



BEA WebLogic Workshop™ Help

Version 8.1 SP4
December 2004

Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software–Restricted Rights Clause at FAR 52.227–19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013, subparagraph (d) of the Commercial Computer Software—Licensing clause at NASA FAR supplement 16–52.227–86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E–Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

Table of Contents

EJBs_ClientApps Samples.....	1
automaticPK Samples.....	2
CustomerControl.jcx Sample.....	3
CustomerControlTest.jws Sample.....	4
beanManagedPersistence Samples.....	6
BMPItemBeanCtrl.jcx Sample.....	7
BMPItemBeanCtrlTest.jws Sample.....	8
finderMethods Samples.....	10
ItemsBeanControl.jcx Sample.....	11
ItemsBeanControlTest.jws Sample.....	12
ManufacturerBeanControl.jcx Sample.....	15
ManufacturerBeanControlTest.jws Sample.....	16
homeMethods Samples.....	19
ItemsBeanControl.jcx Sample.....	20
ItemsBeanControlTest.jws Sample.....	21
messageDriven Samples.....	24
Controller.jpf Sample.....	25
FacadeBeanControl.jcx Sample.....	30
Index.jsp Sample.....	31
MessageSender.java Sample.....	33
MessageSenderImpl.jcs Sample.....	34
SendToQueueControl.jcx Sample.....	36
SendToTopicControl.jcx Sample.....	37

Table of Contents

security Samples.....	38
RoleCheckerControl.jcx Sample.....	39
RoleCheckerControlTest.jws Sample.....	40
selectMethods Samples.....	42
ItemsBeanControl.jcx Sample.....	43
ItemsBeanControlTest.jws Sample.....	44
valueObject Samples.....	47
AdminBeanControl.jcx Sample.....	48
Controller.jspf Sample.....	49
Index.jsp Sample.....	54
ItemsManufacturer.jsp Sample.....	56
NewManufacturer.jsp Sample.....	58

EJBs_ClientApps Samples

This section contains source code for the following samples.

Samples Included in This Section

automaticPK Samples

beanManagedPersistence Samples

finderMethods Samples

homeMethods Samples

messageDriven Samples

security Samples

selectMethods Samples

valueObject Samples

automaticPK Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomerControl.jcx Sample

CustomerControlTest.jws Sample

CustomerControl.jcx Sample

This topic includes the source code for the CustomerControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/automaticPK/

Sample Source Code

```
package automaticPK;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.AutomaticPK_CustomerLocalHome"
 */
public interface CustomerControl
    extends automaticPK.AutomaticPK_CustomerHome,    // home interface
            automaticPK.AutomaticPK_CustomerLocal,  // bean interface
            com.bea.control.EntityEJBControl,        // control interface
            com.bea.control.ControlExtension         // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

CustomerControlTest.jws Sample

This topic includes the source code for the CustomerControlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/automaticPK/

Sample Source Code

```
package automaticPK;

/**
 */
public class CustomerControlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public automaticPK.CustomerControl customerControl;

    /** @common:operation
     * @jws:conversation phase="start"
     */
    public void startTestDrive() {}

    /** @common:operation
     * @jws:conversation phase="finish"
     */
    public void finishTestDrive() {}

    /**
     * The return type of this method was not supported in Web Service operations
     */
    /** @common:operation
     * @jws:conversation phase="continue"
     */
    public void findByPrimaryKey(java.lang.Integer arg0) throws Exception
    { customerControl.findByPrimaryKey(arg0); }

    /**
     * The return type of this method was not supported in Web Service operations
     */
    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * Enter the last name of an existing customer.
     * </font>
     */
    /** @common:operation
     * @jws:conversation phase="continue"
     */
    public void findByLastName(java.lang.String arg0) throws Exception
    { customerControl.findByLastName(arg0); }

    /**
     * The return type of this method was not supported in Web Service operations
     */
    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * Enter a first name and last name to create a new customer. A unique
     * value generated by the table MyNamedSequence is used as primary key
     */
}
```


navEJBs_ClientApps.html Sample

```
* </font>
* @common:operation
* @jws:conversation phase="continue"
*/
public void create(java.lang.String arg0, java.lang.String arg1) throws Exception
{ customerControl.create(arg0, arg1); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public void setLastName(java.lang.String arg0)
{ customerControl.setLastName(arg0); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public void setCustomer_ID(java.lang.Integer arg0)
{ customerControl.setCustomer_ID(arg0); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String getLastName()
{ return customerControl.getLastName(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * After creating a new customer or finding a customer by last name,
 * you can check the Customer ID (the primary key) for the record.
 * </font>
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.Integer getCustomer_ID()
{ return customerControl.getCustomer_ID(); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String getFirstName()
{ return customerControl.getFirstName(); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public void setFirstName(java.lang.String arg0)
{ customerControl.setFirstName(arg0); }
}
```

beanManagedPersistence Samples

This section contains source code for the following samples.

Samples Included in This Section

BMPItemBeanCtrl.jcx Sample

BMPItemBeanCtrlTest.jws Sample

BMPItemBeanCtrl.jcx Sample

This topic includes the source code for the BMPItemBeanCtrl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/beanManagedPersistence/

Sample Source Code

```
package beanManagedPersistence;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.BMPItemLocalHome"
 */
public interface BMPItemBeanCtrl
    extends beanManagedPersistence.BMPItemHome,    // home interface
            beanManagedPersistence.BMPItem,        // bean interface
            com.bea.control.EntityEJBControl,        // control interface
            com.bea.control.ControlExtension         // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

BMPItemBeanCtrlTest.jws Sample

This topic includes the source code for the BMPItemBeanCtrlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/beanManagedPersistence/

Sample Source Code

```
package beanManagedPersistence;

/**
 */
public class BMPItemBeanCtrlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public beanManagedPersistence.BMPItemBeanCtrl bMPItemBeanCtrl;

    /** @common:operation
     * @jws:conversation phase="start"
     */
    public void startTestDrive() {}

    /** @common:operation
     * @jws:conversation phase="finish"
     */
    public void finishTestDrive() {}

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * Enter a price to find the items matching that price.
     * </font>
     *
     * @common:operation
     * @jws:conversation phase="continue"
     */
    public void findByPrice(double price) throws Exception
    { }

    // The return type of this method was not supported in Web Service operations
    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * Enter the number of a previously entered item. The number is the primary key.
     * </font>
     *
     * @common:operation
     * @jws:conversation phase="continue"
     */
    public void findByPrimaryKey(java.lang.Integer number) throws Exception
    { bMPItemBeanCtrl.findByPrimaryKey(number); }

    // The return type of this method was not supported in Web Service operations
```

navEJBs_ClientApps.html Sample

```
/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * Enter a number, name, and price of an item. The number functions as the unique
 * index, that is, the primary key.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public void create(java.lang.Integer number, java.lang.String name, double price) throws Ex
{ bMPItemBeanCtrl.create(number, name, price); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String getItemName()
{ return bMPItemBeanCtrl.getItemName(); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.Integer getItemNumber()
{ return bMPItemBeanCtrl.getItemNumber(); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public double getPrice()
{ return bMPItemBeanCtrl.getPrice(); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public void setItemName(java.lang.String name)
{ bMPItemBeanCtrl.setItemName(name); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public void setItemNumber(java.lang.Integer number)
{ bMPItemBeanCtrl.setItemNumber(number); }

/** @common:operation
 * @jws:conversation phase="continue"
 */
public void setPrice(double price)
{ bMPItemBeanCtrl.setPrice(price); }
}
```

finderMethods Samples

This section contains source code for the following samples.

