

3.3.0 Beta Release Notes

Release Notes for Terracotta 3.3 (Esperance) Beta

Getting Started

1. Download the beta build. If you haven't already, you will need to sign-up to get the download location. Sign-up here: <http://www.terracotta.org/beta/esperance-beta>

NOTE: This beta is based around the enterprise kit. Users wishing to evaluate the open source kit should work with the nightly builds: <http://download.terracotta.org/rss/trunk.rss>

For the enterprise beta version, the accompanying trial license key "product.key" should be downloaded and installed in the Terracotta directory.

2. Review the new beta documentation: <http://www.terracotta.org/documentation/betadocs>

New Features

1. Scale/Performance Improvements (Enterprise Feature)

Terracotta 3.3 Enterprise Edition includes a new storage strategy (an optional mode configured per cache - enabled with a new XML attribute of the <terracotta> subtag - storageStrategy="DCV2"), which when selected is designed to support caches with:

- a large # of entries (depending on the use case - up to 100M items)
- and/or a large # of nodes (maintaining linear scale for high write throughput rate use-cases, even with large numbers - eg >10 - of L1s)

This storage mode works with an optional client side cache (off by default), which has the added benefit of meaning that the memory required on the client is fixed - making it easy to tune.

2. Developer Console Enhancements

- Reorganization of tree hierarchy to simplify usage for new users
- Improved and Simplified Ehcache panel
- Addition of panels for Sessions and Quartz

3. RAAS based authentication to permit access control via LDAP etc to Developer Console and Ops Console (Enterprise Feature)

4. Cluster Configuration Simplification

- Automatic selection of sensible defaults (-n usage is now optional in most cases, port selection is automatic)

5. New Toolkit APIs

The client libraries have been refactored so that now when clustering each of the core use cases a common shared jar is required:

common/terracotta-express-runtime-<version>.jar

This has several advantages - firstly it reduces the number of L1-L2 socket connections from applications that use multiple client types (Ehcache, Sessions etc) and in such cases also reduces permgen heap usage.

Additionally the common library contains a series of useful constructs that can be used within user applications. These include concurrent maps, locks, counters, queues, an evictor etc. For an introduction, see <http://dsoguy.blogspot.com/2010/06/couple-minutes-with-some-toolkit.html> .

- Web Sessions support on Websphere 6.1 and 7.0

Available in Beta2

- Cluster Configuration Simplification - adding passives via Ops Console (Enterprise Feature)
- Operational Simplicity - Operational event monitoring - for cluster management, control and behavior analysis. Delivered via an Events Console and JMX API (Enterprise Feature)
- Operational Simplicity - Simplified configuration of heartbeat and reconnect though template sets

Summary of Changes

Expected Updates and Scheduled Bug Fixes

in the 2.1.1 [EHC Jira Project](#)

in the 3.3.0 [CDV Jira Project](#)

Known Open Issues

Note: The beta is focused on 'Express' usage, rather than 'Custom' (DSO) usage of the Terracotta Platform. Custom DSO users who require specific TIMs compatible with the beta and may get unpredictable results due the fact that the beta is built with SNAPSHOT TIMs. Please contact us with details if you want to try any specific component in DSO mode.

- [CDV-1480](#) If non portable type which is not in the api is added, the map from the toolkit throws a TCNonPortableObjectException.

- [QTZ-51](#) org.quartz.scheduler.instanceId = AUTO must be set to have Quartz panel show in Terracotta Developer Console

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