

Release Notes Terracotta 3.5.0



The Terracotta 3.5.0 Release includes Ehcache Core 2.4.1 and Quartz 2.0.0

Summary of Key Changes

- Addition of Ehcache Search (see below)
- New Ehcache Consistency setting (strong or eventual)
- Improvements to Enterprise Ehcache Disk Store (for use in conjunction with BigMemory)
- New local and local_jta Ehcache transaction modes
- NonStopCache is now in core
- Explicit locking is now in core
- Ehcache rejoin. Distributed Ehcache L1's are now able to rejoin the cluster on disconnect/reconnect from the Terracotta Server Array
- Ehcache Size method performance improvement for distributed caches
- Enhanced L2 BigMemory implementation (with self-tuning)

Quartz Scheduler <http://www.quartz-scheduler.org>

- New fluent configuration API (see below)
- Quartz Where - permitting control of where jobs execute in a clustered environment through node groups and node type constraints
A getting-started page for using Quartz Scheduler Locality API is available: [Getting Started With Quartz Scheduler Locality API](#)
- Clustered Quartz performance improvements

Additional Improvements

- Developer console improvements
- Platform Updates
 - Certification on JDK 1.6.0_24

About Ehcache Search

The Ehcache Search API allows you to execute arbitrarily complex queries against Ehcache. The standalone Ehcache Search implementation is limited to the memoryStore (ie it has no support for diskStore or BigMemory - aka offHeapStore). Ehcache Search support for distributed caches is an Enterprise-only feature, and is designed to work across all Terracotta server storage tiers (memory, BigMemory, and disk).

The easiest way to get started with the API is to try the sample application posted on github. This includes a ready to run bundle for *nix (just download, unpack, and execute with 'sh run.sh'). This bundle also includes a maven pom to rebuild (using the latest SNAPSHOT rather than that in the bundle): <https://github.com/sharrisf/Ehcache-Search-Sample/downloads>

Ehcache Search docs are here: <http://ehcache.org/documentation/search.html>

A getting-started page for using Ehcache Search with clustered caches is also available: [Getting Started With Ehcache Search API For Clustered Caches](#)

New configuration options

NonStopCache is disabled by default, to enable non-stop, update the <terraccotta> element in each <cache>

```
<terraccotta clustered="true">
  <nonstop enabled="true"/>
</terraccotta>
```

Ehcache rejoin on disconnect from Terracotta Server Array is disabled by default and can be enabled as follows in ehcache.xml. Note: that rejoin enable is dependent on NonStopCache being enabled.

```
<terraccottaConfig rejoin="true" url="localhost:9510" />
```

Ehcache now has three transaction modes:

- **local**, which will use a local transaction manager (rather than a JTA one), and permit users to create ACID operations across multiple caches

```
transactionalMode="local"
```

- **xa**, which in Beta1 was temporarily called `jta_local` will use a JTA transaction manager, but will be designed for use with a single resource manager in mind and to work with a lower level of recoverability than the "xa_strict" mode. (it won't recover transaction in the case of catastrophic failure).

```
transactionalMode="xa"
```

The original "xa" mode is now "xa_strict". Transactional caches can be made nonstop only in xa_strict mode.

Local is the fastest mode, followed by xa, followed by xa_strict

A sample that demonstrates the new transaction modes is available here: <https://github.com/sharrissf/Ehcache-Local-Transaction-Sample/downloads>

About Quartz Scheduler 2.0

Please see the [What's new in Quartz 2.0](#) document

Note: [Quartz Where](#) is only supported when using the TerracottaJobStore, and is an enterprise-only feature.

Upgrade Information

Brief descriptions of major changes that may affect upgrades from previous versions of Terracotta product installations are listed in this section. See the [product documentation](#) for more information.

Change to <cache> "coherent" Attribute

The <cache> coherent attribute is deprecated in favor of consistency=[strong | eventual]. For example, where you've used coherent="true" use consistency="strong". For coherent="false" use consistency="eventual" (eventually consistent). If you've set the coherent attribute programmatically, the setting will continue to be honored and mapped as described.