Samples Included in This Section

ItemsBeanControl.jcx Sample

ItemsBeanControlTest.jws Sample

ManufacturerBeanControl.jcx Sample

ManufacturerBeanControlTest.jws Sample

ItemsBeanControl.jcx Sample

This topic includes the source code for the ItemsBeanControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/finderMethods/

Sample Source Code

```
package finderMethods;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.ItemsBean_FHome"
 */
public interface ItemsBeanControl
    extends finderMethods.Items_FHome,    // home interface
            finderMethods.Items_F,       // bean interface
            com.bea.control.EntityEJBControl, // control interface
            com.bea.control.ControlExtension // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

ItemsBeanControlTest.jws Sample

This topic includes the source code for the ItemsBeanControlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/finderMethods/

Sample Source Code

```
package finderMethods;

/**
 */
public class ItemsBeanControlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public finderMethods.ItemsBeanControl itemsBeanControl;

    /** @common:operation
     * @jws:conversation phase="start"
     */
    public void startTestDrive() {}

    /** @common:operation
     * @jws:conversation phase="finish"
     */
    public void finishTestDrive() {}

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * findAboveAverage returns the items that are more expensive than the average
     * price for all the items in the database.
     * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
     * </font>
     *
     * @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String findAboveAverage() throws Exception
    { return (itemsBeanControl.findAboveAverage()).toString(); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * findAll returns all the items known in the items database table.
     * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
     * </font>
     *
     * @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String findAll() throws Exception
    { return (itemsBeanControl.findAll()).toString(); }
```


navEJBs_ClientApps.html Sample

```
/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findByItemName takes a String argument and returns all the items with that name.
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findByItemName(java.lang.String itemname) throws Exception
{ return (itemsBeanControl.findByItemName(itemname)).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findByPriceAndQuantity takes a double price and a int quantity as arguments
 * and returns all the items that are more expensive and have larger quantities
 * available in stock.
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findByPriceAndQuantity(double arg0, int arg1) throws Exception
{ return (itemsBeanControl.findByPriceAndQuantity(arg0, arg1)).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findByPrimaryKey takes a integer itemNumber, the primary key, and returns the matching i
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public void findByPrimaryKey(java.lang.Integer arg0) throws Exception
{ itemsBeanControl.findByPrimaryKey(arg0); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findByUSManufacturer returns all the items whose manufacturer is located in the US,
 * querying the entity relation between items and manufacturers.
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findByUSManufacturer() throws Exception
{ return (itemsBeanControl.findByUSManufacturer()).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findByUSManufacturerOrdered works exactly like findByUSManufacturer but
 * uses the WebLogic keyword ORDERBY to order the items returned by itemName.
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
```

navEJBs_ClientApps.html Sample

```
*/
public java.lang.String findByUSManufacturerOrdered() throws Exception
{ return (itemsBeanControl.findByUSManufacturerOrdered()).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findFiveCheapestItems uses a correlated subquery to returns the
 * five least expensive items.
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findFiveCheapestItems() throws Exception
{ return (itemsBeanControl.findFiveCheapestItems()).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findMostExpensiveItems returns the most expensive items, using a
 * correlated subquery and the keyword EXISTS.
 * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findMostExpensiveItems() throws Exception
{ return (itemsBeanControl.findMostExpensiveItems()).toString(); }
}
```

ManufacturerBeanControl.jcx Sample

This topic includes the source code for the ManufacturerBeanControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/finderMethods/

Sample Source Code

```
package finderMethods;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.ManufacturerBean_FHome"
 */
public interface ManufacturerBeanControl
    extends finderMethods.ManufacturerBean_FHome,    // home interface
           finderMethods.ManufacturerBean_FLocal,    // bean interface
           com.bea.control.EntityEJBControl,        // control interface
           com.bea.control.ControlExtension         // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

ManufacturerBeanControlTest.jws Sample

This topic includes the source code for the ManufacturerBeanControlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/finderMethods/

Sample Source Code

```
package finderMethods;

/**
 */
public class ManufacturerBeanControlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public finderMethods.ManufacturerBeanControl manufacturerBeanControl;

    /** @common:operation
     * @jws:conversation phase="start"
     */
    public void startTestDrive() {}

    /** @common:operation
     * @jws:conversation phase="finish"
     */
    public void finishTestDrive() {}

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * findAll returns all the manufacturers stored in the EJB_manufacturer table.
     * Each manufacturer is shown as: [manufacturerID, manufacturerName, USManufacturer]
     * </font>
     *
     * @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String findAll() throws Exception
    { return (manufacturerBeanControl.findAll()).toString(); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * findAllManufacturers returns all the manufacturers that produce the items known
     * in the database, querying the entity relation between items and manufacturers.
     * Duplicate records indicate that manufacturers produce multiple items.
     * Each manufacturer is shown as: [manufacturerID, manufacturerName, USManufacturer]
     * </font>
     *
     * @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String findAllManufacturers() throws Exception
```

navEJBs_ClientApps.html Sample

```
{ return (manufacturerBeanControl.findAllManufacturers()).toString(); }

// The return type of this method was not supported in Web Service operations
/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findByPrimaryKey take an int manufacturerID and returns the matching manufacturer.
 * Each manufacturer is shown as: [manufacturerID, manufacturerName, USManufacturer]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public void findByPrimaryKey(java.lang.Integer manufacturerID) throws Exception
{ manufacturerBeanControl.findByPrimaryKey(manufacturerID); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findDistinctManufacturer returns all the unique manufacturers that produce the items known
 * in the database, querying the entity relation between items and manufacturers.
 * Duplicate records have been filtered out using the DISTINCT keyword.
 * Each manufacturer is shown as: [manufacturerID, manufacturerName, USManufacturer]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findDistinctManufacturer() throws Exception
{ return (manufacturerBeanControl.findDistinctManufacturer()).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findManufacturerOfItem takes an itemname and returns the manufacturer of that
 * item, using the MEMBER OF keyword.
 * Each manufacturer is shown as: [manufacturerID, manufacturerName, USManufacturer]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findManufacturerOfItem(String itemname) throws Exception
{ return (manufacturerBeanControl.findManufacturerOfItem(itemname)).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findUSManufacturer returns all US manufacturers, using a literal value to test
 * whether a manufacturer is located in the US or not.
 * Each item is shown as: [manufacturerID, manufacturerName, USManufacturer]
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String findUSManufacturer() throws Exception
{ return (manufacturerBeanControl.findUSManufacturer()).toString(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * findWithoutItems returns all manufacturers that do not have any known items
 * using the IS EMPTY operator.
 * Each manufacturer is shown as: [manufacturerID, manufacturerName, USManufacturer]
 * </font>
```

navEJBs_ClientApps.html Sample

```
*
* @common:operation
* @jws:conversation phase="continue"
*/
public java.lang.String findWithoutItems() throws Exception
{ return (manufacturerBeanControl.findWithoutItems()).toString(); }
}
```

homeMethods Samples

This section contains source code for the following samples.

