



BEA WebLogic Workshop™ Help

Version 8.1 SP2
November 2003

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

ExtensionDevKit Samples.....	1
ControlDevKit Samples.....	2
ControlFeatures Samples.....	3
controlEvents Samples.....	4
EventRaiser.java Sample.....	5
EventRaiserImpl.jcs Sample.....	6
EventScheduler.java Sample.....	8
EventSchedulerImpl.jcs Sample.....	9
LifeCycle.java Sample.....	11
LifeCycleImpl.jcs Sample.....	12
insertWizard Samples.....	15
ide Samples.....	16
ServerCheckWizard.java Sample.....	17
ServerCheckWizardPanel.java Sample.....	20
MBeanUtil.java Sample.....	26
ServerCheck-tags.xml Sample.....	27
ServerCheck.java Sample.....	28
ServerCheckImpl.jcs Sample.....	29
insertWizardCustom Samples.....	31
CustomWiz.java Sample.....	32
CustomWizImpl.jcs Sample.....	33
ide Samples.....	34
CustomInsertDialog.java Sample.....	35

Table of Contents

CustomInsertWizard.java Sample.....	49
html Samples.....	54
Introduction.html Sample.....	55
jcxCreate Samples.....	57
ide Samples.....	58
QueryExpressionEditorSimple.java Sample.....	59
XQueryEditorSupport.java Sample.....	61
XQueryWizard.java Sample.....	63
testXML Samples.....	66
Po.xml Sample.....	67
XQuery-tags.xml Sample.....	68
XQuery.java Sample.....	69
XQueryImpl.jcs Sample.....	70
XQueryUtil.java Sample.....	73
propEditor Samples.....	74
CustomerData-tags.xml Sample.....	75
CustomerData.java Sample.....	76
CustomerDataImpl.jcs Sample.....	77
CustomerDBControl.jcx Sample.....	79
ide Samples.....	80
CustIdValidator.java Sample.....	81
CustomerDataEditorSupport.java Sample.....	83
CustomerIdEditorSimple.java Sample.....	84

Table of Contents

ControlTest Samples.....	86
featuresTests Samples.....	87
CustomerDataTest.jws Sample.....	88
EventRaiserTest.jws Sample.....	90
EventSchedulerTest.jws Sample.....	92
JcxTest.jws Sample.....	94
POXQueryControl.jcx Sample.....	95
ServerCheckTest.jws Sample.....	96
DBScripter Samples.....	97
dbScript Samples.....	98
DBScriptRunner-tags.xml Sample.....	99
DBScriptRunner.java Sample.....	101
DBScriptRunnerImpl.jcs Sample.....	102
designUI Samples.....	108
InsertWizCreateJCX.java Sample.....	109
JndiDsPanel.java Sample.....	112
scripts Samples.....	122
Oracle Samples.....	123
PointBase Samples.....	124
SQLServer Samples.....	125
util Samples.....	126
Constants.java Sample.....	127
DsUtils.java Sample.....	128

Table of Contents

docs Samples.....	136
help Samples.....	137
doc Samples.....	138
en Samples.....	139
partners Samples.....	140
DBScripter Samples.....	141
ConDBScriptRunnerControlWizard.html Sample.....	142
java-class Samples.....	143
dbScript Samples.....	144
DBScriptRunner.Callback.html Sample.....	145
DBScriptRunner.html Sample.....	149
DBScriptRunnerImpl.html Sample.....	153
Package-summary.html Sample.....	159
util Samples.....	162
Constants.html Sample.....	163
DsUtils.html Sample.....	173
Package-summary.html Sample.....	181
javadoc-tag Samples.....	184
jc Samples.....	185
Dbscr-connection.html Sample.....	186
Dbscript.html Sample.....	188
NavDBScriptRunner.html Sample.....	191
OvwDBScriptRunner.html Sample.....	193

Table of Contents

Toc.xml Sample.....	195
samples Samples.....	196
partners Samples.....	197
DBScripter Samples.....	198
DBScripterSample Samples.....	199
ScriptTest Samples.....	200
test Samples.....	201
DBScriptTest.jws Sample.....	202
DsPointBaseTest.jcx Sample.....	204
IdeDevKit Samples.....	205
CustomProject Samples.....	206
IdeExtensions Samples.....	207
CustomProject Samples.....	208
php_template Samples.....	209
basic Samples.....	210
scripts Samples.....	211
style Samples.....	212
Template.xml Sample.....	213
src Samples.....	214
ideExtensions Samples.....	215
customProject Samples.....	216
PhpDocument.java Sample.....	217
PhpDocumentImpl.java Sample.....	219

Table of Contents

PhpProjectBuildDriver.java Sample.....	222
PhpProjectDriver.java Sample.....	224
PhpProjectListener.java Sample.....	226
PhpProjectRunDriver.java Sample.....	229
PhpProjectRunPreferences.java Sample.....	234
xml Samples.....	236
DefaultPhpBuild.xml Sample.....	237
DragDropSimple Samples.....	238
src Samples.....	239
ideExtensions Samples.....	240
dragDropSimple Samples.....	241
SimpleTree.java Sample.....	242
SimpleTreeDragDropDriver.java Sample.....	245
TreePanelView.java Sample.....	248
FrameViewSimple Samples.....	250
IdeExtensions Samples.....	251
FrameViewSimple Samples.....	252
src Samples.....	253
ideExtensions Samples.....	254
frameViewSimple Samples.....	255
FrameView.java Sample.....	256
MenuItems Samples.....	258
IdeExtensions Samples.....	259

Table of Contents

MenuItems Samples.....	260
src Samples.....	261
ideExtensions Samples.....	262
menuItems Samples.....	263
AddFavoriteAction.java Sample.....	264
DeleteFavoriteAction.java Sample.....	266
FavoritesGenerator.java Sample.....	268
LaunchBrowserAction.java Sample.....	270
LaunchBrowserAction1.java Sample.....	271
LaunchBrowserAction2.java Sample.....	272
LaunchBrowserAction3.java Sample.....	273
PopupAction Samples.....	274
IdeExtensions Samples.....	275
PopupAction Samples.....	276
lib Samples.....	277
src Samples.....	278
ideExtensions Samples.....	279
popupAction Samples.....	280
FTPPrefsPanel.java Sample.....	281
FTPSettings.java Sample.....	288
UploadAction.java Sample.....	289
strings Samples.....	302
IdeExtensions Samples.....	303

Table of Contents

PopupAction Samples.....	304
PropertyListener Samples.....	305
IdeExtensions Samples.....	306
PropertyListener Samples.....	307
src Samples.....	308
ideExtensions Samples.....	309
propertyListener Samples.....	310
PropertyListenerAction.java Sample.....	311
SearchResults Samples.....	314
IdeExtensions Samples.....	315
SearchResults Samples.....	316
TocView Samples.....	317
src Samples.....	318
tocView Samples.....	319
ToolbarButton Samples.....	320
IdeExtensions Samples.....	321
ToolbarButton Samples.....	322
src Samples.....	323
ideExtensions Samples.....	324
toolbarButton Samples.....	325
LaunchBrowser.java Sample.....	326
TaglibExtDevKit Samples.....	327
TagHandlers Samples.....	328

Table of Contents

TDK Samples.....	329
Barcode.java Sample.....	330
TaglibInstall Samples.....	332
TDK Samples.....	333
TemplateProcessor.java Sample.....	334
template Samples.....	336
Template.xml Sample.....	337
TaglibWebProject Samples.....	338
Controller.jspf Sample.....	339
Demo.jsp Sample.....	340
Error.jsp Sample.....	341
Index.jsp Sample.....	342
TldxHandlers Samples.....	343
help Samples.....	344
en Samples.....	345
tldx Samples.....	346
Index.html Sample.....	347
Overview.html Sample.....	348
Reference.html Sample.....	349
taglib Samples.....	350
www.bea.com Samples.....	351
tdk Samples.....	352
tdk-tags-1.0 Samples.....	353

Table of Contents

Barcode.html Sample.....	354
Toc.xml Sample.....	355
TDK Samples.....	356
BarcodeEncodings.java Sample.....	357
PaletteGenerator.java Sample.....	372
Renderer.java Sample.....	375

ExtensionDevKit Samples

This section contains source code for the following samples.

Samples Included in This Section

ControlDevKit Samples

IdeDevKit Samples

TaglibExtDevKit Samples

ControlDevKit Samples

This section contains source code for the following samples.

Samples Included in This Section

ControlFeatures Samples

ControlTest Samples

DBScripter Samples

ControlFeatures Samples

This section contains source code for the following samples.

Samples Included in This Section

controlEvents Samples

insertWizard Samples

insertWizardCustom Samples

jcxCreate Samples

propEditor Samples

controlEvents Samples

This section contains source code for the following samples.

Samples Included in This Section

EventRaiser.java Sample

EventRaiserImpl.jcs Sample

EventScheduler.java Sample

EventSchedulerImpl.jcs Sample

LifeCycle.java Sample

LifeCycleImpl.jcs Sample

EventRaiser.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/controlEvents/

Sample Source Code

```
package controlEvents;

import com.bea.control.*;

public interface EventRaiser extends Control
{
    public void startTimer();

    interface Callback
    {
        void backEndTimer_onTimeout(long arg0);
    }
}
```

EventRaiserImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/controlEvents/

Sample Source Code

```
package controlEvents;

import com.bea.control.*;

/**
 * EventRaiser illustrates how you can pass a callback
 * from a nested control through to a client. EventRaiser nests
 * a Timer control. When that Timer fires its onTimeout, that
 * callback is simply sent directly to the client through a
 * callback that EventRaiser exposes.
 *
 * @jcs:jc-jar label="EventRaiser"
 *      version="1.0"
 *      icon-16="/images/buffer_16.gif"
 *      icon-32="/images/buffer_32.gif"
 *      group-name="Feature Sample Controls"
 *      description="Starts a timer and calls raiseEvent to callback its container"
 *      palette-priority="7"
 */
public class EventRaiserImpl implements EventRaiser, com.bea.control.ControlSource
{
    /**
     * @jc:timer timeout="5 seconds"
     * @common:control
     */
    private TimerControl backEndTimer;

    /**
     * @common:context
     */
    com.bea.control.ControlContext context;

    /**
     * The startTimer method provides a way for EventRaiser's
     * clients to start its Timer.
     *
     * @common:operation
     */
    public void startTimer()
    {
        backEndTimer.start();
    }

    /**
     * This is the handler for the Timer control's onTimeout

```

navExtensionDevKit.html Sample

```
* callback. When that callback is received, it is sent directly
* to EventRaiser's client.
*/
public void backEndTimer_onTimeout(long arg0)
{
    try{
        /*
         * The raiseEvent method tells WebLogic Server to
         * forward the onTimeout callback to an EventRaiser
         * callback *that has the same name*.
         */
        context.raiseEvent();
    } catch (Exception e){
        throw new ControlException("EventRaiser exception.", e);
    }
}
}
```

EventScheduler.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/controlEvents/

Sample Source Code

```
package controlEvents;

import com.bea.control.*;

public interface EventScheduler extends Control
{
    public void setAlarm(long delayInMsecs);
    public void cancelAlarm();
    /*
     * The onAlarmRinging callback notifies clients when the interval they
     * specified has passed.
     */
    public interface Callback
    {
        void onAlarmRinging(String message, long time);
    }
}
```

EventSchedulerImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/controlEvents/

Sample Source Code

```
package controlEvents;

import com.bea.control.*;
import java.util.Date;

/**
 * EventScheduler illustrates how you can schedule a control's
 * callbacks to occur at specified times. This example receives
 * a request to send a callback after a specified interval. The
 * control's client passes in the interval. Note that this control
 * does not use a Timer control.
 */
@jcs:jc-jar palette-priority="7" label="EventScheduler"
version="1.0"
icon-16="/images/buffer_16.gif"
icon-32="/images/buffer_32.gif"
group-name="Feature Sample Controls"
description="Schedules a callback at a point in the future"
/

public class EventSchedulerImpl implements EventScheduler, com.bea.control.ControlSource
{
    public Callback callback;

    /**
     * @common:context
     */
    com.bea.control.ControlContext context;

    /**
     * The setAlarm method receives an interval value, then uses
     * that value to schedule its callback. The callback will occur
     * on or after the interval's passing.
     */
    @common:operation
    /**
     public void setAlarm(long delayInMsecs)
     {
         /**
          * The alarm is set for the current time plus the interval
          * specified by the client.
          */
         long alarmTime = new Date().getTime() + delayInMsecs;

         /** Arguments for the scheduled callback are stored in
          * an Object array. That array is passed as an argument

```

navExtensionDevKit.html Sample

```
    * to the scheduleEvent method.
    */
    Object[] callbackArgs = new Object[2];
    callbackArgs[0] = "Time to wake up.";
    callbackArgs[1] = new Long(alarmTime);

    /*
     * The scheduleEvent method schedules the callback's firing by
     * specifying the callback's name, its arguments (as an Object array),
     * the time at/after which the callback should be sent, and a boolean
     * value indicating whether the callback should be ignored (and no
     * exception generated) if the callback is sent to a finished
     * instance.
     */
    context.scheduleEvent("onAlarmRinging", callbackArgs, alarmTime, true);
}

/**
 * cancelAlarm provides a way for clients to cancel the alarm.
 * The cancelEvents method specifies which scheduled callback should
 * be canceled.
 *
 * @common:operation
 */
public void cancelAlarm()
{
    context.cancelEvents("onAlarmRinging");
}

}
```

LifeCycle.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/controlEvents/

Sample Source Code

```
package controlEvents;

import com.bea.control.*;

public interface LifeCycle extends Control
{
    public interface Callback
    {
        void onDone(String message);
    }

    /**
     * The startMeUp method provides a place for clients to start using this control.
     * @common:operation
     */
    void startMeUp();
}
```

LifeCycleImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/controlEvents/

Sample Source Code

```
package controlEvents;

import com.bea.control.*;
import java.util.Date;
import java.util.Calendar;

/**
 * The LifeCycle control simply illustrates the events that occur
 * in the life of a Java control. As the client uses this control --
 * acquiring it, calling its method, releasing it -- each of these
 * events is reported to the WebLogic Server console window.
 *
 * @jcs:jcs-jar palette-priority="7" label="LifeCycle"
 *      version="1.0"
 *      icon-16="/images/buffer_16.gif"
 *      icon-32="/images/buffer_32.gif"
 *      group-name="Feature Sample Controls"
 *      description="Illustrates Context life cycle events"
 *      @editor-info:code-gen control-interface="true"
 */
public class LifeCycleImpl implements LifeCycle, com.bea.control.ControlSource
{
    /**
     * @jc:timer timeout="5 seconds"
     * @common:control
     */
    private TimerControl delayTimer;

    public Callback callback;

    long m_currentTime;
    StringBuffer m_cycleMessages = new StringBuffer();
    Calendar m_calendar;

    /**
     * @common:context
     */
    com.bea.control.ControlContext context;

    /**
     * The startMeUp method provides a place for clients to start using this
     * control.
     *
     * @common:operation
     */
}
```


navExtensionDevKit.html Sample

```
public void startMeUp()
{
    String message = "LifeCycle: context.startMeUp method is called. \n";
    System.out.println(message);
    m_cycleMessages.append(message);
}

/*
 * The onCreate callback is received when the control is created by
 * the container. onCreate is a good place to acquire resources that
 * may be needed by the control.
 */
public void context_onCreate() {
    delayTimer.start();
    String message = "LifeCycle: context_onCreate received. Timer started. \n";

    System.out.println(message);
    m_cycleMessages.append(message);
}

/*
 * onAcquire is received when a control operation is about to start.
 */
public void context_onAcquire() {
    String message = "LifeCycle: context_onAcquire received. \n";
    System.out.println(message);
    m_cycleMessages.append(message);
}

/*
 * onReset is implemented to let the container know that this
 * instance can be re-used after a call to onReset().
 */
public void context_onReset() {
    String message = "LifeCycle: context_onReset received. \n";
    System.out.println(message);
    m_cycleMessages.append(message);
}

/*
 * onRelease is received with a control operation has finished.
 */
public void context_onRelease() {
    String message = "LifeCycle: context_onRelease received. \n";
    System.out.println(message);
    m_cycleMessages.append(message);
}

/*
 * onException is received in the event of an unhandled exception.
 * It provides a place to do clean up and notify the client
 * of the error.
 */
public void context_onException(Exception e, String methodName,
    Object[] methodArgs)
{
    throw new ControlException("Exception at: " + methodName, e);
}
```

navExtensionDevKit.html Sample

```
public void delayTimer_onTimeout(long arg0)
{
    String message = "Lifecycle: delayTimer_onTimeout received. " +
        "onDone about to be invoked. \n";
    System.out.println(message);
    callback.onDone(m_cycleMessages.toString());
}
}
```

insertWizard Samples

This section contains source code for the following samples.

Samples Included in This Section

ide Samples

MBeanUtil.java Sample

ServerCheck-tags.xml Sample

ServerCheck.java Sample

ServerCheckImpl.jcs Sample

ide Samples

This section contains source code for the following samples.

Samples Included in This Section

ServerCheckWizard.java Sample

ServerCheckWizardPanel.java Sample

ServerCheckWizard.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizard/ide/

Sample Source Code

```
package insertWizard.ide;

import com.bea.ide.control.ControlWizardSimple;
import javax.swing.JComponent;
import java.util.ArrayList;
import com.bea.ide.control.EditorContext;

/*
 * The insert-wizard-class attribute in the jc-jar file for this
 * control project specifies that this is the class WebLogic Workshop
 * should use to provide a custom insert wizard.
 *
 * This class extends ControlWizardSimple, which provides methods through
 * which WebLogic Workshop retrieves the user interface to use, the
 * annotation values to insert into the control's container, and so on.
 */
public class ServerCheckWizard extends ControlWizardSimple
{
    private ServerCheckWizardPanel m_panel;
    private boolean _createExtension = false;
    private EditorContext _context;

    /*
     * Initializes the user interface. UI components
     * are assembled in the ServerCheckWizardPanel class.
     */
    public ServerCheckWizard ()
    {
        m_panel = new ServerCheckWizardPanel();
    }

    /*
     * Tells WebLogic Workshop whether to display an option
     * to create a JCX file. The constant returned here
     * tells it that an instance is all that is needed.
     */
    public int getConfigurationInfo()
    {
        return CONFIG_INSERT_INSTANCE;
    }

    /**
     * Gives the wizard a way to interact with the IDE.
     */
    public void setContext(EditorContext ctx)
    {

```

navExtensionDevKit.html Sample

```
_context = ctx;
m_panel.setContext(ctx);
}

/**
 * Provides a way for subclasses to get the context as needed.
 */
public EditorContext getContext()
{
    return _context;
}

/**
 * Returns the user interface component to use.
 */
public JComponent getComponent()
{
    return m_panel;
}

/**
 * Provides a place to execute code when the control's user
 * clicks the "Create" button on the insert dialog.
 */
public boolean onFinish()
{
    {
        if (super.onFinish() == false){
            return false;
        } else {
            return true;
        }
    }
}

/**
 * Provides to WebLogic Workshop an ArrayList containing
 * the annotations it should write preceding the control
 * instance.
 */
public ArrayList getInstanceAnnotations()
{
    // An ArrayList for the annotation attributes. There are four.
    ArrayList attributeList = new ArrayList();
    // An ArrayList for this control's two annotations.
    ArrayList annotationsList = new ArrayList();

    /**
     * The attribute ArrayList is filled with TagAttributeValue
     * instances containing attribute name/value mappings. These
     * are used below for the jc:server-data annotation.
     */
    /**
     * Attribute values are retrieved from the insert dialog box UI.
     */
    attributeList.add(new TagAttributeValue("server-name", m_panel.getServerName()));
    attributeList.add(new TagAttributeValue("url", m_panel.getServerURL()));
    attributeList.add(new TagAttributeValue("user-name", m_panel.getUserName()));
    attributeList.add(new TagAttributeValue("password", m_panel.getPassword()));

    /**
     * The annotation ArrayList is filled with TagAttributeList instances
     * containing the annotation name/attribute-list mappings.
     * Note that the second annotation uses the attribute ArrayList.
     */
}
```

navExtensionDevKit.html Sample

```
    */  
    annotationsList.add(new TagAttributeList("common:control", null));  
    annotationsList.add(new TagAttributeList("jc:server-data", attributeList));  
        return annotationsList;  
    }  
}
```

ServerCheckWizardPanel.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizard/ide/

Sample Source Code

```
package insertWizard.ide;

import weblogic.management.MBeanHome;
import java.awt.event.ActionListener;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JPasswordField;
import javax.swing.JLabel;
import javax.swing.JButton;
import insertWizard.MBeanUtil;
import com.bea.ide.control.EditorContext;
import weblogic.management.configuration.ConfigurationMBean;
import javax.management.MBeanServer;
import weblogic.management.RemoteMBeanServer;
import javax.management.MBeanInfo;
import weblogic.management.configuration.WebServerMBean;
import java.util.Set;
import java.util.Iterator;
import java.awt.FocusTraversalPolicy;
import javax.swing.JComponent;
import java.awt.Component;
import java.awt.Container;
import java.net.MalformedURLException;
import java.net.URL;

/*
 * Represents the user interface for the ServerCheck control
 * insert dialog.
 */
public class ServerCheckWizardPanel extends JPanel
    implements ActionListener
{
    JPanel m_pnlServerInfo;
    JTextField m_txtUserName;
    JPasswordField m_pssPassword;
    JLabel m_lblPassword;
    JLabel m_lblUserName;
    JTextField m_txtServerURL;
    JLabel m_lblServerURL;
    JButton m_btnCheckConnection;
    JTextField m_txtServerName;
```


navExtensionDevKit.html Sample

```

        JLabel m_lblServerName;
        EditorContext _ctx;

        public ServerCheckWizardPanel()
        {
            initComponents();
        }

        public void setContext(EditorContext ctx)
        {
            _ctx = ctx;
            m_pnlServerInfo.setFocusCycleRoot(true);
            m_pnlServerInfo.setFocusTraversalPolicy(_newFtp);
        }
        /*
        * Executes when the user clicks the "Check Connection" button
        * in the insert dialog. If the connection does not succeed,
        * relevant messages are forwarded to the user from the MBean.
        */
        public void actionPerformed(ActionEvent e) {
            MBeanHome mBeanFromUserData = null;
            MBeanHome mBeanCurServer = null;
            String message="Connection Check:";
            try {
                if (!_ctx.ensureServerRunning())
                    return;
                mBeanCurServer = (MBeanHome)_ctx.getMBeanHome();

                RemoteMBeanServer mBS = mBeanCurServer.getMBeanServer();
                message += "<br> Server Name current value is: " + mBS.getServerName();

                mBeanFromUserData = MBeanUtil.getMBean(getUserName(), getPassword(),
                    getServerURL(), getServerName());
                message += "<br> Password access verified!";
                this.showConnectionCheckResults(message);
            } catch (IllegalArgumentException iae) {
                message += "<br>" + iae.getMessage();
                this.showConnectionCheckResults(message);
            }
            catch (Exception ex) {
                ex.printStackTrace();
                message += "<br>" + ex.getMessage();
                this.showConnectionCheckResults(message);
            }
        }

        /*
        * Displays a dialog with the results of the "Check Connection"
        * button.
        */
        private void showConnectionCheckResults(String resultsMessage){
            StringBuffer message = new StringBuffer();
            message.append("<html><p>").append(resultsMessage).
                append("</p><p>").append("</p></html>");
            JOptionPane.showMessageDialog(this, message, "Connection Check",
                JOptionPane.INFORMATION_MESSAGE);
        }

        private FocusTraversalPolicy _newFtp = new FocusTraversalPolicy()
        {

