**Writing Enterprise Applications with Java 2
Platform, Enterprise Edition - part7**

Document Information

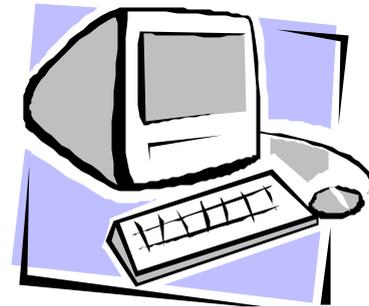|	
Document title:	Writing Enterprise Applications with Java 2 Platform, Enterprise Edition - part7
Document file name:	Writing_Enterprise_Applications_7_JunoYoon
Revision number:	<1.0>
Issued by:	< > junoyoon@orgio.net
Issue Date:	<2001/04/29>
Status:	final

Content Information

Audience	
Abstract	
	BMP .
Reference	Writing Enterprise Applications with JavaTM Platform, Enterprise Edition
	By Monica Pawlan
	(http://developer.java.sun.com/developer/online Training/J2EE/Intro/)
Benchmark information	

Table of Contents

Lesson 7 - JDBC technology   BMP(Bean Managed Persistence)

SQL   Cloudscape

.. ,

Bean   . CMP(Container
Managed Persistence)   ,

.   CMP   (override)
BMP(Bean Managed Persistence)   . BMP   CMP

SQL   SQL

Bean   .   BMP

Bean   1

.

BMP   J2EE   Bean

.

Bean

Lesson 3   BonusBean   Container Managed Bean   BonusBean

.   getBonus,

getSocSec,   bonus   socsec   Bean
ejbCreate   .

.   BMP

.

JDBC   SQL

.

Bean

.   CalcBean   calcBean   getRecord   2

BonusBean   getBonus   getSocsec   2
.   CalcBean   BonusBean   .
                                                                                         B
ean   .

- o setEntityContext: , EntityContext
        Bean   . EntityContext
                  .   Bean
                              , Context
                      .   Bean   setEntityContext   setSe
  ssionContext   .
- o ejbCreate: Bean   Create
                                  .   Create
                              (signature) ejbCreate
  .
- o ejbPostCreate: ejbCreate
                              .   ejbCreate
              ejbPostCreate   .   ejbPostCreate
                          . ejbPostCreate   Bean
        Bean
          .
  .
- o ejbRemove: Bean
                              .   J2EE
                              .
- o unsetEntityContext: ejbRemove   Bean
              ,   . unsetEntityContext
        Bean   .   Bean   unsetSessionContext
          .
- o ejbFindByPrimaryKey: Bean   findByPrimaryK
  ey   .
                  , Bean
      (Signature)   ejbFind <type>   .
- o ejbLoad   ejbStore:
                              .   set   Bean
              (set)   (get) ,   ejbStore
                      ,   ejbLoad
                      .   finder
              ejbLoad
        Bean   .
- o ejbActivate   ejbPassivate: , Bean   A
  ctivation(   ) Passvate(   )   .
  Swap   Bean   Swapping

. ejbPassiv
ate   Bean   Connection   Close
ejbActivate   Open   .

BonusBean

Bean Management Persistence   BonusBean
.   Persistence
.

Import

InitialContext, DataSource   Connection
. PreparedStatement   SQL Request
. ResultSet   query
. FinderException   SQLException   ,
.

```
package Beans;
import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EntityBean;
import javax.ejb.EntityContext;
import javax.naming.InitialContext;
import javax.sql.DataSource;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import javax.ejb.FinderException;
import java.sql.SQLException;
```

. java:comp/env/jdbc/BonusDB   (resourse
reference)   ,   Deploy   Bean   J2EE
.

Cloudscape   (CloudscapeDB)   (alias)   .
CloudscapeDB   BONUS   ,   jdbc/BonusD
B   jdbc/CloudscapeDB   .

```
public class BonusBean implements EntityBean {
private EntityContext context;
```

```
                      private Connection con;
                      private String dbName = "java:comp/env/jdbc/BonusDB";

                      private InitialContext ic = null;
                      private PreparedStatement ps = null;
                      private double bonus;
                      private String socsec;
```

System.out.println

. System.out.println

.

```
                  public double getBonus() {
                    System.out.println("getBonus");
                    return this.bonus;
                  }
                  public String getSocSec() {
                    System.out.println("getSocSec");
                    return this.socsec;
                  }
```

System.out.println   ,

.

ejbCreate

ejbCreate   Signature   CreateException   Remote-
Exception   SQLException   .   ejbCreate
SQL   (SQL
    .)   SQLException   ,
RemoteException   .

