

Eine neue Datenquelle

Verbindung über DriverManager

- 👉 Datenbankverbindungen werden üblicherweise über den `DriverManager` aufgebaut.

```
Connection con = DriverManager.getConnection(...);
```

- 👉 Würden EJBs auf diese Weise Datenbankverbindungen nutzen, so hätte das einige Nachteile:
 - ▶ Die Session-Bean muss die Klasse für den Datenbanktreiber kennen, sowie Verbindungsparameter wie JDBC-URL, Benutzername und Passwort.
 - ▶ Die Verbindungen sollten nicht immer neu aufgebaut werden, sondern aus Performance-Gründen in einem Pool gehalten werden.

Die Schnittstelle DataSource

- ☞ Die Lösung ist eine Abstraktionsschicht durch DataSource-Objekte, die als administrierte Objekte in den Namenskontext gelegt werden.
 - ▶ Das passiert auf der Serverseite, etwa durch XML-Dokumente.
- ☞ Der Client erfragt den Namensdienst über JNDI und bekommt das DataSource-Objekt.
- ☞ DataSource-Objekt liefert die JDBC-Connection.

3

example-ds.xml für JBoss

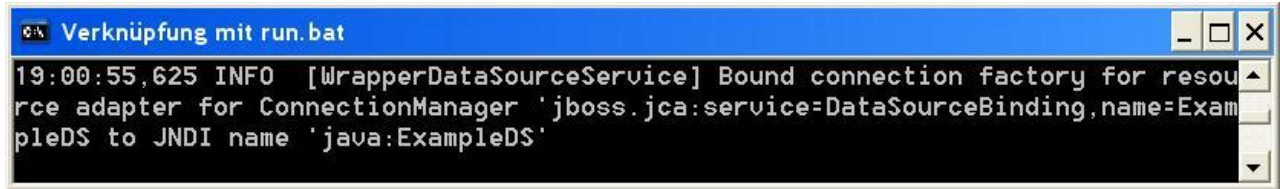
```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>ExampleDS</jndi-name>
    <connection-url>jdbc:hsqldb:file:c:/ExampleDB</connection-url>
    <driver-class>org.hsqldb.jdbcDriver</driver-class>
    <user-name>SA</user-name>
    <password></password>
    <min-pool-size>5</min-pool-size>
    <max-pool-size>20</max-pool-size>
    <idle-timeout-minutes>0</idle-timeout-minutes>
    <track-statements/>
  </local-tx-datasource>
</datasources>
```

Vorlagen für diese Art Dateien finden sich unter *jboss-4.0.x/docs/examples/jca*

4

Deployment

- ☞ Kopiere die Datei *example-ds.xml* in das Verzeichnis *jboss-4.0.y\server\default\deploy*.

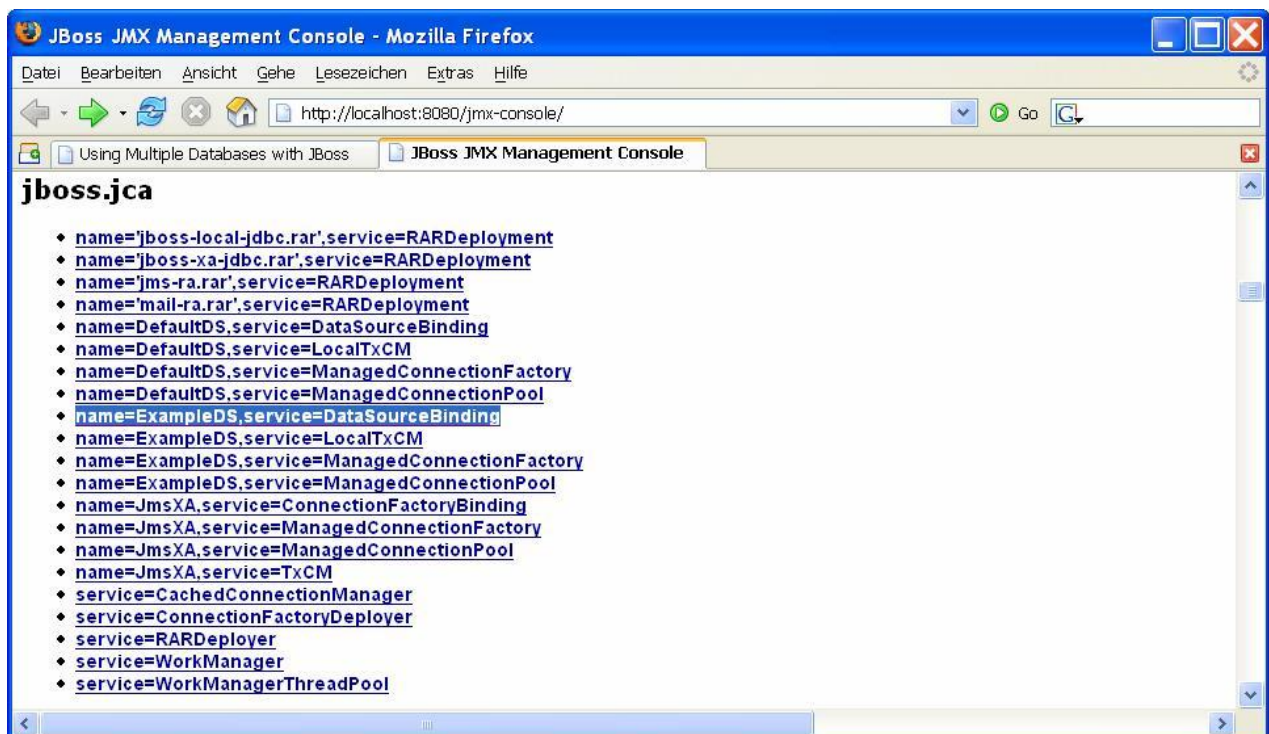


```
Verknüpfung mit run.bat
19:00:55,625 INFO [WrapperDataSourceService] Bound connection factory for resource adapter for ConnectionManager 'jboss.jca:service=DataSourceBinding,name=ExampleDS to JNDI name 'java:ExampleDS''
```

