical FAQ](#)
- [Troubleshooting Guide](#)
- [Gotchas](#)
- [Non-portable Classes](#)
- **Reference**
- [Migrating From DSO](#)
- [Concept and Architecture Guide](#)
- [Examinator Reference Application](#)
- [Clustered Data Structures Guide](#)
- [Integrating Terracotta DSO](#)
- [Clustering Spring Framework](#)
- [Integration Modules Manual](#)
- [AspectWerkz Pattern Language](#)
- [Glossary](#)

Release: 3.6

Publish Date: November, 2011 [Documentation Archive](#) »

- [Introduction](#)
- [Launching the Terracotta Operations Center](#)
 - [The Console Interface](#)
 - [Context-Sensitive Help](#)
 - [Context Menus](#)
- [Connecting and Disconnecting From a Cluster](#)
 - [Connecting to a Secured Cluster](#)
 - [Disconnecting from a Cluster](#)
- [Dashboard](#)
 - [Graphs](#)
 - [Unacknowledged Client Transactions](#)
 - [Activity Gauges](#)
- [Manage](#)
 - [Working with Servers](#)
 - [Working with Clients](#)
- [Backup](#)
 - [Enabling Backups](#)
 - [Restoring a Backup](#)
- [Diagnostics](#)
 - [Distributed Garbage Collection](#)
 - [Operator Events](#)
 - [Thread Dumps](#)
- [Logging](#)

Introduction

The Terracotta Operations Center delivers a full-featured monitoring and diagnostics tool for operators of an application clustered with Terracotta. The Operations Center is available with Enterprise versions of Terracotta.



The Operations Center is designed to monitor and manage a Terracotta cluster in a production environment. In a development environment, use the [Terracotta Developer Console](#).

The Operations Center is a GUI console that includes a JMX client. Using the console, you can:

- Monitor the health of servers and clients under changing conditions.
- Back up the shared-object database
- Force distributed garbage collection operations
- View cluster-wide statistics
- Disconnect problem clients remotely
- Shutdown servers remotely
- Receive console status and server log messages in a static window

These and other console features are described below.



Many of the images in this document can be viewed in a larger format by clicking them.

Launching the Terracotta Operations Center

You can launch the Terracotta Operations Center from a command line.

Microsoft Windows

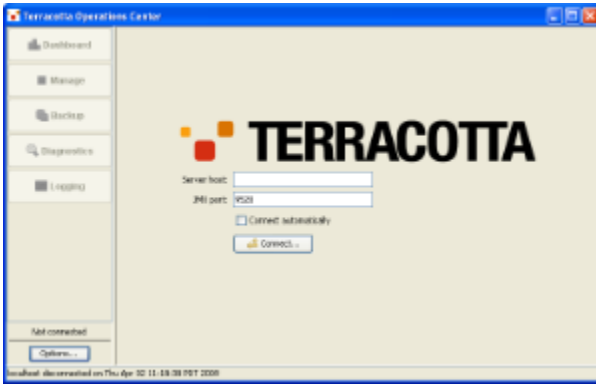
```
[PROMPT] %TERRACOTTA_HOME%\bin\ops-center.bat
```

UNIX/Linux

```
[PROMPT] ${TERRACOTTA_HOME}/bin/ops-center.sh&
```

The Console Interface

When the console is not connected to a server, the selection buttons in the control panel on the left are inactive.





The connection panel contains the last host and JMX port entered. The first time the console is started, default values (host=localhost, jmx-port=9520) are used.

Setting a JMX Port

To learn more about setting JMX ports, see the [Configuration Guide and Reference](#).

Once the console is connected to a cluster, the selection buttons in the control panel become active. The console can connect to only one cluster at a time. The status bar at the bottom of the console window displays messages on the latest changes in the cluster, such as nodes joining or leaving.

Context-Sensitive Help

Context-sensitive help is available wherever  (help button) appears in the Terracotta Operations Center. Click  in a console panel to open a web-browser page containing help on the features in that panel.

Context Menus

Some console features have a context menu that is accessed by right-clicking the feature. For example, to open a text-management context menu in the Logging panel, right-click in the panel.

Connecting and Disconnecting From a Cluster

To connect to an existing cluster, enter the hostname and JMX port number of a Terracotta server in the cluster, then click the **Connect** in the connection panel.

To automatically connect to a cluster whenever the console starts or when at least one of the cluster's servers is running, enable **Connect automatically** in the cluster context menu. Automatic connections are attempted in the background and do not interfere with normal console operation.

Connecting to a Secured Cluster

A Terracotta cluster can be secured for JMX access, requiring authentication before access is granted. Connecting to a secured cluster prompts users to enter a username and password.