New Transactional Modes

The <cache> attribute transactionalMode can now have one of the following values:

- off – Default.
- local – Transactional to a local store; no external XA resource used.
- xa – Basic JTA support.
- xa_strict – Full JTA support.

Note that with a transactional cache, the <cache> attribute coherent must be set to "strong".

Coherency API Deprecated in Favor of Bulk-Load Methods

Deprecated Method	New Method	Notes
<code>isNodeCoherent()</code>	<code>isNodeBulkLoadEnabled()</code>	These methods are opposites: <code>isNodeCoherent() == !isNodeBulkLoadEnabled()</code> .
<code>isClusterCoherent()</code>	<code>isClusterBulkLoadEnabled()</code>	These methods are opposites: <code>isClusterCoherent() == !isClusterBulkLoadEnabled()</code> .
<code>setNodeCoherent()</code>	<code>setNodeBulkLoadEnabled()</code>	These methods are opposites: <code>setNodeCoherent() == !setNodeBulkLoadEnabled()</code> .
<code>waitUntilClusterCoherent()</code>	<code>waitUntilClusterBulkLoadComplete()</code>	

NonStopCache Becomes a <cache> Subelement

NonStopCache is no longer a decorator. Default values for NonStopCache behavior settings have also changed.

Change in default storageStrategy for distributed Ehcache

The DCV2 strategy is now the default. The old strategy is deprecated, but can be accessed by setting storageStrategy=classic.

Note: many of the new features (such as eventual consistency, rejoin, and nonstop) are not compatible with classic.

DCV2 is optimized for large data sets. In certain environments, including ones with small data sets, this optimization may need to be tuned to avoid higher latency and memory usage. If your cluster experiences these issues after upgrading, reduce cache concurrency from the default 2048 to a lower number using the concurrency attribute of the <cache> element's <terracotta> subelement:

```
<cache ... >
  ...
```

```
<terracotta concurrence="10" ... />
</cache>
```

Change in server side eviction precedence for distributed Ehcache

Server side eviction now gives precedence to the maxElementsOnDisk setting over TTI/TTL, and thus if the total number of elements reaches maxElementsOnDisk, entries will be evicted from cache irrespective of whether they have expired or not.

Expected Updates and Scheduled Bug Fixes

CDV Project Jira fixes in Terracotta 3.5.0

EHC Project scheduled fixes for Ehcache 2.4.1

QTZ Project scheduled fixes for Quartz 2.0.0

Known Issues and Limitations

- **CDV-1569** TCNonPortableObjectError using tim-ehcache-2.x-ee and non-EE tims
- **5595** - Quartz Where custom constraints need to be registered using their own prop file, that has to be named just as the property to actually define these.
 - Workaround: needs to have the org.terracotta.quartz.locality.evaluators.properties on this classpath with a single prop in it (org.terracotta.quartz.locality.evaluators) to define custom evaluators
- **FORGE-594** - Quartz avg load monitoring only supported on JDK6
- **CDV-1522** - LinkageError with ehcache jta & jetty (maven plugin)
- **CDV-1564** - Server backup and archive features do not cover search index files
- **CDV-1565** - dirty objectdb move aside functionality does not archive search index files
 - Workaround: have the full jta.jar on jetty's classpath and nowhere else, ie: exclude the geronimo jta jar from BTM's deps and make jta-1.1.jar provided.
- **5234** - Possible java.lang.Error when rapidly shutting down and restarting a Cache that uses the same disk file.
 - Probably related to http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=4938372
- **5050** - Due to <https://issues.apache.org/jira/browse/LUCENE-588> we are unable to support escaped wildcards in Search
- **5573** - If an active terracotta server crashes in the middle of creating a new lucene index it may fail to reload when active restarts
 - java.io.FileNotFoundException: XXX/_terracotta_cache_name.txt (No such file or directory)
 - Delete the incomplete lucene index path and restart server
- **5353** - If the L2 dies during cacheManager initialization itself, the thread will be stuck.

Please email any questions you have regarding the beta to pm <at> terracotta.org