CMP relations between Enterprise Java Beans (EJB) – eclipse, xdoclet, jboss

A step by step example showing how to develop CMP relations between EJBs using eclipse, xdoclet and MyEclipse. We use an example with an 1:n relation between two beans.

Generals

Author:
Sebastian Hennebrüder
Revised by Sascha Wolski

http://www.laliluna.de/tutorial.html – Tutorials for Struts, EJB, xdoclet and eclipse.

Date:
Updated January, 10th 2005

First Edition November, 10st 2004

PDF-Version des Tutorials:

http://www.laliluna.de/assets/tutorials/cmp_relation_between_ejb_en.pdf

Source Code des Projektes:

http://www.laliluna.de/assets/tutorials/cmp-relation-between-ejb.zip

Using the source code.

The source code does not include any libraries but the sources. Create a web project, add the libraries manually or with the help of MyEclipse and the extract the sources we provided to your project.

Development Tools
Eclipse 3.x
MyEclipse plugin 3.8
(A cheap and quite powerful Extension to Eclipse to develop Web Applications and EJB (J2EE) Applications. I think that there is a test version availalable at MyEclipse.)
Database
PostgreSQL 8.0 Beta or MySQL
Application Server
Jboss 3.2.5

Introduction

You should know how to create Entity EJBs. Look at the tutorials simple EJB and session fascade to learn this.

We are going to create the EJBs for a large library application. You are the chief developer and are entirely responsible for the complex modelling.
After a week of work, you know what to do:

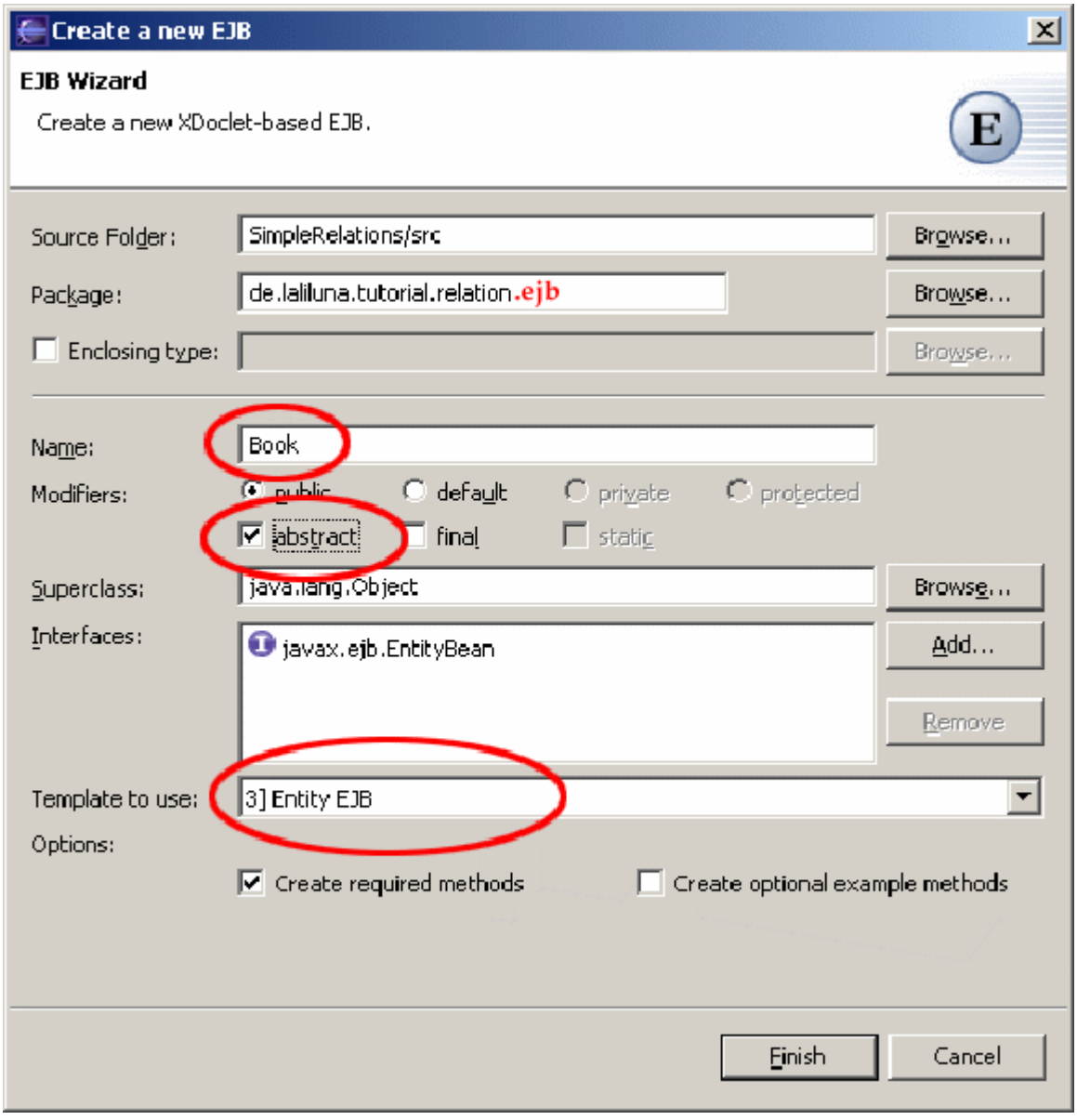
We will need two entity beans. The first one represents the book, the second one the library client. Using your brain you found out that there is a relation between the two EJBs. A book can be lent to a client.