```

navExtensionDevKit.html Sample

```
public Component getComponentAfter(Container focusCycleRoot, Component aComponent)
{
    if (aComponent==m_btnCheckConnection)
        return m_txtServerName;
    if (aComponent==m_txtServerName)
        return m_txtServerURL;
    if (aComponent==m_txtServerURL)
        return m_txtUserName;
    if (aComponent==m_txtUserName)
        return m_pssPassword;
    if (aComponent==m_pssPassword)
        return m_btnCheckConnection;
    return null;
}

public Component getComponentBefore(Container focusCycleRoot, Component aComponent)
{
    if (aComponent==m_txtServerName)
        return m_btnCheckConnection;
    if (aComponent==m_btnCheckConnection)
        return m_pssPassword;
    if (aComponent==m_pssPassword)
        return m_txtUserName;
    if (aComponent==m_txtUserName)
        return m_txtServerURL;
    if (aComponent==m_txtServerURL)
        return m_txtServerName;
    return null;
}

public Component getDefaultComponent(Container focusCycleRoot)
{
    return m_btnCheckConnection;
}

public Component getFirstComponent(Container focusCycleRoot)
{
    return m_txtServerName;
}

public Component getLastComponent(Container focusCycleRoot)
{
    return m_btnCheckConnection;
}
};

/*
 * Assembles user interface components that make up the insert
 * dialog.
 */
private void initComponents() {

    m_pnlServerInfo = new JPanel();
    m_txtUserName = new JTextField();
    m_pssPassword = new JPasswordField();
    m_lblPassword = new JLabel();
    m_lblUserName = new JLabel();
    m_txtServerURL = new JTextField();
    m_lblServerURL = new JLabel();
    m_btnCheckConnection = new JButton();
    m_txtServerName = new JTextField();
```

navExtensionDevKit.html Sample

```
m_lblServerName = new JLabel();

GridBagConstraints gridBagConstraints;

setLayout(new GridBagLayout());
setEnabled(true);

m_lblServerName.setText("Server name:  ");
m_lblServerName.setLabelFor(m_txtServerName);
m_lblServerName.setDisplayedMnemonic('S');

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 7;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.insets = new java.awt.Insets(0, 0, 0, 6);
add(m_lblServerName, gridBagConstraints);

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 7;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_txtServerName, gridBagConstraints);

        m_lblServerURL.setText("Server URL:      ");
m_lblServerURL.setLabelFor(m_txtServerURL);
m_lblServerURL.setDisplayedMnemonic('r');
m_txtServerURL.setFocusTraversalKeysEnabled(true);

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 8;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_lblServerURL, gridBagConstraints);

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 8;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_txtServerURL, gridBagConstraints);

        m_lblUserName.setText("WLS Admin user:  ");
m_lblUserName.setLabelFor(m_txtUserName);
m_lblUserName.setDisplayedMnemonic('A');

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 9;
```

navExtensionDevKit.html Sample

```
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_lblUserName, gridBagConstraints);

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 9;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_txtUserName, gridBagConstraints);

        m_lblPassword.setText("Password:      ");
m_lblPassword.setLabelFor(m_pssPassword);
m_lblPassword.setDisplayedMnemonic('P');

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 10;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_lblPassword, gridBagConstraints);

gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 10;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 0);
add(m_pssPassword, gridBagConstraints);

        m_btnCheckConnection.setText("Check Connection");
m_btnCheckConnection.setMnemonic('C');
m_btnCheckConnection.addActionListener(this);
gridBagConstraints = new GridBagConstraints();
gridBagConstraints.gridx = 1;
gridBagConstraints.gridy = 11;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.fill = GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = GridBagConstraints.WEST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.insets = new java.awt.Insets(6, 0, 6, 250);
add(m_btnCheckConnection, gridBagConstraints);

}

/*
 * A set of accesses that simplify retrieving the values
 * that the control's user has entered into the insert dialog.
 */
```

navExtensionDevKit.html Sample

```
public String getPassword(){
    return m_pssPassword.getText();
}

public String getUsername(){
    return m_txtUserName.getText();
}

public String getServerName(){
    return m_txtServerName.getText();
}

public String getServerURL() throws IllegalArgumentException
{
    try
    {
        URL serverURL = new URL(m_txtServerURL.getText());
        return serverURL.toString();
    }
    catch (MalformedURLException mfue)
    {
        throw new IllegalArgumentException("Server URL must be of form http://localhost:7000");
    }
}
}
```

MBeanUtil.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizard/

Sample Source Code

```
package insertWizard;

import weblogic.management.MBeanHome;
import weblogic.management.Helper;

/*
 * Represents the server's MBean to the control.
 */
public class MBeanUtil
{
    public static MBeanHome getMBean(String userName, String password, String serverURL,
        String serverName) throws IllegalArgumentException
    {
        MBeanHome requestedMBean = null;
        try {
            requestedMBean = (MBeanHome)Helper.getMBeanHome(userName, password,
                serverURL, serverName);
        } catch (IllegalArgumentException iae) {
            throw iae;
        }
        return requestedMBean;
    }
}
```

ServerCheck-tags.xml Sample

This topic includes the source code for the ServerCheck-tags.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizard/

Sample Source Code

```
<?xml version="1.0"?>
<control-tags xmlns="http://www.bea.com/2003/03/controls/">
  <control-tag name="server-data">
    <description>Location and login information for the server to access.</description>
    <attribute name="url" required="true">
      <description>The URL of the server to access.</description>
      <type>
        <URL/>
      </type>
    </attribute>
    <attribute name="server-name" required="true">
      <description>The name of the server to access.</description>
      <type>
        <text/>
      </type>
    </attribute>
    <attribute name="user-name" required="false">
      <description>The username of a user who has administrative access to this server.</description>
      <type>
        <text/>
      </type>
    </attribute>
    <attribute name="password" required="false">
      <description>The password of a user who has administrative access to this server.</description>
      <type>
        <text/>
      </type>
    </attribute>
  </control-tag>
</control-tags>
```

ServerCheck.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizard/

Sample Source Code

```
package insertWizard;

import com.bea.control.*;

public interface ServerCheck extends Control
{
    /**
     * The getDomainName method is the control's only operation. It uses an external class, MBE
     * @common:operation
     */
    java.lang.String getDomainName();
}
```


ServerCheckImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizard/

Sample Source Code

```
package insertWizard;

import com.bea.control.*;
import weblogic.management.MBeanHome;
import weblogic.management.Helper;
import weblogic.management.WebLogicMBean;
import weblogic.management.WebLogicObjectName;

/**
 * The ServerCheck control illustrates how to connect a custom
 * dialog box that will be used to collect attribute values
 * when the control is added to a project. This control doesn't
 * create a JCX file; its property annotations are specified
 * at the control instance level, inside the control's container.
 *
 * The ServerCheckWizard class provides the hooks needed by WebLogic
 * Workshop to show a dialog box, get and set attribute values, and
 * so on. The ServerCheckWizardPanel provides the insert wizard's user
 * interface components.
 *
 * This control is designed to use a WebLogic Server MBean to
 * return the server's domain name. The control's property
 * attributes are the server's name and URL, as well as a user name
 * and password to be used if needed.
 *
 * @jcs:control-tags file="ServerCheck-tags.xml"
 * @jcs:jc-jar label="ServerCheck"
 *     insert-wizard-class="insertWizard.ide.ServerCheckWizard"
 *     version="0.8"
 *     icon-16="/images/hello_16.gif"
 *     icon-32="/images/hello_32.gif"
 *     palette-priority="6"
 *     group-name="Feature Sample Controls"
 *     description="Illustrates custom insert wizard"
 *     @editor-info:code-gen control-interface="true"
 */
public class ServerCheckImpl implements ServerCheck, com.bea.control.ControlSource
{
    /**
     * @common:context
     */
    com.bea.control.ControlContext context;

    /**
     * Variables to hold values specified by the control's user.
     */
}
```

navExtensionDevKit.html Sample

```
    */
    String m_serverURL = null;
    String m_userName = null;
    String m_password = null;
    String m_serverName = null;
    transient MBeanHome m_localMBean = null;

    /**
     * The getDomainName method is the control's only operation.
     * It uses an external class, MBeanUtil, to get the server's
     * domain name.
     *
     * @common:operation
     */
    public String getDomainName ()
    {
        /*
         * The context.getControl attribute method retrieves values
         * specified in annotations on the control instance declaration.
         * These annotations are written into the control's container.
         */
        m_userName = context.getControlAttribute("jc:server-data","user-name");
        m_password = context.getControlAttribute("jc:server-data","password");
        m_serverURL = context.getControlAttribute("jc:server-data","url");
        m_serverName = context.getControlAttribute("jc:server-data","server-name");

        try {
            m_localMBean = MBeanUtil.getMBean(m_userName, m_password, m_serverURL, m_serverName);
        } catch (IllegalArgumentException iae) {
            context.getLogger("ServerCheck");
            throw new ControlException("ServerCheck: Error getting the domain name.", iae);
        }
        String domainName = m_localMBean.getDomainName();
        return domainName;
    }
}
```

insertWizardCustom Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomWiz.java Sample

CustomWizImpl.jcs Sample

ide Samples

CustomWiz.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizardCustom/

Sample Source Code

```
package insertWizardCustom;

import com.bea.control.Control;

public interface CustomWiz extends Control
{
}
```

CustomWizImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizardCustom/

Sample Source Code

```
package insertWizardCustom;

import com.bea.control.*;
import java.lang.reflect.Method;
/**
 * @editor-info:code-gen control-interface="true"
 * @jcs:jcs-jar group-name="Feature Sample Controls" insert-wizard-class="insertWizardCustom.ide
 */
public class CustomWizImpl implements CustomWiz, ControlSource, Extensible
{
    static final long serialVersionUID = 1L;

    public Object invoke(Method method, Object[] args) throws Throwable
    {
        if (method.getName()=="echoInput")
            return (String)args[0];
        else
            return "Unknown method";
    }
}
```

ide Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomInsertDialog.java Sample

CustomInsertWizard.java Sample

html Samples

CustomInsertDialog.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizardCustom/i

Sample Source Code

```
package insertWizardCustom.ide;

import com.bea.control.Issue;
import com.bea.ide.Application;
import com.bea.ide.control.ControlWizard;
import com.bea.ide.core.MessageSvc;
import com.bea.ide.core.ResourceSvc;
import com.bea.ide.ui.browser.BrowserSvc;
import com.bea.ide.ui.output.OutputSvc;
import com.bea.ide.util.swing.DialogUtil;
import java.awt.Frame;
import java.awt.Insets;
import java.awt.KeyEventDispatcher;
import java.awt.KeyboardFocusManager;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import java.awt.event.KeyEvent;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.net.URL;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JDialog;
import javax.swing.JEditorPane;
import javax.swing.JFileChooser;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.event.HyperlinkEvent;
import javax.swing.event.HyperlinkListener;

/**
 * A dialog implementation for the IDE to display when the user chooses to
 * add or insert a CustomWiz control.
 */
public class CustomInsertDialog extends JDialog
{
    protected String m_variableName = new String();
    protected String m_jcxName = new String();
}
```

navExtensionDevKit.html Sample

```
* This variable is used in this dialog's logic to indicate
* whether a new JCX may be created. This value is retrieved
* by the CustomInsertWizard instance for this control for
* passing to the IDE through the wizard's implementation of the
* onFinish method.
*/
protected boolean m_controlInsertRequested = false;
private boolean m_jcxNameIsGood = false;
private boolean m_variableNameIsGood = false;
private CustomInsertWizard m_wizard = null;
private Frame m_parent = null;

// Declare variables for the user interface components.
private JButton m_btnCancel;
private JButton m_btnCreate;
private JEditorPane m_editIntroduction;
private JLabel m_lblJcxOptions;
private JLabel m_lblNewJcx;
private JLabel m_lblVariableName;
private JPanel m_pnlCreateCancel;
private JPanel m_pnlMessages;
private JPanel m_pnlTab1;
private JPanel m_pnlTab2;
private JPanel m_pnlVariableJCXName;
private JScrollPane m_scrIntroduction;
private JScrollPane m_scrMessages;
private JTabbedPane m_tabTabbedPane;
private JTextField m_txtNewJcx;
private JTextField m_txtVariableName;
private JTextArea m_txtaMessages;
private DialogLinkListener m_linkListener;
private DialogFocusListener m_focusListener;
private DialogMouseListener m_mouseListener;
private DialogDocumentListener m_documentListener;

/**
 * Constructs the dialog with a parent (the WebLogic Workshop
 * window) and an instance of the wizard. The wizard instance will be
 * used to retrieve information about the current context.
 *
 * @param parent A Frame representing the WebLogic Workshop main window.
 * @param wizard The current wizard instance.
 */
public CustomInsertDialog(Frame parent, CustomInsertWizard wizard)
{
    super(parent, true);
    // Variables for the control's wizard and for the WebLogic Workshop frame.
    m_wizard = wizard;
    m_parent = parent;
    // Assemble the UI components.
    initComponents();
    // Set this dialog's size.
    this.setSize(499, 561);
    // This dialog cannot be resized.
    this.setResizable(false);
    // Set the title bar text.
    this.setTitle("Insert Control - CustomWiz");
}

/**
 * Displays this dialog.
```


navExtensionDevKit.html Sample

```

    */
    public void show()
    {
        /**
         * Add a key event dispatcher to listen for ENTER and ESC keystrokes.
         */
        DialogKeyEventDispatcher dked = new DialogKeyEventDispatcher();
        KeyboardFocusManager.getCurrentKeyboardFocusManager().addKeyEventDispatcher(dked);
        super.show();
        KeyboardFocusManager.getCurrentKeyboardFocusManager().removeKeyEventDispatcher(dked);
    }

    /**
     * Assemble the user interface for this dialog. All of the work
     * to assemble this dialog's user interface is done in this method.
     */
    private void initComponents()
    {
        java.awt.GridBagConstraints gridBagConstraints;

        /**
         * Initialize the user interface components.
         */
        m_tabbedPane = new JTabbedPane();
        m_pnlTab1 = new JPanel();
        m_scrIntroduction = new JScrollPane();
        m_editIntroduction = new JEditorPane();
        m_pnlTab2 = new JPanel();
        m_pnlVariableJCXName = new JPanel();
        m_lblVariableName = new JLabel();
        m_txtVariableName = new JTextField();
        m_lblJcxOptions = new JLabel();
        m_lblNewJcx = new JLabel();
        m_txtNewJcx = new JTextField();
        m_pnlMessages = new JPanel();
        m_scrMessages = new JScrollPane();
        m_txtaMessages = new JTextArea();
        m_pnlCreateCancel = new JPanel();
        m_btnCreate = new JButton();
        m_btnCancel = DialogUtil.getCancelButton();
        m_linkListener = new DialogLinkListener();
        m_focusListener = new DialogFocusListener();
        m_mouseListener = new DialogMouseListener();
        m_documentListener = new DialogDocumentListener();

        /**
         * Set the layout style for this dialog.
         */
        getContentPane().setLayout(new java.awt.GridBagLayout());

        // Specify what should happen when the dialog closes.
        setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
        addWindowListener(new java.awt.event.WindowAdapter()
        {
            public void windowClosing(java.awt.event.WindowEvent evt)
            {
                closeDialog(evt);
            }
        });
    }

    /**

```

navExtensionDevKit.html Sample

```
* Assemble the panel that makes up the first tab of this dialog.
*/
m_pnlTab1.setLayout(new java.awt.GridBagLayout());

/*
 * m_editIntroduction is the JEditorPane that displays the HTML
 * introduction to this sample. m_scrIntroduction is the JScrollPane
 * that contains it and provides a vertical scroll bar.
 */
m_editIntroduction.setContentType("text/html");
// Connect a listener to handle clicks to links in the introduction.
m_editIntroduction.addHyperlinkListener(m_linkListener);
m_editIntroduction.setEditable(false);
m_editIntroduction.setMargin(new Insets(0, 0, 0, 8));
m_editIntroduction.setSize(m_editIntroduction.getSize());
m_scrIntroduction.setViewportView(m_editIntroduction);
m_scrIntroduction.setAutoscrolls(true);

/*
 * Use the application's classloader to find the HTML file
 * to load in the introduction pane. Set that URL as the
 * content of the pane.
 */
ClassLoader loader = Thread.currentThread().getContextClassLoader();
URL introUrl = loader.getResource("insertWizardCustom/ide/html/introduction.html");
try
{
    m_editIntroduction.setPage(introUrl);
}
catch (java.io.IOException ioe)
{
    ioe.printStackTrace();
}

/*
 * Swing provides several "layout styles" -- ways to structure the components
 * of a user interface. For the most part, this dialog uses GridBagLayout
 * as its style. In a gridbag layout, components are structured based on a grid
 * and their position in the grid. This includes X and Y axes, number of
 * grid points in width or height, "weight" relative to other components,
 * where the component is anchored (top, right, left, etc), and so on.
 *
 * You use a GridBagConstraint instance to set these component properties,
 * then add the component to its parent along with the constraint instance.
 *
 * Here, the constraint instance is used to set layout properties for
 * m_scrIntroduction, the JScrollPane that contains the pane used to
 * display introduction HTML.
 */
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 1.0;
gridBagConstraints.insets = new java.awt.Insets(4, 4, 4, 4);
// Add the scroll pane to the panel on the first tab.
m_pnlTab1.add(m_scrIntroduction, gridBagConstraints);

m_tabTabbedPane.setName("tabs");

/*
 * Add the first tab panel as the UI component to the tabbed pane.
```

navExtensionDevKit.html Sample

```
    */
    m_tabTabbedPane.addTab("Introduction", m_pnlTab1);

    /*
     * Assemble the panel that makes up the first tab of this dialog.
     */
    m_pnlTab2.setLayout(new java.awt.GridBagLayout());

    /*
     * Set layout of components in this panel to null. Its components will be
     * arranged through explicit coordinates.
     */
    m_pnlVariableJCXName.setLayout(null);

    /*
     * Define the variable name label and box and add them to the panel.
     */
    m_lblVariableName.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
    m_lblVariableName.setText("Variable name for this control:");
    m_lblVariableName.setBounds(0, 20, 163, 16);
    m_pnlVariableJCXName.add(m_lblVariableName);

    m_txtVariableName.setHorizontalAlignment(javax.swing.JTextField.LEFT);
    m_txtVariableName.setName("variable_name_box");
    m_txtVariableName.setToolTipText("Enter the name for a variable of this control type.")
    // Connect a listener that will handle receiving focus.
    m_txtVariableName.addFocusListener(m_focusListener);
    // Connect a listener that will handle changes to the box's contents.
    m_txtVariableName.getDocument().addDocumentListener(m_documentListener);
    m_txtVariableName.setBounds(160, 20, 295, 20);
    m_pnlVariableJCXName.add(m_txtVariableName);

    /**
     * Define the JCX name label and box and add them to the panel.
     */
    m_lblNewJcx.setText("JCX name for this control:");
    m_lblNewJcx.setBounds(0, 62, 163, 16);
    m_pnlVariableJCXName.add(m_lblNewJcx);

    m_txtNewJcx.setToolTipText("Enter the name of the JCX to create");
    m_txtNewJcx.setName("new_jcx_name_box");
    // Connect a listener that will handle receiving focus.
    m_txtNewJcx.addFocusListener(m_focusListener);
    // Connect a listener that will handle changes to the box's contents.
    m_txtNewJcx.getDocument().addDocumentListener(m_documentListener);
    m_txtNewJcx.setBounds(140, 62, 315, 20);
    m_pnlVariableJCXName.add(m_txtNewJcx);

    /**
     * Define this panel's relationship to other components on
     * the tab, then add the panel to the tab.
     */
    gridBagConstraints = new java.awt.GridBagConstraints();
    gridBagConstraints.gridx = 0;
    gridBagConstraints.gridy = 0;
    gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
    gridBagConstraints.insets = new java.awt.Insets(0, 12, 0, 12);
    gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTH;
    gridBagConstraints.weightx = 1.0;
    gridBagConstraints.weighty = 1.0;
    m_pnlTab2.add(m_pnlVariableJCXName, gridBagConstraints);
```

navExtensionDevKit.html Sample

```
/**
 * Define the message panel and text area, then add them to the
 * tab.
 */
m_pnlMessages.setLayout(new java.awt.GridBagLayout());
m_pnlMessages.setPreferredSize(m_pnlMessages.getPreferredSize());
m_txtaMessages.setLineWrap(true);
m_txtaMessages.setWrapStyleWord(true);
m_txtaMessages.setMargin(new Insets(6, 6, 6, 6));
m_txtaMessages.setSize(m_txtaMessages.getSize());
m_txtaMessages.setEditable(false);
m_scrMessages.setHorizontalScrollBarPolicy(javax.swing.JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
m_scrMessages.setViewportView(m_txtaMessages);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 1.0;
m_pnlMessages.add(m_scrMessages, gridBagConstraints);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 1;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.insets = new java.awt.Insets(0, 10, 10, 10);
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTH;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 3.5;
m_pnlTab2.add(m_pnlMessages, gridBagConstraints);

// Add the tab to the group of tabs.
m_tabTabbedPane.addTab("Name the Control", m_pnlTab2);

/**
 * Add the tabs to the dialog. getContentPane returns this
 * dialog's content pane.
 */
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 0;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.gridheight = 1;
gridBagConstraints.fill = java.awt.GridBagConstraints.BOTH;
gridBagConstraints.anchor = java.awt.GridBagConstraints.NORTH;
gridBagConstraints.insets = new java.awt.Insets(6, 6, 15, 6);
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 1.0;
// Connect a listener that will handle mouse clicks.
m_tabTabbedPane.addMouseListener(m_mouseListener);
getContentPane().add(m_tabTabbedPane, gridBagConstraints);

/*
 * Define the panel that holds the Create and Cancel buttons,
 * then add it to the dialog.
 */
m_pnlCreateCancel.setLayout(new java.awt.GridBagLayout());
m_btnCreate.setText("Create");
m_btnCreate.setName("create_button");
// Connect a listener that will handle mouse clicks.
m_btnCreate.addMouseListener(m_mouseListener);
```

navExtensionDevKit.html Sample

```
// Because the first tab is visible by default, disable the Create button.
m_btnCreate.setEnabled(false);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 2;
gridBagConstraints.insets = new java.awt.Insets(0, 0, 12, 100);
gridBagConstraints.anchor = java.awt.GridBagConstraints.SOUTHEAST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 1.0;
m_pnlCreateCancel.add(m_btnCreate, gridBagConstraints);
m_btnCancel.setText("Cancel");
m_btnCancel.setName("cancel_button");
// Connect a listener that will handle mouse clicks.
m_btnCancel.addMouseListener(m_mouseListener);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 2;
gridBagConstraints.insets = new java.awt.Insets(0, 0, 12, 12);
gridBagConstraints.anchor = java.awt.GridBagConstraints.SOUTHEAST;
gridBagConstraints.weightx = 1.0;
gridBagConstraints.weighty = 1.0;
m_pnlCreateCancel.add(m_btnCancel, gridBagConstraints);
gridBagConstraints = new java.awt.GridBagConstraints();
gridBagConstraints.gridx = 0;
gridBagConstraints.gridy = 3;
gridBagConstraints.gridwidth = 2;
gridBagConstraints.gridheight = java.awt.GridBagConstraints.REMAINDER;
gridBagConstraints.fill = java.awt.GridBagConstraints.HORIZONTAL;
gridBagConstraints.anchor = java.awt.GridBagConstraints.SOUTH;
getContentPane().add(m_pnlCreateCancel, gridBagConstraints);

/*
 * If the user tried to add a CustomWiz control by clicking File -> New
 * (or by right-clicking in the Application pane), then don't show the second tab.
 * The user does not want to add this control as an instance to a container (yet),
 * and has already been prompted for a JCX name.
 *
 * If the user has clicked the Insert menu (or right-clicked Design View), the second
 * tab will be displayed. In that scenario, the IDE will set configuration as
 * a sum of CONFIG_CREATE_EXTENSION_FILE and CONFIG_INSERT_INSTANCE.
 */
if (m_wizard.m_requestedConfig == CustomInsertWizard.CONFIG_CREATE_EXTENSION_FILE)
{
    // Disable the tab at index 1 (the second tab).
    m_tabTabbedPane.setEnabledAt(1, false);
    // Hide the Create button because it's not relevant in this case.
    m_btnCreate.setVisible(false);
    // Select the Cancel button (which is relabeled below).
    m_btnCancel.setSelected(true);
    // Relabel the Cancel button to Done.
    m_btnCancel.setLabel("Done");
}

// Sizes the window to preferred size.
pack();
}

/** Closes the dialog */
private void closeDialog(java.awt.event.WindowEvent evt)
{
    setVisible(false);
}
```

navExtensionDevKit.html Sample

```

}

/**
 * Prints messages to the dialog's bottom pane as the user
 * shifts focus to components in the user interface.
 */
private void printActionMessage(String messageText)
{
    m_txtaMessages.append(messageText + "\n");
}

/**
 * Displays a dialog that contains messages resulting from
 * validating JCX or variable names. This method is called if there
 * are validation issues when the Create button is clicked.
 *
 * @param issues Issues that arose while validating dialog values.
 * @param proposedName The name that was validated.
 */
private void displayIssues(Issue[] issues, String proposedName)
{
    StringBuffer issueMessages = new StringBuffer();
    for (int i = 0; i < issues.length; i++)
    {
        issueMessages.append(issues[i].getDescription() + "\n");
        if (issues[i].getPrescription() != null)
        {
            issueMessages.append(issues[i].getPrescription() + "\n");
        }
    }
    DialogUtil.showErrorDialog(issueMessages.toString());
}

/**
 * Enables the Create button if there are values in both the
 * variable and JCX name boxes; otherwise, disables the Create
 * button.
 */
private void checkForEntries()
{
    if (!m_txtNewJcx.getText().equals("") && !m_txtVariableName.getText().equals(""))
    {
        m_btnCreate.setEnabled(true);
    }
    else
    {
        m_btnCreate.setEnabled(false);
    }
}

/**
 * Evaluates the contents of the dialog's text boxes,
 * validating those boxes' values.
 *
 * @param componentName The name of the UI component that
 * received an action -- ie, the Create or Cancel button.
 */
private void evaluateDialog(String componentName)
{
    /**
     * If the user presses return when focus is on the create button,