Samples Included in This Section

ItemsBeanControl.jcx Sample

ItemsBeanControlTest.jws Sample

ItemsBeanControl.jcx Sample

This topic includes the source code for the ItemsBeanControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/homeMethods/

Sample Source Code

```
package homeMethods;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.ItemsBean_HHome"
 */
public interface ItemsBeanControl
    extends homeMethods.Items_HHome,    // home interface
            homeMethods.Items_H,        // bean interface
            com.bea.control.EntityEJBControl,    // control interface
            com.bea.control.ControlExtension    // control extension marker
{
    static final long serialVersionUID = 1L;
}
```


ItemsBeanControlTest.jws Sample

This topic includes the source code for the ItemsBeanControlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/homeMethods/

Sample Source Code

```
package homeMethods;

/**
 */
public class ItemsBeanControlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public homeMethods.ItemsBeanControl itemsBeanControl;

    /** @common:operation
     * @jws:conversation phase="start"
     */
    public void startTestDrive() {}

    /** @common:operation
     * @jws:conversation phase="finish"
     */
    public void finishTestDrive() {}

    // The return type of this method was not supported in Web Service operations
    /** @common:operation
     * @jws:conversation phase="continue"
     */
    public void findByPrimaryKey(java.lang.Integer arg0) throws Exception
    { itemsBeanControl.findByPrimaryKey(arg0); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * Enter a valid price as a double (e.g., 44.99) to obtain the average
     * price of the items that are more expensive than the entered price,
     * returned as a String. If there are no expensive items, 0.00 is returned.
     * </font>
     */
    /** @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String getAverageExpensiveItems(double arg0)
    { return itemsBeanControl.getAverageExpensiveItems(arg0); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * Enter a valid price as a double (e.g., 44.99) to find out if there are
```

navEJBs_ClientApps.html Sample

```
* any items in stock that are more expensive than the entered price,
* returned as a boolean. If there are no expensive items, false is returned.
* </font>
*
* @common:operation
* @jws:conversation phase="continue"
*/
public boolean getExpensiveItemsInStock(double arg0)
{ return itemsBeanControl.getExpensiveItemsInStock(arg0); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * Enter a valid price as a double (e.g., 44.99) to obtain the number
 * of items that are more expensive than the entered price, returned as an int.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
*/
public int getNumberOfExpensiveItems(double arg0)
{ return itemsBeanControl.getNumberOfExpensiveItems(arg0); }

/** @common:operation
 * @jws:conversation phase="continue"
*/
public java.lang.String getItemname()
{ return itemsBeanControl.getItemname(); }

/** @common:operation
 * @jws:conversation phase="continue"
*/
public java.lang.Integer getItemnumber()
{ return itemsBeanControl.getItemnumber(); }

// The return type of this method was not supported in Web Service operations

/** @common:operation
 * @jws:conversation phase="continue"
*/
public double getPrice()
{ return itemsBeanControl.getPrice(); }

/** @common:operation
 * @jws:conversation phase="continue"
*/
public int getQuantityavailable()
{ return itemsBeanControl.getQuantityavailable(); }

/** @common:operation
 * @jws:conversation phase="continue"
*/
public void setItemname(java.lang.String arg0)
{ itemsBeanControl.setItemname(arg0); }

/** @common:operation
 * @jws:conversation phase="continue"
*/
public void setItemnumber(java.lang.Integer arg0)
{ itemsBeanControl.setItemnumber(arg0); }

// This method includes types that are not supported in Web Service operations
```

navEJBs_ClientApps.html Sample

```
//      /** @common:operation
//      * @jws:conversation phase="continue"
//      */
//      public void setManufacturer(ManufacturerClass.Manufacturer arg0)
//      { itemsBeanControl.setManufacturer(arg0); }

      /** @common:operation
      * @jws:conversation phase="continue"
      */
      public void setPrice(double arg0)
      { itemsBeanControl.setPrice(arg0); }

      /** @common:operation
      * @jws:conversation phase="continue"
      */
      public void setQuantityavailable(int arg0)
      { itemsBeanControl.setQuantityavailable(arg0); }
}
```

messageDriven Samples

This section contains source code for the following samples.

Samples Included in This Section

Controller.jspf Sample

FacadeBeanControl.jcx Sample

Index.jsp Sample

MessageSender.java Sample

MessageSenderImpl.jcs Sample

SendToQueueControl.jcx Sample

SendToTopicControl.jcx Sample

Controller.jspf Sample

This topic includes the source code for the Controller.jspf Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
package messageDriven;
import com.bea.wlw.netui.pageflow.Forward;
import com.bea.wlw.netui.pageflow.PageFlowController;
import java.util.Collection;
import javax.ejb.FinderException;
import javax.jms.JMSException;

/**
 * @jpf:controller
 * @jpf:view-properties view-properties::
 * <!-- This data is auto-generated. Hand-editing this section is not recommended. -->
 * <view-properties>
 * <pageflow-object id="pageflow:/messageDriven/Controller.jspf"/>
 * <pageflow-object id="action:begin.do">
 *   <property value="60" name="x"/>
 *   <property value="240" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:refreshPage.do">
 *   <property value="260" name="x"/>
 *   <property value="80" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:refreshPageOrdered.do">
 *   <property value="80" name="x"/>
 *   <property value="140" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:addViaQueue.do">
 *   <property value="400" name="x"/>
 *   <property value="240" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:addViaTopic.do">
 *   <property value="400" name="x"/>
 *   <property value="120" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:deleteViaQueue.do">
 *   <property value="260" name="x"/>
 *   <property value="280" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action-call:@page:index.jsp#@@action:addViaQueue.do@">
 *   <property value="296,330,330,364" name="elbowsX"/>
 *   <property value="172,172,232,232" name="elbowsY"/>
 *   <property value="East_1" name="fromPort"/>
 *   <property value="West_1" name="toPort"/>
 * </pageflow-object>
 * <pageflow-object id="action-call:@page:index.jsp#@@action:refreshPageOrdered.do@">
 *   <property value="224,170,170,116" name="elbowsX"/>
```

navEJBs_ClientApps.html Sample

