

# Java 2 Enterprise Edition



# Message-driven beans

*Helder da Rocha*  
[www.argonavis.com.br](http://www.argonavis.com.br)

- *Este capítulo é uma breve introdução aos Message-Driven Beans (MDB).*
- *MDBs são consumidores de mensagens JMS*
  - *Já sabemos como criar produtores e consumidores JMS*
  - *Neste capítulo veremos apenas como transformá-los em enterprise Java Beans*
- *Para quem deseja se aprofundar mais no assunto, veja o capítulo "Message Driven Beans" no J2EE Tutorial e as implementações de MDBs no Duke's bank.*

# O que é um Message-Driven Bean

- Bean guiado por mensagens, ou eventos
  - Um event-handler distribuído em forma de EJB
- MDBs **consomem** mensagens enviadas a filas e canais JMS
  - Não é possível enviar uma mensagem diretamente a um message-driven bean (envia-se mensagens a um canal que o bean escuta)
  - Acoplamento entre clientes e MDB resume-se à conhecimento de canal comum de comunicações
- Um message-driven bean é um EJB mas
  - **Não possui interfaces** Home ou Remote
  - Possui apenas **um método** que recebe qualquer tipo de mensagem
  - **Não devolve tipos ou exceções**
  - Não possui estado (são **stateless**)
  - Podem ser ouvintes de uma **fila** ou assinantes duráveis ou não-duráveis de um **canal** (Topic) JMS

# Desenvolvendo MDB

- *Message-driven beans implementam duas interfaces*

```
public interface javax.jms.MessageListener
```

```
public interface javax.ejb.MessageDrivenBean
```

```
extends javax.ejb.EnterpriseBean
```

- *Métodos de MessageDrivenBean*

- **ejbRemove()**: chamado pelo container quando o message-driven bean está sendo removido do pool

- **ejbCreate()**: chamado quando um novo message-driven bean é instanciado

- **setMessageDrivenContext(MessageDrivenContext context)**: chamado antes do **ejbCreate()** quando o bean está sendo adicionado ao pool

- *Método de MessageListener*

- **onMessage(Message m)**: chamado cada vez que uma mensagem é enviada para o canal do bean (se o bean estiver ativado).

# Exemplo: implementação de um MDB

```
import javax.ejb.*;
import javax.jms.*;
```

```
public class LogBean implements MessageDrivenBean, MessageListener {
```

```
    protected MessageDrivenContext ctx;
```

```
    public void setMessageDrivenContext(MessageDrivenContext ctx) {
        this.ctx = ctx;
```

```
    }
```

```
    public void ejbCreate() {}
```


```
    public void ejbRemove() {}
```

Métodos da interface  
MessageDrivenBean




```
    public void onMessage(Message msg) {
        TextMessage tm = (TextMessage) msg;
        try {
            String text = tm.getText();
            System.err.println("Received new message : " + text);
        } catch (JMSEException e) {
            e.printStackTrace();
        }
    }
}
```

