29. Legacy BootstrappingThe legacy way to bootstrap a SessionFactory is via the org.hibernate.cfg.Configuration object. Configuration represents, essentially, a single point for specifying all aspects of building the SessionFactory: everything from settings, to mappings, to strategies, etc. I like to think of Configuration as a big pot to which we add a bunch of stuff (mappings, settings, etc) and from which we eventually get a SessionFactory. There are some significant drawbacks to the legacy bootstrapping mechanism which led to its deprecation and the development of the new approach, which is discussed in Native Bootstrapping. Configuration is semi-deprecated but still available for use, in a limited form that eliminates these drawbacks. "Under the covers", Configuration uses the new bootstrapping code, so the things available there are also available here in terms of auto-discovery. You can obtain the Configuration by instantiating it directly. You then specify mapping metadata (XML mapping documents, annotated classes) that describe your applications object model and its mapping to a SQL database. Configuration cfg = new Configuration()
// addResource does a classpath resource lookup
.addResource( "Item.hbm.xml" )
.addResource( "Bid.hbm.xml" )
// calls addResource using "/org/hibernate/auction/User.hbm.xml"
.addClass( org.hibernate.auction.User.class )
// parses Address class for mapping annotations
.addAnnotatedClass( Address.class )
// reads package-level (package-info.class) annotations in the named package
.addPackage( "org.hibernate.auction" )
.setProperty( "hibernate.dialect", "org.hibernate.dialect.H2Dialect" )
.setProperty( "hibernate.connection.datasource", "java:comp/env/jdbc/test" )
.setProperty( "hibernate.order\_updates", "true" ); There are other ways to specify Configuration information, including: Place a file named hibernate.properties in a root directory of the classpath Pass an instance of java.util.Properties to Configuration#setProperties Via a hibernate.cfg.xml file System properties using Java -Dproperty=value