```
* <property value="161,161,132,132" name="elbowsY" />
* <property value="West_0" name="fromPort" />
* <property value="East_1" name="toPort" />
* </pageflow-object>
* <pageflow-object id="action-call:@page:index.jsp#@action:addViaTopic.do@">
*   <property value="296,330,330,364" name="elbowsX" />
*   <property value="172,172,112,112" name="elbowsY" />
*   <property value="East_1" name="fromPort" />
*   <property value="West_1" name="toPort" />
* </pageflow-object>
* <pageflow-object id="action-call:@page:index.jsp#@action:deleteViaQueue.do@">
*   <property value="260,260,260,260" name="elbowsX" />
*   <property value="224,230,230,236" name="elbowsY" />
*   <property value="South_1" name="fromPort" />
*   <property value="North_1" name="toPort" />
* </pageflow-object>
* <pageflow-object id="action-call:@page:index.jsp#@action:refreshPage.do@">
*   <property value="260,260,260,260" name="elbowsX" />
*   <property value="136,130,130,124" name="elbowsY" />
*   <property value="North_1" name="fromPort" />
*   <property value="South_1" name="toPort" />
* </pageflow-object>
* <pageflow-object id="page:index.jsp">
*   <property value="260" name="x" />
*   <property value="180" name="y" />
* </pageflow-object>
* <pageflow-object id="forward:path#success#index.jsp#@action:begin.do@">
*   <property value="96,160,160,224" name="elbowsX" />
*   <property value="232,232,172,172" name="elbowsY" />
*   <property value="East_1" name="fromPort" />
*   <property value="West_1" name="toPort" />
*   <property value="success" name="label" />
* </pageflow-object>
* <pageflow-object id="return-to:@forward:return-to#success#currentPage#@action:refreshPage.do@">
*   <property value="180" name="x" />
*   <property value="80" name="y" />
* </pageflow-object>
* <pageflow-object id="forward:return-to#success#currentPage#@action:refreshPage.do@">
*   <property value="224,220,220,216" name="elbowsX" />
*   <property value="72,72,72,72" name="elbowsY" />
*   <property value="West_1" name="fromPort" />
*   <property value="East_1" name="toPort" />
*   <property value="success" name="label" />
* </pageflow-object>
* <pageflow-object id="return-to:@forward:return-to#success#currentPage#@action:refreshPageOrdered.do@">
*   <property value="80" name="x" />
*   <property value="40" name="y" />
* </pageflow-object>
* <pageflow-object id="forward:return-to#success#currentPage#@action:refreshPageOrdered.do@">
*   <property value="80,80,80,80" name="elbowsX" />
*   <property value="96,90,90,84" name="elbowsY" />
*   <property value="North_1" name="fromPort" />
*   <property value="South_1" name="toPort" />
*   <property value="success" name="label" />
* </pageflow-object>
* <pageflow-object id="return-to:@forward:return-to#success#currentPage#@action:addViaQueue.do@">
*   <property value="520" name="x" />
*   <property value="240" name="y" />
* </pageflow-object>
* <pageflow-object id="forward:return-to#success#currentPage#@action:addViaQueue.do@">
*   <property value="436,460,460,484" name="elbowsX" />
```

navEJBs_ClientApps.html Sample

```

*   <property value="232,232,232,232" name="elbowsY" />
*   <property value="East_1" name="fromPort" />
*   <property value="West_1" name="toPort" />
*   <property value="success" name="label" />
* </pageflow-object>
* <pageflow-object id="return-to:@forward:return-to#success#currentPage#@action:addViaTopic.do"
*   <property value="520" name="x" />
*   <property value="120" name="y" />
* </pageflow-object>
* <pageflow-object id="forward:return-to#success#currentPage#@action:addViaTopic.do@">
*   <property value="436,460,460,484" name="elbowsX" />
*   <property value="112,112,112,112" name="elbowsY" />
*   <property value="East_1" name="fromPort" />
*   <property value="West_1" name="toPort" />
*   <property value="success" name="label" />
* </pageflow-object>
* <pageflow-object id="return-to:@forward:return-to#success#currentPage#@action:deleteViaQueue"
*   <property value="140" name="x" />
*   <property value="320" name="y" />
* </pageflow-object>
* <pageflow-object id="forward:return-to#success#currentPage#@action:deleteViaQueue.do@">
*   <property value="224,200,200,176" name="elbowsX" />
*   <property value="272,272,312,312" name="elbowsY" />
*   <property value="West_1" name="fromPort" />
*   <property value="East_1" name="toPort" />
*   <property value="success" name="label" />
* </pageflow-object>
* <pageflow-object id="control:messageDriven.FacadeBeanControl#facadeBeanControl">
*   <property value="53" name="x" />
*   <property value="34" name="y" />
* </pageflow-object>
* <pageflow-object id="control:messageDriven.MessageSender#messageSender">
*   <property value="46" name="x" />
*   <property value="34" name="y" />
* </pageflow-object>
* </view-properties>
* ::
*/
public class Controller extends PageFlowController
{
    /**
     * @common:control
     */
    private messageDriven.FacadeBeanControl facadeBeanControl;

    /**
     * @common:control
     */
    private messageDriven.MessageSender messageSender;

    public transient Collection allTokens;

    // Uncomment this declaration to access Global.app.
    //
    //     protected global.Global globalApp;
    //

    // For an example of page flow exception handling see the example "catch" and "exception-ha
    // annotations in {project}/WEB-INF/src/global/Global.app

```

navEJBs_ClientApps.html Sample

```
/**
 * This method represents the point of entry into the pageflow
 * @jpf:action
 * @jpf:forward name="success" path="index.jsp"
 */
protected Forward begin() throws FinderException
{
    allTokens = facadeBeanControl.findAllTokens();
    return new Forward("success");
}

/**
 * @jpf:action
 * @jpf:forward name="success" return-to="currentPage"
 */
protected Forward refreshPage() throws FinderException
{
    allTokens = facadeBeanControl.findAllTokens();
    return new Forward("success");
}

/**
 * @jpf:action
 * @jpf:forward name="success" return-to="currentPage"
 */
protected Forward refreshPageOrdered() throws FinderException
{
    allTokens = facadeBeanControl.findAllTokensOrdered();
    return new Forward("success");
}

/**
 * @jpf:action
 * @jpf:forward name="success" return-to="currentPage"
 */
protected Forward addViaQueue() throws FinderException, JMSEException
{
    messageSender.add20ViaQueue(allTokens.size());
    return new Forward("success");
}

/**
 * @jpf:action
 * @jpf:forward name="success" return-to="currentPage"
 */
protected Forward addViaTopic() throws FinderException, JMSEException
{
    messageSender.add20ViaTopic(allTokens.size());
    return new Forward("success");
}