                              Primary key   String
                                          Bonus
                .   Primary Key
    Bonus   .

```
                  public String ejbCreate(double bonus, String socsec)
                      throws RemoteException,
                      CreateException,
                      SQLException {
                    this.socsec=socsec;
```

```
     this.bonus=bonus;

     System.out.println("Create Method");
     try {
//Establish database connection
   ic = new InitialContext();
   DataSource ds = (DataSource) ic.lookup(dbName);
   con = ds.getConnection();
//Use PreparedStatement to form SQL INSERT statement
//to insert into BONUS table
   ps = con.prepareStatement(
       "INSERT INTO BONUS VALUES (? , ? ) ");
//Set 1st PreparedStatement value marked by ?  , with
//socsec and the 2nd value marked by ? ) with bonus
   ps.setString(1, socsec);
   ps.setDouble(2, bonus);
   ps.executeUpdate();
   } catch (javax.naming.NamingException ex) {
     ex.printStackTrace();
   } finally {
//Close database connection
   ps.close();
   con.close();
   }
//Return primary key
   return socsec;
   }
```

ejbPostCreate

           ejbCreate   Signature

                                                              .


```
public void ejbPostCreate(double bonus,
           String socsec)
           throws RemoteException,
           CreateException,
           SQLException {
System.out.println("Post Create");
   }
```

ejbFindByPrimaryKey

BonusBean   CMP   ejbFindByPrimaryKey   .
 CMP   Primary Key   ,   Pri
mary Key   .
   BonusBean   BMP   ,

SQLException   . CMP   RemoteException   FinderExcepti
on   .
Find Operation   ejbFindByPrimaryKey   Primary Key
    Primary Key   Primary Key   ejbLoad
        bonus   socsec   BonusBean   .
                  Primary Key   Type   String
                              Bonus
    .   ,   Primart Key   Bonus
            .

```java
public String ejbFindByPrimaryKey(String primaryKey)
        throws RemoteException, FinderException,
        SQLException {
  System.out.println("Find by primary key");
  try {
//Establish database connection
     ic = new InitialContext();
     DataSource ds = (DataSource) ic.lookup(dbName);
     con = ds.getConnection();
//Use PreparedStatement to form SQL SELECT statement
//to select from BONUS table
     ps = con.prepareStatement(
     "SELECT socsec FROM BONUS WHERE socsec = ?  ");
     ps.setString(1, primaryKey);
//Use ResultSet to capture SELECT statement results
     ResultSet rs = ps.executeQuery();
//If ResultSet has a value, the find was successful,
//and so initialize and return key
     if(rs.next()) {
        key = primaryKey;
     } else {
        System.out.println("Find Error");
     }
  } catch (javax.naming.NamingException ex) {
     ex.printStackTrace();
  } finally {
//Close database connection
     ps.close();
     con.close();
  }
//Return primary key
  return key;
}
```

ejbLoad

ejbFindByPrimaryKey

Bean   .

```java
public void ejbLoad() {
  System.out.println("Load method");
   try {
//Establish database connection
     ic = new InitialContext();
     DataSource ds = (DataSource) ic.lookup(dbName);
     con = ds.getConnection();
//Use PreparedStatement to form SQL SELECT statement
//to select from BONUS table
     ps = con.prepareStatement(
        "SELECT * FROM BONUS WHERE SOCSEC = ? ");
     ps.setString(1, this.socsec);
//Use ResultSet to capture SELECT statement results
     ResultSet rs = ps.executeQuery();
//If ResultSet has a value, the find was successful
     if(rs.next()){
        this.bonus = rs.getDouble(2);
     } else {
        System.out.println("Load Error");
     }
   } catch (java.sql.SQLException ex) {
     ex.printStackTrace();
   } catch (javax.naming.NamingException ex) {
     ex.printStackTrace();
   } finally {
     try {
//Close database connection
        ps.close();
        con.close();
     } catch (java.sql.SQLException ex) {
        ex.printStackTrace();
     }
   }
}
```

ejbStore

Bean   (set)   (get)