OK, let's do it.

Create the beans

After creating a EJB project, create the book EJB. We don't give here any explanations to the basics. Look for the basic tutorial at http://www.lalilluna.de/tutorials.html

But don't forget: your package should end with .ejb = dot ejb !!!



Create abstract getters for the fields of the Book bean.

- String id
- String title
- boolean available

Type _gs and then press „Strg+Space“. Nothing happens?

;-) Are you a lazy type of person? Then think about using the following template:

```
/**
 * @ejb.interface-method view-type = "both"
 * @ejb.persistence column-name = "f${lowerName}"
 * @return
 */
public abstract ${type} get${upperName}();

/**
 * @ejb.interface-method view-type = "both"
 * @param ${lowerName}
 */
public abstract void set${upperName}(${type} ${lowerName});
```

Go to Menu „Windows -> Preferences“ and then choose java->editor->template, paste the template, give it a name and use it. I called it _gs.

Use „Tab“ to jump from field to field when you use a template.

Your primary key should look like:

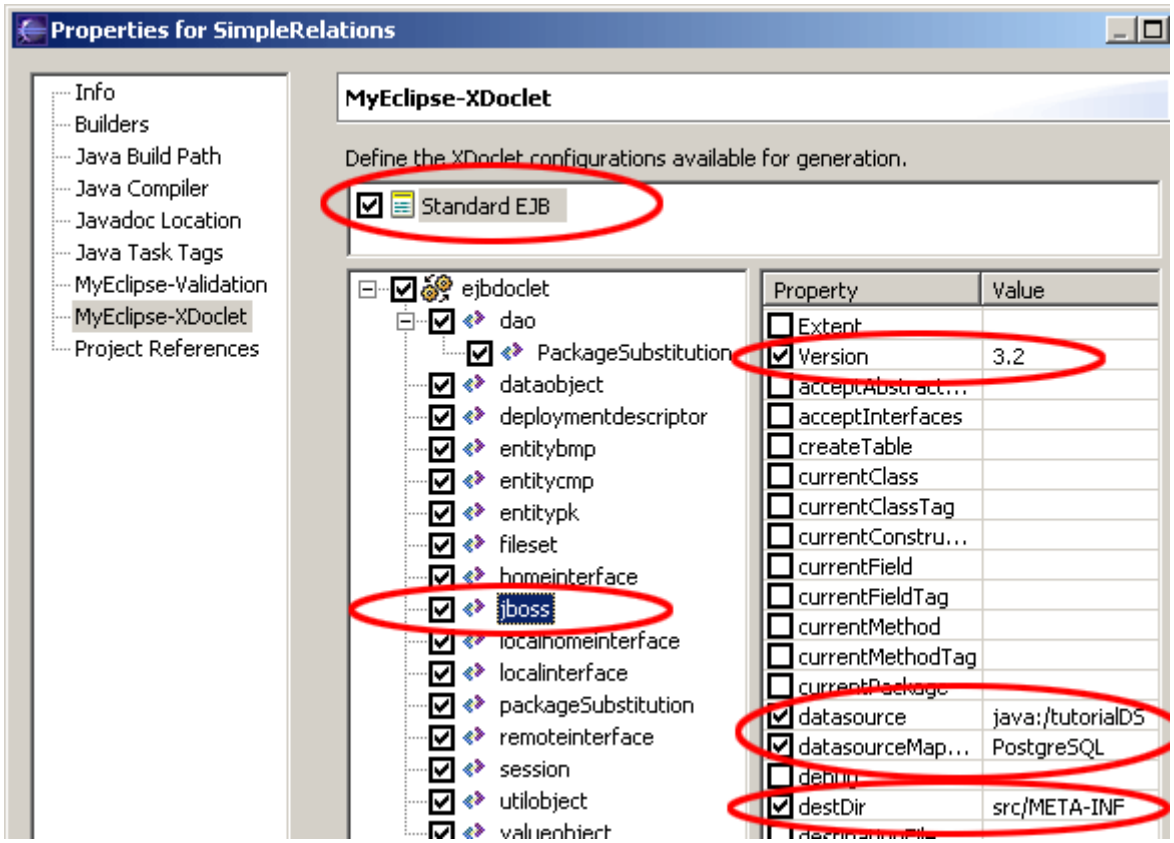
```
  /**
   * @ejb.interface-method view-type = "both"
   * @ejb.pk-field
   * @ejb.persistence column-name = "fid"
   * @return
   */
  public abstract String getId();

  /**
   * @ejb.interface-method view-type = "both"
   * @param id
   */
  public abstract void setId(String id);
```