/**
 * @jpf:action
 * @jpf:forward name="success" return-to="currentPage"
 */
protected Forward deleteViaQueue() throws FinderException, JMSEException
{
    messageSender.deleteAllViaQueue();
    return new Forward("success");
}
```


}

FacadeBeanControl.jcx Sample

This topic includes the source code for the FacadeBeanControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
package messageDriven;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.FacadeLocalHome_M"
 */
public interface FacadeBeanControl
    extends messageDriven.FacadeLocalHome_M,    // home interface
            messageDriven.FacadeLocal_M,        // bean interface
            com.bea.control.SessionEJBControl,   // control interface
            com.bea.control.ControlExtension     // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

Index.jsp Sample

This topic includes the source code for the Index.jsp Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="netui-tags-databinding.tld" prefix="netui-data"%>
<%@ taglib uri="netui-tags-html.tld" prefix="netui"%>
<%@ taglib uri="netui-tags-template.tld" prefix="netui-template"%>
<netui:html>
  <head>
    <title>
      Message Driven Page
    </title>
  </head>
  <body>
    <p><font size="-1"><em>Add 20 via Queue</em> sends 20 JMS messages to a queue and orders
    the creation of tokens. <em>Add 20 via Topic</em> sends 20 JMS messages to a topic and o
    the creation of tokens. After sending JMS message, press the <em>Refresh (Ordered) </em>
    repeatedly to see the gradual creation of tokens. For more information, see the
    Message-Driven Bean Sample topic in WebLogic help.</font>
    <netui:form action="refreshPage">
      <netui:button value="Refresh" type="submit" action="refreshPage"/>
      <netui:button value="Refresh Ordered" type="submit" action="refreshPageOrdered"/>
      <netui:button value="Add 20 via Queue" type="submit" action="addViaQueue"/>
      <netui:button value="Add 20 via Topic" type="submit" action="addViaTopic"/>
      <netui:button value="Delete All via Queue" type="submit" action="deleteViaQueue"/>
    </netui:form>
    </p>
    <netui-data:repeater dataSource="{pageFlow.allTokens}">
      <netui-data:repeaterHeader>
        <p>The following tokens are currently known:</p>
        <p><table border="1" cellpadding="4">
          <tr>
            <td><b>Name</b></td>
            <td><b>Date Entered</b></td>
            <td><b>Entered By Class</b></td>
            <td><b>Object Hash</b></td>
          </tr>
        </netui-data:repeaterHeader>
        <netui-data:repeaterItem>
          <tr>
            <td>
              <netui:label value="{container.item.name}" />
            </td>
            <td>
              <netui:label value="{container.item.date}" />
            </td>
            <td>
              <netui:label value="{container.item.enteredByClass}" />
            </td>
            <td>
              <netui:label value="{container.item.objectHash}" />
            </td>
          </tr>
        </netui-data:repeaterItem>
      </netui-data:repeater>
    </p>
  </body>
</netui:html>
```

navEJBs_ClientApps.html Sample

```
</td>
<td>
    <netui:label value="{container.item.objectHash}" />
</td>
</tr>
</netui-data:repeaterItem>
<netui-data:repeaterFooter>
    </table>
    <p><font size="-1"><em>Name</em> is the name of the token, and reflects
    the order of the JMS messages as they were sent. <em>Entered By Class</em>
    shows which message-driven bean class received the message, while <em>Object
    Hash</em> shows the actual object which received the message.</font></p>
</netui-data:repeaterFooter>
</netui-data:repeater>
</body>
</netui:html>
```

MessageSender.java Sample

This topic includes the source code for the MessageSender.java Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
package messageDriven;

import com.bea.control.Control;

public interface MessageSender extends Control
{
    void deleteAllViaQueue() throws javax.jms.JMSEException;

    void add20ViaTopic(int currentNum) throws javax.jms.JMSEException;

    void add20ViaQueue(int currentNum) throws javax.jms.JMSEException;
}
```

MessageSenderImpl.jcs Sample

This topic includes the source code for the MessageSenderImpl.jcs Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
package messageDriven;

import com.bea.control.*;
import javax.jms.JMSEException;
import javax.jms.MapMessage;
/**
 * @editor-info:code-gen control-interface="true"
 */
public class MessageSenderImpl implements MessageSender, ControlSource
{
    /**
     * @common:control
     */
    private messageDriven.sendToQueueControl queueSend;

    /**
     * @common:control
     */
    private messageDriven.sendToTopicControl topicSend;

    static final long serialVersionUID = 1L;

    /**
     * @common:operation
     * @common:message-buffer enable="true"
     */
    public void add20ViaQueue(int currentNum) throws JMSEException
    {
        String name;
        for(int i = 0; i < 20; i++) {
            MapMessage msg = queueSend.getSession().createMapMessage();
            msg.setStringProperty("Command", "Add");
            name = Integer.toString(currentNum + i);
            msg.setString("tokenName", name);
            queueSend.sendJMSMessage(msg);
        }
    }

    /**
     * @common:operation
     * @common:message-buffer enable="true"
     */
    public void add20ViaTopic(int currentNum) throws JMSEException
    {
        String name;
```

navEJBs_ClientApps.html Sample

```
        for(int i = 0; i < 20; i++) {
            name = Integer.toString(currentNum + i);
            topicSend.sendMessage(name);
        }
    }

    /**
     * @common:operation
     * @common:message-buffer enable="true"
     */
    public void deleteAllViaQueue() throws JMSEException
    {
        MapMessage msg = queueSend.getSession().createMapMessage();
        msg.setStringProperty("Command", "Delete");
        queueSend.sendMessage(msg);
    }
}
```

SendToQueueControl.jcx Sample

This topic includes the source code for the SendToQueueControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
package messageDriven;

import com.bea.control.*;
import java.io.Serializable;
import javax.jms.Message;

/**
 * @jc:jms send-type="queue" send-jndi-name="jms.SamplesAppMDBQ"
 *          connection-factory-jndi-name="weblogic.jws.jms.QueueConnectionFactory"
 */
public interface sendToQueueControl extends JMSControl, com.bea.control.ControlExtension
{
    /**
     * this method will send a javax.jms.Message to send-jndi-name
     */
    public void sendJMSMessage(Message msg);
    static final long serialVersionUID = 1L;
}
```


SendToTopicControl.jcx Sample

This topic includes the source code for the SendToTopicControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/messageDriven/

Sample Source Code

```
package messageDriven;

import com.bea.control.*;
import java.io.Serializable;

/**
 * @jc:jms send-type="topic" send-jndi-name="jms.SamplesAppMDBTopic"
 * connection-factory-jndi-name="weblogic.jws.jms.QueueConnectionFactory"
 */
public interface sendToTopicControl extends JMSControl, com.bea.control.ControlExtension
{
    /**
     * this method will send a javax.jms.TextMessage to send-jndi-name
     */
    public void sendTextMessage(String payload);
    static final long serialVersionUID = 1L;
}
```

security Samples

This section contains source code for the following samples.

Samples Included in This Section

RoleCheckerControl.jcx Sample

RoleCheckerControlTest.jws Sample

RoleCheckerControl.jcx Sample

This topic includes the source code for the RoleCheckerControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/security/

Sample Source Code

```
package security;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.RoleCheckerLocalHome"
 */
public interface RoleCheckerControl
    extends security.RoleCheckerLocalHome,    // home interface
           security.RoleCheckerLocal,        // bean interface
           com.bea.control.SessionEJBControl, // control interface
           com.bea.control.ControlExtension   // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

RoleCheckerControlTest.jws Sample

This topic includes the source code for the RoleCheckerControlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/security/