Bean

.

```java
public void ejbStore() {
  system.out.println("Store method");
   try {
```

```
//Establish database connection
    DataSource ds = (DataSource) ic.lookup(dbName);
    con = ds.getConnection();
//Use PreparedStatement to form SQL UPDATE statement
//to update BONUS table
    ps = con.prepareStatement(
    "UPDATE BONUS SET BONUS = ?  WHERE SOCSEC = ? ");
//Set 1st PreparedStatement value marked by ?  with
//bonus and the 2nd value marked by ? ) with socsec
    ps.setDouble(1, bonus);
    ps.setString(2, socsec);
    int rowCount = ps.executeUpdate();
  } catch (javax.naming.NamingException ex) {
    ex.printStackTrace();
  } catch (java.sql.SQLException ex) {
    ex.printStackTrace();
  } finally {
    try {
//Close database connection
        ps.close();
        con.close();
    } catch (java.sql.SQLException ex) {
      ex.printStackTrace();
    }
  }
}
```

ejbRemove

Bean

.   JavaBean   BonusBean

.

ejbRemove   .   ejbRemove   ejbRemove
socsec   Primary Key (socsec)   ,   Bean

.

```
public void ejbRemove()
      throws RemoteException {
System.out.println("Remove method");
try {
  DataSource ds = (DataSource) ic.lookup(dbName);
  con = ds.getConnection();
  ps = con.prepareStatement(
      "DELETE FROM BONUS WHERE SOCSEC = ? ");
  ps.setString(1, socsec);
  ps.executeUpdate();
} catch (java.sql.SQLException ex) {
```

```
      ex.printStackTrace();
    } catch (Exception ex) {
      ex.printStackTrace();
      try {
        ps.close();
        con.close();
      } catch (java.sql.SQLException ex) {
        ex.printStackTrace();
      }
    }
```

ejbActivate

Bean   ,   Bean   Bean
      Temporary   .   Bean
                            ,   Bean   .
EntityContext   getPrimaryKey   Bean   Primary
Key   .   ,   Primary Key
      Bean   .

```
    public void ejbActivate() {
      System.out.println("Activate method");
      socsec = (String) context.getPrimaryKey();
    }
```

ejbPassivate

Bean   ,   Bean   Bean
Temporary   .   Bean
                      ,   Bean   .   Bean
                              Primary Key   null   .

```
    public void ejbPassivate() {
      System.out.println("Passivate method");
      socsec = null;
    }
```

setEntityContext

                                        Bean   Context   .
ejbActivate   Context   getPrimarykey
    Bean   .

```
    public void setEntityContext(
              javax.ejb.EntityContext ctx){
      System.out.println("setEntityContext me thod");
```

```
        this.context = ctx;
    }
```

unsetEntityContext

Bean   ejbRemove   Context
null   . unsetEntityContext
              Bean   .

```
public void unsetEntityContext(){
    System.out.println("unsetEntityContext method");
    ctx = null;
}
```

CalcBean   JBonusBean

BonusBean   SQL   BonusBean
      CalcBean.calcbonus   java.sql.SQLException
        .   .

```
public class CalcBean implements SessionBean {
    BonusHome homebonus;

    public Bonus calcBonus(int multiplier,
                double bonus, String socsec)
                throws RemoteException,
                SQLException,
                CreateException {

    Bonus theBonus = null;
    double calc = (multiplier*bonus);
    try {
        InitialContext ctx = new InitialContext();
        Object objref = ctx.lookup("bonus");
        homebonus = (BonusHome)
                PortableRemoteObject.narrow(
                objref, BonusHome.class);
        } catch (Exception NamingException) {
            NamingException.printStackTrace();
        }
//Store data in entity Bean
    theBonus=homebonus.create(calc, socsec);
    return theBonus;
    }
```

