Annotations for DSO

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.

Release: 3.6
Publish Date: November, 2011 **Documentation Archive »**

Annotations for Terracotta DSO

Introduction

Terracotta Distributed Shared Objects (DSO) configuration can normally be implemented by identifying classes in your application that need to be instrumented and locked using the appropriate XML in the Terracotta configuration file (`tc-config.xml` by default).

Annotations provide a mechanism for developers to mark class instrumentation, root configuration, transient fields, and locking directly in Java class files. In this case, there is no requirement to configure these in `tc-config.xml`.

One advantage to using Annotations is that the Terracotta definitions for your classes will stay with your code. Encapsulating all information about a class — including its Terracotta nature — in the class file improves the readability of your application. Also, refactoring annotated classes does not require changing the `tc-config.xml` file.

If you are replacing configuration properties in `tc-config.xml` with annotations, be sure to remove or comment out the replaced properties.

Installing Annotations

To use annotations, you must install the Terracotta Toolkit JAR using the tim-get tool:

```
[PROMPT] tim-get.sh install terracotta-toolkit-1.0
```

You should see output similar to the following:

```
Terracotta 3.3.0, as of 20100716-150712 (Revision 15922 by cruise@su10mo5 from 3.3)

Installing terracotta-toolkit-1.0 1.0.0 [org.terracotta.toolkit]...
   INSTALLED: terracotta-toolkit-1.0 1.0.0 [org.terracotta.toolkit] - Ok

Done. (Make sure to update your tc-config.xml with the new/updated version if necessary)
```

This example uses version the Terracotta Toolkit with API version 1.0, available with Terracotta kit version 3.3.0. The version you install may differ.

Configuring Your Application To Use Annotations

After installing the Terracotta Toolkit, you must add the JAR file to the Terracotta configuration file (`tc-config.xml` by default) and to your application's classpath. You can use tim-get to confirm the installation of the Terracotta Toolkit JAR file, locate its installation path, and obtain the configuration snippet that must be added to `tc-config.xml`:

```
[PROMPT] tim-get.sh info terracotta-toolkit-1.0

Terracotta 3.3.0, as of 20100716-150712 (Revision 15922 by cruise@su10mo5 from 3.3)

(+) terracotta-toolkit-1.0 1.0.0 [org.terracotta.toolkit]

   Installed: YES

   Author   : Terracotta, Inc.
   Homepage : http://forge.terracotta.org/releases
   Docs     : http://forge.terracotta.org/releases
   Download : http://www.terracotta.org/download/reflector/releases/org/terracotta/toolkit/
terracotta-toolkit-1.0/1.0.0/terracotta-toolkit-1.0-1.0.0.jar
   Status   : Supported
   Internal : false
   Terracotta Toolkit
   Compatible with Terracotta 3.3.0

   Dependencies:

      None.

   Maven Coordinates:

      groupId   : org.terracotta.toolkit
      artifactId: terracotta-toolkit-1.0
      version   : 1.0.0

   Configuration:

   <?xml version="1.0" encoding="UTF-8"?>
   <modules>
   <module group-id="org.terracotta.toolkit" name="terracotta-toolkit-1.0" version="1.0.0"/>
   </modules>
   Note: If you are installing the newest or only version, the version may be omitted.

   Installed at mystuff/terracotta-3.3.0/platform/modules/org/terracotta/toolkit/terracotta-toolkit-1.0/1.0.0
   This is the latest version.

   There are no other versions of this TIM that are compatible with the current installation.

(+) Installed  (-) Not installed  (!) Installed but newer version exists
```

Module Versions Are Optional

Since the tim-get script finds the optimal version for the current installation of the Terracotta kit, module versions are optional.

The `Configuration` section gives the appropriate configuration required to add to the `clients` section of `tc-config.xml`. It also shows the JAR file's installation path.

Using Maven or an IDE

Terracotta Annotations define Java annotations, which means your application must include the Terracotta Toolkit as a normal Java library. If you are using Maven, this is done by adding the appropriate maven configuration to your `pom.xml` file. For example:

```
    <dependency>
      <groupId>org.terracotta.toolkit</groupId>
      <artifactId>terracotta-toolkit-1.0</artifactId>
      <version>1.0.0</version>
    </dependency>
```

You'll also have to configure the appropriate repository:

```
<repositories>
  <repository>
    <id>terracotta-snapshots</id>
    <url>http://www.terracotta.org/download/reflector/snapshots</url>
    <releases><enabled>false</enabled></releases>
    <snapshots><enabled>true</enabled></snapshots>
  </repository>
  <repository>
    <id>terracotta-releases</id>
    <url>http://www.terracotta.org/download/reflector/releases</url>
    <releases><enabled>true</enabled></releases>
    <snapshots><enabled>false</enabled></snapshots>
  </repository>
</repositories>
```

If you are using a snapshot version, the snapshop repository must be included.

In an IDE such as Eclipse, configure the dependency as you do for any other Java library.

Using Terracotta Annotations

After the Terracotta Toolkit is installed and configured, annotations can be used.

There are currently 8 total annotations, each corresponds to a configuration element in the `tc-config.xml` file. They are:

- InstrumentedClass
- HonorTransient
- Root
- AutolockRead
- AutolockWrite
- AutolockConcurrentWrite
- AutolockSynchronousWrite
- DMI

InstrumentedClass

Use this annotation to mark a class instrumented.

```
import org.terracotta.modules.annotations.*;

@InstrumentedClass
public class MyClass {
    ...
}
```

HonorTransient

Use this annotation to honor transient fields in the instrumented class.

```
import org.terracotta.modules.annotations.*;

@InstrumentedClass
@HonorTransient
public class MyClass {
    ...
}
```

Root

Use this annotation to mark a field as a root.

```
import org.terracotta.modules.annotations.*;

public class MyClass {
    @Root
    private final Map map = new HashMap();

    ...
}
```

Autolock

Use the AutolockXXX annotations to autolock methods.

```
import org.terracotta.modules.annotations.*;

@InstrumentedClass
public class MyClass {
    @AutolockRead
    public synchronized void getFoo() { return foo; }

    @AutolockWrite
    public synchronized void setFoo(int foo) { this.foo = foo; }

    @AutolockConcurrentWrite
    public synchronized void setFooConcurrent(int foo) { this.foo = foo; }

    @AutolockSynchronousWrite
    public synchronized void setFooSynchronous(int foo) { this.foo = foo; }

    ...
}
```

If you are replacing autolock properties in `tc-config.xml` with AutolockXXX annotations, you may find that the lock level has not been specified. The default lock level for autolock properties in `tc-config-xml` is "write".

DMI

Use the DMI annotation to mark a method for DMI.

```
import org.terracotta.modules.annotations.*;

@InstrumentedClass
public class MyClass {
    @DMI
    public void notify() { &hellip; }

    ...
}
```