Sample Source Code

```
package security;

/**
 * @common:security run-as="manager"
 */
public class RoleCheckerControlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public security.RoleCheckerControl roleCheckerControl;

    // The return type of this method was not supported in Web Service operations
    /** @common:operation
     */
    public void create() throws Exception
    { roleCheckerControl.create(); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * The <em>ceoOnly</em> method can only be invoked by the ceo security role. Because the we
     * runs under the manager role, a <em>javax.ejb.AccessLocalException</em> security violatio
     * exception will be thown.
     * </font>
     */
    /** @common:operation
     */
    public java.lang.String ceoOnly()
    { return roleCheckerControl.ceoOnly(); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * The <em>discoverCallerRole</em> method checks the security role of the invoking client,
     * determines that the web service runs under the manager role.
     * </font>
     */
    /** @common:operation
     */
    public java.lang.String discoverCallerRole()
    { return roleCheckerControl.discoverCallerRole(); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * The <em>discoverRoleCheckersRole</em> method checks the security role of the RoleChecker
     * The RoleCheckerBean runs under the engineer security role.
     */
}
```

navEJBs_ClientApps.html Sample

```
* </font>
*
* @common:operation
*/
public java.lang.String discoverRoleCheckersRole()
{ return roleCheckerControl.discoverRoleCheckersRole(); }

/**
* <font face="Geneva, Arial, Helvetica, sans-serif">
* The <em>managerOnly</em> method can only be invoked by the manager security role. Because
* runs under the manager role, the method executes normally.
* </font>
*
* @common:operation
*/
public java.lang.String managerOnly()
{ return roleCheckerControl.managerOnly(); }
}
```

selectMethods Samples

This section contains source code for the following samples.

Samples Included in This Section

ItemsBeanControl.jcx Sample

ItemsBeanControlTest.jws Sample

ItemsBeanControl.jcx Sample

This topic includes the source code for the ItemsBeanControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/selectMethods/

Sample Source Code

```
package selectMethods;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.ItemsBean_SHome"
 */
public interface ItemsBeanControl
    extends selectMethods.Items_SHome,    // home interface
            selectMethods.Items_S,        // bean interface
            com.bea.control.EntityEJBControl,    // control interface
            com.bea.control.ControlExtension    // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

ItemsBeanControlTest.jws Sample

This topic includes the source code for the ItemsBeanControlTest.jws Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/selectMethods/

Sample Source Code

```
package selectMethods;

/**
 */
public class ItemsBeanControlTest implements com.bea.jws.WebService
{
    static final long serialVersionUID = 1L;

    /** @common:control */
    public selectMethods.ItemsBeanControl itemsBeanControl;

    /** @common:operation
     * @jws:conversation phase="start"
     */
    public void startTestDrive() {}

    /** @common:operation
     * @jws:conversation phase="finish"
     */
    public void finishTestDrive() {}

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * The home method selectAll uses ejbSelectAll to return all the items known in the database.
     * Each item is shown as: [itemNumber, itemName, price, quantityAvailable]
     * </font>
     */
    /** @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String selectAll()
    { return itemsBeanControl.selectAll(); }

    /**
     * <font face="Geneva, Arial, Helvetica, sans-serif">
     * The home method selectByPrimaryKey take an integer itemnumber, the primary key, and uses it
     * to return the name of the item matching that value.
     * </font>
     */
    /** @common:operation
     * @jws:conversation phase="continue"
     */
    public java.lang.String selectByPrimaryKey(int itemnumber)
    { return itemsBeanControl.selectByPrimaryKey(itemnumber); }
```


navEJBs_ClientApps.html Sample

```
/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * The home method selectByUSManufacturer uses ejbSelectByUSManufacturer
 * to return all the item names of the items whose manufacturer is located in the US,
 * querying the entity relation between items and manufacturers.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String selectByUSManufacturer()
{ return itemsBeanControl.selectByUSManufacturer(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * The home method selectByItemName uses ejbSelectByItemName
 * to return the names of all the items known in the database.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String selectItemNames()
{ return itemsBeanControl.selectItemNames(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * The home method selectByItemNameOrdered uses ejbSelectByItemNameOrdered
 * to return the names of all the items known in the database,
 * ordered by price in descending order (starting with the most expensive items)
 * and quantity available in ascending order,
 * using the WebLogic keyword ORDERBY.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String selectItemNamesOrdered()
{ return itemsBeanControl.selectItemNamesOrdered(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * The home method selectManufacturerPriceN uses ejbSelectManufacturerPriceN
 * to return the names of all manufacturers, the number of items offered
 * offered by a manufacturer, and the average price of these items. These results are
 * ordered by manufacturer name.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String selectManufacturerPriceN()
{ return itemsBeanControl.selectManufacturerPriceN(); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * The home method selectNameByPrice take a double price and uses ejbSelectNameByPrice
 * to return the names of all the more expensive items.
 * </font>
 *

```

navEJBs_ClientApps.html Sample

```
* @common:operation
* @jws:conversation phase="continue"
*/
public java.lang.String selectNameByPrice(double arg0)
{ return itemsBeanControl.selectNameByPrice(arg0); }

/**
 * <font face="Geneva, Arial, Helvetica, sans-serif">
 * The home method selectNameQuantity uses ejbSelectNameQuantity
 * to return the names and quantity in stock of all known items.
 * </font>
 *
 * @common:operation
 * @jws:conversation phase="continue"
 */
public java.lang.String selectNameQuantity()
{ return itemsBeanControl.selectNameQuantity(); }
}
```

valueObject Samples

This section contains source code for the following samples.

Samples Included in This Section

AdminBeanControl.jcx Sample

Controller.jspf Sample

Index.jsp Sample

ItemsManufacturer.jsp Sample

NewManufacturer.jsp Sample

AdminBeanControl.jcx Sample

This topic includes the source code for the AdminBeanControl.jcx Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/valueObject/

Sample Source Code

```
package valueObject;

import com.bea.control.*;

/**
 * @jc:ejb home-jndi-name="ejb.AdminRemoteHome"
 */
public interface AdminBeanControl
    extends valueObject.AdminHome,    // home interface
            valueObject.Admin,        // bean interface
            com.bea.control.SessionEJBControl,    // control interface
            com.bea.control.ControlExtension    // control extension marker
{
    static final long serialVersionUID = 1L;
}
```

Controller.jspf Sample

This topic includes the source code for the Controller.jspf Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/valueObject/

Sample Source Code

```
package valueObject;
import com.bea.wlw.netui.pageflow.FormData;
import com.bea.wlw.netui.pageflow.Forward;
import com.bea.wlw.netui.pageflow.PageFlowController;
import finderMethods.ManufacturerVO;
import java.rmi.RemoteException;
import java.util.ArrayList;
import java.util.Collection;