Change the javaDoc in front of the class. We will repeatedly change the table structure. So its a good idea to specify create-table = "true" remove-table = "true" to create/remove tables when deploying/undeploying your project!

```
 * @ejb.bean name = "Book"
 *           type = "CMP"
 *           cmp-version = "2.x"
 *           display-name = "Book"
 *           description = "Book EJB"
 *           view-type = "both"
 *           jndi-name = "ejb/BookHome"
 *           local-jndi-name = "ejb/BookLocalHome"
 * primkey-field = "id"
 * @ejb.persistence table-name = "tbook"
 * @jboss.persistence table-name = "tbook"
 *                    create-table = "true"
 *                    remove-table = "true"
 * @ejb:util
 *      generate="physical"
 */
public abstract class Book implements EntityBean {
```

Configure the datasource. (this is explained in the basis tutorials)

Configure xDoclet for Jboss and run xDoclet by Myeclipse a first time.



It will create a BookUtils class. With this class we will create a unique primary key. So change your EJB create method to:

```
public String ejbCreate() throws CreateException {
  this.setId(BookUtil.generateGUID(this));
  return null;
}
```

Create the Client bean exactly like the Book bean.

Fields are:

- name
- id

Make sure that you successfully run Xdoclet before continuing. Deploy your beans to jboss to test if the deployment is successful. Now we go on to the more complex things!

Create the relation 1:n

We are having a 1:n relation. Each client may have several books. Add the following code to the client bean.

```
/**
 * @ejb.interface-method view-type = "local"
 * @ejb.relation name = "book-client" role-name = "client lends books"
 *
 * @return a Collection of BookLocal
 */
public abstract Collection getBook();

/**
 * @ejb.interface-method view-type = "local"
 * @param Collection of books
 */
public abstract void setBook(Collection books);
```

In your book bean add:

```
/**
 * @ejb.interface-method view-type = "local"
 * @ejb.relation name = "book-client" role-name = "book is lent by client"
 * @jboss.relation related-pk-field = "id" fk-column = "client_id"
 *                 fk-constraint = "true"
 * @return
 */
public abstract ClientLocal getClientLocal();

/**
 * @ejb.interface-method view-type = "local"
 * @param clientLocal
 */
public abstract void setClientLocal(ClientLocal clientLocal);
```

It is important that the relation name are exactly the same! Run Xdoclet to create your beans and try to deploy them. I saw the following quite often at the beginning of a class. (example for Book class)

```
* @ejb.ejb-ref view-type = "local" ejb-name = "Client" ref-name = "ejb/Client"
```

It is working without this, but just when you may have a problem one day, thing about it. CMP relations work only with local interfaces. So make sure that you specify your view-type="local" ! Do not allow yourself to make a bug by calling a relation getter from remote! But when I cannot call them from remote, how do I use the relations. The solution is about 1,7 cm below this line. ;-)

Create Business Logic

We will create a LibaryManager session bean to provide all the business logic we need. There are two methods lendBook and returnBook.