```

navExtensionDevKit.html Sample

```
* variable name box, or either of the JCX name boxes, do the same
* thing: check that there are allowable variable names and
* JCX names, and handle errors accordingly.
*/
if (componentName.equals("create_button"))
{
    /*
    * If the second tab is visible, validate the input in the
    * variable and JCX name boxes.
    */
    if (m_pnlTab2.isVisible())
    {
        /*
        * If there's text in the variable name box,
        * assign that text value to a variable. If there's no text,
        * prompt the user to enter some. Also, set a boolean
        * variable to false so it can be checked along with the
        * presence of a JCX name value later.
        */
        if (!m_txtVariableName.getText().equals(""))
        {
            m_variableName = m_txtVariableName.getText();
        }
        else if (m_txtVariableName.getText().equals(""))
        {
            m_variableNameIsGood = false;
            DialogUtil.showErrorDialog("Please specify a variable name.");
        }
        /*
        * If there's text in the JCX name box,
        * assign that text value to a variable. If there's no text,
        * prompt the user to enter some. Also, set a boolean
        * variable to false so it can be checked along with the
        * presence of a variable name value later.
        */
        if (!m_txtNewJcx.getText().equals(""))
        {
            m_jcxName = m_txtNewJcx.getText();
        }
        else if (m_txtNewJcx.getText().equals(""))
        {
            m_jcxNameIsGood = false;
            DialogUtil.showErrorDialog("Please specify a JCX name.");
        }
        /*
        * If both boxes have values, validate those values for use as
        * Java code in WebLogic Workshop. If there are issues,
        * report the issues to the user.
        */
        if (!m_jcxName.equals("") && !m_variableName.equals(""))
        {
            Issue[] jcxIssues = m_wizard.getNameValidator().validateExtensionName(m_jcxName);
            Issue[] variableIssues = m_wizard.getNameValidator().validateInstanceName(m_variableName);
            if (jcxIssues != null)
            {
                m_jcxNameIsGood = false;
                displayIssues(jcxIssues, m_jcxName);
            }
            if (variableIssues != null)
            {

```

navExtensionDevKit.html Sample

```

        m_variableNameIsGood = false;
        displayIssues(variableIssues, m_variableName);
    }
    if (variableIssues == null)
    {
        m_variableNameIsGood = true;
    }
    if (jcxIssues == null)
    {
        m_jcxNameIsGood = true;
    }
    /*
     * After validation, if both the variable and JCX name boxes have
     * valid values, signal that it's okay to insert the control
     * and close the dialog.
     */
    if (m_jcxNameIsGood && m_variableNameIsGood)
    {
        m_controlInsertRequested = true;
        closeDialog(null);
    }
}

/*
 * The "Cancel" button needs to be handled as "OK" in one
 * circumstance.
 */
if (componentName.equals("cancel_button"))
{
    /*
     * If CONFIG_CREATE_EXTENSION_FILE is the requested configuration,
     * it means that the users reached this dialog through File -> New -> Java Control
     * (or a related command). They've already named the control, so we
     * change the Cancel button to an OK button (see code in initComponents), set
     * "requested" to true, and close the dialog.
     */
    if (m_wizard.m_requestedConfig == CustomInsertWizard.CONFIG_CREATE_EXTENSION_FILE)
    {
        m_controlInsertRequested = true;
        closeDialog(null);
    }
    /*
     * For other configurations, we presented a cancel button and
     * set "requested" to false when it is clicked.
     */
    else
    {
        m_controlInsertRequested = false;
        closeDialog(null);
    }
}

/**
 * A m_focusListener implementation simply to present messages about
 * how different pieces of the dialog user interface contribute to the
 * control's insertion.
 */

```


navExtensionDevKit.html Sample

```
* Note that it can be it can be difficult to debug a focus listener with
* WebLogic Workshop's debugger. This is because a breakpoint set in the
* listener will immediately be hit again when you attempt to switch back
* to the instance launched when you started debugging. Instead of using
* the debugger, you can print messages to the Output pane using the
* message service, in this way:
*
*      MessageSvc.get().debugLog("My debugging message.");
*
* This is an alternative to using System.out, which is unavailable when
* a console window is not displayed.
*/
private class DialogFocusListener implements FocusListener
{
    /**
     * Received when a component that has added this listener receives
     * focus. This event handler is used to print messages to the
     * dialog's message window as the variable and JCX name boxes
     * receive focus.
     *
     * @param event The details of the event.
     */
    public void focusGained(FocusEvent event)
    {
        String componentName = ((JComponent)event.getSource()).getName();

        /**
         * If the component that received focus is a text field, then it
         * must be the variable or JCX name box. Print a descriptive
         * message.
         */
        if (event.getSource() instanceof JTextField)
        {
            if (componentName.equals("variable_name_box"))
            {
                String message = "Control variable name: \n" +
                    "The name entered here is stored in a dialog " +
                    "variable. When the Create button is clicked, the value is " +
                    "checked and validated within this dialog. If the " +
                    "value is valid, the control wizard retrieves the value from " +
                    "from this dialog and holds it for retrieval by the IDE through " +
                    "the wizard's getInstanceName method.\n";
                printActionMessage(message);
            }
            if (componentName.equals("new_jcx_name_box"))
            {
                String message = "New JCX name: \n" +
                    "The name entered here is stored in a dialog " +
                    "variable. When the Create button is clicked, the value is " +
                    "checked and validated within this dialog. If the " +
                    "value is valid, the control wizard retrieves the value from " +
                    "from this dialog and holds it for retrieval by the IDE through " +
                    "the wizard's getExtensionName method.\n";
                printActionMessage(message);
            }
        }
    }

    /**
     * Not implemented. This dialog doesn't do anything special when focus
     * leaves a component.
     */
}
```

navExtensionDevKit.html Sample

```

    */
    public void focusLost(FocusEvent arg0){}
}

/**
 * Handles hyperlinks events within the JEditorPane component
 * in the "Introduction" pane of the dialog. This
 * code launches a browser to display the HTML page at the other
 * end of the clicked link.
 */
private class DialogLinkListener implements HyperlinkListener
{
    /**
     * Handles a hyperlink click.
     *
     * @param event The details of the event.
     */
    public void hyperlinkUpdate(HyperlinkEvent event)
    {
        if (event.getEventType() == HyperlinkEvent.EventType.ACTIVATED)
        {
            // Use the browser service to display the page.
            BrowserSvc.get().invokeBrowser(event.getURL(), true);
        }
    }
}

/**
 * An event listener to listen for mouse actions in this dialog.
 * In particular, if the event is a mouse click on one of the dialog's
 * tabs, this listener enables or disables the "Create" button. If
 * the mouse click is on the first tab ("Introduction"), the button is disabled. It would
 * be counterintuitive to allow the user to click Create from the first tab, which
 * does not prompt for information needed to create the control.
 */
private class DialogMouseListener implements MouseListener
{
    /**
     * Handles a mouse click event.
     *
     * @param event The click event.
     */
    public void mouseClicked(MouseEvent event)
    {
        /**
         * If a tab was clicked, enable or disable the Create button to
         * based on which tab was clicked and whether there are values
         * in the variable and JCX name boxes.
         */
        if (event.getSource() instanceof JTabbedPane)
        {
            /*
             * If the first tab was clicked, disable the Create button.
             */
            if (((JTabbedPane)event.getSource()).getSelectedIndex() == 0)
            {
                m_btnCreate.setEnabled(false);
            }
            /*
             * If the second tab was clicked, call a function that finds out
             * if there's text in the variable and JCX name boxes, and

```

navExtensionDevKit.html Sample

```

        * enables or disables the Create button accordingly.
        */
        else if (((JTabbedPane)event.getSource()).getSelectedIndex() == 1)
        {
            checkForEntries();
        }
    }
    /*
    * If a the Create or Cancel button was clicked, handle it with the
    * evaluateDialog function.
    */
    else if (event.getSource() instanceof JButton)
    {
        String componentName = ((JComponent)event.getSource()).getName();
        evaluateDialog(componentName);
    }
}

// These methods aren't needed in this dialog.
public void mouseEntered(MouseEvent event){}
public void mouseExited(MouseEvent event){}
public void mousePressed(MouseEvent event){}
public void mouseReleased(MouseEvent event){}
}

/**
 * A listener to handle ENTER and ESC keystrokes. A key event dispatcher
 * handles keystrokes before they can be consumed by other components in the
 * user interface.
 */
private class DialogKeyEventDispatcher implements KeyEventDispatcher
{
    /**
     * Receives key strokes sent to the dialog, handling only the ENTER
     * and ESC keys.
     *
     * @param event The details of the event.
     */
    public boolean dispatchKeyEvent(KeyEvent event)
    {
        /*
        * Handle an ENTER keystroke.
        */
        if (event.getKeyCode() == KeyEvent.VK_ENTER)
        {
            /*
            * If the Create button is enabled, then the second tab must be
            * visible and there are values in the variable and JCX name
            * boxes. Do the same name value checking that would be done if the
            * Create button had been clicked.
            */
            if (m_btnCreate.isEnabled())
            {
                String componentName = ((JComponent)event.getSource()).getName();
                evaluateDialog("create_button");
            }
            /*
            * If the current configuration specifies creating a new JCX file, then
            * the user reached this dialog through File -> New -> Java Control. Just
            * signal that the control may be added, and close the dialog.
            */

```

navExtensionDevKit.html Sample

```
        if (m_wizard.m_requestedConfig == ControlWizard.CONFIG_CREATE_EXTENSION_FILE)
        {
            m_controlInsertRequested = true;
            closeDialog(null);
        }
    }
    /*
    * Handle an ESC keystroke. Signal that the control may not be inserted
    * and close the dialog.
    */
    if (event.getKeyCode() == KeyEvent.VK_ESCAPE)
    {
        m_controlInsertRequested = false;
        closeDialog(null);
    }
    return false;
}

/**
 * A listener to handle keystrokes in the variable and JCX name
 * text boxes. All three methods of this class simply call a function
 * that enables or disables the Create button depending on whether
 * there's text in the boxes.
 */
private class DialogDocumentListener implements DocumentListener
{
    public void changedUpdate(DocumentEvent event)
    {
        checkForEntries();
    }
    public void insertUpdate(DocumentEvent event)
    {
        checkForEntries();
    }
    public void removeUpdate(DocumentEvent event)
    {
        checkForEntries();
    }
}
}
```

CustomInsertWizard.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizardCustom/i

Sample Source Code

```
package insertWizardCustom.ide;

import com.bea.ide.Application;
import com.bea.ide.control.ControlWizard;
import com.bea.ide.control.EditorContext;
import com.bea.ide.filesystem.FileSvc;
import com.bea.ide.filesystem.IFile;
import com.bea.ide.ui.IURISelectionContext;
import com.bea.ide.util.URIUtil;
import com.bea.ide.util.swing.DialogUtil;
import com.bea.ide.workspace.WorkspaceSvc;
import java.awt.Dialog;
import java.awt.Frame;
import java.io.File;
import java.net.URI;
import java.net.URL;
import java.text.MessageFormat;
import java.util.StringTokenizer;
import javax.swing.JDialog;
import javax.swing.JFileChooser;
import javax.swing.JLabel;

/**
 * A control wizard implementation that returns a full dialog to prompt
 * the control's user for information when they are inserting or creating
 * the control. Contrast this with the insertWizard.ide.ServerCheckWizard
 * class in this application, which returns only a panel that will be
 * included in a dialog constructed by the IDE. The full dialog that
 * this implementation uses is defined as insertWizardCustom.ide.CustomInsertDialog.
 *
 * When implementing ControlWizard, you are responsible for prompting the
 * user for control instance and JCX names, and for validating the names that
 * the user specifies.
 *
 * Note that a ControlWizard implementation does not support allowing the user
 * to specify an existing JCX file -- a new file must always be created.
 */
public class CustomInsertWizard extends ControlWizard
{
    protected String m_variableName = null;
    protected String m_jcxName = null;
    protected String m_jcxPackageName = null;
    protected String m_currentProjectPath = null;
    protected URI m_activeDocumentUri = null;
    private EditorContext m_context = null;
}
```

navExtensionDevKit.html Sample

```
private ControlWizard.NameValidator m_validator = null;
protected boolean m_okayToInsert = false;
protected boolean m_nameHasBeenSet = false;
protected boolean m_packageHasBeenSet = false;
protected int m_requestedConfig = -1;
private Frame m_container = null;

public CustomInsertDialog m_insertDialog;

/**
 * Called by the IDE to get possible configurations.
 */
public int getConfigurationInfo()
{
    return (CONFIG_CREATE_EXTENSION_FILE | CONFIG_INSERT_INSTANCE);
}

/**
 * Called by the IDE to set the insert configuration
 * requested by the user.
 *
 * @param config A sum of the ControlWizardSimple configuration constants that
 * represent the user's insert request.
 */
public void setConfiguration(int config)
{
    m_requestedConfig = config;
}

/**
 * Called by the IDE to get the custom dialog to show.
 *
 * @return The insert dialog to display.
 */
public Dialog getDialog(Frame container)
{
    m_insertDialog = new CustomInsertDialog(container, this);
    m_container = container;
    return m_insertDialog;
}

/**
 * Called by the IDE to get the name of the interface defined
 * in the JCX.
 *
 * @return The control JCX and interface name.
 */
public String getExtensionName()
{
    m_jcxName = m_insertDialog.m_jcxName;
    return m_jcxName;
}

/**
 * Called by the IDE to get the control instance name specified by
 * the user in the insert dialog.
 *
 * @return The control instance name specified by the user.
 */
public String getInstanceName()
{

```

navExtensionDevKit.html Sample

```

    m_variableName = m_insertDialog.m_variableName;
    return m_variableName;
}

/**
 * Called by the IDE to set the package name. This value is
 * used in the JCX template's <code>package</code> statement.
 *
 * @param packageName The package name.
 */
public void setPackage(String packageName)
{
    if (!m_packageHasBeenSet)
    {
        if (m_activeDocumentUri != null)
        {
            String uriPath = m_activeDocumentUri.getPath();
            if (uriPath.startsWith("/"))
            {
                uriPath = uriPath.substring(uriPath.indexOf("/") + 1);
            }
        }
        if (packageName.endsWith("/"))
        {
            packageName = packageName.substring(0, packageName.lastIndexOf("/"));
        }
        if (packageName.endsWith(".jcx"))
        {
            packageName = packageName.substring(0, packageName.lastIndexOf("/"));
        }
        if (packageName.indexOf("/") > 0)
        {
            if (packageName.endsWith("/"))
            {
                packageName = packageName.substring(0, packageName.lastIndexOf("/"));
            }
            packageName = packageName.replaceAll("/", ".");
        }
        m_jcxPackageName = packageName;
        m_packageHasBeenSet = true;
    }
}

/**
 * Displays a yes/no dialog. Used to ask the user if they want to choose a folder
 * for their JCX.
 *
 * @return <code>true</code> if the user selected "Yes" in the dialog; <code>false</code>
 * if they selected "No".
 */
private boolean showPackageAlert()
{
    String alertText = "A Java control must reside in a subfolder of the " +
        "project. Do you want to choose a subfolder in " +
        "which to create this control?";
    return DialogUtil.showYesNoDialog(m_container, alertText);
}

/**
 * Called to set the name of the JCX file to create.
 */

```

navExtensionDevKit.html Sample

```
* @param name The name specified by the user.
*/
public void setName(String name)
{
    if (!m_nameHasBeenSet)
    {
        m_jcxName = name;
        m_nameHasBeenSet = true;
    }
}

/**
 * Called by the IDE to pass in a validator that can be used to validate
 * the names of control variable and JCX names.
 *
 * @param validator The validator instance.
 */
public void setNameValidator(ControlWizard.NameValidator validator)
{
    m_validator = validator;
}

/**
 * Gets the validator passed in by the IDE. This is called from the
 * insert dialog.
 *
 * @return The validator instance.
 */
public ControlWizard.NameValidator getNameValidator()
{
    return m_validator;
}

/**
 * Called by the IDE just before it inserts the control.
 *
 * @return <code>true</code> if the control may be inserted; otherwise, <code>false</code>.
 */
public boolean onFinish()
{
    if (m_insertDialog.m_controlInsertRequested)
    {
        if (!m_nameHasBeenSet)
        {
            m_jcxName = m_insertDialog.m_jcxName;
        }
        m_okayToInsert = true;
    }
    m_insertDialog.dispose();
    return m_okayToInsert;
}

/**
 * Called by the IDE to pass in an EditorContext instance that contains
 * information about, for example, the current project.
 *
 * @param context Details about the context into which this control
 * is being inserted.
 */
public void setContext(EditorContext context)
{

```


navExtensionDevKit.html Sample

```
m_context = context;
m_currentProjectPath = URIUtil.getFilePath(URIUtil.fromFile(m_context.getProjectDir()))
if (Application.getActiveDocument() != null)
{
    m_activeDocumentUri = Application.getActiveDocument().getURI();
}
}

/**
 * Provides a place for the IDE to retrieve the content that it
 * should insert into newly created JCX files. Here, the package name
 * and interface name passed in by the IDE are inserted into a
 * template using the MessageFormat class.
 *
 * @return The text of the JCX file to insert.
 */
public String getExtensionFileContent()
{
    String jcxTemplate = this.getTemplate();
    String jcxContent =
        MessageFormat.format(jcxTemplate, new Object[] {m_jcxPackageName, m_jcxName});
    return jcxContent;
}

/**
 * Returns a template for text to be inserted into a new JCX file.
 *
 * @return The template.
 */
private String getTemplate()
{
    String template =
        "package {0}; \n\n" +

        "public interface {1} extends com.bea.control.ControlExtension, insertWizardCustom."
        "'{' \n\n" +
        "    static final long serialVersionUID = 1L; \n\n" +
        "    \n\n" +
        "    public String echoInput(String input); \n\n" +
        "}' \n ";
    return template;
}
}
```

html Samples

This section contains source code for the following samples.

Samples Included in This Section

Introduction.html Sample

Introduction.html Sample

This topic includes the source code for the Introduction.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/insertWizardCustom/i

Sample Source Code

```
<html>
  <head>

<link href="workshop.css" rel="stylesheet" type="text/css">
</head>
  <body>

<h1>Custom Insert Wizard Sample Overview</h1>
<p><font size="2">This sample is an example of a fully-custom dialog for inserting
  a Java control. The control itself isn't functional. The interesting code is in
  the implementation of ControlWizard and the dialog that accompanies it.</font></p>
<p><font size="2">As you use the "Name the Control" tab of the wizard dialog, the dialog's
  bottom pane will display messages that describe how your choices impact what
  the wizard does.</font></p>
<p><font size="2">This wizard dialog uses several Swing components. For more information
  about these, see the following:</font></p>
<p><font size="2">Tabbed panes:</font></p>
<blockquote>
  <p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/components/tabbedp
    to Use Tabbed Panes</a></font></p>
</blockquote>
<p><font size="2">Panels that contain other components:</font></p>
<blockquote>
  <p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/components/panel.h
    to Use Panels</a></font></p>
</blockquote>
<p><font size="2">Buttons and radio buttons:</font></p>
<blockquote>
  <p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/components/button.
    to Use Buttons, Check Boxes, and Radio Buttons</a></font></p>
</blockquote>
<p><font size="2">Text boxes, editing boxes (such as the one on the Step 1 tab),
  and simple text areas (such as the one on the bottom of the wizard dialog:</font></p>
<blockquote>
  <p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/components/text.ht
    Text Components</a></font></p>
</blockquote>
<p><font size="2">Listening for events, such as when a UI component receives focus
  or gets clicked:</font></p>
<blockquote>
  <p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/events/">Lesson:
    Writing Event Listeners</a></font></p>
</blockquote>
<p><font size="2">Arranging user interface components.</font></p>
<blockquote>
```

navExtensionDevKit.html Sample

```
<p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/layout/gridbag.htm
</blockquote>
<blockquote>
  <p><font size="2"><a href="http://java.sun.com/docs/books/tutorial/uiswing/layout/visual.html
</blockquote>
<p>&nbsp;</p>
</body>
</html>
```

jcxCreat Samples

This section contains source code for the following samples.

Samples Included in This Section

ide Samples

testXML Samples

XQuery-tags.xml Sample

XQuery.java Sample

XQueryImpl.jcs Sample

XQueryUtil.java Sample

ide Samples

This section contains source code for the following samples.

Samples Included in This Section

QueryExpressionEditorSimple.java Sample

XQueryEditorSupport.java Sample

XQueryWizard.java Sample

QueryExpressionEditorSimple.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/ide/

Sample Source Code

```
package jcxCreate.ide;

import com.bea.ide.control.EditorSupport;
import java.awt.Component;
import java.awt.GridBagConstraints;
import java.awt.Insets;
import javax.swing.JFormattedTextField;
import com.bea.ide.control.AttributeEditorSimple;
import com.bea.control.Issue;

/*
 * Represent a dialog for editing the expression attribute. This
 * dialog is displayed when the control's user clicks the ... corresponding
 * to the attribute in the Property Editor, or double-clicks a method
 * in the JCX file.
 */
public class QueryExpressionEditorSimple extends javax.swing.JPanel
    implements AttributeEditorSimple
{

    javax.swing.JEditorPane edpQueryExpression = new javax.swing.JEditorPane();
    javax.swing.JLabel lblXQuery = new javax.swing.JLabel();
    javax.swing.JTextArea txtDescription = new javax.swing.JTextArea();

    /*
     * Call the initComponents method to assemble the dialog's
     * pieces. Receive the expression attribute original value
     * to display in the dialog.
     */
    public QueryExpressionEditorSimple(String origValue)
    {
        super();
        this initComponents(origValue);
    }

    public JFormattedTextField.AbstractFormatter getFormatter()
    {
        return null;
    }

    /*
     * Provides a way for WebLogic Workshop to retrieve the dialog's
     * components for display.
     */
    public Component getEditorComponent()
```

navExtensionDevKit.html Sample

```
{
    return this;
}

/*
 * Used by WebLogic Workshop to retrieve the new expression attribute
 * value. edpQueryExpression is the JEditorPane in the dialog box
 * that contains the expression as entered by the control's user.
 */
public String getNewAttributeValue()
{
    return edpQueryExpression.getText();
}

/*
 * Provides a place for code that should execute when the control's
 * user clicks OK in the expression editor dialog.
 */
public Issue[] onFinish()
{
    return null;
}

/*
 * Assembles components of the dialog.
 */
public void initComponents(String value) {
    java.awt.GridBagConstraints gridBagConstraints;
    setLayout(new java.awt.GridBagLayout());
    setEnabled(true);

    txtDescription.setText("Editing your XQuery expression in the box below.");
    txtDescription.setLineWrap(true);
    txtDescription.setBackground(null);
    txtDescription.setEditable(false);
    gridBagConstraints = new java.awt.GridBagConstraints(
        0, 0, 4, 1, 1, 0,
        GridBagConstraints.NORTH, GridBagConstraints.HORIZONTAL,
        new Insets(5, 5, 5, 5), 0, 0);
    add(txtDescription, gridBagConstraints);

    lblXQuery.setText("XQuery expression:");
    lblXQuery.setHorizontalAlignment(javax.swing.JLabel.RIGHT);
    gridBagConstraints = new GridBagConstraints(
        0, 1, 1, 1, 0, 0,
        GridBagConstraints.NORTHEAST, GridBagConstraints.NONE,
        new Insets(5, 5, 5, 5), 0, 0);
    add(lblXQuery, gridBagConstraints);

    edpQueryExpression.setText(value);
    gridBagConstraints = new GridBagConstraints(
        0, 2, 3, 1, 1, 1,
        GridBagConstraints.NORTH, GridBagConstraints.BOTH,
        new Insets(5, 0, 5, 5), 0, 0);
    add(edpQueryExpression, gridBagConstraints);
}
}
```