For instructions on how to secure your Terracotta cluster for JMX, see the [Configuration Guide and Reference](#).

Disconnecting from a Cluster

To disconnect from a cluster, click **Disconnect** in the control panel.

Dashboard

The Dashboard provides live measurements of the most important metrics affecting the cluster. This panel is where you monitor the components and health of your cluster.

This graph is useful for quickly locating and investigating clients that maintain a large or growing number of unacknowledged Terracotta transaction broadcasts. In a healthy cluster, clients clear unacknowledged transactions as they process transactions. The bar representing each client's unacknowledged-transactions volume fluctuates as each client processes its acknowledgements. However, clients experiencing trouble may not be able to clear unacknowledged transactions, and the bars corresponding to those clients remain in place or even grow. Such clients could slow down an entire cluster and may need to be disconnected.

Activity Gauges

The cluster activity gauges provide real-time readings of crucial cluster metrics.



Each gauge has the following characteristics:

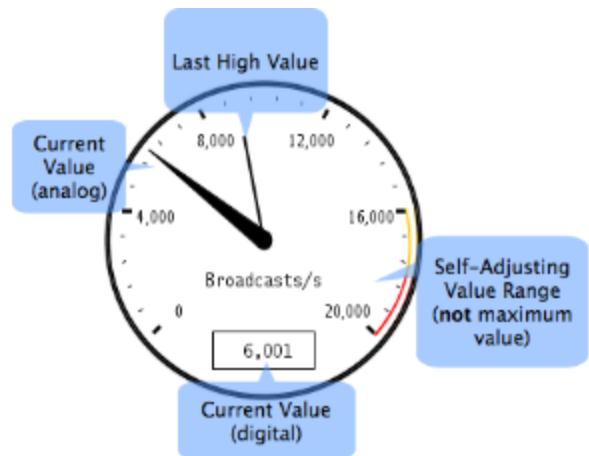
- Yellow and red zones on the dial indicate when the metric value has reached warning or extreme levels.
- A digital readout field displays the metric's current value.
- A tooltip shows the metric's full name, last maximum value, and average value (over all samples).
- By default, values are sampled over one-second intervals (except for **Unacked Txns**). The sample rate can be changed in the [Options Dialog](#).
- A "high-water" mark tracks the last high value, fading after several seconds.
- A self-adjusting value range uses a built-in multiplier to automatically scale with the cluster.

The left-most gauge (the large dial with the red needle) measures the rate of write transactions, which reflects the work being done in the cluster, based on [Terracotta transactions](#). This gauge may have a high value or trend higher in a busy cluster. An indication that the cluster may be overloaded or out of tune is when this gauge is constantly at the top of its range.

The remaining gauges, which measure "impeding factors" in your cluster, typically fluctuate or remain steady at a low value. If any impeding factors consistently trend higher over time, or remain at a high value, a problem may exist in the cluster.

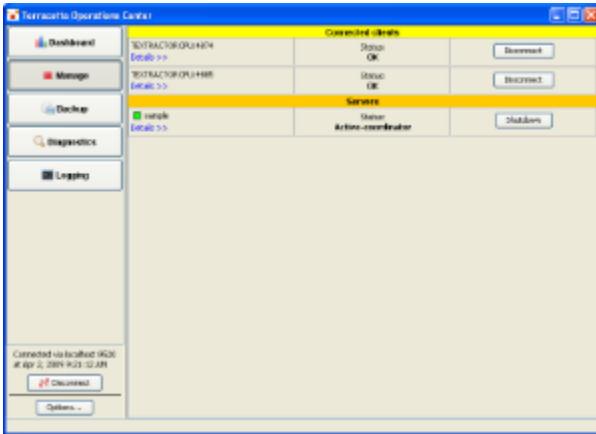
These gauges are listed below:

- **Objects Created/s** – The rate of shared objects being created. A rising trend can have a negative impact on performance by reducing available memory and necessitating more garbage collection.
- **Lock Recalls/s** – The number of locks being recalled by Terracotta servers. Growing lock recalls result from a high contention for shared objects, and have a negative performance impact. Higher locality of reference can usually lower the rate of lock recalls.
- **Broadcasts/s** – The number of object changes being communicated by the server to affected clients. High broadcast rates raise network traffic and can have a negative performance impact. Higher locality of reference can usually lower the need for broadcasts.
- **Faults/s** – Rate of faulting objects from servers to all connected clients. A high or increasing value can indicate one or more clients running low on memory or poor locality of reference.
- **Flushes/s** – Rate of flushing objects from all connected clients to servers. A high or increasing value can indicate one or more clients running low on memory.
- **Transaction Size KB/s** – Average size of total transactions.
- **Unacked Txns** – The current count of unacknowledged client transactions. A high or increasing value can indicate one more troubled clients.