/**
 * @jpf:controller
 * @jpf:view-properties view-properties::
 * <!-- This data is auto-generated. Hand-editing this section is not recommended. -->
 * <view-properties>
 * <pageflow-object id="pageflow:/valueObject/Controller.jspf"/>
 * <pageflow-object id="action:begin.do">
 *   <property value="100" name="x"/>
 *   <property value="280" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:toNewManufacturer.do">
 *   <property value="360" name="x"/>
 *   <property value="280" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:addNewManufacturer.do#valueObject.Controller.AddNewManufacturerF
 *   <property value="220" name="x"/>
 *   <property value="420" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:showItemsManufacturer.do">
 *   <property value="360" name="x"/>
 *   <property value="200" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action:backToIndex.do">
 *   <property value="220" name="x"/>
 *   <property value="60" name="y"/>
 * </pageflow-object>
 * <pageflow-object id="action-call:@page:index.jsp#@@action:toNewManufacturer.do@">
 *   <property value="256,290,290,324" name="elbowsX"/>
 *   <property value="252,252,272,272" name="elbowsY"/>
 *   <property value="East_1" name="fromPort"/>
 *   <property value="West_1" name="toPort"/>
 * </pageflow-object>
 * <pageflow-object id="action-call:@page:index.jsp#@@action:showItemsManufacturer.do@">
 *   <property value="256,290,290,324" name="elbowsX"/>
 *   <property value="252,252,192,192" name="elbowsY"/>
 *   <property value="East_1" name="fromPort"/>

```

navEJBs_ClientApps.html Sample

```
* <property value="West_1" name="toPort"/>
* </pageflow-object>
* <pageflow-object id="page:index.jsp">
* <property value="220" name="x"/>
* <property value="260" name="y"/>
* </pageflow-object>
* <pageflow-object id="action-call:@page:newManufacturer.jsp#@action:addNewManufacturer.do#valueObject">
* <property value="324,290,290,256" name="elbowsX"/>
* <property value="412,412,412,412" name="elbowsY"/>
* <property value="West_1" name="fromPort"/>
* <property value="East_1" name="toPort"/>
* </pageflow-object>
* <pageflow-object id="page:newManufacturer.jsp">
* <property value="360" name="x"/>
* <property value="420" name="y"/>
* </pageflow-object>
* <pageflow-object id="action-call:@page:itemsManufacturer.jsp#@action:backToIndex.do@">
* <property value="324,290,290,256" name="elbowsX"/>
* <property value="52,52,52,52" name="elbowsY"/>
* <property value="West_1" name="fromPort"/>
* <property value="East_1" name="toPort"/>
* </pageflow-object>
* <pageflow-object id="page:itemsManufacturer.jsp">
* <property value="360" name="x"/>
* <property value="60" name="y"/>
* </pageflow-object>
* <pageflow-object id="forward:path#success#index.jsp#@action:begin.do@">
* <property value="136,160,160,184" name="elbowsX"/>
* <property value="272,272,252,252" name="elbowsY"/>
* <property value="East_1" name="fromPort"/>
* <property value="West_1" name="toPort"/>
* <property value="success" name="label"/>
* </pageflow-object>
* <pageflow-object id="forward:path#success#newManufacturer.jsp#@action:toNewManufacturer.do@">
* <property value="360,360,360,360" name="elbowsX"/>
* <property value="324,350,350,376" name="elbowsY"/>
* <property value="South_1" name="fromPort"/>
* <property value="North_1" name="toPort"/>
* <property value="success" name="label"/>
* </pageflow-object>
* <pageflow-object id="forward:path#success#index.jsp#@action:addNewManufacturer.do#valueObject">
* <property value="220,220,220,220" name="elbowsX"/>
* <property value="376,340,340,304" name="elbowsY"/>
* <property value="North_1" name="fromPort"/>
* <property value="South_1" name="toPort"/>
* <property value="success" name="label"/>
* </pageflow-object>
* <pageflow-object id="forward:path#success#itemsManufacturer.jsp#@action:showItemsManufacture">
* <property value="360,360,360,360" name="elbowsX"/>
* <property value="156,130,130,104" name="elbowsY"/>
* <property value="North_1" name="fromPort"/>
* <property value="South_1" name="toPort"/>
* <property value="success" name="label"/>
* </pageflow-object>
* <pageflow-object id="forward:path#success#index.jsp#@action:backToIndex.do@">
* <property value="220,220,220,220" name="elbowsX"/>
* <property value="104,160,160,216" name="elbowsY"/>
* <property value="South_1" name="fromPort"/>
* <property value="North_1" name="toPort"/>
* <property value="success" name="label"/>
* </pageflow-object>
```

navEJBs_ClientApps.html Sample

```

* <pageflow-object id="control:valueObject.AdminBeanControl#adminBeanControl">
*   <property value="52" name="x"/>
*   <property value="34" name="y"/>
* </pageflow-object>
* <pageflow-object id="formbeanprop:valueObject.Controller.AddNewManufacturerForm#name#java.la
* <pageflow-object id="formbeanprop:valueObject.Controller.AddNewManufacturerForm#usManufactur
* <pageflow-object id="formbeanprop:valueObject.Controller.AddNewManufacturerForm#listnumber#j
* <pageflow-object id="formbean:valueObject.Controller.AddNewManufacturerForm"/>
* </view-properties>
* ::
*/
public class Controller extends PageFlowController
{
    /**
     * @common:control
     */
    private valueObject.AdminBeanControl adminBeanControl;

    public transient Collection manufacturers;
    public transient ManufacturerVO mVO;

    // Uncomment this declaration to access Global.app.
    //
    //     protected global.Global globalApp;
    //

    // For an example of page flow exception handling see the example "catch" and "exception-ha
    // annotations in {project}/WEB-INF/src/global/Global.app

    /**
     * This method represents the point of entry into the pageflow
     * @jpf:action
     * @jpf:forward name="success" path="index.jsp"
     */
    protected Forward begin() throws RemoteException
    {
        manufacturers = adminBeanControl.getManufacturersVO();
        return new Forward("success");
    }

    /**
     * @jpf:action
     * @jpf:forward name="success" path="newManufacturer.jsp"
     */
    protected Forward toNewManufacturer()
    {
        return new Forward("success");
    }

    /**
     * @jpf:action
     * @jpf:forward name="success" path="index.jsp"
     */
    protected Forward addNewManufacturer(AddNewManufacturerForm form) throws RemoteException
    {
        mVO = new ManufacturerVO(null,form.getListnumber(), form.getName(),form.getUsManufactur
        adminBeanControl.addManufacturerVO(mVO);
        manufacturers.add(mVO);
        return new Forward("success");
    }
}

```

navEJBs_ClientApps.html Sample

```
/**
 * @jpf:action
 * @jpf:forward name="success" path="itemsManufacturer.jsp"
 */
protected Forward showItemsManufacturer() throws RemoteException
{
    Integer PK = new Integer(getRequest().getParameter("manufacturerPK"));
    mVO = adminBeanControl.getManufacturer(PK);
    return new Forward("success");
}

/**
 * @jpf:action
 * @jpf:forward name="success" path="index.jsp"
 */
protected Forward backToIndex()
{
    return new Forward("success");
}

public String getBooleanValue(int booleanInt) {
    if(booleanInt == 0)
        return "No";
    else
        return "Yes";
}

/**
 * FormData get and set methods may be overwritten by the Form Bean editor.
 */
public static class AddNewManufacturerForm extends FormData
{
    private Integer listnumber;

    private Integer usManufacturer;

    private String name;

    public void setName(String name)
    {
        this.name = name;
    }

    public String getName()
    {
        return this.name;
    }

    public void setUsManufacturer(Integer usManufacturer)
    {
        this.usManufacturer = usManufacturer;
    }

    public Integer getUsManufacturer()
    {
        return this.usManufacturer;
    }

    public void setListnumber(Integer listnumber)
    {
        this.listnumber = listnumber;
    }
}
```


navEJBs_ClientApps.html Sample