JBonusBean   CalcBean   SQLException
.   DuplicateKeyExcpetion   CreateException   catch
(javax.ejb.CreateException e)   .

```java
public double getBonusAmt() {
  if(strMult ! = null){
    Integer integerMult = new Integer(strMult);
    int multiplier = integerMult.intValue();
    try {
      double bonus = 100.00;
      theCalculation = homecalc.create();
      Bonus theBonus = theCalculation.calcBonus(
                multiplier, bonus, socsec);
      Bonus record = theCalculation.getRecord(
                socsec);
      bonusAmt = record.getBonus();
      socsec = record.getSocSec();
    } catch (java.sql.SQLException e) {
      this.bonusAmt = 0.0;
      this.socsec = "000";
      this.message = e.getMessage();
    } catch (javax.ejb.CreateException e) {
      this.bonusAmt = 0.0;
      this.socsec = "000";
      this.message = e.getMessage();
    } catch (java.rmi.RemoteException e) {
      this.bonusAmt = 0.0;
      this.socsec = "000";
      this.message = e.getMessage();
    }
    genXML();
    return this.bonusAmt;
  } else {
    this.bonusAmt = 0;
    this.message = "None. ";
    return this.bonusAmt;
  }
}
```

BMP   , CloudscapeDB   BONUS
. CMP   ,
.

createTable.sql   cloudTable.sh (Unix)

cloudTable.bat (Windows/NT)   2   .
                    , createTable.sql   ~/J2EE/Beans
cloudTable.sh (Unix)   cloudTable.bat (Windows/NT)   ~/J2EE
                        .

                    , Beans   ,   .

Unix:
    ../cloudTable.sh
Windows/NT:
    ..\cloudTable.bat

createTable.sql

                                                                                .

    drop table bonus;

    create table bonus
    (socsec varchar(9) constraint pk_bonus primary key, bonus decimal(10,2));

    exit;

cloudTable.bat

                                                                                .


```
rem cloudTable.bat
rem Creates BONUS table in CloudscapeDB.
rem
rem Place this script in ~\J2EE
rem To run: cd ~\J2EE\cloudTable.sh
rem
rem Change this next line to point to *your*
rem j2sdkee1. 2.1 installation
rem
set J2EE_HOME=\home \monicap\J2EE\j2sdkee1. 2.1
rem
rem Everything below goes on one line
java -Dij.connection.CloudscapeDB=
jdbc:rmi://localhost:1099/jdbc:cloudscape:
CloudscapeDB\;create=true -Dcloudscape.system.home=
```

```
%J2EE_HOME%\cloudscape -classpath
%J2EE_HOME%_ib\cloudscape\client.jar;
%J2EE_HOME%_ib\cloudscape\ tools.jar;
%J2EE_HOME%_ib\cloudscape\cloudscape.jar;
%J2EE_HOME%_ib\cloudscape\RmiJdbc.jar;
%J2EE_HOME%_ib\cloudscape_icense.jar;
%CLASSPATH% -ms16m -mx32m
COM.cloudscape.tools.ij createTable.sql
```

cloudTable.sh

.

```
#! /bin/sh
#
# cloudTable.sh
# Creates BONUS table in CloudscapeDB.
#
# Place this script in ~\J2EE
# To run: cd ~\J2EE\cloudTable.sh
#
# Change this next line to point to *your*
# j2sdkee1. 2.1 installation
#
J2EE_HOME=/home/monicap/J2EE/j2sdkee1. 2
#
# Everything below goes on one line
java -Dij.connection.CloudscapeDB=jdbc:rmi:
//localhost:1099/jdbc:cloudscape:CloudscapeDB\;
create=true -Dcloudscape.system.home=
$J2EE_HOME/cloudscape -classpath
$J2EE_HOME/lib/cloudscape/client.jar:
$J2EE_HOME/lib/cloudscape/tools.jar:
$J2EE_HOME/lib/cloudscape/cloudscape.jar:
$J2EE_HOME/lib/cloudscape/RmiJdbc.jar:
$J2EE_HOME/lib/cloudscape/license.jar:
${CLASSPATH} -ms16m -mx32m
COM.cloudscape.tools.ij createTable.sql
```

JAR

Bean   Bean JAR   .   Bean
1   JAR   , 2BeansJar   JAR
                    . CalcBean   Bean   JAR
(Lesson 3)   .

