Jndi2x

Embedded JNDI provider

BTM is bundled with a read-only JNDI provider since version 1.3. You can use it as a convenience as it sometimes facilitates integration with some frameworks in non-application server environments.

Using the JNDI provider is obviously not mandatory. As long as you do not use it, it won't conflict with any other JNDI provider nor consume any resource.

Contents

- Accessing resources and transaction manager via JNDI
- Changing the default transaction manager's JNDI name
- Configuring BTM's JNDI provider as the default one

Accessing resources and transaction manager via JNDI

You can access configured resources (JMS and JDBC) as well as the transaction manager from the JNDI provider.

Here is the quick way to lookup resources from it:

```
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY,
"bitronix.tm.jndi.BitronixInitialContextFact
ory");
Context ctx = new InitialContext(env);

DataSource ds = (DataSource)
ctx.lookup("myDataSourceUniqueName");
UserTransaction ut = (UserTransaction)
ctx.lookup("java:comp/UserTransaction");
```

This expects that you configured one JDBC datasource with unique name `myDataSourceUniqueName`. Resources are automatically bound when created and unbound when closed.

The transaction manager itself (which implements both `UserTransaction` and `TransactionManager` interfaces) is available by default under the standard `java:comp/UserTransaction` name.

Changing the default transaction manager's JNDI name

Sometimes it is desirable to bind the transaction manager under a different name than the standard `java:comp/Us`

`erTransaction` one, for instance when there already is a JNDI URL handler configured for `java:`, like in most application servers.

It is recommended that you bind BTM and it resources in the application server's JNDI server but in case you don't want to or can't, you can still use BTM's JNDI provider and bind the transaction under another name by setting this configuration property (see here for more details):

```
bitronix.tm.jndi.userTransactionName=btmTran
sactionManager
```

Configuring BTM's JNDI provider as the default one

You just have to create a `jndi.properties` file in your classpath with this content:

```
java.naming.factory.initial=bitronix.tm.jndi
.BitronixInitialContextFactory
```

and then you don't have to specify an environment when creating an `InitialContext`:

```
Context ctx = new InitialContext();

DataSource ds = (DataSource)
ctx.lookup("myDataSourceUniqueName");
UserTransaction ut = (UserTransaction)
ctx.lookup("java:comp/UserTransaction");
```

See BitronixInitialContextFactory for more details.