FAQ

Frequently Asked Questions

JDK support

- what are the JDK Requirements

jUDDI 0.7 requires a 1.4.x JDK. You can compile and run with a 1.3.x JDK, if you remove the package org.apache.juddi.auth.crypt from your codebase.

Embedded Databases

- Can i use an embedded database

You can use the juddi.useConnectionPool property (and other related properties) to avoid using the JNDI lookup. Here's a sample setup of juddi.properties

Required JDBC DataSource properties

- juddi.dataSource=java:comp/env/jdbc/juddiDB

Required JDBC Driver properties

- juddi.useConnectionPool=true
- juddi.jdbcDriver=org.hsqldb.jdbcDriver
- juddi.jdbcURL=jdbc:hsqldb:C:/jakarta-tomcat/server/webapps/juddi/WEB-INF/hsql/juddidb
- juddi.jdbcUser=sa
- juddi.jdbcPassword=

optional JDBC Driver connection pool properties

- juddi.jdbcMinConnections = 3
- juddi.jdbcMaxConnections = 50
- juddi.jdbcMaxWaitTime = 15000
- juddi.jdbcRetryInterval = 1000

How do I configure jUDDI to run with Tomcat, MySQL and use a JNDI DataSource?

- How do I configure jUDDI to run with Tomcat, MySQL and use a JNDI DataSource?

A good setup is discussed here http://sourceforge.net/mailarchive/forum.php?thread_id=2429672&forum_id=7138

Writing pluggable authentication modules

- How do I write a pluggable authentication module?

You can configure jUDDI to use one of the three Authentication modules supplied or you can write your own Authenticator that integrates jUDDI with your organizations authentication mechanism. All you have to do to create your own Authenticator is: Create a class that implements the Authenticator interface (your Authenticator); create a class that extends the AuthenticatorFactory abstract class (your Authenticator's Factory); specify your AuthenticatorFactory in jUDDI's property file ("juddi.properties").

For example:
juddi.authFactory = org.apache.juddi.auth.simple.SimpleAuthenticatorFactory In the example above you'll need to replace "org.apache.juddi.auth.simple.SimpleAuthenticatorFactory" with the fully qualified name of your AuthenticatorFactory class. Take a look at the SimpleAuthenticator and SimpleAuthenticationFactory classes in the "org.apache.juddi.auth.simple" package for an example

Using a datastore without JDBC

- How do I access a datastore without using JDBC

jUDDI is designed to allow different persistence (datastore) implementations to be "plugged-in". The JDBCDataStore included with jUDDI attempts to support as many as the mainstream RDBMS's as possible (Oracle, DB2, Sybase, MySQL - even HSQLDB). If you're interested in using something other than JDBC or if you want to take advantages of proprietary features of your RDBMS (i.e. using stored procedures to improve performance) then you will need to develop your own datastore implementation. To do so you'll need to create implementations of the DataStoreFactory and DataStore interfaces in the org.apache.juddi.datastore package and make a change to the jUDDI properties file to use your new DataStoreFactory implementation (i.e. the judd.dataStoreFactory property) instead of the one supplied. Check out jUDDI's JDBCDataStoreFactory and JDBCDataStore for examples.

Application server support

- Which application servers does jUDDI support?

jUDDI is known to work with a number of application servers like Tomcat, Sybase EAServer and Borland Enterprise Server 5.2 and 6.0.

FAQ Contributors

- Steve Viens
- Anou Manavalan
- Rupesh Ramachandran
- Andy Cutright