Manage

The Manage panel displays a table of all connected servers and clients in the cluster. This panel is where you can check the status of servers and clients, remove them from the cluster, and get more detailed information.



To open the Manage panel, click **Manage** in the control panel.

Working with Servers

The Manage panel lists each server by name and gives its current status. There must be at least one active server instance for the cluster to function. In a Terracotta server array, a typical mirror group has one active server instance and one passive (or "hot" standby) backup. A server's role within the cluster is shown in the center column.

A status light next to each server's name indicates the server's *connection* status. The light's color indicates the server's current connection status. A cluster can have one server, or be configured with multiple servers that communicate state over the network or use a shared file-system.

The following table summarizes the connection status lights.

Status Light	Server Status	Notes
GREEN	Active	The server is connected and ready for work.
RED	Unreachable	The server, or the network connection to the server, is down.
YELLOW	Starting or Standby	A server is starting up; in a disk-based multi-server cluster, a passive server goes into standby mode until a file lock held by the active server is released.
ORANGE	Initializing	In a network-based multi-server cluster, a passive server must initialize its state before going into standby mode.
CYAN	Standby	In a network-based multi-server cluster, a passive server is ready to become active if the active server fails.

Server Panel

Click **Details >>** under a specific server to display tabbed subpanels with detailed information about that server. Click **<< Back** to return to the main Manage panel.

The **Main** tab displays the server status and a list of properties, including the server's IP address, version, license, and persistence and failover modes.

The **Environment** tab displays the server's JVM system properties and provides a **Find** tool.

The **Config** tab displays the Terracotta configuration the server is using and provides a **Find** tool.

Connecting and Disconnecting from a Server

The console is connected to all of a cluster's servers when it's connected to the cluster. Being connected to a server means that the console is listening for JMX events coming from that server.



If you have confirmed that a Terracotta server is running, but the console is unable to connect to it, a firewall on your network may be blocking the server's JMX port.

The console is disconnected from a cluster's servers when it's disconnected from the cluster. The console is also disconnected from a server when that server is shut down, but the server still appears as part of the cluster. However, its [connection status](#) changes.

To shut down a server, click the **Shutdown** button in that server's row. A dialog prompts to you confirm the shutdown.

Working with Clients

The Manage panel lists each client by name and gives its current status.

Client Panel

Click **Details >>** under a specific client to display tabbed subpanels with detailed information about that client. Click **<< Back** to return to the main Manage panel.

The **Main** tab displays a list of client properties such as hostname and DSO port.

The **Environment** tab displays the client's JVM system properties and provides a **Find** tool.

The **Config** tab displays the Terracotta configuration the client is using and provides a **Find** tool.

Connecting and Disconnecting Clients

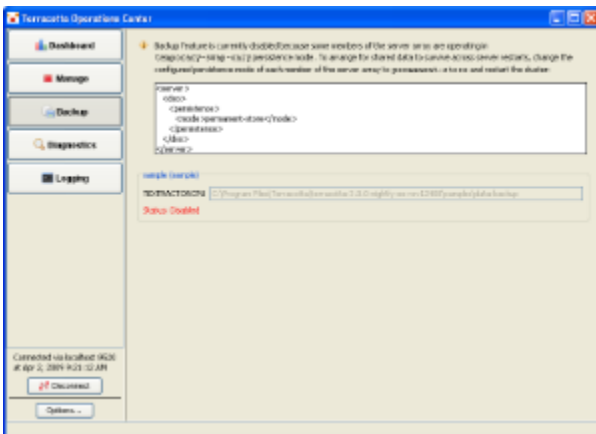
When started up properly, a Terracotta client is automatically added to the appropriate cluster.

When a Terracotta client is shut down or disconnects from a server, that client is automatically removed from the cluster and no longer appears in the console.


To disconnect a client from the cluster, click the **Disconnect** button in that client's row. A dialog prompts to you confirm the shutdown.

Backup

Using the Terracotta backup feature, you can create a backup of the data being shared by your application. The backup feature is available from the Backup panel. To open the Backup panel, click **Backup** in the control panel. If the cluster's persistence mode is not set to `permanent-store`, then the backup feature is disabled (see [#Enabling Backups](#)).