XQueryEditorSupport.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/ide/

Sample Source Code

```
package jcxCreate.ide;

import com.bea.ide.control.ControlAttribute;
import com.bea.ide.control.EditorSupport;
import com.bea.ide.control.ControlMethod;
import com.bea.ide.control.ControlBehaviorContext;
import com.bea.ide.control.DefaultEditorSupport;

/*
 * Represents IDE support for XQuery controls added to a project.
 * An "editor support" class specifies various "behaviors" within the IDE.
 * These include what to do when the user double-clicks on control methods,
 * when the user attempts to add or edit methods and callbacks, and what icon
 * to display on a method.
 */
public class XQueryEditorSupport extends DefaultEditorSupport
{
    /*
     * Provides a way for WebLogic Workshop to retrieve the behavior associated
     * with specified actions in the IDE. Each action has a corresponding
     * constant in the EditorSupport interface, which is implemented by the
     * DefaultEditorSupport class this class extends. A ControlBehaviorContext
     * argument specifies whether the context of the action is a method, attribute,
     * interface, instance, and so on.
     */
    public Object getBehavior(String behavior, ControlBehaviorContext context)
    {
        /*
         * If the user asks to edit a JCX method, return true.
         */
        if (behavior.equals(EditorSupport.BEHAVIOR_EDIT_METHOD))
        {
            return Boolean.TRUE;
        }
        /*
         * If the user asks to edit a callback, return false. XQuery JCX files do
         * not support callbacks.
         */
        else if (behavior.equals(EditorSupport.BEHAVIOR_EDIT_CALLBACK))
        {
            return Boolean.FALSE;
        }
        /*
         * If the user clicks an edit ... in the Property Editor, and if the attribute
         * corresponding to the ... is expression, return an instance of the expression
         */
    }
}
```

navExtensionDevKit.html Sample

```
    * edit dialog box.
    */
else if (behavior.equals(EditorSupport.BEHAVIOR_ATTRIBUTE_EDITOR))
{
    if (context instanceof ControlAttribute && ((ControlAttribute)context).getName().eq
    {
        return new QueryExpressionEditorSimple(((ControlAttribute)context).getValue());
    }
}
return super.getBehavior(behavior, context);
}
```

XQueryWizard.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/ide/

Sample Source Code

```
package jcxCreate.ide;

import java.text.MessageFormat;
import com.bea.ide.control.ControlWizardSimple;
import javax.swing.JComponent;

/*
 * Represents an insert dialog for the XQuery control. While
 * this dialog doesn't provide any special user interface, it
 * does provide the text that the IDE should insert into
 * newly created JCX files.
 */
public class XQueryWizard extends ControlWizardSimple
{
    private boolean m_createExtension = false;

    String m_jcxPackage = null;
    String m_jcxName = null;

    /*
     * Tell the IDE how to configure the insert dialog. This dialog
     * should provide an option to use an existing JCX file, or to
     * create a new one.
     */
    public int getConfigurationInfo()
    {
        return CONFIG_CREATE_EXTENSION_FILE |
            CONFIG_INSERT_INSTANCE;
    }

    /*
     * Provide a place for the IDE to tell this code whether the
     * user has asked to create a new JCX or not.
     */
    public void setConfiguration(int config)
    {
        m_createExtension =
            ((config & ControlWizardSimple.CONFIG_CREATE_EXTENSION_FILE) != 0);
    }

    /*
     * Tell the IDE what to use for custom insert dialog user interface.
     * None is needed here.
     */
    public JComponent getComponent()
```

navExtensionDevKit.html Sample

```
{
    return null;
}

/*
 * Provide a place for code that should execute when the user clicks the
 * Create button on the insert dialog.
 */
public boolean onFinish()
{
    if (super.onFinish() == false){
        return false;
    } else {
        return true;
    }
}

/*
 * Provides a way for the IDE to pass in the package name
 * that should be specified at the top of a new JCX file.
 */
public void setPackage(String packageName)
{
    m_jcxPackage = packageName;
}

/*
 * Provides a way for the IDE to pass in the name of the
 * interface defined in a new JCX file.
 */
public void setName(String name)
{
    m_jcxName = name;
}

/*
 * Provides a place for the IDE to retrieve the content that it
 * should insert into newly created JCX files. Here, the package name
 * and interface name passed in by the IDE are inserted into a
 * template using the MessageFormat class.
 */
public String getExtensionFileContent(){
    // did they ask for an extension?
    String jcxTemplate = this.getTemplate();

    String jcxContent =
        MessageFormat.format(jcxTemplate, new Object[] {m_jcxPackage, m_jcxName});

    return jcxContent;
}

/*
 * Returns a template for text to be inserted into a new JCX file.
 */
private String getTemplate(){
    String template =
        "package {0}; \n\n " +
        "import com.bea.control.*; \n " +
        "import com.bea.xml.XmlCursor; \n " +
        "import jcxCreate.XQuery; \n\n " +
```

navExtensionDevKit.html Sample

```
"public interface {1} extends XQuery, com.bea.control.ControlExtension \n " +
"'{' \n \n" +

"    /* \n" +
"        * A version number for this JCX. This will be incremented in new versions of
"        * this control to ensure that conversations for instances of earlier \n" +
"        * versions were invalid. \n" +
"        */ \n" +
"    static final long serialVersionUID = 1L; \n \n" +

"    /* Replace the following method with your own. Be sure \n" +
"        * that your method returns an XmlCursor and that its \n" +
"        * parameter is a File object to contain the XML against \n" +
"        * which to execute the XQuery expression. \n" +
"        */ \n \n" +

"    /** \n" +
"        * @jc:query expression=\"$this/purchase-order/line-item[price <= 20.00]\" \n" +
"        */ \n" +
"    XmlCursor selectLineItem(String filePath); \n \n" +

"    }' \n ";
return template;
}

}
```

testXML Samples

This section contains source code for the following samples.

Samples Included in This Section

Po.xml Sample

Po.xml Sample

This topic includes the source code for the Po.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/testXML/

Sample Source Code

```
<purchase-order>
  <customer>
    <name>Gladys Cravits</name>
    <address>Anytown, PA</address>
  </customer>
  <date>2001-12-17T09:30:47-05:00</date>
  <line-item>
    <description>Burnham's Celestial Handbook, Vol 1</description>
    <per-unit-ounces>5</per-unit-ounces>
    <price>21.79</price>
    <quantity>2</quantity>
  </line-item>
  <line-item>
    <description>Burnham's Celestial Handbook, Vol 2</description>
    <per-unit-ounces>5</per-unit-ounces>
    <price>19.89</price>
    <quantity>2</quantity>
  </line-item>
  <line-item>
    <description>Burnham's Celestial Handbook, Vol 3</description>
    <per-unit-ounces>5</per-unit-ounces>
    <price>19.89</price>
    <quantity>1</quantity>
  </line-item>
  <shipper>
    <name>UPS</name>
    <per-ounce-rate>0.74</per-ounce-rate>
  </shipper>
</purchase-order>
```

XQuery-tags.xml Sample

This topic includes the source code for the XQuery-tags.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/

Sample Source Code

```
<?xml version="1.0"?>
<control-tags xmlns="http://www.bea.com/2003/03/controls/" editor-class="jcxCreate.ide.XQueryEd">
  <method-tag name="query" method-location="interface-method">
    <description>Information about the XQuery.</description>
    <attribute name="expression" required="true">
      <description>The XQuery expression.</description>
      <type>
        <text/>
      </type>
    </attribute>
  </method-tag>
</control-tags>
```


XQuery.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/

Sample Source Code

```
package jcxCreate;

import com.bea.control.*;

public interface XQuery extends Control
{
}
```

XQueryImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/

Sample Source Code

```
package jcxCreate;

import com.bea.control.*;
import com.bea.xml.XmlCursor;
import com.bea.xml.XmlObject;
import java.lang.reflect.Method;
import java.lang.reflect.InvocationTargetException;
import java.io.File;
import java.io.InputStream;
import java.io.FileInputStream;

/**
 * The XQuery control illustrates how to create a control
 * that generates a JCX file. A JCX file enables the control's
 * user to customize the control's interface.
 *
 * This control also includes a dialog for editing an attribute
 * containing the XQuery expression. Each method of the control's
 * extended interface is annotation with this attribute.
 *
 * @jcs:control-tags file="XQuery-tags.xml"
 * @jcs:jc-jar label="XQuery"
 *     insert-wizard-class="jcxCreate.ide.XQueryWizard"
 *     version="0.8"
 *     icon-16="/images/xml.gif"
 *     palette-priority="5"
 *     group-name="Feature Sample Controls"
 *     description="Extensible control that creates a jcx and runs an XQuery."
 *     @editor-info:code-gen control-interface="true"
 */
public class XQueryImpl implements XQuery, com.bea.control.Extensible, com.bea.control.Controls
{
    /**
     * @common:context
     */
    com.bea.control.ControlContext context;

    /**
     * A String to hold the XQuery expression specified by the control's
     * user.
     */
    String m_expression = null;

    /**
     * Constants representing annotation and attribute names. The
```

navExtensionDevKit.html Sample

```
* jc:query annotation will be written into the JCX file for
* each XQuery expression the control's user associates with a
* control method.
*/
    public static final String TAG_XQUERY = "jc:query";
    public static final String ATTR_EXPRESSION = "expression";

/*
* The invoke method is implemented from the Extensible interface.
* It is a key aspect of any control that provides a customizable
* interface. When a method on a control's JCX file is called, that
* call is delegated to the invoke method. The name and arguments for
* the JCX method called are passed to invoke as arguments. Within
* the invoke method, you extract and process the method and arguments,
* returning a value if appropriate.
*
* A JCX file for the XQuery control would contain methods whose argument
* is an object containing the XML on which to execute the query. An
* annotation on that method would specify the XQuery expression to use.
*/
public Object invoke(Method method, Object[] args) throws Throwable {
    XmlCursor resultCursor;
    try{
        // Create an instance of the class designed to handle the XQuery.
        XQueryUtil queryUtil = new XQueryUtil();

        // Retrieve the XQuery expression from the annotation in the JCX.
        m_expression = context.getMethodAttribute(TAG_XQUERY, ATTR_EXPRESSION);

        // Get the XML Stream named by the first parameter.
        // First look to see if it is a resource in the jar that will be created from this
        // Any non-source files in this project will be moved to the jar by default
        String filePath = new String().valueOf(args[0]);
        InputStream iXml= null;
        if (filePath.substring(0,1).equals("/"))
            iXml= this.getClass().getClassLoader().getResourceAsStream(filePath);
        // try as a file
        if (iXml==null)
        {
            File f=new File(filePath);
            if (f.exists())
                iXml = new FileInputStream(f);
            else
            {
                throw new ControlException("file not found at " + f.getAbsolutePath());
            }
        }
        // Load the XML using the XMLBeans API.
        XmlObject instanceDoc = XmlObject.Factory.parse(iXml);

        /*
        * Pass the value to the XQuery utility class, and return the results
        * in an XMLBeans XmlCursor object.
        */
        resultCursor = queryUtil.runXQueryExpression(instanceDoc, m_expression);
        return resultCursor;
    } catch (InvocationTargetException ite) {
        throw new ControlException("Error while executing the query: " + ite.getMessage(),
    } catch (Exception e) {
        throw new ControlException("XQueryControl: Exception while executing expression.",
    }
}
```

navExtensionDevKit.html Sample

```
}

/*
 * Implement onException for any unhandled exceptions.
 */
public void context_onException(Exception e, String methodName, Object [] args){
    throw new ControlException("XQuery control exception in " + methodName + ": ", e);
}
}
```

XQueryUtil.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/jcxCreate/

Sample Source Code

```
package jcxCreate;

import com.bea.xml.XmlCursor;
import java.lang.reflect.InvocationTargetException;
import com.bea.xml.XmlObject;

/*
 * Represents XQuery functionality to the XQuery control.
 */
public class XQueryUtil
{
    /*
     * Returns an XMLBeans XmlCursor object containing the results of
     * the query. If results are not possible, perhaps because the expression
     * was incorrect, the method throws an exception.
     */
    public XmlCursor runXQueryExpression(XmlObject xml, String expression)
        throws InvocationTargetException, Exception
    {
        XmlCursor queryResults = null;
        try{
            XmlCursor itemCursor = xml.newCursor().execQuery(expression);
            queryResults = itemCursor;
        } catch (Exception e){
            throw e;
        }
        return queryResults;
    }
}
```

propEditor Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomerData-tags.xml Sample

CustomerData.java Sample

CustomerDataImpl.jcs Sample

CustomerDBControl.jcx Sample

ide Samples

CustomerData-tags.xml Sample

This topic includes the source code for the CustomerData-tags.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/

Sample Source Code

```
<?xml version="1.0"?>
<control-tags xmlns="http://www.bea.com/2003/03/controls/" editor-class="propEditor.ide.Custome
  <control-tag name="customer-db" >
    <description>Information needed to retrieve customer data.</description>
    <attribute name="customer-id" required="true">
      <description>Unique identifier for the customer in the database.</description>
      <type>
        <custom name="custidType" class-name="propEditor.ide.CustIdValidator"/>
      </type>
    </attribute>
  </control-tag>
</control-tags>
```

CustomerData.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/

Sample Source Code

```
package propEditor;

import com.bea.control.*;

public interface CustomerData extends Control
{
    /**
     * Returns the customer's company name using the nested Database control.
     * @common:operation
     */
    java.lang.String getCustomerName();

    /**
     * Returns a list of items the customer has ordered. The database includes tables for custo
     * @common:operation
     */
    java.util.ArrayList getItemsOrdered();
}
```


CustomerDataImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/

Sample Source Code

```
package propEditor;

import com.bea.control.*;
import java.util.HashMap;
import java.util.ArrayList;
import java.sql.ResultSet;
import java.sql.SQLException;

/**
 * The CustomerData control illustrates how to connect a custom attribute
 * editing/validation dialog. The dialog ensures that the customer-id
 * attribute is a six-digit number.
 *
 * After specifying a value for the customer-id attribute, you can use
 * the control's method's to get information about the customer.
 *
 * @jcs:control-tags file="CustomerData-tags.xml"
 * @jcs:jc-jar label="CustomerData"
 *     version="0.8"
 *     icon-16="/images/hello_16.gif"
 *     icon-32="/images/hello_32.gif"
 *     palette-priority="6"
 *     group-name="Feature Sample Controls"
 *     description="Illustrates custom property editor"
 *     @editor-info:code-gen control-interface="true"
 */
public class CustomerDataImpl implements CustomerData, com.bea.control.ControlSource
{
    /**
     * @common:control
     */
    private propEditor.CustomerDBControl customerDB;

    private String m_custIDProp;
    private int m_custIDInt;

    /**
     * @common:context
     */
    com.bea.control.ControlContext context;

    /**
     * Use the onCreate callback handler to retrieve the attribute value set
     * by the control's user. With the customer-id attribute stored internally,
     * it can be used by other control methods. Remember that the onCreate callback

```

navExtensionDevKit.html Sample

```
* is received when the control is created, before any control methods are called.
*/
public void context_onCreate() {
    m_custIDProp = context.getControlAttribute("jc:customer-db", "customer-id");
    m_custIDInt = new Integer(m_custIDProp).intValue();
}

/**
 * Returns the customer's company name using the nested Database control.
 *
 * @common:operation
 */
public String getCustomerName()
{
    String custIDProp = context.getControlAttribute("jc:customer-db", "customer-id");
    int custIDInt = new Integer(custIDProp).intValue();
    String customerName = customerDB.selectCustomerName(custIDInt);
    return customerName;
}

/**
 * Returns a list of items the customer has ordered. The database includes
 * tables for customers, items, items by order, and customers by order.
 * This is method nests queries against these tables to produce joined
 * results.
 *
 * @common:operation
 */
public ArrayList getItemsOrdered()
{
    ArrayList itemNamesArray = new ArrayList();
    ResultSet orderIDs = customerDB.selectOrdersByCustomer(m_custIDInt);
    try{
        while (orderIDs.next()){
            int orderID = orderIDs.getInt("orderid");
            ResultSet itemNumbers = customerDB.selectItemNumbersByOrder(orderID);
            while (itemNumbers.next()){
                int itemNumber = itemNumbers.getInt("itemnumber");
                ResultSet itemNames = customerDB.selectItemNameByItemNumber(itemNumber);
                while (itemNames.next()){
                    itemNamesArray.add(itemNames.getString("itemname"));
                }
            }
        }
        return itemNamesArray;
    } catch(SQLException se){
        throw new ControlException("Error while getting items ordered.", se);
    }
}
}
```

CustomerDBControl.jcx Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/

Sample Source Code

```
package propEditor;

import com.bea.control.*;
import java.sql.SQLException;
import java.sql.ResultSet;

/**
 * @jc:connection data-source-jndi-name="cgSampleDataSource"
 * @jcs:jc-jar display-in-palette="false"
 */
public interface CustomerDBControl extends com.bea.control.ControlExtension, DatabaseControl
{
    /**
     * @jc:sql statement="SELECT customername FROM customers WHERE customerid={customerID}"
     */
    String selectCustomerName(int customerID);

    /**
     * @jc:sql statement="SELECT itemnumber FROM po_items WHERE orderid={orderID}" array-max-le
     */
    ResultSet selectItemNumbersByOrder(int orderID);

    /**
     * @jc:sql statement="SELECT itemname FROM items WHERE itemnumber={itemNumber}" array-max-l
     */
    ResultSet selectItemNameByItemNumber(int itemNumber);

    /**
     * @jc:sql statement="SELECT orderid FROM po_customers WHERE customerid={customerID}" array
     */
    ResultSet selectOrdersByCustomer(int customerID);
}
```

ide Samples

This section contains source code for the following samples.

Samples Included in This Section

CustIdValidator.java Sample

CustomerDataEditorSupport.java Sample

CustomerIdEditorSimple.java Sample

CustIdValidator.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/ide/

Sample Source Code

```
package propEditor.ide;

import com.bea.control.Issue;
import com.bea.control.ValidateAttribute;
import java.util.Map;

public class CustIdValidator implements ValidateAttribute
{
    int legalValues[] = { 987654, 987655, 987658, 987659 };

    static class CustIdIssue implements Issue
    {
        String _message;
        private CustIdIssue(String message) { _message = message; }
        public boolean isError() { return true; }
        public String getDescription() { return _message; }
        public String getPrescription()
        {
            return "Provide one of the following values: " +
                "987654, 987655, 987658, 987659. that's all";
        }
    }

    // called by the attribute editor.
    public Issue[] validateId(String value)
    {
        return validateDuringCompile(null, value, null);
    }

    public Issue[] validateDuringCompile(String attributeType, String value, Map context)
    {
        Issue[] issues = new Issue[1];
        int val;
        try
        {
            val = Integer.parseInt(value);
        }
        catch(Exception e)
        {
            issues[0] = new CustIdIssue(e.getMessage());
            return issues;
        }

        for(int i = 0; i < legalValues.length; i++)
```

navExtensionDevKit.html Sample

```
{
    if(legalValues[i] == val)
    {
        return null;
    }
}

issues[0] = new CustIdIssue("unsupported value error");
return issues;
}

public Issue[] validateDuringEdit(String attributeType, String value)
{
    return null;
}
}
```

CustomerDataEditorSupport.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/ide/

Sample Source Code

```
package propEditor.ide;

import com.bea.ide.control.ControlBehaviorContext;
import com.bea.ide.control.EditorSupport;
import com.bea.ide.control.ControlAttribute;
import com.bea.ide.control.DefaultEditorSupport;

/*
 * Represents support for actions in the IDE. In particular, this
 * class provides code that executes when the user clicks the ... in the
 * Property Editor to edit the customer-id attribute.
 */
public class CustomerDataEditorSupport extends DefaultEditorSupport
{
    public Object getBehavior(String behavior, ControlBehaviorContext ctx)
    {
        // if both editor and validator are specified only editor is used
        // TODO: validator example.
        if (behavior.equals(EditorSupport.BEHAVIOR_ATTRIBUTE_EDITOR))
        {
            if (ctx instanceof ControlAttribute &&
                ((ControlAttribute)ctx).getName().equals("customer-id"))
            {
                return new CustomerIdEditorSimple(((ControlAttribute)ctx).getValue());
            }
        }
        return super.getBehavior(behavior, ctx);
    }
}
```

CustomerIdEditorSimple.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlFeatures/propEditor/ide/

Sample Source Code

```
package propEditor.ide;

import java.util.StringTokenizer;
import java.awt.BorderLayout;
import java.awt.Component;
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JFormattedTextField;
import javax.swing.JPanel;
import com.bea.ide.control.EditorSupport;
import javax.swing.JTextPane;
import com.bea.ide.control.AttributeEditorSimple;
import com.bea.control.Issue;
import com.bea.control.DefaultIssue;

/*
 * Represents an attribute editing/validation dialog for the
 * customer-id attribute.
 */
public class CustomerIdEditorSimple extends JPanel implements AttributeEditorSimple
{
    private JTextField m_field;

    /*
     * Constructs this dialog using the existing value for
     * the customer-id attribute.
     */
    public CustomerIdEditorSimple(String origValue)
    {
        super(new BorderLayout());
        m_field = new JTextField(origValue);
        String messageText = "This sample supports the following values: \n" +
            "987654      987655 \n" +
            "987658      987659 \n"
            +"that's all \n";

        JTextPane valueMessage = new JTextPane();
        valueMessage.setText(messageText);
        valueMessage.setBackground(null);
        valueMessage.setEditable(false);
        this.add(m_field, BorderLayout.SOUTH);
        this.add(valueMessage, BorderLayout.NORTH);
    }

    public JFormattedTextField.AbstractFormatter getFormatter()
```


navExtensionDevKit.html Sample

```
{
    return null;
}

/*
 * Returns the component to use for the customer-id
 * editing dialog. The IDE calls this method to display
 * the dialog.
 */
public Component getEditorComponent()
{
    return this;
}

/*
 * Provides a way for the IDE to retrieve the newly
 * entered attribute value.
 */
public String getNewAttributeValue()
{
    return m_field.getText();
}

/*
 * Provides a place for code that should execute when the
 * user clicks OK. This method calls code that validates the
 * attribute value, ensuring that it is a six-digit numeric
 * value.
 */
public Issue[] onFinish()
{
    CustIdValidator cIdV = new CustIdValidator();
    return cIdV.validateId(getNewAttributeValue());
}

/*
 * Removes any nonnumeric characters in the new attribute value.
 */
private String removeNonNumeric(String stringNumber) {
    StringBuffer delimiters = new StringBuffer();
    StringBuffer cleanString = new StringBuffer();

    for (int i = 0; i < stringNumber.length(); i++) {
        if (stringNumber.charAt(i) < '0' || stringNumber.charAt(i) > '9')
            delimiters.append(stringNumber.charAt(i));
    }
    StringTokenizer tokens = new StringTokenizer(stringNumber, delimiters.toString());

    while(tokens.hasMoreTokens()) {
        cleanString.append(tokens.nextToken());
    }
    return cleanString.toString();
}
}
```

ControlTest Samples

This section contains source code for the following samples.