```
public Integer getListnumber()  
{  
    return this.listnumber;  
}  
}
```

Index.jsp Sample

This topic includes the source code for the Index.jsp Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/valueObject/

Sample Source Code

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="netui-tags-databinding.tld" prefix="netui-data"%>
<%@ taglib uri="netui-tags-html.tld" prefix="netui"%>
<%@ taglib uri="netui-tags-template.tld" prefix="netui-template"%>
<netui:html>
  <head>
    <title>
      List of Items
    </title>
  </head>
  <body>
    <netui-data:repeater dataSource="{pageFlow.manufacturers}">
      <netui-data:repeaterHeader>
        <p>The following manufacturers are currently known:</p>
        <p><table border="1" cellpadding="4">
          <tr>
            <td><b>List Number</b></td>
            <td><b>Name</b></td>
            <td><b>US Manufacturer</b></td>
            <td><b>Link To Items</b></td>
          </tr>
        </netui-data:repeaterHeader>
        <netui-data:repeaterItem>
          <netui-data:choiceMethod object="{pageFlow}" method="getBooleanValue">
            <netui-data:methodParameter value="{container.item.usManufacturer}"/>
          </netui-data:choiceMethod>
          <netui-data:choice value="Yes">
            <tr>
              <td>
                <netui:label value="{container.item.manufacturerId}" />
              </td>
              <td>
                <netui:label value="{container.item.manufacturername}" />
              </td>
              <td>Yes</td>
              <td>
                <netui:anchor action="showItemsManufacturer">
                  Show Items
                  <netui:parameter name="manufacturerPK" value="{container.item.manufacturerId}" />
                </netui:anchor>
              </td>
            </tr>
          </netui-data:choice>
          <netui-data:choice value="No">
            <tr>
```

navEJBs_ClientApps.html Sample

```
<td>
    <netui:label value="{container.item.manufacturerId}" />
</td>
<td>
    <netui:label value="{container.item.manufacturername}" />
</td>
<td>No</td>
<td>
    <netui:anchor action="showItemsManufacturer">
        Show Items
        <netui:parameter name="manufacturerPK" value="{container.item.manufacturer
    </netui:anchor>
</td>
</tr>
</netui-data:choice>
</netui-data:repeaterItem>
<netui-data:repeaterFooter>
    <table>
</netui-data:repeaterFooter>
</netui-data:repeater>
<p>
<netui:form action="toNewManufacturer">
    <netui:button value="New Manufacturer" type="submit"/>
</netui:form>
</p>

<p><font size="-1">This table is created on the basis of a Collection of
'ManufacturerValue' value objects holding manufacturer information. This collection is
returned by the AdminBean's getManufacturersVO method. The value objects do not
contain information about the items created by a manufacturer, as this information
is not needed in this table.</font></p>
</body>
</netui:html>
```

ItemsManufacturer.jsp Sample

This topic includes the source code for the ItemsManufacturer.jsp Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/valueObject/

Sample Source Code

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="netui-tags-databinding.tld" prefix="netui-data"%>
<%@ taglib uri="netui-tags-html.tld" prefix="netui"%>
<%@ taglib uri="netui-tags-template.tld" prefix="netui-template"%>
<netui:html>
  <head>
    <title>
      Web Application Page
    </title>
  </head>
  <body>
    <p>The following items are currently known for <netui:content value="{pageFlow.mVO.manu
    <netui-data:repeater dataSource="{pageFlow.mVO.items}">
      <netui-data:repeaterHeader>
        <p><table border="1" cellpadding="4">
          <tr>
            <td><b>List Number</b></td>
            <td><b>Name</b></td>
            <td><b>Price</b></td>
            <td><b>Quantity Available</b></td>
          </tr>
        </netui-data:repeaterHeader>
        <netui-data:repeaterItem>
          <tr>
            <td>
              <netui:label value="{container.item.itemnumber}" />
            </td>
            <td>
              <netui:label value="{container.item.itemname}" />
            </td>
            <td>
              <netui:label value="{container.item.price}" />
            </td>
            <td>
              <netui:label value="{container.item.quantityavailable}" />
            </td>
          </tr>
        </netui-data:repeaterItem>
        <netui-data:repeaterFooter>
          </table>
        </netui-data:repeaterFooter>
      </netui-data:repeater>
      <p>
        <netui:anchor action="backToIndex">
          To Index Page
```

navEJBs_ClientApps.html Sample

```
</netui:anchor>
<p><font size="-1">This page is created on the basis of one
'ManufacturerValue' value object holding all manufacturer information. This value object is
returned by the AdminBean's getManufacturer method. The value object contains a collection
of ItemsValue value objects, storing information about the items created by the
manufacturer.</font></p>
</body>
</netui:html>
```

NewManufacturer.jsp Sample

This topic includes the source code for the NewManufacturer.jsp Sample.

Sample Location

This sample is located in the following directory in your WebLogic Workshop installation:

BEA_HOME/weblogic81/samples/workshop/SamplesApp/EJBs_ClientApps/valueObject/

Sample Source Code

```
<%@ page language="java" contentType="text/html; charset=UTF-8"%>
<%@ taglib uri="netui-tags-databinding.tld" prefix="netui-data"%>
<%@ taglib uri="netui-tags-html.tld" prefix="netui"%>
<%@ taglib uri="netui-tags-template.tld" prefix="netui-template"%>
<netui:html>
  <head>
    <title>
      Web Application Page
    </title>
  </head>
  <body>
    <p>Please enter a new manufacturer:</p>
    <netui:form action="addNewManufacturer">
      <table>
        <tr valign="top">
          <td>List Number:</td>
          <td>
            <netui:textBox dataSource="{actionForm.listnumber}"/>
          </td>
        </tr>
        <tr valign="top">
          <td>Name:</td>
          <td>
            <netui:textBox dataSource="{actionForm.name}"/>
          </td>
        </tr>
        <tr valign="top">
          <td>US Manufacturer:</td>
          <td>
            <netui:select defaultValue="1" dataSource="{actionForm.usManufacturer}">
              <netui:selectOption value="0">No</netui:selectOption>
              <netui:selectOption value="1">Yes</netui:selectOption>
            </netui:select>
          </td>
        </tr>
      </table>
      <br/>&nbsp;
      <netui:button value="Add" type="submit"/>
    </netui:form>
    <p><font size="-1">After you enter the information and click the 'Add' button,
    the entered information is stored in a 'ManufacturerVO' value object and transmitted
    to the AdminBean EJB. The AdminBean's addManufacturer method in turn invokes the
    Manufacturer EJB create method and attempts to store the new manufacturer information.
    This attempt will fail if the entered list number (which is the Manufacturer EJB's
    primary key) is not unique.</font></p>
  </body>
</netui:html>
```

navEJBs_ClientApps.html Sample

```
</body>  
</netui:html>
```