- ☞ Die Datenquelle wird jetzt bei JBoss angemeldet.
- ☞ Anschließend können wir die neue Datenquelle unter der JBoss Management Console besichtigen.

5

Die neue DataSource in der Console



```
JBoss JMX Management Console - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://localhost:8080/jmx-console/
Using Multiple Databases with JBoss JBoss JMX Management Console
jboss.jca
• name='jboss-local-jdbc.rar',service=RARDeployment
• name='jboss-xa-jdbc.rar',service=RARDeployment
• name='jms-ra.rar',service=RARDeployment
• name='mail-ra.rar',service=RARDeployment
• name=DefaultDS,service=DataSourceBinding
• name=DefaultDS,service=LocalTxCM
• name=DefaultDS,service=ManagedConnectionFactory
• name=DefaultDS,service=ManagedConnectionPool
• name=ExampleDS,service=DataSourceBinding
• name=ExampleDS,service=LocalTxCM
• name=ExampleDS,service=ManagedConnectionFactory
• name=ExampleDS,service=ManagedConnectionPool
• name=JmsXA,service=ConnectionFactoryBinding
• name=JmsXA,service=ManagedConnectionFactory
• name=JmsXA,service=ManagedConnectionPool
• name=JmsXA,service=TxCM
• service=CachedConnectionManager
• service=ConnectionFactoryDeployer
• service=RARDeployer
• service=WorkManager
• service=WorkManagerThreadPool
```

6

@ejb.resource-ref

Physikalischen auf logischen Namen abbilden

- ☞ Im Allgemeinen wird man eine zusätzliche Indirektion wählen.
 - ▶ Der Namen der Datenquelle soll von dem physikalisch gewählten Namen unabhängig bleiben.
- ☞ Erst definiert man das Mapping vom physikalischen Namen auf den logischen und dann gibt man ihn frei.

```
@jboss.resource-ref
```

```
res-ref-name="jdbc/javatutor/Datenbank"
```

```
jndi-name="java:/ExampleDS"
```

```
@ejb.resource-ref
```

```
res-name="jdbc/javatutor/Datenbank"
```

```
res-type="javax.sql.DataSource"
```

Der Deployment-Deskriptor

- 👉 Der Teil von `@ejb.resource-ref` führt zu einem Fragment in der Datei `ejb-jar.xml`; der Teil mit `@jboss.resource-ref` ist spezifisch für JBoss und führt zu einem Eintrag in `jboss.xml`.
- 👉 Eigentlich ist unsere Trennung zwischen physikalischem Namen und logischen Namen nicht perfekt.
 - ▶ Wenn sich tatsächlich der Name der Datenquelle ändern, müssen wir in den Quellcode und XDoclet laufen lassen, egal ob wir einen physikalischen oder logischen Namen nutzen.
 - ▶ Denkt man vor die Zeit von XDoclet, als man Quellcode und Deployment-Deskriptoren getrennt verwaltet hat, war das eleganter: Man ändert nur einen Eintrag im Deployment-Deskriptor. Deswegen macht die Trennung ja auch Sinn!