Samples Included in This Section

featuresTests Samples

featuresTests Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomerDataTest.jws Sample

EventRaiserTest.jws Sample

EventSchedulerTest.jws Sample

JcxTest.jws Sample

POXQueryControl.jcx Sample

ServerCheckTest.jws Sample

CustomerDataTest.jws Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlTest/featuresTests/

Sample Source Code

```
package featuresTests;

import com.bea.control.JwsContext;
import java.util.HashMap;
import java.util.ArrayList;

/*
 * This web service provides a way to test the CustomerData control.
 * By specifying a customer ID as an attribute of that control,
 * you can use the control to query a database for information about
 * the customer.
 *
 * The control illustrates how to implement an attribute editing/validation
 * dialog. To see this control's dialog in use, click the control in
 * Design View, then click the ... for the customer-id attribute.
 */
public class CustomerDataTest implements com.bea.jws.WebService
{

    /**
     * @common:control
     * @jc:customer-db customer-id="987659"
     */
    private propEditor.CustomerData thisCustomer;

    /**
     * Returns the customers name.
     *
     * @common:operation
     */
    public String getCustName()
    {
        return thisCustomer.getCustomerName();
    }

    /**
     * Returns the list of items the customer has purchased.
     *
     * @common:operation
     */
    public ArrayList getItemsPurchased()
    {
        ArrayList results = new ArrayList();
        ArrayList items = thisCustomer.getItemsOrdered();
        return items;
    }
}
```

```
}    }
```

EventRaiserTest.jws Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlTest/featuresTests/

Sample Source Code

```
package featuresTests;

import com.bea.control.JwsContext;
import java.util.Date;

/*
 * This web service uses the EventRaiser control. That control is designed
 * to simply forward the onTimeout event it receives from a Timer control.
 *
 * To test the control, run the web service. When Test View appears, click
 * the getStarted button, then click the refresh button until the response
 * appears. The control is designed to respond after 5 seconds.
 */
public class EventRaiserTest implements com.bea.jws.WebService
{
    public Callback callback;

    /** @common:context */
    JwsContext context;

    /**
     * @common:control
     */
    private controlEvents.EventRaiser eventRaiserControl;

    /**
     * This method tells the EventRaiser control to start its timer.
     *
     * @common:operation
     * @jws:conversation phase="start"
     */
    public void getStarted()
    {
        eventRaiserControl.startTimer();
    }

    /**
     * This code handles a callback exposed by the EventRaiser control.
     * The onTimeout fired by EventRaiser's Timer control is simply
     * forwarded through this callback. The value received by the
     * callback is wrapped in a Date object and converted to a String
     * for display in Test View.
     */
    public void eventRaiserControl_backEndTimer_onTimeout(long arg0)
    {

```

navExtensionDevKit.html Sample

```
        callback.onDone(new Date(arg0).toLocaleString());
    }

    public interface Callback
    {
        /**
         * @jws:conversation phase="finish"
         */
        void onDone(String response);
    }
}
```

EventSchedulerTest.jws Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlTest/featuresTests/

Sample Source Code

```
package featuresTests;

import com.bea.control.JwsContext;
import java.util.Date;

/*
 * This web service uses the EventScheduler control to request notification
 * after a specified interval. This service pass the interval from the client
 * to the control, which in turn schedules a callback based on the interval.
 */
public class EventSchedulerTest implements com.bea.jws.WebService
{
    public Callback callback;

    /**
     * @common:control
     */
    private controlEvents.EventScheduler scheduleControl;

    /** @common:context */
    JwsContext context;

    /**
     * The setWakeUpCall method provides a place for clients to specify the time
     * after which they would like to receive the call.
     *
     * To test control, run the web service. Enter a value in milliseconds
     * as the intervalInMilliseconds parameter (eg, 5000 for 5 seconds).
     * Click the setWakeUpCall button, then click Refresh until a response
     * is displayed.
     *
     * @common:operation
     * @jws:conversation phase="start"
     */
    public void setWakeUpCall(long intervalInMilliseconds)
    {
        scheduleControl.setAlarm(intervalInMilliseconds);
    }

    /**
     * This callback handler receives the alarm notification from the
     * EventScheduler control. It packages the time value received
     * into a message, then sends the message to its client through
     * its onTimeToWakeUp callback.
     */
}
```


navExtensionDevKit.html Sample

```
public void scheduleControl_onAlarmRingin(java.lang.String responseMessage, long time)
{
    callback.onTimeToWakeUp(responseMessage + " You scheduled a wake-up call for " +
        new Date(time).toLocaleString() + ". It is now " + new Date().toLocaleString());
}

/**
 * The cancelWakeUp method provides a way for clients to cancel
 * the wake up call. This calls an EventScheduler method that
 * cancels the scheduled callback.
 *
 * @common:operation
 * @jws:conversation phase="finish"
 */
public void cancelWakeUp()
{
    scheduleControl.cancelAlarm();
}

/**
 * The onTimeToWakeUp callback notifies clients that the interval has
 * passed.
 */
public interface Callback
{
    /**
     * @jws:conversation phase="finish"
     */
    void onTimeToWakeUp(String response);
}
}
```

JcxTest.jws Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlTest/featuresTests/

Sample Source Code

```
package featuresTests;

import com.bea.control.JwsContext;
import com.bea.control.ControlException;
import java.io.File;
import com.bea.xml.XmlCursor;
import com.bea.xml.XmlObject;
import java.io.InputStream;

public class JcxTest implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private featuresTests.POXQueryControl pOXQueryControl;

    /**
     * Executes an XQuery expression against the XML specified by the xmlFilePath argument.
     * The query expression itself is specified as an attribute value on the control
     * method called by this code.
     * <br/>
     * Note that for debugging purposes, control exceptions are not caught by this service.
     * This means that if there is a problem with the query expression, it will be
     * written to Test View.
     * <br/>
     * When testing with this method, the method's argument can point to a file
     * in the control's jar (use "/jcxCreate/testXML/po.xml" to find this). The
     * parameter can also give an absolute path a po.xml file included
     * in the testXML folder of this feature test project.
     *
     * @common:operation
     */
    public String getXMLData(String filePath)
    {
        String resultsXML = null;
        XmlCursor queryResults = pOXQueryControl.selectLineItem(filePath);
        resultsXML = queryResults.xmlText();
        return resultsXML;
    }
}
```

POXQueryControl.jcx Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlTest/featuresTests/

Sample Source Code

```
package featuresTests;

import com.bea.control.*;
import com.bea.xml.XmlCursor;
import jcxCreate.XQuery;

public interface POXQueryControl extends XQuery, com.bea.control.ControlExtension
{

    /* Replace the following method with your own. Be sure
     * that your method returns an XmlCursor and that its
     * parameter is a File object to contain the XML against
     * which to execute the XQuery expression.
     */

    /**
     * @jc:query expression="$input/purchase-order/line-item[price <= 20.00]"
     */
    XmlCursor selectLineItem(String filePath);

}
```

ServerCheckTest.jws Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/ControlTest/featuresTests/

Sample Source Code

```
package featuresTests;

import com.bea.control.JwsContext;

/*
 * This web service provides a way to test the ServerCheck control.
 * ServerCheck simply returns the domain name of the server specified
 * by its property attributes. In the case of this instance, that
 * server is the one used by WebLogic Workshop samples.
 *
 * The ServerCheck control illustrates how to connect a custom
 * control insert dialog box. To see that dialog in action, add a
 * new instance of the control to this web service.
 *
 * To test the control, run it, then click the getDomainName
 * button. Test View will refresh to display the domain name as a
 * response.
 */
public class ServerCheckTest implements com.bea.jws.WebService
{
    /**
     * @common:control
     * @jc:server-data password="weblogic" user-name="weblogic" url="http://localhost:7001" ser
     */
    private insertWizard.ServerCheck thisServerCheck;

    /**
     * @common:operation
     */
    public String getDomainName()
    {
        return thisServerCheck.getDomainName();
    }
}
```

DBScripter Samples

This section contains source code for the following samples.

Samples Included in This Section

dbScript Samples

docs Samples

dbScript Samples

This section contains source code for the following samples.

Samples Included in This Section

DBScriptRunner-tags.xml Sample

DBScriptRunner.java Sample

DBScriptRunnerImpl.jcs Sample

designUI Samples

scripts Samples

util Samples

DBScriptRunner-tags.xml Sample

This topic includes the source code for the DBScriptRunner-tags.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/

Sample Source Code

```
<?xml version="1.0" encoding="UTF-8"?>

<control-tags
xmlns="http://www.bea.com/2003/03/controls/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <control-tag name="dbscr-connection" multiple="false" allow-declaration-override="true">
    <description>Class-level attributes of datasource, set on JCX</description>
    <attribute name="data-source-jndi-name" required="false">
      <description/>
      <type>
        <text max-length="255" long="false"/>
      </type>
    </attribute>
    <attribute name="batch-size" required="false">
      <description>Sets the number of INSERT statements the control will batch</description>
      <type>
        <text/>
      </type>
      <default-value>1</default-value>
    </attribute>
  </control-tag>
  <method-tag name="dbscript" multiple="false" method-location="interface-method">
    <description>Tags on JCX methods, determine whether the method a) defines a data source</description>
    <attribute name="new-data-source-type" required="false">
      <description>Type of DataSource to create. Determines driver class. If present, the control will create the DataSource</description>
      <type>
        <text max-length="255" long="false"/>
      </type>
    </attribute>
    <attribute name="new-data-source-name" required="false">
      <description>Name of DataSource to create. </description>
      <type>
        <text max-length="255" long="false"/>
      </type>
    </attribute>
    <attribute name="data-source-url-defaults" required="false">
      <description>URL of DataSource to create, if not provided as method parameters.</description>
      <type>
        <text max-length="255" long="false"/>
      </type>
    </attribute>
    <attribute name="data-source-properties" required="false">
      <description>Properties of DataSource to create, if not provided as method parameters.</description>
      <type>
        <text max-length="1024" long="true"/>
      </type>
    </attribute>
  </method-tag>
</control-tags>
```

navExtensionDevKit.html Sample

```

        </type>
    </attribute>
    <attribute name="prompt-for-url" required="false">
        <description>Whether parts of URL for DataSource are provided as method parameter
        <type>
            <boolean/>
        </type>
    </attribute>
    <attribute name="prompt-for-username" required="false">
        <description>Whether database username and password for DataSource are provided
        <type>
            <boolean/>
        </type>
    </attribute>
    <attribute name="create-notx-data-source" required="false">
        <description>Create a second datasource that does not participate in global transaction
        </description>
        <type>
            <boolean/>
        </type>
    </attribute>
    <attribute name="wls-admin-user" required="false">
        <description>WLS admin user with permission to create DataSources. Used at runtime
        <type>
            <text/>
        </type>
    </attribute>
    <attribute name="wls-admin-pwd" required="false">
        <description>Password for WLS admin user.</description>
        <type>
            <text/>
        </type>
    </attribute>
        <attribute name="script-file" required="false">
            <description>A SQL Script file compiled into the jar</description>
            <type>
                <text max-length="255" long="false"/>
            </type>
            <default-value/>
        </attribute>
    </method-tag>
</control-tags>

```


DBScriptRunner.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/

Sample Source Code

```
package dbScript;

import com.bea.control.Control;

/**
 * DBScriptRunner sample custom Java control; see
 * {@link <a href="../../navDBScriptRunner.html">DBScriptRunner Control</a>}.
 *
 * This class supports two functions:
 * defining datasources and running scripts.
 *
 * @see <a href="../../navDBScriptRunner.html">DBScriptRunner Control</a>.
 */

public interface DBScriptRunner extends Control
{
    interface Callback
    {
        // callback methods that should appear on all
        // control instances would go here
    }
}
```

DBScriptRunnerImpl.jcs Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/

Sample Source Code

```
package dbScript;

import com.bea.control.*;
import java.sql.Connection;
import java.sql.Statement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.sql.SQLWarning;
import javax.sql.DataSource;
import java.util.*;
import javax.naming.InitialContext;
import java.sql.DatabaseMetaData;
import java.io.*;

import java.lang.reflect.Method;
import dbScript.util.DsUtils;
import dbScript.util.Constants;

/**
 * Implementation of DBScriptRunner sample custom Java control; see
 * {@link <a href="../../navDBScriptRunner.html">DBScriptRunner Control</a>}.
 *
 * @see <a href="../../navDBScriptRunner.html">DBScriptRunner Control</a>.
 *
 * @jcs:control-tags file="DBScriptRunner-tags.xml"
 * @jcs:jc-jar label="Script Runner"
 *         version="0.8"
 *         insert-wizard-class="dbScript.designUI.InsertWizCreateJCX"
 *         icon-16="/images/dbScript.gif"
 *         palette-priority="9"
 *         group-name="SQL Utilities"
 *         description="Creates a control to define datasources and run SQL scripts"
 *         @editor-info:code-gen control-interface="false"
 */
public class DBScriptRunnerImpl
    implements DBScriptRunner,
        com.bea.control.Extensible,
        com.bea.control.ControlSource
{

    /**
```

navExtensionDevKit.html Sample

```

    * @common:context
    */
    com.bea.control.ControlContext context;

    transient Object[] _args = null;
    transient String _dataSource = null;
    transient DataSource _ds = null;
    transient Connection _conn = null;
    transient int _iBatchSize = 1;

    public void context_onAcquire() throws Exception
    {
        try
        {
            _dataSource = context.getControlAttribute(Constants.TAG_CONN, Constants.ATTR_DS);
            String sBatchSize = context.getControlAttribute(Constants.TAG_CONN, Constants.ATTR_UB);
            _iBatchSize = Integer.parseInt(sBatchSize);
            return;
        } catch (NumberFormatException nfe) {
        } catch (Exception e) {
            e.printStackTrace();
            throw e;
        }
    }
    /**
     * Extensible.invoke handles all extended interface methods
     * Have to use this in order to have method-level annotations.
     */
    public Object invoke(Method method, Object[] args) throws Exception
    {
        try
        {
            String mName = method.getName();
            _args = args;
            String script = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.ATTR_SF);
            String dsType = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.ATTR_NEWD

            if ((script==null) && (dsType!=null))
            {
                // CreateDS method
                DsUtils dsu = new DsUtils();
                String url= context.getMethodAttribute(Constants.TAG_DBSCR, Constants.ATTR_URL);
                String sProps = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.ATTR_P
                Properties props = dsu.getPropertiesFromString(sProps);
                String sPromptUrl = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.A
                String sPromptUser = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.
                String sCreateNoTxDs = context.getMethodAttribute(Constants.TAG_DBSCR, Constant
                boolean fPromptUrl = (new Boolean(sPromptUrl)).booleanValue();
                boolean fPromptUser = (new Boolean(sPromptUser)).booleanValue();
                boolean fCreateNoTxDs = (new Boolean(sCreateNoTxDs)).booleanValue();
                String sAdminUser = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.A
                String sAdminPwd = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.AT
                String sNewDSName = context.getMethodAttribute(Constants.TAG_DBSCR, Constants.A
                String encryptedPassword=null;
                String[] aUrl=url.split(";");
                int i=0;
                if (fPromptUrl)
                {
                    for (i=0;i<aUrl.length;i++)

```

navExtensionDevKit.html Sample

```

        aUrl[i]=(String)args[i];
    }
    if (fPromptUser)
    {
        assert (args.length ==i+2);
        props.setProperty("user", (String)args[i]);
        i++;
        if (props.containsKey("password"))
            props.setProperty("password", (String)args[i]);
        else
            encryptedPassword = (String)args[i];
    }
    if (null==sNewDSName)
        sNewDSName = "Ds" + dsType + "Test";

    return dsu.createDataSource(dsType, sNewDSName,
                               aUrl, props, encryptedPassword,
                               sAdminUser, sAdminPwd,
                               fCreateNoTxDs);

}
if ((script!=null) && (script.length()>0))
{
    runScriptInternal(script, mName, true);
    return null;
}
if (args.length==1)
{
    runScriptInternal((String)args[0], mName, false);
    return null;
}
throw new Exception ("Unknown method signataure");

} catch (Exception e) {
    e.printStackTrace();
    throw e;
}
}

private void runScriptInternal(String script, String method, boolean fOpenAsResource)
{
    StringBuffer sbMsg = new StringBuffer(2000);
    sbMsg.append("Method ").append(method).append("invoked.\n");

    String cmdBatch="";
    boolean bEOF=false;
    int numErrs=0;
    String sInsertCmd = "INSERT";
    int [] numRows;
    int totRows=0;
    long start = System.currentTimeMillis();

    // SQL resources
    Statement st = null;
    Connection conn = null;
    InputStream is = null;
    Reader reader = null;

    try

```

navExtensionDevKit.html Sample

```
{
    Properties h = new Properties();
    h.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY, "weblogic.jndi.WLInitialContextFactory");
    InitialContext jndiContext= new InitialContext (h);
    _ds = (DataSource)jndiContext.lookup(_dataSource);

    conn = _ds.getConnection();
    DatabaseMetaData dbmd = conn.getMetaData();

    if (!(dbmd.supportsBatchUpdates()))
    {
        sbMsg.append("Batch inserts not supported; setting batch to 1 \n");
        _iBatchSize = 1;
    }

    if (fOpenAsResource)
    {
        // read the SQL script from the current class loader
        is = getClass().getClassLoader().getResourceAsStream(script);
        if(is == null) throw new RuntimeException("Could not load resource: " + script);
    }
    else
    {
        is= new java.io.FileInputStream(script);
    }
    // use platform encoding
    reader = new BufferedReader(new InputStreamReader(is));

    // read all commands
    List stmts = readCmds(reader);

    st = conn.createStatement();

    Iterator iterator = stmts.iterator();
    while(iterator.hasNext())
    {
        try
        {
            st.clearBatch();
            int sz = 0;
            // create the next batch
            while(sz < _iBatchSize && iterator.hasNext())
            {
                cmdBatch = (String)iterator.next();
                st.addBatch(cmdBatch);
                sz++;

                // non-INSERT commands have a batch size of 1
                if(!cmdBatch.startsWith(sInsertCmd))
                    break;
            }
            // execute the batch
            numRows = st.executeBatch();

            //keep track of rowcounts
            for (int k=0;k<numRows.length;k++)
            {
                if (numRows[k] > 0)
                    totRows += numRows[k];
            }
        }
    }
}
```

navExtensionDevKit.html Sample

```

        // keep track of any warnings
        SQLWarning sw = st.getWarnings();
        while (sw!=null)
        {
            sbMsg.append(sw.getMessage() + "\n");
            sw=sw.getNextWarning();
        }
    }
    catch (SQLException se)
    {
        String sqlState=se.getSQLState();
        int nativeError=se.getErrorCode();

        //ignore "object not found" errors on drop
        if (cmdBatch.toLowerCase().startsWith("drop"))
        {
            //sbMsg.append(se.getMessage() + "\n");
        }
        else
        {
            sbMsg.append("Batch failed : \n");
            sbMsg.append(cmdBatch + "\n");
            sbMsg.append(se.getMessage() + "\n");
            sbMsg.append("SQLSTATE: " + se.getSQLState() + "\n\n");
            numErrs++;
            if (numErrs > 25)
                bEOF=true;
        }
    }
} // end of statement iterator

long end = System.currentTimeMillis();
sbMsg.append("\n\nScript " + script + " executed successfully \n");
if (totRows > 0)
    sbMsg.append((new Integer(totRows).toString()) + " rows affected\n");
sbMsg.append(((end-start)/1000) + " seconds elapsed\n");
sbMsg.append("Used batch size of ").append(new Integer(_iBatchSize)).append("\n");
}
catch (Exception e)
{
    e.printStackTrace();
    sbMsg.append(e.getMessage());
    sbMsg.append("\n \n Batch Terminated \n\n Last stmt attempted: \n\n");
    sbMsg.append(cmdBatch);
}

// Send the notification that the batch is done. Since the callback is defined on the
// JCX we don't have its interface at compile time. So we use the reverse of the
// invoke function above by calling context.sendEvent().
// alternatively we could call getCallbackInterface();
try
{
    Object [] cbArgs = new Object[1];
    cbArgs[0] = sbMsg.toString();
    context.sendEvent("onScriptDone", cbArgs);
}
catch (Exception e)
{
    System.out.println("Unable to send callback; error : " + e.getMessage());
}

```

navExtensionDevKit.html Sample

```
// release all resources
try{if(st != null) st.close();}catch(SQLException ignore){}
try{if(conn != null) conn.close();}catch(SQLException ignore){}
try{if(is != null) is.close();}catch(IOException ignore){}
try{if(reader != null) reader.close();}catch(IOException ignore){}
return;

}

private List readCmds(Reader is)
    throws IOException
{
    List list = new ArrayList();
    StringBuffer buf = new StringBuffer();

    int ch;
    boolean inQuote = false;
    while((ch = is.read()) > -1)
    {
        if(ch == '\\')
            inQuote = !inQuote;

        // handle CR || LF, replace CR with ' ', skip LF
        if(ch == 13)
            ch = ' ';
        else if(ch == 10)
            continue;

        if(ch == ';' && !inQuote)
        {
            list.add(buf.toString().trim());
            buf = new StringBuffer(512);
        }
        else buf.append((char)ch);
    }

    return list;
}

}
```

designUI Samples

This section contains source code for the following samples.

Samples Included in This Section

InsertWizCreateJCX.java Sample

JndiDsPanel.java Sample

InsertWizCreateJCX.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/designUI/

Sample Source Code

```
package dbScript.designUI;

import com.bea.ide.control.ControlWizard;
import java.util.ArrayList;
import javax.swing.*;
import java.awt.BorderLayout;
import java.util.prefs.Preferences;
import com.bea.ide.control.EditorSupport;
import java.io.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JList;
import javax.swing.ListSelectionModel;
import java.text.MessageFormat;
import java.awt.Dimension;
import java.beans.PropertyChangeListener;
import java.beans.PropertyChangeEvent;
import com.bea.ide.control.EditorContext;

public class InsertWizCreateJCX extends com.bea.ide.control.ControlWizardSimple
{
    private JndiDsPanel _jndiPanel;

    private JList _scripts;
    private JList _scriptsSelected;
    private JButton _btnSelect;
    private JButton _btnRemove;
    private String _sFileName;

    private int _panelNum = 0;
    public static boolean _fPanelIsInited=false;
    private String _packageName;
    private String _jcxName;
    private int _configReq=this.CONFIG_INSERT_INSTANCE;
    public static EditorContext _ctx;

    public ArrayList getInstanceAnnotations()
    {
        ArrayList list = super.getInstanceAnnotations();
        return list;
    }

    public boolean onFinish()
```

navExtensionDevKit.html Sample

```

{
    if (!_jndiPanel.isValidState())
        return false;

    return super.onFinish();
}

public int getConfigurationInfo()
{
    return (this.CONFIG_INSERT_INSTANCE + this.CONFIG_CREATE_EXTENSION_FILE);
}

public EditorContext getEditorContext()
{
    return super.getContext();
}

public JComponent getComponent()
{
    if (_jndiPanel == null)
    {
        _jndiPanel = new JndiDsPanel(this);
        try
        {
            _jndiPanel.buildUI();
            _jndiPanel.retrieveDataSourceNames();
            _jndiPanel.addPropertyChangeListener("enabled", new PropertyChangeListener()
            {
                public void propertyChange(PropertyChangeEvent e)
                {
                    if ((e.getNewValue().toString()=="true") && !_fPanelIsInitd)
                    {
                        _jndiPanel.setInitialEnabledState();
                    }
                }
            });
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return _jndiPanel;
}

public void setPackage(String packageName)
{
    _packageName = packageName;
}

public void setName(String JcxName)
{
    _jcxName = JcxName;
}

public void setConfiguration(int iCfg)
{
    _configReq = iCfg;
}

public String getExtensionFileContent()
{
    return _jndiPanel.getJcxFileContent(_packageName, _jcxName);
}

```

```
}  
}
```

JndiDsPanel.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/designUI/

Sample Source Code

```
package dbScript.designUI;

import com.bea.ide.control.EditorSupport.*;
import java.awt.Component;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.JComponent;
import javax.swing.JLabel;
import java.util.Collection;
import javax.swing.JComboBox;
import javax.swing.DefaultComboBoxModel;

import java.awt.GridBagLayout;
import java.awt.GridBagConstraints;
import javax.swing.JSeparator;
import java.awt.Font;
import java.awt.Insets;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.JButton;
import dbScript.util.DsUtils;
import javax.swing.JOptionPane;
import javax.swing.ButtonGroup;
import javax.swing.JRadioButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import javax.swing.JCheckBox;
import java.text.MessageFormat;
import java.util.Properties;

import dbScript.util.Constants;
import javax.swing.SwingUtilities;
import com.bea.ide.control.EditorContext;
import weblogic.management.MBeanHome;
import java.io.File;
import java.net.URL;
import weblogic.management.configuration.ConfigurationMBean;

public class JndiDsPanel extends javax.swing.JPanel
{
    private String Title = "Method Configuration ";

    private JLabel dstLabel;
    private JLabel dsLabel;
    private JComboBox cmbExistingDS;
```

navExtensionDevKit.html Sample

```

private JComboBox cmbNewDSTypes;
private JRadioButton rdoNewDS;
private JRadioButton rdoExistingDS;
protected JLabel lblMessage;

private Collection dataSources;
private String _selectedDSType=Constants.DS_DEFAULT_TYPE;
private dbScript.util.DsUtils _dsu;

private JCheckBox chkInstNwind;
private JCheckBox chkInstAny;
private JCheckBox chkPromptServer;
private JCheckBox chkPromptUsername;
private JLabel scriptFilePath;
private static InsertWizCreateJCX _wiz;

public JndiDsPanel(InsertWizCreateJCX wiz)
{
    _wiz=wiz;
}

public void setInitialEnabledState()
{
    InsertWizCreateJCX._fPanelIsInited=true;
    SwingUtilities.invokeLater (
        new Runnable ()
        {
            public void run ()
            {
                try
                {
                    rdoExistingDS.setSelected(false);
                    cmbExistingDS.setEnabled(false);
                    rdoNewDS.setSelected(true);
                    enableNewDS(true);
                    setMessage(" ");
                }
                catch ( Exception e )
                {
                    return;
                }
            }
        }
    );
}

public void retrieveDataSourceNames ()
{
    EditorContext ctx = _wiz.getEditorContext();
    if ((null==ctx) || !ctx.ensureServerRunning())
    {
        setMessage("Unable to retrieve Data Source names");
        return;
    }
    final MBeanHome home = (MBeanHome)ctx.getMBeanHome();

    SwingUtilities.invokeLater (
        new Runnable ()
        {
            public void run ()
            {
                try

