Bytecode Enhancement

Table of Contents

This guide covers Hibernate's ability to enhance an applications domain model, the ways to perform that enhancement and the capabilities introduced into the domain model by the enhancement.

The capabilities

Hibernate will enhance the classes in an application's domain model in order to add one or more of the following capabilities:

1. Lazy state initialization
2. Dirtiness tracking
3. Automatic bi-directional association management
4. Performance optimizations

todo : explain each in detail

Performing enhancement

Ultimately all enhancement is handled by the `org.hibernate.bytecode.enhance.spi.Enhancer` class. Custom means to enhancement can certainly be crafted on top of Enhancer, but that is beyond the scope of this guide. Here we will focus on the means Hibernate already exposes for performing these enhancements.

Run-time enhancement

Currently run-time enhancement of the domain model is only supported in managed JPA environments following the JPA defined SPI for performing class transformations. Even then, this support is disabled by default. In this scenario, run-time enhancement can be enabled by specifying `hibernate.ejb.use_class_enhancer=true` as a persistent unit property.

Build-time enhancement

Hibernate also offers the ability to integrate the enhancement of the domain model as part of the normal build cycle of that domain model. Gradle, Ant and Maven are all supported. One possible benefit of this approach is that the enhanced classes are what gets added to the jar and can then be used on both sides of serialization.

Gradle Plugin

Hibernate provides a Gradle plugin that is capable of providing build-time enhancement of the domain model as they are compiled as part of a Gradle build. To use the plugin a project would first need to apply it:

Example 1. Apply the plugin

```groovy
ext {
  hibernateVersion = 'hibernate-version-you-want'
}
buildscript {
  dependencies {
    classpath "org.hibernate:hibernate-gradle-plugin:$hibernateVersion"
  }
}
```

At the moment there is not much to configure with regard to the Enhancer, but what is configurable is exposed through a DSL extension registered. By default enhancement will not be done (in preparation for when this Gradle plugin offers additional capabilities). To enable it you must configure the following DSL extension:

Example 2. Apply the plugin

```groovy
hibernate {
  enhance {
    // any configuration goes here
  }
}
```

Currently the "enhance" extension supports 4 properties:

- `enableLazyInitialization`
- `enableDirtyTracking`
- `enableAssociationManagement`
- `enableExtendedEnhancement`

Once enhancement overall is enabled, the default for the first 3 properties is `true`. Field access is not enhanced by default, as it can potentially trigger enhancement of code outside the entities, and also because it assumes that all the target entities are enhanced, which may not always be the case.

Ant Task

Maven Plugin

The Hibernate Maven plugin provides a convenient way to enhance the domain model at build-time when using Maven as the build environment. Much like the Gradle plugin, the current bytecode enhancement capabilities that can be configured on the plugin are:

- `enableLazyInitialization`
- `enableDirtyTracking`
- `enableAssociationManagement`
- `enableExtendedEnhancement`

Field access is not enhanced by default, because it can potentially trigger enhancement of code outside the entities. Other capabilities are enabled by default. Even if the plugin is enabled, the bytecode enhancement can be bypassed by disabling all the capabilities.

There is also a parameter `failOnError` that controls what happens in case of error. Default behavior is to fail the build, but it can be set so that only a warning is issued.

To use the Hibernate Maven plugin on a build it must added to the model (pom.xml) along with other plugins that may be already in use. The XML snippet below is an example of how to declare and configure the plugin.

Example 3. Apply the plugin

```xml
<build>
    <plugins>
        [...]
        <plugin>
            <groupId>org.hibernate.orm.tooling</groupId>
            <artifactId>hibernate-enhance-maven-plugin</artifactId>
            <version>$currentHibernateVersion</version>
            <executions>
                <execution>
                    <configuration>
                        <failOnError>true</failOnError>
                        <enableLazyInitialization>true</enableLazyInitialization>
                        <enableDirtyTracking>true</enableDirtyTracking>
                        <enableAssociationManagement>true</enableAssociationManagement>
                        <enableExtendedEnhancement>false</enableExtendedEnhancement>
                    </configuration>
                    <goals>
                        <goal>enhance</goal>
                    </goals>
                </execution>
            </executions>
        </plugin>
        [...]
    </plugins>
</build>
```

Last updated 2017-01-19 00:14:28 EST