BonusBean   ,

.

2   JAR   Bean   , BonusBean   J
AR   JAR   .

BonusBean   JAR

.

[EJB JAR]:

- o   [Add] ([Contents]   )   .
- o   Beans   .
- o   [Bonus.class]   .
- o   [Add]   .
- o   [BonusBean.class]   .
- o   [Add]   .
- o   [BonusHome.class]   .
- o   [Add]   .

[Enterprise Bean JAR classes]:

- o   Beans/Bonus.class ,Beans/BonusHome.class   Beans/BonusBean.class

  .

- o   [OK]   .

[EJB JAR]:

- o   [Next]   .

[General]:

- o   Beans.BonusBean   ,Beans.BonusHome   , Be
      ans.Bonus   .
- o   BonusBean   .
- o   [Entity]   .
- o   [Next]   .

[Entity Settings]:

- o   [Bean-managed persistence]   .
- o   Primary Key   java.lang.String   .   Primary Key
                                 . Private   Primary Key   .
- o   [Next]   .

[Environment Entries]:

o   [Next]   .   Bean   (   )
                            .

[Enterprise Bean References]:

o   [Next]   .

[**Resource References]:**

o   [Add]   .
o   [Coded Name]   1   jdbc/BonusDB   . [Type]   ja
    vax.sql.DataSource   [Authentication]   Container
                       .

o   [Next]   .

[Security]:

o   [Next]   .   Bean   Security Role
                    .

[Transaction Management]:

o   [Container-managed transactions]   . (   )
o   create ,findByPrimaryKey, getBonus   getSocSec   requ ired
                    .

                                                            Commit   .
                                ,   Enterprise JavaBeans Developer's Guid
    e   Chapter 6   .
o   [Next]   .

[Review Settings]:

o   [Finish]   .

[Inspecting]   :

o   2BeansApp   [JNDI names]   .
o   CalcBean calcs BonusBean   bonus ,   jdbc/BonusDB   jdbc/Cloud
    scape   .

(Verifier)   .

Verifier   Bean   compiler
                                                    .

            - Verify   Save   ,
                                                    .

Verify :

- o   2BeansApp   ,**[Tools]**   **[Verifier]**
        .

- o   [OK]   .
                                                    .

- o
                                Verifier   .

- o
        -        1.2. 1   ,tests app.WebURI
            .   , Deploy   WAR   WAR   . war
  (extension)   .
                ,   J2EE
        .

            :

- o   [Tools]   [Deploy Application]   . [Deploy BonusApp]
                                        .

- o   J2EE
                                        .

- o   [Return Client Jar]   .   ,
            Bean   JAR
        .

- o   [Next]   .
- o   CalcBean   JNDI   calcs   ,BonusBean   JNDI   bonus
                            . , BonusDB   JNDI   jdbc/Cloudscape
                    .   JNDI   Ret
        urn   .
- o   [Next]   . Context Root   JSPRoot
                .   Return
        .

- o   [Next]   .
- o   [Finish]   .

.

o   [OK]   .

Web   8000   . bonus.jsp
http://localhost:8000/JSPRoot/bonus.jsp   .   Deploy   JSP
.

o   .
o   [Submit]   . Bonus.jsp
HTML   .

, J2EE
.

. Cloudscape
.

,   .

Cannot find principal mapping information for data source with JNDI name
jdbc/Cloudscape

J2EE   (   )

setEntityContext method
Create Method
Post Create
setEntityContext method
Find by primary key
Load method
getBonus
Store method
Load method
getSocSec
Store method
Find by primary key
Load method
getSocSec
Store method
Load method
getBonus
Store method
<?xml version="1. 0"? >

```
        <bonusCalc ssnum="777777777" bonusAmt="300. 0" />
    </report>
```

Bean   BMP   .

http://java.sun.com/j2ee/j2sdkee/techdocs/guides/ejb/html/Entity.fm.html

,   .

http://java.sun.com/j2ee/j2sdkee/techdocs/guides/ejb/html/Database.fm.html