```

navExtensionDevKit.html Sample

```

        {
            setMessage("Fetching DataSource list from server...");
        }
        catch ( Exception e )
        {
            return;
        }
    }
} );

SwingUtilities.invokeLater (
    new Runnable ()
    {
        public void run ()
        {
            try
            {
                dataSources = _dsu.getDatasourceJNDINames(home);
                cmbExistingDS.setModel ( new DefaultComboBoxModel ( dataSources.toArray() ) );
                setMessage(" ");
            }
            catch ( Exception e )
            {
                setMessage("Error fetching DataSource names: " + e.getMessage());
                return;
            }
        }
    }
);

}

public void retrieveNewDSTypes ()
{
    try
    {
        _dsu = new DsUtils();
        cmbNewDSTypes.setModel ( new DefaultComboBoxModel ( _dsu.getDriverList() ) );
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}

public void buildUI ()
{
    if ( getComponents () != null && getComponents ().length > 0 ) removeAll ();

    GridBagConstraints constraints = new GridBagConstraints ();
    GridBagLayout layout = new GridBagLayout ();

    setLayout ( layout );
    constraints.anchor = GridBagConstraints.WEST;
    constraints.fill = GridBagConstraints.HORIZONTAL;
    constraints.gridy= 0;
    constraints.gridx = 0;
    constraints.gridwidth=2;
    constraints.ipadx=0;
    constraints.ipady=0;
    JLabel titleLabel = new JLabel ( "Data Source options", JLabel.LEFT );

```

navExtensionDevKit.html Sample

```

titleLabel.setFont(getHeaderFont());
add( titleLabel, constraints);

constraints.gridx= 0;
constraints.gridx = 2;
constraints.gridwidth=1;
constraints.ipadx=0;
constraints.ipady=0;
constraints.anchor = GridBagConstraints.EAST;
constraints.fill = GridBagConstraints.NONE;
JButton btnHelp= new JButton( "Help");
btnHelp.setFont(getHeaderFont());
btnHelp.setMnemonic('H');
btnHelp.addActionListener ( new ActionListener ()
{
    public void actionPerformed ( ActionEvent e )
    {
        try
        {
            EditorContext ctx=_wiz.getContext();
            if (null==ctx)
            {
                setMessage("Can't access editor context.");
                return;
            }
            File dir = ctx.getComponentsDir();

            File helpfile = new File(dir.getAbsolutePath()
                                    + "/" + Constants.VENDORNAME
                                    + "/help/en/index.html");
            ctx.invokeBrowser(helpfile.toURL());
        }
        catch (Exception ex)
        {
            System.out.println(ex.getMessage());
        }
    }
} );

add( btnHelp, constraints);

// datasource
ButtonGroup group = new ButtonGroup ();
constraints.gridx = 0;
constraints.gridy++;
constraints.anchor = GridBagConstraints.WEST;
constraints.fill = GridBagConstraints.HORIZONTAL;
constraints.insets = new Insets(10,0,0,0);
rdoExistingDS = new JRadioButton ( "Use existing DataSource", true );
rdoExistingDS.setMnemonic('U');
rdoExistingDS.setEnabled ( true );
rdoExistingDS.addActionListener ( new ActionListener ()
{
    public void actionPerformed ( ActionEvent e )
    {
        enableNewDS(false);
    }
} );
group.add ( rdoExistingDS );
add ( rdoExistingDS, constraints );

```

navExtensionDevKit.html Sample

```
constraints.insets = new Insets(0,15,0,0);
constraints.gridy++;
constraints.gridwidth=1;
add ( dsLabel = new JLabel ( "DataSource:", JLabel.LEFT ), constraints );

constraints.gridx=1;
constraints.gridwidth=2;
constraints.ipadx=85;
add ( cmbExistingDS=new JComboBox(), constraints );
cmbExistingDS.setEditable ( false );
cmbExistingDS.addActionListener ( new ActionListener ()
{
    public void actionPerformed ((ActionEvent e )
    {
        setDsType();
    }
} );
dsLabel.setDisplayedMnemonic('D');
dsLabel.setLabelFor(cmbExistingDS);

constraints.insets = new Insets(0,0,0,0);
constraints.gridx = 0;
constraints.gridy++;
constraints.gridwidth=3;
constraints.ipadx=0;
rdoNewDS = new JRadioButton ( "Create new DataSource (using name of new JCX file)", false );
rdoNewDS.setMnemonic('C');

rdoNewDS.addActionListener ( new ActionListener ()
{
    public void actionPerformed ((ActionEvent e )
    {
        enableNewDS(true);
    }
} );
rdoNewDS.setEnabled ( true);

group.add ( rdoNewDS );
add ( rdoNewDS, constraints );

constraints.insets = new Insets(0,15,0,0);
constraints.gridx = 0;
constraints.gridy++;
constraints.gridwidth=1;
add ( dstLabel = new JLabel ( "DB Type", JLabel.LEFT ), constraints );

constraints.gridx =1;
constraints.gridwidth=2;
constraints.ipadx=85;
add( cmbNewDSTypes=new JComboBox(), constraints );
cmbNewDSTypes.setEditable ( false );
retrieveNewDSTypes();
cmbNewDSTypes.setSelectedItem(Constants.DS_DEFAULT_TYPE);

cmbNewDSTypes.addActionListener ( new ActionListener ()
{
    public void actionPerformed ((ActionEvent e )
    {
        setDsType();
    }
} );
```


navExtensionDevKit.html Sample

```

    }
} );
dstLabel.setDisplayedMnemonic('T');
dstLabel.setLabelFor(cmbNewDSTypes);

constraints.gridx = 0;
constraints.gridy++;
constraints.gridwidth=3;
constraints.ipadx=0;
chkPromptServer = new JCheckBox ( "Prompt for DataSource URL", false);
chkPromptServer.setMnemonic('P');
add ( chkPromptServer, constraints );

constraints.gridy++;
chkPromptUsername = new JCheckBox ( "Prompt for Database User and Password", true);
add ( chkPromptUsername, constraints );
chkPromptUsername.setMnemonic('r');

JLabel scriptLabel = new JLabel("Script methods to create:");
constraints.insets=new Insets(10,0,0,0);
constraints.gridx = 0;
constraints.gridy++;
constraints.gridwidth=3;
add (scriptLabel, constraints);

constraints.insets=new Insets(0,15,0,0);
constraints.gridx = 0;
constraints.gridy++;
constraints.gridwidth=1;
chkInstNwind = new JCheckBox ( "Built-in script:  ", true);
chkInstNwind.setMnemonic('B');
chkInstNwind.setEnabled ( true );
add ( chkInstNwind, constraints );

constraints.gridx=1;
constraints.gridwidth=2;
add ( scriptFilePath=new JLabel ("..." , JLabel.LEFT ), constraints );

constraints.gridx=0;
constraints.gridy++;
constraints.gridwidth=3;
chkInstAny = new JCheckBox ( "Run any script file", true);
chkInstAny.setMnemonic('a');
chkInstAny.setEnabled ( true );
add ( chkInstAny, constraints );

constraints.insets = new Insets(10,0,0,0);
constraints.gridx=0;
constraints.gridy++;
constraints.gridwidth=3;
add ( lblMessage = new JLabel ( "", JLabel.LEFT ) , constraints);
InsertWizCreateJCX._fPanelIsInitd=false;

}
public void enableNewDS(boolean fNewDSSelected)
{
    cmbExistingDS.setEnabled (!fNewDSSelected);
    dsLabel.setEnabled(!fNewDSSelected);
    cmbNewDSTypes.setEnabled (fNewDSSelected );
    dstLabel.setEnabled(fNewDSSelected);
}

```

navExtensionDevKit.html Sample

```
        chkPromptServer.setEnabled (fNewDSSelected);
        chkPromptUsername.setEnabled (fNewDSSelected);
        setDsType();
    }

    public boolean isValidState ()
    {
        if (rdoExistingDS.isSelected() )
        {
            String dsn = ( String ) cmbExistingDS.getSelectedItem ();
            if ( dsn == null || dsn.length () < 1 )
            {
                setMessage ("Choose a datasource");
                return false;
            }
        }
        else
        {
            String dst = ( String ) cmbNewDSTypes.getSelectedItem ();
            if ( dst == null || dst.length () < 1 )
            {
                setMessage ("Choose a driver type");
                return false;
            }
        }
        return true;
    }

    protected String getHeaderFontName ()
    {
        // todo should be a property
        return "SansSerif";
    }

    protected Font getHeaderFont ()
    {
        return new Font ( getHeaderFontName (), Font.BOLD, 12 ) ;
    }

    protected void setMessage ( String message)
    {
        if ( lblMessage == null ) return;
        if (!lblMessage.isEnabled())
            lblMessage.setEnabled(true);
        lblMessage.setText ( message );
        lblMessage.repaint ();
    }

    private String getPathForScript( String scriptName)
    {
        return ("/dbScript/scripts/" + _selectedDSType + "/" + scriptName + ".sql");
    }
    private void setDsType()
    {
        String sel=null;
        try
        {
            if (rdoNewDS.isSelected())
            {
```

navExtensionDevKit.html Sample

```

        sel = (String)cmbNewDSTypes.getSelectedItemAt();
    }
    else
    {
        sel = (String)(cmbExistingDS.getSelectedItemAt());
        if (sel != null)
            sel = _dsu.getTypeForDSName(sel);
    }
}
catch (Exception e)
{
    System.out.println("failed in setDsType");
    e.printStackTrace();
}

if (sel==null)
    _selectedDSType = Constants.DS_DEFAULT_TYPE;
else
    _selectedDSType = sel;

if (scriptFilePath!=null)
    scriptFilePath.setText(getPathForScript("instnwind"));
}

public String getJcxFileContent(String packageName, String jcxName)
{
    String dsName;

    // if creating a new datasource, use jcxName as base;
    if (rdoExistingDS.isSelected())
        dsName=(String)cmbExistingDS.getSelectedItemAt();
    else
        dsName=jcxName;

    StringBuffer sb = new StringBuffer();
    sb.append(MessageFormat.format(
        getHeaderTemplate(),
        new Object [] { packageName,
            _dsu.getDsName(dsName,true), //use the NoTx Name for running script
            jcxName }));

    if (rdoNewDS.isSelected())
    {
        String sProps=_dsu.getDriverProperties(_selectedDSType);
        String urlDefaults=_dsu.getDriverURLDefaults(_selectedDSType);
        String [] aPrompts=_dsu.getDriverURLPrompts(_selectedDSType);
        StringBuffer sbParams=new StringBuffer();
        String methodName = Constants.METH_CREATEDS + dsName;

        if (chkPromptServer.isSelected())
        {
            String sc=", ";
            for (int i=0;i<aPrompts.length;i++)
            {
                if ((i==(aPrompts.length -1)) && (!chkPromptUsername.isSelected()))
                    sc=" ";
                sbParams.append("String ").append(aPrompts[i]).append(sc);
            }
        }
        if (chkPromptUsername.isSelected())

```

navExtensionDevKit.html Sample

```

        sbParams.append("String user, String password");

        sb.append(MessageFormat.format(
            getCreatedDSMethodTemplate(),
            new Object [] { cmbNewDSTypes.getSelectedItem(),
                           urlDefaults, sProps,
                           Boolean.toString(chkPromptServer.isSelected()),
                           Boolean.toString(chkPromptUsername.isSelected()),
                           Constants.WLSADMIN, Constants.WLSADMINPWD,
                           dsName, methodName, sbParams.toString()}));
    }
    if (chkInstNwind.isSelected())
    {
        String scriptPath = getPathForScript("instnwind");
        sb.append(MessageFormat.format(
            getScriptMethodTemplate(),
            new Object [] { scriptPath, "Instnwind" }));
    }
    if (chkInstAny.isSelected())
        sb.append(getFileMethodTemplate());

    if ((chkInstNwind.isSelected()) || (chkInstAny.isSelected()))
        sb.append(getCallbackTemplate());

    sb.append("} \n");
    return sb.toString();
}

private String getHeaderTemplate()
{
    String sTemplate =
        "package {0}; \n\n " +
        "import com.bea.control.*; \n " +

        " /** \n" +
        "  * @" + Constants.TAG_CONN + " " + Constants.ATTR_DS + "=\ \"{1}\" \n" +
        "  */ \n" +
        "public interface {2} extends dbScript.DBScriptRunner, com.bea.control.ControlExtension {
        "'{ ' \n \n";
    return sTemplate;
}

private String getScriptMethodTemplate()
{
    String sTemplate =
        " /** \n" +
        "  * @" + Constants.TAG_DBSCR + " " + Constants.ATTR_SF + "=\ \"{0}\" \n" +
        "  * \t\t @common:message-buffer enable=\ "true\" \n" +
        "  */ \n" +
        "  public void " + Constants.METH_RUNSCRIPT + "{1}(); \n \n" ;
    return sTemplate;
}

private String getFileMethodTemplate()
{
    String sTemplate =
        " /** \n" +
        "  * \t\t @common:message-buffer enable=\ "true\" \n" +
        "  */ \n" +

        "  public void " + Constants.METH_RUNFILE + "(String filePath); \n \n" ;
    return sTemplate;
}

```

navExtensionDevKit.html Sample

```

}
private String getCallbackTemplate()
{
    String sTemplate =
        "\n\t\t interface Callback extends dbScript.DBScriptRunner.Callback \n" +
        "    \t\t { \n" +
        "        \t\t\t void onScriptDone(String msg); \n" +
        "        \t\t } \n\n"
        ;
    return sTemplate;
}

private String getCreateDSMethodTemplate()
{
    String sTemplate =
        "    /** \n" +
        "        * @" + Constants.TAG_DBSCR + " " + Constants.ATTR_NEWDSTYPE + "=\">{0}\\" +
        "        * \t\t " + Constants.ATTR_URL + "=\">{1}\\" +
        "        * \t\t " + Constants.ATTR_PROPS + "=\">{2}\\" +
        "        * \t\t " + Constants.ATTR_PROMPTURL + "={3} \n" +
        "        * \t\t " + Constants.ATTR_PROMPTUSER + "={4} \n" +
        "        * \t\t " + Constants.ATTR_ADMINUSER + "=\">{5}\\" +
        "        * \t\t " + Constants.ATTR_ADMINPWD + "=\">{6}\\" +
        "        * \t\t " + Constants.ATTR_NEWDSNAME + "=\">{7}\\" +
        "        * \t\t " + Constants.ATTR_CREATENOTX + "=true \n" +
        "        */ \n" +
        "        public String {8}({9} ); \n \n" ;
    return sTemplate;
}
}

```

scripts Samples

This section contains source code for the following samples.

Samples Included in This Section

Oracle Samples

PointBase Samples

SQLServer Samples

Oracle Samples

This section contains source code for the following samples.

Samples Included in This Section

PointBase Samples

This section contains source code for the following samples.

Samples Included in This Section

SQLServer Samples

This section contains source code for the following samples.

Samples Included in This Section

util Samples

This section contains source code for the following samples.

Samples Included in This Section

Constants.java Sample

DsUtils.java Sample

Constants.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/util/

Sample Source Code

```
package dbScript.util;

public interface Constants
{
    public static final String TAG_CONN = "jc:dbscr-connection";
    public static final String TAG_DBSCR = "jc:dbscript";
    public static final String ATTR_DS = "data-source-jndi-name";
    public static final String ATTR_SF = "script-file";
    public static final String ATTR_NEWDSTYPE = "new-data-source-type";
    public static final String ATTR_NEWDNAME = "new-data-source-name";
    public static final String ATTR_PROPS = "data-source-properties";
    public static final String ATTR_URL = "data-source-url-defaults";
    public static final String ATTR_PROMPTURL = "prompt-for-url";
    public static final String ATTR_PROMPTUSER = "prompt-for-username";
    public static final String ATTR_CREATENOTX = "create-notx-data-source";
    public static final String ATTR_ADMINUSER = "wls-admin-user";
    public static final String ATTR_ADMINPWD = "wls-admin-pwd";
    public static final String ATTR_UB = "batch-size";
    public static final String METH_RUNSCRIPT="runScript";
    public static final String METH_RUNFILE="runFileScript";
    public static final String METH_CREATEDS="create";
    public static final String DS_DEFAULT_NOTXSUFFIX="NoTx";
    public static final String DS_DEFAULT_POOLSUFFIX="Pool";
    public static final String DS_DEFAULT_TYPE="PointBase";

    public static final String WLSADMIN="weblogic";
    public static final String WLSADMINPWD="weblogic";
    public static final String VENDORNAME="WorkshopSampleControl";
}
```

DsUtils.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/dbScript/util/

Sample Source Code

```
package dbScript.util;

import com.bea.control.JwsContext;
import java.sql.*;
import java.util.*;

import javax.naming.*;
import javax.management.*;
import weblogic.jndi.Environment;
import weblogic.management.*;
import weblogic.management.configuration.*;
import weblogic.management.deploy.*;
import weblogic.management.runtime.*;
import java.text.MessageFormat;
import dbScript.util.Constants;
import com.bea.ide.control.EditorContext;

public class DsUtils
{
    private transient Map mAvailConns;

    // this list of available drivers and their connection attributes
    // should come from the file BEAHOME/weblogic81/lib/jdbcdrivers.xml
    private static JDBCConn _connOracle_Thin = new JDBCConn
    (
        "oracle.jdbc.driver.OracleDriver",
        "jdbc:oracle:thin:@{0}:{1}:{2}",
        "server;port;database",
        "localhost;1521;sample", // or use ip addr
        "user=guest;protocol=thin",
        "DUAL"
    );

    private static JDBCConn _connOracle_WLS = new JDBCConn
    (
        "weblogic.jdbc.oci.Driver",
        "jdbc:oracle:thin:@{0}:{1}:{2}",
        "server;port;database",
        "localhost;1521;sample", // or use ip addr
        "user=guest;dll=ocijdbc9;protocol=thin",
        "DUAL"
    );

    private static JDBCConn _connSQLServer_DataDirect = new JDBCConn
    (
        "com.ddtek.jdbc.sqlserver.SQLServerDriver",
```

navExtensionDevKit.html Sample

```
"jdbc:microsoft:sqlsever:{0}:{1};DatabaseName={2}",
"server;port;database",
"//localhost;1433;Northwind",
"user=sa;password=notnull",
"systypes"
    );

private static JDBCConn _connSQLServer_WLS = new JDBCConn
(
    "weblogic.jdbc.mssqlserver4.Driver",
    //"jdbc:weblogic:mssqlserver4:Northwind@localhost:1433",
    "jdbc:weblogic:mssqlserver4:{0}@{1}:{2}",
    "database;server;port",
    "Northwind;localhost;1433",
    "user=sa",
    "systypes"
    );

private static JDBCConn _connPointBase = new JDBCConn
(
    "com.pointbase.jdbc.jdbcUniversalDriver",
    //"jdbc:pointbase:server://localhost:9093/cajun",
    "jdbc:pointbase:server:{0}:{1}/{2}",
    "server;port;database",
    "://localhost;9093;cajun",
    "user=PBPUBLIC;password=PBPUBLIC",
    "POINTBASE.SYSSCHEMATA"
    );

private static String _nameServer = null;

private static class JDBCConn
{
    JDBCConn(String d, String u, String urlP, String urlD, String p, String t)
    {
        _driver = d;
        _url = u;
        _urlPrompts=urlP;
        _urlDefaults=urlD;
        _properties = p;
        _testTable = t;
    }
    public Properties getProperties()
    {
        Properties p = new Properties();
        String[] a = _properties.split(";");
        for (int i=0 ; i<a.length ; i++)
        {
            String s = a[i];
            int ind = s.indexOf('=');
            p.put(s.substring(0,ind), s.substring(ind+1));
        }
        return p;
    }
    public String _driver;
    public String _url;
    public String _urlPrompts;
    public String _urlDefaults;
    public String _properties;
    public String _testTable;
}
```

```

public DsUtils()
{
    mAvailConns=new HashMap();
    mAvailConns.put("PointBase", _connPointBase);
    mAvailConns.put("Oracle", _connOracle_Thin);
    mAvailConns.put("SQLServer", _connSQLServer_WLS);

    mAvailConns.put("Oracle_Thin", _connOracle_Thin);
    mAvailConns.put("Oracle_WLS", _connOracle_WLS);
    mAvailConns.put("SQLServer_DataDirect", _connSQLServer_DataDirect);
    mAvailConns.put("SQLServer_WLS", _connSQLServer_WLS);
}

public String [] getDriverList()
{
    Object [] oa;
    oa=mAvailConns.keySet().toArray();
    String [] sa=new String [oa.length];
    for (int i=0;i<oa.length;i++)
        sa[i]=oa[i].toString();

    return sa;
}

public String [] getDriverURLPrompts(String driverType)
{
    JDBCConn c = (JDBCConn)mAvailConns.get(driverType);
    return c._urlPrompts.split(";");
}

public String getDriverURLDefaults(String driverType)
{
    JDBCConn c = (JDBCConn)mAvailConns.get(driverType);
    return c._urlDefaults;
}

public String getDriverProperties(String driverType)
{
    JDBCConn c = (JDBCConn)mAvailConns.get(driverType);
    return c._properties;
}

public Properties getPropertiesFromString(String props)
{
    Properties p = new Properties();
    String[] a = props.split(";");
    for (int i=0 ; i<a.length ; i++)
    {
        String s = a[i];
        int ind = s.indexOf('=');
        p.put(s.substring(0,ind), s.substring(ind+1));
    }
    return p;
}

public String createDataSource(
    String driverType,    // Oracle, PointBase, etc.
    String baseName,      //The DSName to be created
    String [] urlValues,  // url parts
    Properties p,         // connection properties
    String passwordToEncrypt, // stored as encrypted
    String wlsAdminUser,  // e.g. installadministrator
    String wlsAdminPassword, // e.g. installadministrator
    boolean fCreateNonTxDs //

```

navExtensionDevKit.html Sample

```

        ) throws Exception
    {
        MBeanHome home = _getMBeanHome(wlsAdminUser, wlsAdminPassword);
        return _createConnection(home, driverType, baseName , urlValues, p, passwordToEncrypt,
    }

    private void _dropMBean(MBeanHome home, DeploymentMBean bean) throws Exception
    {
        // Remove dynamically created connection pool from the server
        TargetMBean[] targets = bean.getTargets();
        for (int i=targets.length ; --i >= 0 ; )
            bean.removeTarget(targets[i]);
        // Remove dynamically created connection pool from the configuration
        home.deleteMBean(bean);
    }

    private MBeanHome _getMBeanHome(String user, String password) throws Exception
    {
        Environment env = new Environment();
        if (null != user)
        {
            env.setSecurityPrincipal(user);
            env.setSecurityCredentials(password);
        }

        Context c = env.getInitialContext();
        MBeanHome home = (MBeanHome)c.lookup("weblogic.management.home.localhome");

        if (null == _nameServer)
        {
            RemoteMBeanServer serverMBean = home.getMBeanServer();
            _nameServer = serverMBean.getServerName();
        }

        return home;
    }

    private String _createConnection(
        MBeanHome home,
        String driverType,
        String baseName,
        String [] urlValues,
        Properties p,
        String passwordToEncrypt,
        boolean fBothTxTypes) throws Exception
    {
        StringBuffer sb = new StringBuffer();
        boolean fCreateAttempted=false;

        JDBCConn conn = (JDBCConn)mAvailConns.get(driverType);
        if (null==conn)
            throw new Exception("Unknown Driver Type");

        JDBCConnectionPoolMBean jdbcBean = null;

        String poolName = getPoolName(baseName);
        String dsNameNoTx = getDsName(baseName, true);
        String dsTxName = getDsName(baseName, false);

        sb.append("\n-- Drop Existing Data Sources and Pool --");
        sb.append(dropExistingObjects(home, dsNameNoTx, dsTxName, poolName));
    }