Método da interface  
MessageListener

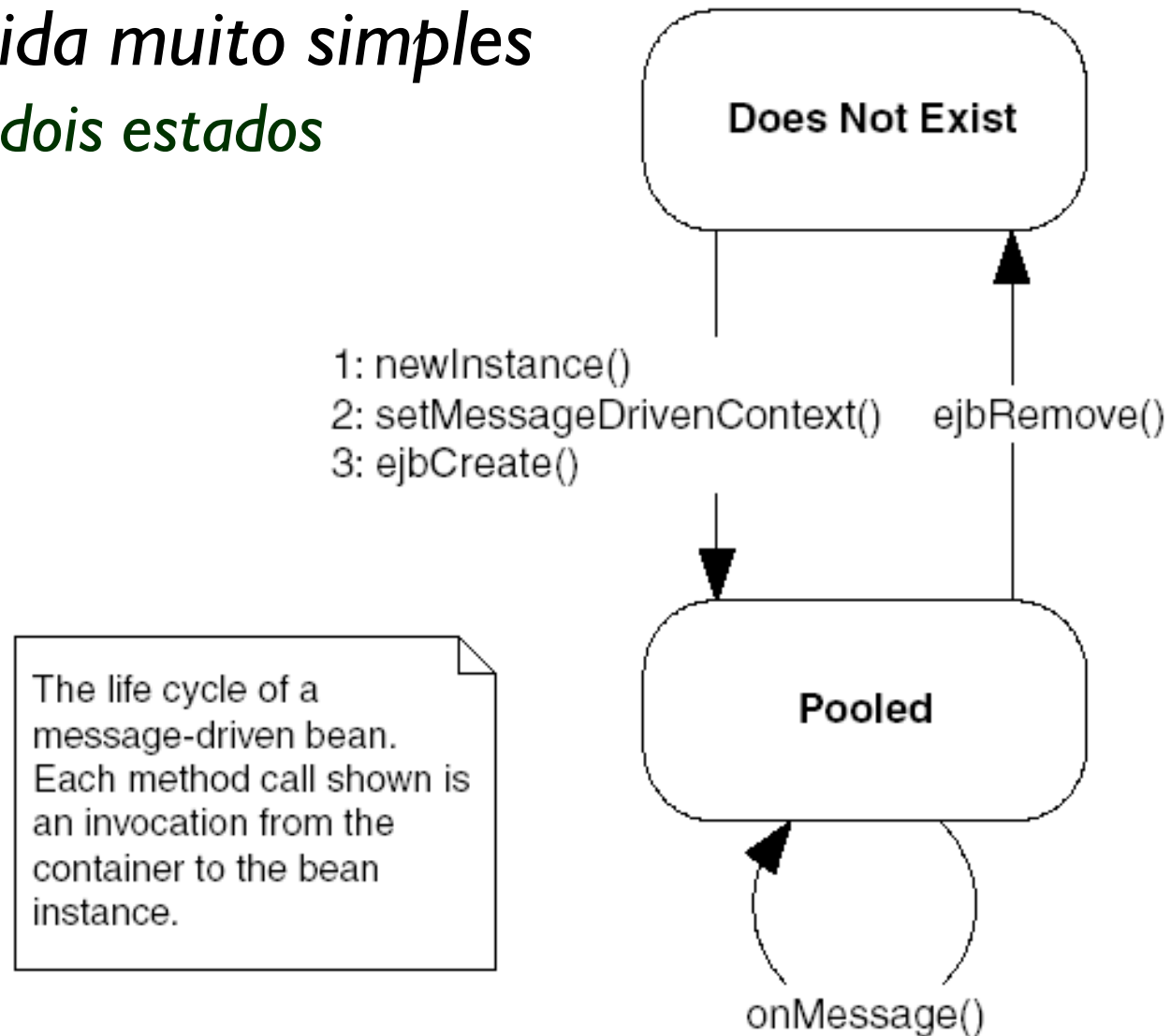


MDB não pode  
provocar exceção  
alguma



Este bean simplesmente  
imprime o texto contido na  
mensagem recebida

- *Ciclo de vida muito simples*
  - *Apenas dois estados*



**Figure 8.5** Life cycle of a message-driven bean.

# Deployment Descriptor

```
<ejb-jar>
  <enterprise-beans>
    <message-driven>

      <ejb-name>LogEJB</ejb-name>
      <ejb-class>examples.LogBean</ejb-class>
      <transaction-type>Container</transaction-type>

      <message-driven-destination>
        <destination-type>javax.jms.Topic</destination-type>
        <subscription-durability>NonDurable</subscription-durability>
      </message-driven-destination>

      <resource-ref>
        <res-ref-name>jms/TopicFactory</res-ref-name>
        <res-type>javax.jms.TopicConnectionFactory</res-type>
        <res-auth>Container</res-auth>
      </resource-ref>

    </message-driven>
  </enterprise-beans>
</ejb-jar>
```

*Canal (apenas a classe e tipo)*

*Fábrica de filas e canais*

# Vendor-specific

- Os nomes JNDI dos recursos que serão acessados globalmente (nome da fila ou canal, por exemplo), são definidos na configuração do fabricante
- No JBoss:

```
<jboss>
  <enterprise-beans>
    <message-driven>
      <ejb-name>LogEJB</ejb-name>
      <destination-jndi-name>
        topic/testTopic
      </destination-jndi-name>
      <resource-ref>
        <res-ref-name>jms/TopicFactory</res-ref-name>
        <jndi-name>ConnectionFactory</jndi-name>
      </resource-ref>
    </message-driven>
  </enterprise-beans>
</jboss>
```

Nome JNDI global do destino compartilhado pelo bean e pelos clientes

Referência do ENC java:comp/env

Nome JNDI global default da fábrica de destinos (fila ou canal) no JBoss



# Cliente é um cliente JMS comum

- Pode ser outro bean, cliente Web ou standalone

```
public class Client {  
  
    public static void main(String[] args) throws Exception {  
  
        Context ctx = new InitialContext(System.getProperties());  
        TopicConnectionFactory factory =  
            (TopicConnectionFactory) ctx.lookup("ConnectionFactory");  
        TopicConnection connection =  
            factory.createTopicConnection();  
        TopicSession session =  
            connection.createTopicSession(  
                false, Session.AUTO_ACKNOWLEDGE);  
  
        Topic topic = (Topic) ctx.lookup("topic/testTopic");  
        TopicPublisher publisher = session.createPublisher(topic);  
        TextMessage msg = session.createTextMessage();  
        msg.setText("This is a test message.");  
        publisher.publish(msg);  
    }  
}
```

Cliente externo (fora do ENC) usa nomes JNDI globais



# Execução dos exemplos

- *Diretório cap12/mejb2/*
- *Para montar e instalar a aplicação, use*
  - > ant jboss.deploy*
- *Uma vez instalada, rode o cliente:*
  - > ant run.jboss.client*
- *A execução não deve causar nenhuma reação no cliente, já que foi assíncrona "one-way".*
- *No servidor, aparece a mensagem enviada:*
  - [STDERR] Received new message : This is a test message.*
- *Veja também o exemplo do J2EE Tutorial (usando Queue) no subdiretório cap12/sun/*

- *1. Transforme os consumidores de mensagens JMS usados como exemplo no capítulo 4 em message-driven beans*
  - *a) Chame os beans a partir de um session bean que recebe a mensagem e o número de vezes em que ela é chamada de um cliente standalone ou Web.*

- [1] Ed Roman. *Mastering EJB 2. Chapter 8 - Introduction to Message Driven beans.*
- [2] J2EE Tutorial. *Message-driven Beans.*

*helder@ibpinet.net*

***www.argonavis.com.br***