If backups are enabled, click **Backup DB** to perform a backup. A dialog box appears where you can confirm the backup destination directory or enter a new destination. The backup is saved to the directory `objectdb` at the destination. The database is always backed up to a directory called `objectdb`, which is automatically created if it does not exist at the destination.

 Backups can also be initiated using the `backup-data` script (see *Tools Catalog* in

Error rendering macro 'html'

Notify your Confluence administrator that "Bob Swift Atlassian Add-ons - HTML" requires a valid license. Reason: EXPIRED

). The script is useful for automating scheduled backups.

To change the default backup directory path, edit the `<data-backup>` property in the Terracotta server's configuration file with the path to your preferred backup directory:

```
<server>
<data-backup>path/to/my/backup/directory</data-backup>
</server>
```

Enabling Backups

If a Terracotta server is not configured for permanent-store persistence, the **Backup DB** button is disabled and the following message appears on the **Backup database** panel:

```
Backup feature is currently disabled because the cluster is
operating in temporary-swap-only persistence mode.
```

To enable the **Backup DB** button, change the value of the [persistence mode property](#) in the Terracotta server's configuration file to `permanent-store`:

```
<server>
  <dso>
    <persistence>
      <mode>permanent-store</mode>
    </persistence>
  </dso>
</server>
```

Restoring a Backup

Terracotta maintains a copy of shared in-memory data on disk. In most server-failure cases, Terracotta automatically restores that shared data by loading it from the copy, recreating the application state.

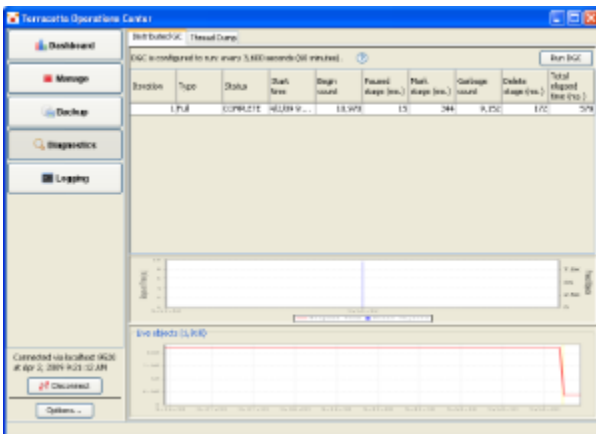
However, if you encounter a situation in which the data files are missing, you can restore them from backups.

To restore data files from a backup:

1. Shut down the Terracotta cluster.
2. Make copy of any existing data files.
3. Delete the existing data files.
4. Copy the backup data files to the directory from which you deleted the original (existing) data files.
5. Restart the Terracotta cluster.

Diagnostics

The Diagnostics panels offers tools for monitoring and analysis. There are three tabbed panels available: one for monitoring and triggering the Terracotta Distributed Garbage Collector (DGC), one for viewing cluster events, and one for taking thread dumps.




Operator Events

The cluster events are shown under the Operator Events panel. The button for accessing this panel, which is not shown in the screenshot above, is labeled **Operator Events** and appears between the **Distributed GC** and **Thread Dump** buttons.

To open the Diagnostics panels, click **Diagnostics** in the control panel.

Distributed Garbage Collection

Objects in a DSO root object graph may become unreferenced and no longer exist in the Terracotta client's heap. These objects are eventually marked as garbage in a Terracotta server instance's heap and from persistent storage by the DGC. The DGC is unrelated to the Java garbage collector.

 For more information on the DGC, see the [Terracotta Concept and Architecture Guide](#).

To view a history table of DGC activity in the current cluster, click **Distributed garbage collection** in the cluster list. The history table is automatically refreshed each time a collection occurs. Each row in the history table represents one distributed garbage collection cycle, with the following columns:

Column	Definition	Values
Iteration	The index number of the DGC cycle	Sequential integer
Type	The type of cycle	Full – Running a full collection cycle targeting all eligible objects. Young – Running a collection cycle targeting Young Generation objects.
Status	The collection cycle's current state	START – Monitoring for object reference changes and collecting statistics such as the object begin count. MARK – Determining which objects should be collected and which should not. PAUSE – Determining if any marked objects should not be collected. MARK COMPLETED – Stops checking for reference changes (finalizing marked object list). DELETE – Deleting objects. COMPLETE – Completed cycle.
Start time	The date and time the cycle began	Date and time stamp (local server time)
Begin count	The total number of shared objects held by the server	Integer counter
Paused stage	The total time the DGC paused	Milliseconds
Mark stage	The total time the DGC took to mark objects for collection	Milliseconds
Garbage count	The number of shared objects marked for collection	Integer counter
Delete stage	The total time the DGC took to collect marked objects	Milliseconds
Total elapsed time	The total time the DGC took to pause, mark objects, and collect marked objects	Milliseconds

The DGC graph combines a real-time line graph (with history) displaying the DGC total elapsed time with a bar graph showing the total number of freed objects.