```

navExtensionDevKit.html Sample

```

        try
        {
sb.append("\n\n-- Create New Connection Pool and Data Sources--");
fCreateAttempted=true;
        jdbcBean = (JDBCConnectionPoolMBean)home.createAdminMBean
                (
                        poolName,
                        "JDBCConnectionPool"
                );
String url = MessageFormat.format(conn._url, urlValues);
        jdbcBean.setURL(url);
        jdbcBean.setDriverName(conn._driver);
        jdbcBean.setProperties(p);
if (passwordToEncrypt != null)
        jdbcBean.setPassword(passwordToEncrypt);
        jdbcBean.setLoginDelaySeconds(1);
        jdbcBean.setInitialCapacity(1);
        jdbcBean.setMaxCapacity(10);
        jdbcBean.setCapacityIncrement(1);
        jdbcBean.setShrinkingEnabled(true);
        jdbcBean.setShrinkPeriodMinutes(10);
        jdbcBean.setRefreshMinutes(10);
        jdbcBean.setTestTableName(conn._testTable);

sb.append("\n New Connection Pool ");
sb.append("\n\t Driver: " + conn._driver);
sb.append("\n\t URL: " + url);
        sb.append("\n\t Properties: ").append(p);
        jdbcBean.addTarget((ServerMBean)home.getAdminMBean(_nameServer, "Server
        sb.append("\n\t\t CREATED");

sb.append("\n New DataSource (tx): ").append("\t" + dsTxName);
createTxDS(home, conn, poolName, dsTxName);
        sb.append("\t\t CREATED");

if (fBothTxTypes)
{
        sb.append("\n New DataSource (NoTx): ").append("\t" + dsNameNoTx);
        createDS(home, conn, poolName, dsNameNoTx);
        sb.append("\t\t CREATED");
}
        jdbcBean=null;
sb.append("\n");
return sb.toString();

}

        catch (Exception x)
        {
Exception ex = new Exception("Exception After " + sb.toString() + "\n\n", x);
// failed create leaves half-created pool
if (fCreateAttempted)
        dropExistingObjects(home, dsNameNoTx, dsTxName, poolName);
throw ex;
        }
}

private void createTxDS(MBeanHome home, JDBCConn conn, String poolName, String dsName) thro
{
        JDBCDataSourceMBean dsTxBean = null;
        dsTxBean = (JDBCDataSourceMBean)home.createAdminMBean
                (

```


navExtensionDevKit.html Sample

```

        dsName,
        "JDBCTxDataSource"
    );
    dsTxBean.setJNDIName(dsName);
    dsTxBean.setPoolName(poolName);
    dsTxBean.addTarget((ServerMBean)home.getAdminMBean(_nameServer, "Server"));
}

private void createDS(MBeanHome home, JDBCConn conn, String poolName, String dsNameNoTx) throws Exception {
    JDBCDataSourceMBean dsBean = null;
    dsBean = (JDBCDataSourceMBean)home.createAdminMBean(
        (
            dsNameNoTx,
            "JDBCDataSource"
        )
    );
    dsBean.setJNDIName(dsNameNoTx);
    dsBean.setPoolName(poolName);
    dsBean.addTarget((ServerMBean)home.getAdminMBean(_nameServer, "Server"));
}

private String dropExistingObjects(MBeanHome home, String dsNameNoTx, String dsTxName, String poolName) throws Exception {
    StringBuffer sb = new StringBuffer();
    JDBCConnectionPoolMBean jdbcBean = null;
    try {
        {
            JDBCDataSourceMBean dsBean = null;
            sb.append("\n DataSource (noTx) : ").append(dsNameNoTx);
            dsBean = (JDBCDataSourceMBean)home.getAdminMBean(dsNameNoTx, "JDBCDataSource");
            _dropMBean(home, dsBean);
            sb.append("    DROPPED");
            dsBean = null;
        }
        catch (InstanceNotFoundException x) {sb.append("    NOT FOUND");}

    try {
        JDBCTxDataSourceMBean dsTxBean = null;
        sb.append("\n DataSource (Tx) ").append(dsTxName);
        dsTxBean = (JDBCTxDataSourceMBean)home.getAdminMBean(dsTxName, "JDBCTxDataSource");
        _dropMBean(home, dsTxBean);
        sb.append("    DROPPED");
        dsTxBean = null;
    }
    catch (InstanceNotFoundException x) {sb.append("    NOT FOUND");}

    try {
        sb.append("\n ConnectionPool: ").append(poolName);
        jdbcBean = (JDBCConnectionPoolMBean)home.getAdminMBean(poolName, "JDBCConnectionPool");

        sb.append("\n\t").append(jdbcBean.getDriverName());
        sb.append("\n\t").append(jdbcBean.getURL());
        sb.append("\n\t").append(jdbcBean.getProperties());

        _dropMBean(home, jdbcBean);
    }
    catch (InstanceNotFoundException x) {}
}

```

navExtensionDevKit.html Sample

```

        sb.append("\n\t\t\t\t DROPPED");
        jdbcBean = null;
    }
    catch (InstanceNotFoundException x) {sb.append("\n\t\t\t\t NOT FOUND");}
catch (UndeploymentException ux)
{
    if (ux.getMessage().indexOf("Unknown Connection Pool")>=0)
        sb.append("\n\t\t\t\t NOT FOUND ");
    else
        throw ux;
}
catch (DistributedManagementException dx)
{
    if (dx.getMessage().indexOf("Unknown Connection Pool")>=0)
        sb.append("\n\t\t\t\t NOT FOUND ");
    else
        throw dx;
}

jdbcBean = null;
return sb.toString();
}

public Collection getDatasourceJNDINames (MBeanHome home)
    throws Exception
{
    Set allMBeans = home.getMBeansByType ( "JDBCTxDataSource" ); //Config
    Iterator iter = allMBeans.iterator ();

    ArrayList list = new ArrayList ();
    while ( iter.hasNext () )
    {
        WebLogicMBean mbean = ( WebLogicMBean ) iter.next ();
        list.add ( mbean.getName () );
    }
    allMBeans = home.getMBeansByType ( "JDBCDataSource" ); //Config
    iter = allMBeans.iterator ();

    while ( iter.hasNext () )
    {
        WebLogicMBean mbean = ( WebLogicMBean ) iter.next ();
        list.add ( mbean.getName () );
    }
    return list;
}

public String getTypeForDSName (String dsName)
    throws Exception
{
    Iterator iter = mAvailConns.keySet().iterator();
    String dsType=null;
    while (iter.hasNext())
    {
        dsType = (String) iter.next();
        if (dsName.indexOf(dsType)>=0)
            return dsType;
    }
    return null;
}

public String getDsName(String baseName, boolean fNoTx)

```

navExtensionDevKit.html Sample

```
{
    String name = baseName.substring(0,1).toUpperCase() + baseName.substring(1);

    if (fNoTx)
        return name + Constants.DS_DEFAULT_NOTXSUFFIX;
    else
        return name;
}
private String getPoolName(String baseName)
{
    String name = baseName.substring(0,1).toUpperCase() + baseName.substring(1);
    return name + Constants.DS_DEFAULT_POOLSUFFIX;
}
}
```

docs Samples

This section contains source code for the following samples.

Samples Included in This Section

help Samples

samples Samples

help Samples

This section contains source code for the following samples.

Samples Included in This Section

doc Samples

doc Samples

This section contains source code for the following samples.

Samples Included in This Section

en Samples

en Samples

This section contains source code for the following samples.

Samples Included in This Section

partners Samples

partners Samples

This section contains source code for the following samples.

Samples Included in This Section

DBScripter Samples

DBScripter Samples

This section contains source code for the following samples.

Samples Included in This Section

ConDBScriptRunnerControlWizard.html Sample

java-class Samples

javadoc-tag Samples

NavDBScriptRunner.html Sample

OvwDBScriptRunner.html Sample

Toc.xml Sample

ConDBScriptRunnerControlWizard.html Sample

This topic includes the source code for the ConDBScriptRunnerControlWizard.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<head>
<title>DBScriptRunner Control Wizard</title>
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<link href="../../workshop.css" rel="stylesheet" type="text/css">
<a href="../../core/index.html" id="index"></a>
<script language="JavaScript" src="../../core/topicInfo.js"></script>
<script language="JavaScript" src="../../core/CookieClass.js"></script>
<script language="JavaScript" src="../../core/displayContent.js"></script>
</head>

<body>

<script language="JavaScript">
displayInFrames();
</script>

<h1 class="Title">DBScriptRunner Control Wizard</h1>
<p>Help on Wizard Panel.</p>

<script language="JavaScript">
writeTopicInfo();
</script>

</body>
```

java-class Samples

This section contains source code for the following samples.

Samples Included in This Section

dbScript Samples

dbScript Samples

This section contains source code for the following samples.

Samples Included in This Section

DBScriptRunner.Callback.html Sample

DBScriptRunner.html Sample

DBScriptRunnerImpl.html Sample

Package-summary.html Sample

util Samples

DBScriptRunner.Callback.html Sample

This topic includes the source code for the DBScriptRunner.Callback.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 14:44:38 PDT 2003 -->
<TITLE>
DBScriptRunner.Callback
</TITLE>
<META NAME="keywords" CONTENT="dbScript.DBScriptRunner.Callback,DBScriptRunner.Callback interfa
<LINK REL="stylesheet" TYPE="text/css" HREF="../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="DBScriptRunner.Callback";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <A HREF="../overview-summary.html"><FONT CLASS=
    <A HREF="package-summary.html"><FONT CLASS="Nav
    &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <A HREF="package-tree.html"><FONT CLASS="NavBar
    <A HREF="../deprecated-list.html"><FONT CLASS="
    <A HREF="../index-all.html"><FONT CLASS="NavBar
    <A HREF="../help-doc.html"><FONT CLASS="NavBarF
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">
&nbsp;<A HREF="../dbScript/DBScriptRunner.html"><B>PREV CLASS</B></A>&nbsp;</TD>
```

navExtensionDevKit.html Sample

```
&nbsp;NEXT CLASS</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">
  <A HREF=" ../index.html" TARGET="_top"><B>FRAMES</B></A> &nbsp;
&nbsp;<A HREF="DBScriptRunner.Callback.html" TARGET="_top"><B>NO FRAMES</B></A> &nbsp;
&nbsp;
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF=" ../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF=" ../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
<TR>
<TD VALIGN="top" CLASS="NavBarCell13"><FONT SIZE="-2">
  SUMMARY:&nbsp;&nbsp;NESTED&nbsp;&nbsp;|&nbsp;&nbsp;FIELD&nbsp;&nbsp;|&nbsp;&nbsp;CONSTR&nbsp;&nbsp;|&nbsp;&nbsp;METHOD</FONT></TD>
<TD VALIGN="top" CLASS="NavBarCell13"><FONT SIZE="-2">
DETAIL:&nbsp;&nbsp;FIELD&nbsp;&nbsp;|&nbsp;&nbsp;CONSTR&nbsp;&nbsp;|&nbsp;&nbsp;METHOD</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>
<!-- ===== START OF CLASS DATA ===== -->
<H2>
<FONT SIZE="-1">
dbScript</FONT>
<BR>
Interface DBScriptRunner.Callback</H2>
<DL>
<DT><B>Enclosing interface:</B><DD><A HREF=" ../dbScript/DBScriptRunner.html">DBScriptRunner</A>
</DL>
<HR>
<DL>
<DT>public static interface <B>DBScriptRunner.Callback</B></DL>

<P>
<HR>

<P>
<!-- ===== NESTED CLASS SUMMARY ===== -->

<!-- ===== FIELD SUMMARY ===== -->

<!-- ===== CONSTRUCTOR SUMMARY ===== -->

<!-- ===== METHOD SUMMARY ===== -->

<P>
<!-- ===== FIELD DETAIL ===== -->
```

navExtensionDevKit.html Sample

```
<!-- ===== CONSTRUCTOR DETAIL ===== -->

<!-- ===== METHOD DETAIL ===== -->

<!-- ===== END OF CLASS DATA ===== -->
<HR>

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_bottom"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_bottom_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../overview-summary.html"><FONT CLASS=
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-summary.html"><FONT CLASS="Nav
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../deprecated-list.html"><FONT CLASS=
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../index-all.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../help-doc.html"><FONT CLASS="NavBarF
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;<A HREF="../dbScript/DBScriptRunner.html"><B>PREV CLASS</B></A>&nbsp;<
&nbsp;<NEXT CLASS</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../index.html" TARGET="_top"><B>FRAMES</B></A>  &nbsp;<
&nbsp;<A HREF="DBScriptRunner.Callback.html" TARGET="_top"><B>NO FRAMES</B></A>  &nbsp;<
&nbsp;<
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');
  }
  <!-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
<TR>
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
  SUMMARY:&nbsp;<NOSTED&nbsp;<|&nbsp;<FIELD&nbsp;<|&nbsp;<CONSTR&nbsp;<|&nbsp;<METHOD</FONT></TD>
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
DETAIL:&nbsp;<FIELD&nbsp;<|&nbsp;<CONSTR&nbsp;<|&nbsp;<METHOD</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>
```

```
</BODY>  
</HTML>
```


DBScriptRunner.html Sample

This topic includes the source code for the DBScriptRunner.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 16:04:27 PDT 2003 -->
<TITLE>
DBScriptRunner
</TITLE>
<META NAME="keywords" CONTENT="dbScript.DBScriptRunner,DBScriptRunner interface">
<LINK REL="stylesheet" TYPE="text/css" HREF="../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="DBScriptRunner";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TR>
    <TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&PREV CLASS&nbsp;&nbsp;&nbsp;&nbsp;&

```

```
<!-->  
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">  
    <A HREF="../dbScript/DBScriptRunner.Callback.html"><B>NEXT CLASS</B></A></FONT></TD>  
<A HREF="../../index.html" TARGET="_top"><B>FRAMES</B></A>   <br>  
<A HREF="DBScriptRunner.html" TARGET="_top"><B>NO FRAMES</B></A>   <br>  
<SCRIPT>  
    <!--  
        if(window==top) {  
            document.writeln('<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');  
        }  
    //-->  
</SCRIPT>  
<NOSCRIPT>  
    <A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>  
</NOSCRIPT>  
</FONT></TD>  
</TR>  
<TR>  
<TD VALIGN="top" CLASS="NavBarCell13"><FONT SIZE="-2">  
    SUMMARY:<br><A HREF="#nested_class_summary">NESTED</A><br>|<br>FIELD<br>|<br>CONSTR  
<TD VALIGN="top" CLASS="NavBarCell13"><FONT SIZE="-2">  
DETAIL:<br>FIELD<br>|<br>CONSTR<br>|<br>METHOD</FONT></TD>  
</TR>  
</TABLE>  
<!-- ===== END OF NAVBAR ===== -->  
  
<HR>  
<!-- ===== START OF CLASS DATA ===== -->  
<H2>  
<FONT SIZE="-1">  
dbScript</FONT>  
<BR>  
Interface DBScriptRunner</H2>  
<DL>  
<DT><B>All Known Implementing Classes:</B> <DD><A HREF="../dbScript/DBScriptRunnerImpl.html">DB  
</DL>  
<HR>  
<DL>  
<DT>public interface <B>DBScriptRunner</B></DL>  
  
<P>  
DBSriptRunner sample custom Java control; see  
    <a href="../../navDBScriptRunner.html">DBScriptRunner Control</a>.   
  
This class supports two functions:  
defining datasources and running scripts.  
<P>  
  
<P>  
<DL>  
<DT><B>See Also:</B><DD><a href="../../navDBScriptRunner.html">DBScriptRunner Control</a>.</DL>  
<HR>  
  
<P>  
<!-- ===== NESTED CLASS SUMMARY ===== -->  
  
<A NAME="nested_class_summary"><!-- --></A>  
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">  
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">  
<TD COLSPAN=2><FONT SIZE="+2">  
<B>Nested Class Summary</B></FONT></TD>
```

navExtensionDevKit.html Sample

```
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">
<CODE>static interface</CODE></FONT></TD>
<TD><CODE><B><A HREF="../../dbScript/DBScriptRunner.Callback.html">DBScriptRunner.Callback</A></B>

<BR>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</TD>
</TR>
</TABLE>
&nbsp;&nbsp;&
<!-- ===== FIELD SUMMARY ===== -->

<!-- ===== CONSTRUCTOR SUMMARY ===== -->

<!-- ===== METHOD SUMMARY ===== -->

<P>

<!-- ===== FIELD DETAIL ===== -->

<!-- ===== CONSTRUCTOR DETAIL ===== -->

<!-- ===== METHOD DETAIL ===== -->

<!-- ===== END OF CLASS DATA ===== -->
<HR>

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_bottom"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_bottom_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../overview-summary.html"><FONT CLASS=
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-summary.html"><FONT CLASS="Nav
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev">    &nbsp;&nbsp;&<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../deprecated-list.html"><FONT CLASS="
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../index-all.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../help-doc.html"><FONT CLASS="NavBarF
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">
&nbsp;&nbsp;&PREV CLASS&nbsp;&nbsp;&
&nbsp;&nbsp;&<A HREF="../../dbScript/DBScriptRunner.Callback.html"><B>NEXT CLASS</B></A></FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">
```

navExtensionDevKit.html Sample

```
<A HREF="../index.html" TARGET="_top"><B>FRAMES</B></A>    &nbsp;
&nbsp;<A HREF="DBScriptRunner.html" TARGET="_top"><B>NO FRAMES</B></A>    &nbsp;
&nbsp;
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
<TR>
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
  SUMMARY:&nbsp;<A HREF="#nested_class_summary">NESTED</A>&nbsp;|&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;CONSTR
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
DETAIL:&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;CONSTR&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;METHOD</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>

</BODY>
</HTML>
```

DBScriptRunnerImpl.html Sample

This topic includes the source code for the DBScriptRunnerImpl.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 16:04:27 PDT 2003 -->
<TITLE>
DBScriptRunnerImpl
</TITLE>
<META NAME="keywords" CONTENT="dbScript,DBScriptRunnerImpl,DBScriptRunnerImpl class">
<LINK REL ="stylesheet" TYPE="text/css" HREF="../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="DBScriptRunnerImpl";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../overview-summary.html"><FONT CLASS=
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-summary.html"><FONT CLASS="Nav
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../deprecated-list.html"><FONT CLASS="
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../index-all.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../help-doc.html"><FONT CLASS="NavBarF
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;&nbsp;&nbsp;<PREV CLASS&nbsp;&nbsp;&
```

navExtensionDevKit.html Sample

```

&nbsp;NEXT CLASS</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">
  <A HREF=" ../index.html" TARGET="_top"><B>FRAMES</B></A> &nbsp;
&nbsp;<A HREF="DBScriptRunnerImpl.html" TARGET="_top"><B>NO FRAMES</B></A> &nbsp;
&nbsp;
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF=" ../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF=" ../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
<TR>
<TD VALIGN="top" CLASS="NavBarCell13"><FONT SIZE="-2">
  SUMMARY:&nbsp;&nbsp;NESTED&nbsp;&nbsp;|&nbsp;&nbsp;FIELD&nbsp;&nbsp;|&nbsp;&nbsp;<A HREF="#constructor_summary">CONSTR</A>&nbsp;
<TD VALIGN="top" CLASS="NavBarCell13"><FONT SIZE="-2">
DETAIL:&nbsp;&nbsp;FIELD&nbsp;&nbsp;|&nbsp;&nbsp;<A HREF="#constructor_detail">CONSTR</A>&nbsp;&nbsp;<A HREF="#m
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>
<!-- ===== START OF CLASS DATA ===== -->
<H2>
<FONT SIZE="-1">
dbScript</FONT>
<BR>
Class DBScriptRunnerImpl</H2>
<PRE>
java.lang.Object
|
+--<B>dbScript.DBScriptRunnerImpl</B>
</PRE>
<DL>
<DT><B>All Implemented Interfaces:</B> <DD><A HREF=" ../dbScript/DBScriptRunner.html">DBScriptRu
</DL>
<HR>
<DL>
<DT>public class <B>DBScriptRunnerImpl</B><DT>extends java.lang.Object<DT>implements <A HREF=" .
<P>
Implementation of DBSriptRunner sample custom Java control; see
  <a href=" ../ ../navDBScriptRunner.html">DBScriptRunner Control</a>.
<P>

<P>
<DL>
<DT><B>See Also:</B><DD><a href=" ../ ../navDBScriptRunner.html">DBScriptRunner Control</a>.</DL>
<HR>

<P>
<!-- ===== NESTED CLASS SUMMARY ===== -->

<A NAME="nested_class_summary"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">

```

navExtensionDevKit.html Sample

```

<TD COLSPAN=2><FONT SIZE="+2">
<B>Nested Class Summary</B></FONT></TD>
</TR>
</TABLE>
  <A NAME="nested_classes_inherited_from_class_dbScript.DBScriptRunner"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#EEEEFF" CLASS="TableSubHeadingColor">
<TD><B>Nested classes inherited from class dbScript.<A HREF="../dbScript/DBScriptRunner.html">DBScriptRunner
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD><CODE><A HREF="../dbScript/DBScriptRunner.Callback.html">DBScriptRunner.Callback</A></CODE>
</TR>
</TABLE>
  <!-- ===== FIELD SUMMARY ===== -->

<!-- ===== CONSTRUCTOR SUMMARY ===== -->

<A NAME="constructor_summary"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Constructor Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD><CODE><B><A HREF="../dbScript/DBScriptRunnerImpl.html#DBScriptRunnerImpl()">DBScriptRunnerImpl
<BR>
  <!-- ===== METHOD SUMMARY ===== -->

<A NAME="method_summary"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Method Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">
<CODE>&nbsp;&nbsp;&nbsp;&nbsp;&void</CODE></FONT></TD>
<TD><CODE><B><A HREF="../dbScript/DBScriptRunnerImpl.html#context_onAcquire()">context_onAcquire
<BR>
  <!-- ===== METHOD SUMMARY ===== -->

<A NAME="method_summary"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Method Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">
<CODE>&nbsp;&nbsp;&nbsp;&nbsp;&java.lang.Object</CODE></FONT></TD>
<TD><CODE><B><A HREF="../dbScript/DBScriptRunnerImpl.html#invoke(java.lang.reflect.Method, java
    java.lang.Object[]&nbsp;&nbsp;&args)">invoke
<BR>
  <!-- ===== METHOD SUMMARY ===== -->

<A NAME="method_summary"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Method Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">
<CODE>&nbsp;&nbsp;&nbsp;&nbsp;&Extensible.invoke handles all extensions
Have to use this in order to have method-level annotations.</CODE></TD>
</TR>
</TABLE>
  <!-- ===== METHOD SUMMARY ===== -->

<A NAME="method_summary"><!-- --></A>

```

navExtensionDevKit.html Sample

```
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#EEEEFF" CLASS="TableSubHeadingColor">
<TD><B>Methods inherited from class java.lang.Object</B></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD><CODE>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
</TR>
</TABLE>
&nbsp;
<P>

<!-- ===== FIELD DETAIL ===== -->

<!-- ===== CONSTRUCTOR DETAIL ===== -->

<A NAME="constructor_detail"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=1><FONT SIZE="+2">
<B>Constructor Detail</B></FONT></TD>
</TR>
</TABLE>

<A NAME="DBScriptRunnerImpl()"><!-- --></A><H3>
DBScriptRunnerImpl</H3>
<PRE>
public <B>DBScriptRunnerImpl</B>()</PRE>
<DL>
</DL>

<!-- ===== METHOD DETAIL ===== -->

<A NAME="method_detail"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=1><FONT SIZE="+2">
<B>Method Detail</B></FONT></TD>
</TR>
</TABLE>

<A NAME="context_onAcquire()"><!-- --></A><H3>
context_onAcquire</H3>
<PRE>
public void <B>context_onAcquire</B>()
                throws java.lang.Exception</PRE>
<DL>
<DD><DL>
</DL>
</DD>
<DD><DL>

<DD><CODE>java.lang.Exception</CODE></DL>
</DD>
</DL>
<HR>

<A NAME="invoke(java.lang.reflect.Method, java.lang.Object[])"><!-- --></A><H3>
invoke</H3>
<PRE>
public java.lang.Object <B>invoke</B>(java.lang.reflect.Method&nbsp;method,
```



```
java.lang.Object[]&nbsp;args)
throws java.lang.Exception</PRE>
```

DBScriptRunnerImpl.html Sample

navExtensionDevKit.html Sample

```
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
  SUMMARY:&nbsp;&nbsp;&nbsp;NESTED&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#constructor_summary">CONSTR</A>&
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
DETAIL:&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#constructor_detail">CONSTR</A>&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#m
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>

</BODY>
</HTML>
```

Package-summary.html Sample

This topic includes the source code for the Package-summary.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 14:44:38 PDT 2003 -->
<TITLE>
dbScript
</TITLE>
<META NAME="keywords" CONTENT="dbScript package">
<LINK REL="stylesheet" TYPE="text/css" HREF="../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="dbScript";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> <A HREF="../overview-summary.html"><FONT CLASS=
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Package</
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <FONT CLASS="NavBarFont1">Class</FONT>&nbsp;</T
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="../deprecated-list.html"><FONT CLASS="
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="../index-all.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="../help-doc.html"><FONT CLASS="NavBarF
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;&nbsp;&nbsp;<A HREF="#">PREV</A> <A HREF="#">PACKAGE</A> &nbsp;&nbsp;&

```

navExtensionDevKit.html Sample