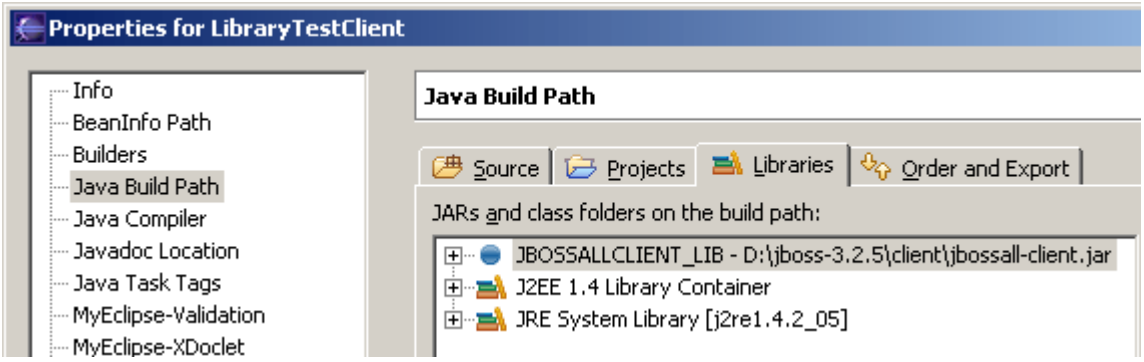
```
  /**
   * lend a book by client and sets availability to false
   * @ejb.interface-method view-type = "remote"
   * @param clientId
   * @param bookId
   * @throws EJBException
   */
  public void lendBook(String clientId, String bookId) throws EJBException{
    try {
      InitialContext initialContext = new InitialContext();

      // [laliluna] 01.10.2004 get homeinterface by JNDI lookup and find client
by id
      ClientLocalHome clientLocalHome = (ClientLocalHome) initialContext.lookup
(ClientLocalHome.JNDI_NAME);
      ClientLocal clientLocal = clientLocalHome.findByPrimaryKey(clientId);

      // [laliluna] 01.10.2004 get homeinterface by JNDI lookup and find book by
id
      BookLocalHome bookLocalHome = (BookLocalHome) initialContext.lookup
(BookLocalHome.JNDI_NAME);
      BookLocal bookLocal = bookLocalHome.findByPrimaryKey(bookId);

      if (!bookLocal.getAvailable())
        System.out.println("Book is not available");
      else
      {
      bookLocal.setClientLocal(clientLocal);
      bookLocal.setAvailable(false);
      }

      /*
       * [laliluna] 01.10.2004
       * alternative you may use
       * clientLocal.getBook().add(bookLocal);
       */

    } catch (NamingException e) {
      e.printStackTrace();
    } catch (FinderException e) {
      e.printStackTrace();
    }


  }

  /**
   * returns a book from a client and sets availability to true
   * @ejb.interface-method view-type = "remote"
   * @param clientId
   * @param bookId
   * @throws EJBException
   */
  public void returnBook(String clientId, String bookId) throws EJBException{
    try {
      InitialContext initialContext = new InitialContext();

      // [laliluna] 01.10.2004 get homeinterface by JNDI lookup and find client
by id
      ClientLocalHome clientLocalHome = (ClientLocalHome) initialContext.lookup
(ClientLocalHome.JNDI_NAME);
      ClientLocal clientLocal = clientLocalHome.findByPrimaryKey(clientId);

      // [laliluna] 01.10.2004 get homeinterface by JNDI lookup and find book by
```

```
id
      BookLocalHome bookLocalHome = (BookLocalHome) initialContext.lookup
(BookLocalHome.JNDI_NAME);
      BookLocal bookLocal = bookLocalHome.findByPrimaryKey(bookId);

      bookLocal.setAvailable(true);
      bookLocal.setClientLocal(null);
    } catch (NamingException e) {
      e.printStackTrace();
    } catch (FinderException e) {
      e.printStackTrace();
    }
  }
```