Live Object Count

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 may fail. Upward trends indicate a problem with application logic, garbage collection, or a tuning issue on one or more clients. The total number of live objects is given in the graph's title.



DGC cycles should occur often enough to keep the total number of live objects from continuing to grow.

Triggering a DGC Cycle

The DGC panel displays a message stating the configured frequency of DGC cycles. To manually trigger a DGC cycle, click **Run DGC**.

Operator Events

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

Events are listed in a table with the following columns:

- Event Type - The level of the event (INFO, WARN, DEBUG, ERROR, CRITICAL) along with a color-coded light corresponding to the severity of the event.
- Time of Event - The event's date and time stamp.
- Node - The server receiving the event.
- Event System - The Terracotta subsystem that generated the event. The choices are MEMORY MANAGER (virtual memory manager), DGC (distributed garbage collector), LOCK MANAGER (cluster-wide locks manager), and HA (server array).
- Message - Message text reporting on a discrete event.

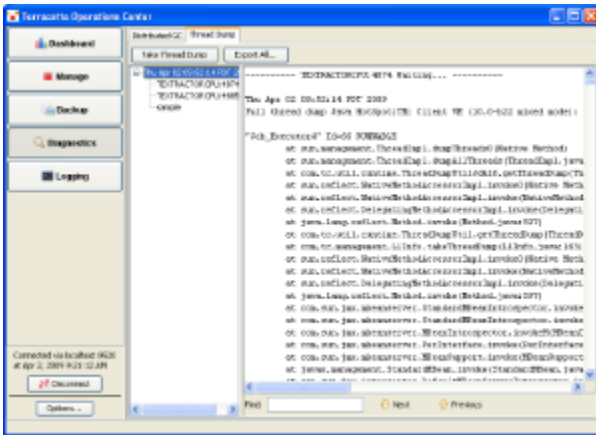
An event appears in bold text until it is manually selected (highlighted). The text of an event that has been selected is displayed in regular weight.

The Operator Events panel has the following controls:

- Mark All Viewed - Click this button to change the text of all listed events from bold to regular weight.
- Export - Click this button to export a text file containing all listed events.
- Select View - Use this drop-down menu to filter the list of displayed events. You can filter the list of events based on type (level) or by the generating system (or subsystem). For example, if you choose INFO from the menu, only events with this event type are displayed in the event list.

Thread Dumps

You can get a snapshot of the state of each server and client in the Terracotta cluster using thread dumps. To generate a thread dump, click **Take Thread Dump**.



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. A **Find** tool is available for searching through the currently displayed thread dump.

To delete a thread dump (or all thread dumps) from the thread-dump navigation pane, right-click the thread dump in the thread-dumps navigation pane, then choose **Delete** (or **Delete All**) from the context menu. You can also delete a selected thread dump with your keyboard's Delete key.



Servers that appear in the console but are not running produce empty thread dumps.

Saving Thread Dumps

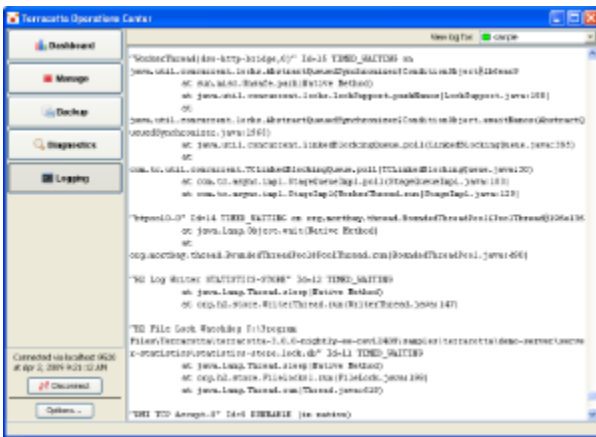
Thread dumps are deleted each time the console is restarted.

To save all existing thread dumps to disk as a ZIP file, click **Export All...**

To save any thread dump as a text file, right-click the thread dump in the thread-dumps navigation pane, then choose **Export As Text...** from the context menu.

Logging

The Logging panel displays log messages for each server. From the **View log for** menu, choose the server whose logs you want to view.



To open the Logging panel, click **Logging** in the control panel.

You can select and, using the **Copy** command from the context menu, copy any portion of the log. The context menu also offers **Select All** and **Clear** commands.