

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.
- 👉 Eine DataSource wird serverseitig als **administriertes Objekt** in den Namenskontext (also JNDI) gelegt.
 - ▶ Üblicherweise beschreibt eine ein XML-Dokumente den Namen der DataSource und Verbindungsparameter.
- 👉 Der Client erfragt den Namensdienst über JNDI und bekommt das DataSource-Objekt.
 - ▶ Das DataSource-Objekt liefert dann JDBC-Connection.

chris-ds.xml für JBoss

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
  <local-tx-datasource>
    <jndi-name>ChrisDS</jndi-name>
    <connection-url>jdbc:hsqldb:file:c:/ChrisDB</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*.

Deployment

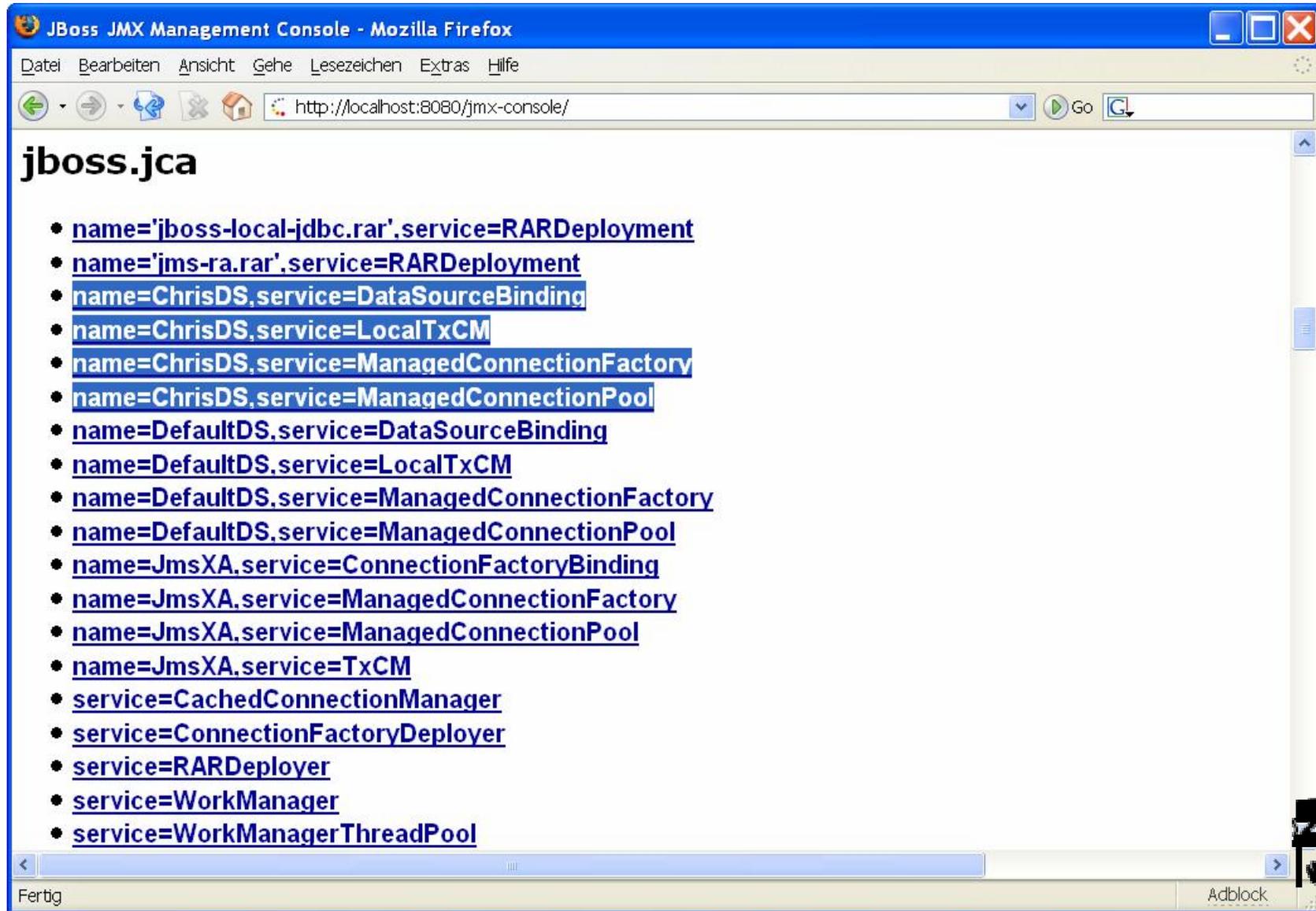
- ☞ Zum Deployment wird die Datei *chris-ds.xml* in das Verzeichnis *jboss-4.0.y\server\default\deploy* kopiert.
- ☞ JBoss erkennt die angelegte Datei und meldet die Datenquelle automatisch an.



```
C:\WINDOWS\system32\cmd.exe
12:06:11,156 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jboss.jca:service=DataSourceBinding,name=ChrisDS' to JNDI name 'java:ChrisDS'
```

- ☞ Anschließend können wir die neue Datenquelle unter der JBoss Management Console besichtigen.

Die neue DataSource in der Console



JBoss JMX Management Console - Mozilla Firefox

http://localhost:8080/jmx-console/

jboss.jca

- [name='jboss-local-jdbc.rar',service=RARDeployment](#)
- [name='jms-ra.rar',service=RARDeployment](#)
- [name=ChrisDS,service=DataSourceBinding](#)
- [name=ChrisDS,service=LocalTxCM](#)
- [name=ChrisDS,service=ManagedConnectionFactory](#)
- [name=ChrisDS,service=ManagedConnectionPool](#)
- [name=DefaultDS,service=DataSourceBinding](#)
- [name=DefaultDS,service=LocalTxCM](#)
- [name=DefaultDS,service=ManagedConnectionFactory](#)
- [name=DefaultDS,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](#)

Fertig Adblock

Zugriff auf die Datenquelle

Zugriff auf die DataSource

- 👉 Die DataSource ist unter dem Namen **java:/ChrisDS** verfügbar.
 - ▶ Also nur lokal und nicht für einen remote Client.
- 👉 Für den Zugriff gibt es zwei Möglichkeiten:
 - ▶ Der Server injiziert die DataSource über ein Attribut oder Setter. Das ist der EJB 3-Stil.
 - ▶ Das Programm holt sich manuell die DataSource über den JNDI-Kontext.

Injizierung mit @Resource

- ☞ Die Injizierung ist am Einfachsten:

```
@Resource( mappedName="java:/ChrisDS" )  
DataSource chrisDS;
```

- ☞ Das Attribut `mappedName` ist bei Datenquellen nötig und bezeichnet den JNDI-Namen der DataSource.

Aktives Holen über den JNDI-Kontext

```
Context naming = new InitialContext();  
DataSource chrisDS = (DataSource)naming.lookup(  
    "java:/ChrisDS");  
Connection con = chrisDS.getConnection();  
...  
con.close();
```