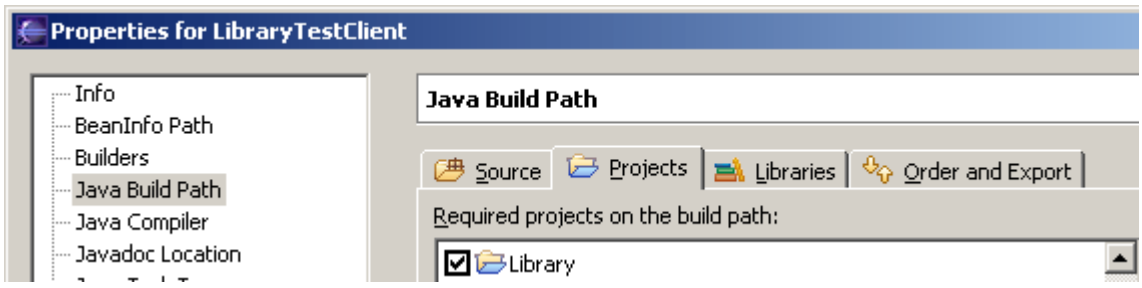
Create the session bean, deploy it and check that the deployment works. See the session bean facade tutorial when you need more information about creating them.

Create a test client

Create a java project and a Test class. You will need some libraries. Add them in the project properties. See the picture below.



Add the EJB project to the build path.



Implement something like the following to test your business logic.

```
package de.laliluna.tutorial.relation.client;

import java.rmi.RemoteException;
import java.util.Properties;

import javax.ejb.CreateException;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.rmi.PortableRemoteObject;

import de.laliluna.tutorial.library.interfaces.Book;
import de.laliluna.tutorial.library.interfaces.BookHome;
import de.laliluna.tutorial.library.interfaces.Client;
import de.laliluna.tutorial.library.interfaces.ClientHome;
import de.laliluna.tutorial.library.interfaces.LibraryManager;
import de.laliluna.tutorial.library.interfaces.LibraryManagerHome;

/**
 * @author HS
 *
 *
 */
public class TestClient {

  private Properties properties;

  public TestClient() {
    // [laliluna] the properties specifies where the JNDI lookups for the EJBs
can be made
    properties = new Properties();
    properties.put("java.naming.factory.initial",
        "org.jnp.interfaces.NamingContextFactory");
    properties.put("java.naming.factory.url.pkgs",
        "org.jboss.naming:org.jnp.interfaces");
    properties.put("java.naming.provider.url", "jnp://localhost:1099");
    properties.put("jnp.disableDiscovery", "true");
  }

  public static void main(String[] args) {
    TestClient testClient = new TestClient();
    testClient.testRelation();
  }

  public void testRelation() {
    try {
      InitialContext context = new InitialContext(properties);
      Object object;
      // [laliluna] 01.10.2004 get home interfaces
      object = context.lookup(BookHome.JNDI_NAME);
      BookHome bookHome = (BookHome) PortableRemoteObject.narrow(object,
          BookHome.class);

      object = context.lookup(ClientHome.JNDI_NAME);
      ClientHome clientHome = (ClientHome) PortableRemoteObject.narrow(object,
          ClientHome.class);

      object = context.lookup(LibraryManagerHome.JNDI_NAME);
      LibraryManagerHome libraryManagerHome = (LibraryManagerHome)
PortableRemoteObject
          .narrow(object, LibraryManagerHome.class);

      // [laliluna] 01.10.2004 create a dummy book and a client
      Book book1 = bookHome.create();
      book1.setTitle("Be nice");
      book1.setAvailable(true);
      String bookId1 = book1.getId();
```

```
      Book book2 = bookHome.create();
      book2.setTitle("Be very nice");
      book2.setAvailable(true);
      String bookId2 = book2.getId();

      Client client = clientHome.create();
      client.setName("Nasty boy");
      String clientId = client.getId();

      // [laliluna] 01.10.2004 know do some business logic
      LibraryManager libraryManager = libraryManagerHome.create();
      libraryManager.lendBook(clientId, bookId1);
      libraryManager.lendBook(clientId, bookId2);
      libraryManager.returnBook(clientId, bookId1);

    } catch (ClassCastException e) {
      e.printStackTrace();
    } catch (NamingException e) {
      e.printStackTrace();
    } catch (RemoteException e) {
      e.printStackTrace();
    } catch (CreateException e) {
      e.printStackTrace();
    }

  }
}
```

That's it. Congratulations.

How much time did you save with this tutorial? If it was a lot please think about a donation at http://www.laliluna.de/tutorials.html? We are pleased about every cent. ;-)