```

   <A HREF="../dbScript/util/package-summary.html"><B>NEXT PACKAGE</B></A></FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../index.html" TARGET="_top"><B>FRAMES</B></A>      
   <A HREF="package-summary.html" TARGET="_top"><B>NO FRAMES</B></A>      
   
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>
<H2>
Package dbScript
</H2>

<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Interface Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD WIDTH="15%"><B><A HREF="DBScriptRunner.html"><I>DBScriptRunner</I></A></B></TD>
<TD>This class supports two functions:
  defining datasources and running scripts.</TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD WIDTH="15%"><B><A HREF="DBScriptRunner.Callback.html"><I>DBScriptRunner.Callback</I></A></B></TD>
<TD>&nbsp;</TD>
</TR>
</TABLE>
   

<P>

<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Class Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD WIDTH="15%"><B><A HREF="DBScriptRunnerImpl.html">DBScriptRunnerImpl</A></B></TD>
<TD>&nbsp;</TD>
</TR>
</TABLE>
   

<P>
<HR>

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_bottom"><!-- --></A>

```

```
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_bottom_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../overview-summary.html"><FONT CLASS=
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Package</
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <FONT CLASS="NavBarFont1">Class</FONT>&nbsp;</T
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../deprecated-list.html"><FONT CLASS="
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../index-all.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../help-doc.html"><FONT CLASS="NavBarF
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;<PREV PACKAGE&nbsp;</TD>
&nbsp;<A HREF="../dbScript/util/package-summary.html"><B>NEXT PACKAGE</B></A></FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../index.html" TARGET="_top"><B>FRAMES</B></A>   &nbsp;</TD>
&nbsp;<A HREF="package-summary.html" TARGET="_top"><B>NO FRAMES</B></A>   &nbsp;</TD>
&nbsp;</TD>
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>');
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>

</BODY>
</HTML>
```

util Samples

This section contains source code for the following samples.

Samples Included in This Section

Constants.html Sample

DsUtils.html Sample

Package-summary.html Sample

Constants.html Sample

This topic includes the source code for the Constants.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 14:44:38 PDT 2003 -->
<TITLE>
Constants
</TITLE>
<META NAME="keywords" CONTENT="dbScript.util.Constants, Constants interface">
<LINK REL="stylesheet" TYPE="text/css" HREF="../../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="Constants";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
      <A HREF="../../overview-summary.html"><FONT CLA
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
      <A HREF="package-summary.html"><FONT CLASS="Nav
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
      <A HREF="../../deprecated-list.html"><FONT CLAS
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
      <A HREF="../../index-all.html"><FONT CLASS="Nav
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
      <A HREF="../../help-doc.html"><FONT CLASS="NavB
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;&nbsp;&nbsp;<PREV CLASS&nbsp;&nbsp;&
```

[illegible]

	ATTR_ADMINUSER
	ATTR_CREATENOTX
	ATTR_DS
	ATTR_NEWSNAME
	ATTR_NEWDSTYPE
	ATTR_PROMPTURL
	ATTR_PROMPTUSER
	ATTR_PROPS

[illegible]

```
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#METH_RUNFILE">METH_RUNFILE</A></B></CODE>  
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>static&nbsp;&nbsp;&nbsp;&java.lang.String</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#METH_RUNSCRIPT">METH_RUNSCRIPT</A></B></CODE>  
  
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>static&nbsp;&nbsp;&nbsp;&java.lang.String</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#TAG_CONN">TAG_CONN</A></B></CODE>  
  
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>static&nbsp;&nbsp;&nbsp;&java.lang.String</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#TAG_DBSCR">TAG_DBSCR</A></B></CODE>  
  
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&~  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>static&nbsp;&nbsp;&nbsp;&java.lang.String</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#VENDORNAME">VENDORNAME</A></B></CODE>  
  
<BR>  
&nbsp;&nbsp;&~  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>static&nbsp;&nbsp;&~  
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#WLSADMIN">WLSADMIN</A></B></CODE>  
  
<BR>  
&nbsp;&~  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>static&nbsp;&~  
<TD><CODE><B><A HREF="../../dbScript/util/Constants.html#WLSADMINPWD">WLSADMINPWD</A></B></CODE>  
  
<BR>  
&~  
</TR>  
</TABLE>  
  
&nbsp;&~  
<!-- ===== CONSTRUCTOR SUMMARY ===== -->
```

```
<!-- ===== METHOD SUMMARY ===== -->
```

navExtensionDevKit.html Sample

<P>

<!-- ===== FIELD DETAIL ===== -->

<!-- -->

<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">

<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">

<TD COLSPAN=1>

Field Detail</TD>

</TR>

</TABLE>

<!-- --><H3>

TAG_CONN</H3>

<PRE>

public static final java.lang.String TAG_CONN</PRE>

<DL>

<DL>

<DT>See Also:<DD>C

</DL>

<HR>

<!-- --><H3>

TAG_DBSCR</H3>

<PRE>

public static final java.lang.String TAG_DBSCR</PRE>

<DL>

<DL>

<DT>See Also:<DD>

</DL>

<HR>

<!-- --><H3>

ATTR_DS</H3>

<PRE>

public static final java.lang.String ATTR_DS</PRE>

<DL>

<DL>

<DT>See Also:<DD>Co

</DL>

<HR>

<!-- --><H3>

ATTR_SF</H3>

<PRE>

public static final java.lang.String ATTR_SF</PRE>

<DL>

<DL>

<DT>See Also:<DD>Co

</DL>

<HR>

<!-- --><H3>

ATTR_NEWDSTYPE</H3>

<PRE>

public static final java.lang.String ATTR_NEWDSTYPE</PRE>

<DL>

<DL>

<DT>See Also:<DD><A HREF="../../../constant-values.html#dbScript.util.Constants.ATTR_NEWDST

</DL>

<HR>

navExtensionDevKit.html Sample

```
<A NAME="ATTR_NEWDSDNAME"><!-- --></A><H3>
ATTR_NEWDSDNAME</H3>
<PRE>
public static final java.lang.String <B>ATTR_NEWDSDNAME</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.ATTR_NEWDSDNAME">
</DL>
<HR>

<A NAME="ATTR_PROPS"><!-- --></A><H3>
ATTR_PROPS</H3>
<PRE>
public static final java.lang.String <B>ATTR_PROPS</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.ATTR_PROPS">
</DL>
<HR>

<A NAME="ATTR_URL"><!-- --></A><H3>
ATTR_URL</H3>
<PRE>
public static final java.lang.String <B>ATTR_URL</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.ATTR_URL">C
</DL>
<HR>

<A NAME="ATTR_PROMPTURL"><!-- --></A><H3>
ATTR_PROMPTURL</H3>
<PRE>
public static final java.lang.String <B>ATTR_PROMPTURL</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.ATTR_PROMPTURL">
</DL>
<HR>

<A NAME="ATTR_PROMPTUSER"><!-- --></A><H3>
ATTR_PROMPTUSER</H3>
<PRE>
public static final java.lang.String <B>ATTR_PROMPTUSER</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.ATTR_PROMPTUSER">
</DL>
<HR>

<A NAME="ATTR_CREATENOTX"><!-- --></A><H3>
ATTR_CREATENOTX</H3>
<PRE>
public static final java.lang.String <B>ATTR_CREATENOTX</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.ATTR_CREATENOTX">
</DL>
<HR>
```

navExtensionDevKit.html Sample

```
<A NAME="ATTR_ADMINUSER"><!-- --></A><H3>
ATTR_ADMINUSER</H3>
<PRE>
public static final java.lang.String <B>ATTR_ADMINUSER</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../../constant-values.html#dbScript.util.Constants.ATTR_ADMINUSER">
</DL>
<HR>

<A NAME="ATTR_ADMINPWD"><!-- --></A><H3>
ATTR_ADMINPWD</H3>
<PRE>
public static final java.lang.String <B>ATTR_ADMINPWD</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../../constant-values.html#dbScript.util.Constants.ATTR_ADMINPWD">
</DL>
<HR>

<A NAME="ATTR_UB"><!-- --></A><H3>
ATTR_UB</H3>
<PRE>
public static final java.lang.String <B>ATTR_UB</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../../constant-values.html#dbScript.util.Constants.ATTR_UB">Co
</DL>
<HR>

<A NAME="METH_RUNSCRIPT"><!-- --></A><H3>
METH_RUNSCRIPT</H3>
<PRE>
public static final java.lang.String <B>METH_RUNSCRIPT</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../../constant-values.html#dbScript.util.Constants.METH_RUNSCRIPT">
</DL>
<HR>

<A NAME="METH_RUNFILE"><!-- --></A><H3>
METH_RUNFILE</H3>
<PRE>
public static final java.lang.String <B>METH_RUNFILE</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../../constant-values.html#dbScript.util.Constants.METH_RUNFILE">
</DL>
<HR>

<A NAME="METH_CREATEDS"><!-- --></A><H3>
METH_CREATEDS</H3>
<PRE>
public static final java.lang.String <B>METH_CREATEDS</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../../constant-values.html#dbScript.util.Constants.METH_CREATEDS">
</DL>
<HR>

<A NAME="DS_DEFAULT_NOTXSUFFIX"><!-- --></A><H3>
```

navExtensionDevKit.html Sample

```
DS_DEFAULT_NOTXSUFFIX</H3>
<PRE>
public static final java.lang.String <B>DS_DEFAULT_NOTXSUFFIX</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.DS_DEFAULT_
</DL>
<HR>

<A NAME="DS_DEFAULT_POOLSUFFIX"><!-- --></A><H3>
DS_DEFAULT_POOLSUFFIX</H3>
<PRE>
public static final java.lang.String <B>DS_DEFAULT_POOLSUFFIX</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.DS_DEFAULT_
</DL>
<HR>

<A NAME="DS_DEFAULT_TYPE"><!-- --></A><H3>
DS_DEFAULT_TYPE</H3>
<PRE>
public static final java.lang.String <B>DS_DEFAULT_TYPE</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.DS_DEFAULT_
</DL>
<HR>

<A NAME="WLSADMIN"><!-- --></A><H3>
WLSADMIN</H3>
<PRE>
public static final java.lang.String <B>WLSADMIN</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.WLSADMIN">C
</DL>
<HR>

<A NAME="WLSADMINPWD"><!-- --></A><H3>
WLSADMINPWD</H3>
<PRE>
public static final java.lang.String <B>WLSADMINPWD</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.WLSADMINPWD
</DL>
<HR>

<A NAME="VENDORNAME"><!-- --></A><H3>
VENDORNAME</H3>
<PRE>
public static final java.lang.String <B>VENDORNAME</B></PRE>
<DL>
<DL>
<DT><B>See Also:</B><DD><A HREF="../../constant-values.html#dbScript.util.Constants.VENDORNAME "
</DL>

<!-- ===== CONSTRUCTOR DETAIL ===== -->
```

navExtensionDevKit.html Sample

```

<!-- ===== METHOD DETAIL ===== -->

<!-- ===== END OF CLASS DATA ===== -->
<HR>

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_bottom"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_bottom_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../overview-summary.html"><FONT CLA
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-summary.html"><FONT CLASS="Nav
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;&nbsp;&nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../deprecated-list.html"><FONT CLAS
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../index-all.html"><FONT CLASS="Nav
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../help-doc.html"><FONT CLASS="NavB
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;&nbsp;&nbsp;PREV CLASS&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;NEXT CLASS</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../../index.html" TARGET="_top"><B>FRAMES</B></A>   &nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;<A HREF="Constants.html" TARGET="_top"><B>NO FRAMES</B></A>   &nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>')
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>

<TR>
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
  SUMMARY:&nbsp;&nbsp;&nbsp;NESTED&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#field_summary">FIELD</A>&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;CONSTR&nbsp;&nbsp;&nbsp;|
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
  DETAIL:&nbsp;&nbsp;&nbsp;<A HREF="#field_detail">FIELD</A>&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;CONSTR&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;METHOD</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>

</BODY>
</HTML>

```


DsUtils.html Sample

This topic includes the source code for the DsUtils.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 14:44:38 PDT 2003 -->
<TITLE>
DsUtils
</TITLE>
<META NAME="keywords" CONTENT="dbScript.util.DsUtils,DsUtils class">
<LINK REL ="stylesheet" TYPE="text/css" HREF="../../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="DsUtils";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../overview-summary.html"><FONT CLA
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-summary.html"><FONT CLASS="Nav
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../deprecated-list.html"><FONT CLAS
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../index-all.html"><FONT CLASS="Nav
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../help-doc.html"><FONT CLASS="NavB
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;&nbsp;&nbsp;PREV CLASS&nbsp;&nbsp;&nbsp;
```

navExtensionDevKit.html Sample

```

&nbsp;NEXT CLASS</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../../index.html" TARGET="_top"><B>FRAMES</B></A> &nbsp;
&nbsp;<A HREF="DsUtils.html" TARGET="_top"><B>NO FRAMES</B></A> &nbsp;
&nbsp;
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>')
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
<TR>
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
  SUMMARY:&nbsp;&nbsp;&nbsp;NESTED&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#constructor_summary">CONSTR</A>&nbsp;&nbsp;&nbsp;
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">
DETAIL:&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#constructor_detail">CONSTR</A>&nbsp;&nbsp;&nbsp;|&nbsp;&nbsp;&nbsp;<A HREF="#m
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>
<!-- ===== START OF CLASS DATA ===== -->
<H2>
<FONT SIZE="-1">
dbScript.util</FONT>
<BR>
Class DsUtils</H2>
<PRE>
java.lang.Object
|
+--<B>dbScript.util.DsUtils</B>
</PRE>
<HR>
<DL>
<DT>public class <B>DsUtils</B><DT>extends java.lang.Object</DL>

<P>
<HR>

<P>
<!-- ===== NESTED CLASS SUMMARY ===== -->

<!-- ===== FIELD SUMMARY ===== -->

<!-- ===== CONSTRUCTOR SUMMARY ===== -->

<A NAME="constructor_summary"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Constructor Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">

```

```
<TD><CODE><B><A HREF="../../dbScript/util/DsUtils.html#DsUtils()">DsUtils</A></B>()</CODE>
```

```
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</TD>  
</TR>  
</TABLE>  
  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
<!-- ===== METHOD SUMMARY ===== -->  
  
<A NAME="method_summary"><!-- --></A>  
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">  
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">  
<TD COLSPAN=2><FONT SIZE="+2">  
<B>Method Summary</B></FONT></TD>  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/DsUtils.html#createDataSource(java.lang.String, java.  
          java.lang.String&nbsp;&nbsp;&nbsp;&nbsp;&baseName,  
          java.lang.String[]&nbsp;&nbsp;&nbsp;&urlValues,  
          java.util.Properties&nbsp;&nbsp;&nbsp;&p,  
          java.lang.String&nbsp;&nbsp;&nbsp;&passwordToEncrypt,  
          java.lang.String&nbsp;&nbsp;&nbsp;&wlsAdminUser,  
          java.lang.String&nbsp;&nbsp;&nbsp;&wlsAdminPassword,  
          boolean&nbsp;&nbsp;&nbsp;&fCreateNonTxDs)</CODE>  
  
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</TD>  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/DsUtils.html#getDatasourceJNDINames(MBeanHome)">getDa  
  
<BR>  
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</TD>  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/DsUtils.html#getDriverList()">getDriverList</A></B>( )  
  
<BR>  
&nbsp;&nbsp;&nbsp;&~</TD>  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>&nbsp;&nbsp;&nbsp;&~</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/DsUtils.html#getDriverProperties(java.lang.String)">g  
  
<BR>  
&nbsp;&nbsp;&~</TD>  
</TR>  
<TR BGCOLOR="white" CLASS="TableRowColor">  
<TD ALIGN="right" VALIGN="top" WIDTH="1%"><FONT SIZE="-1">  
<CODE>&nbsp;&nbsp;&~</CODE></FONT></TD>  
<TD><CODE><B><A HREF="../../dbScript/util/DsUtils.html#getDriverURLDefaults(java.lang.String)">  
  
<BR>  
&nbsp;&~</TD>
```

[illegible]

navExtensionDevKit.html Sample

```
DsUtils</H3>
<PRE>
public <B>DsUtils</B>()</PRE>
<DL>
</DL>

<!-- ===== METHOD DETAIL ===== -->

<A NAME="method_detail"><!-- --></A>
<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=1><FONT SIZE="+2">
<B>Method Detail</B></FONT></TD>
</TR>
</TABLE>

<A NAME="getDriverList()"><!-- --></A><H3>
getDriverList</H3>
<PRE>
public java.lang.String[] <B>getDriverList</B>()</PRE>
<DL>
<DD><DL>
</DL>
</DD>
</DL>
<HR>

<A NAME="getDriverURLPrompts(java.lang.String)"><!-- --></A><H3>
getDriverURLPrompts</H3>
<PRE>
public java.lang.String[] <B>getDriverURLPrompts</B>(java.lang.String&nbsp;driverType)</PRE>
<DL>
<DD><DL>
</DL>
</DD>
</DL>
<HR>

<A NAME="getDriverURLDefaults(java.lang.String)"><!-- --></A><H3>
getDriverURLDefaults</H3>
<PRE>
public java.lang.String <B>getDriverURLDefaults</B>(java.lang.String&nbsp;driverType)</PRE>
<DL>
<DD><DL>
</DL>
</DD>
</DL>
<HR>

<A NAME="getDriverProperties(java.lang.String)"><!-- --></A><H3>
getDriverProperties</H3>
<PRE>
public java.lang.String <B>getDriverProperties</B>(java.lang.String&nbsp;driverType)</PRE>
<DL>
<DD><DL>
</DL>
</DD>
</DL>
<HR>

<A NAME="getPropertiesFromString(java.lang.String)"><!-- --></A><H3>
```

navExtensionDevKit.html Sample

```
getPropertiesFromString</H3>
<PRE>
public java.util.Properties <B>getPropertiesFromString</B>( java.lang.String&nbsp;props)</PRE>
<DL>
<DD><DL>
</DL>
</DD>
</DL>
<HR>

<A NAME="createDataSource( java.lang.String, java.lang.String, java.lang.String[], java.util.Pro
createDataSource</H3>
<PRE>
public java.lang.String <B>createDataSource</B>( java.lang.String&nbsp;driverType,
                                java.lang.String&nbsp;baseName,
                                java.lang.String[ ]&nbsp;urlValues,
                                java.util.Properties&nbsp;p,
                                java.lang.String&nbsp;passwordToEncrypt,
                                java.lang.String&nbsp;wlsAdminUser,
                                java.lang.String&nbsp;wlsAdminPassword,
                                boolean&nbsp;fCreateNonTxDs)
                                throws java.lang.Exception</PRE>

<DL>
<DD><DL>

<DD><CODE>java.lang.Exception</CODE></DL>
</DD>
</DL>
<HR>

<A NAME="getDatasourceJNDINames(MBeanHome)"><!-- --></A><H3>
getDatasourceJNDINames</H3>
<PRE>
public java.util.Collection <B>getDatasourceJNDINames</B>(MBeanHome&nbsp;home)
                                throws java.lang.Exception</PRE>

<DL>
<DD><DL>

<DD><CODE>java.lang.Exception</CODE></DL>
</DD>
</DL>
<HR>

<A NAME="getTypeForDSName( java.lang.String)"><!-- --></A><H3>
getTypeForDSName</H3>
<PRE>
public java.lang.String <B>getTypeForDSName</B>( java.lang.String&nbsp;dsName)
                                throws java.lang.Exception</PRE>

<DL>
<DD><DL>

<DD><CODE>java.lang.Exception</CODE></DL>
</DD>
</DL>
<HR>

<A NAME="getDsName( java.lang.String, boolean)"><!-- --></A><H3>
getDsName</H3>
<PRE>
public java.lang.String <B>getDsName</B>( java.lang.String&nbsp;baseName,
                                boolean&nbsp;fNoTx)</PRE>
```

```
<DL>  
<DD><DL>  
</DL>  
</DD>  
</DL>  
  
<!-- ===== END OF CLASS DATA ===== -->  
<HR>  
  
<!-- ===== START OF NAVBAR ===== -->  
<A NAME="navbar_bottom"><!-- --></A>  
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">  
<TR>  
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">  
<A NAME="navbar_bottom_firstrow"><!-- --></A>  
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">  
    <TR ALIGN="center" VALIGN="top">  
        <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">          <A HREF="../../overview-summary.html"><FONT CLA  
        <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">          <A HREF="package-summary.html"><FONT CLASS="Nav  
        <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev">      &nbsp;&nbsp;&nbsp;<FONT CLASS="NavBarFont1Rev"><B>Class</B>  
        <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">          <A HREF="package-tree.html"><FONT CLASS="NavBar  
        <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">          <A HREF="../../deprecated-list.html"><FONT CLAS  
        <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">          <A HREF="../../index-all.html"><FONT CLASS="Nav  
        <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">          <A HREF="../../help-doc.html"><FONT CLASS="NavB  
    </TR>  
</TABLE>  
</TD>  
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>  
</EM>  
</TD>  
</TR>  
  
<TR>  
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">  
&nbsp;&nbsp;&nbsp;PREV CLASS&nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;NEXT CLASS</FONT></TD>  
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">  
    <A HREF="../../index.html" TARGET="_top"><B>FRAMES</B></A>   &nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;<A HREF="DsUtils.html" TARGET="_top"><B>NO FRAMES</B></A>   &nbsp;&nbsp;&nbsp;  
&nbsp;&nbsp;&nbsp;  
<SCRIPT>  
    <!--  
    if(window==top) {  
        document.writeln('<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>'  
    }  
    //-->  
</SCRIPT>  
<NOSCRIPT>  
<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>  
</NOSCRIPT>  
</FONT></TD>  
</TR>  
  
<TR>  
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">  
    SUMMARY:&nbsp;&nbsp;&nbsp;NESTED&nbsp;&nbsp;&nbsp;; |&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;; |&nbsp;&nbsp;&nbsp;<A HREF="#constructor_summary">CONSTR</A>&nbsp;&nbsp;&nbsp;  
<TD VALIGN="top" CLASS="NavBarCell3"><FONT SIZE="-2">  
DETAILS:&nbsp;&nbsp;&nbsp;FIELD&nbsp;&nbsp;&nbsp;; |&nbsp;&nbsp;&nbsp;<A HREF="#constructor_detail">CONSTR</A>&nbsp;&nbsp;&nbsp;; |&nbsp;&nbsp;&nbsp;<A HREF="#" m  
</TR>  
</TABLE>  
  
<!-- ===== END OF NAVBAR ===== -->  
  
<HR>
```

```
</BODY>  
</HTML>
```


Package-summary.html Sample

This topic includes the source code for the Package-summary.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" "http://www.w3.org/TR/REC-html40/1
<!--NewPage-->
<HTML>
<HEAD>
<!-- Generated by javadoc on Fri Oct 10 14:44:38 PDT 2003 -->
<TITLE>
dbScript.util
</TITLE>
<META NAME="keywords" CONTENT="dbScript.util package">
<LINK REL="stylesheet" TYPE="text/css" HREF="../../stylesheet.css" TITLE="Style">
</HEAD>
<SCRIPT>
function asd()
{
parent.document.title="dbScript.util";
}
</SCRIPT>
<BODY BGCOLOR="white" onload="asd();">

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_top"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_top_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
  <TR ALIGN="center" VALIGN="top">
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
    <TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<A HREF="../../overview-summary.html"><FONT CLA
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="package-tree.html"><FONT CLASS="NavBar
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="../../deprecated-list.html"><FONT CLAS
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="../../index-all.html"><FONT CLASS="Nav
    <TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1"> <A HREF="../../help-doc.html"><FONT CLASS="NavB
  </TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell12"><FONT SIZE="-2">
&nbsp;<A HREF="../../dbScript/package-summary.html"><B>PREV PACKAGE</B></A>&nbsp;</TD>
```

navExtensionDevKit.html Sample

```
&nbsp;NEXT PACKAGE</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../../index.html" TARGET="_top"><B>FRAMES</B></A>   &nbsp;
&nbsp;<A HREF="package-summary.html" TARGET="_top"><B>NO FRAMES</B></A>   &nbsp;
&nbsp;
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>')
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>
<H2>
Package dbScript.util
</H2>

<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Interface Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD WIDTH="15%"><B><A HREF="Constants.html"><I>Constants</I></A></B></TD>
<TD>&nbsp;</TD>
</TR>
</TABLE>
&nbsp;

<P>

<TABLE BORDER="1" CELLPADDING="3" CELLSPACING="0" WIDTH="100%">
<TR BGCOLOR="#CCCCFF" CLASS="TableHeadingColor">
<TD COLSPAN=2><FONT SIZE="+2">
<B>Class Summary</B></FONT></TD>
</TR>
<TR BGCOLOR="white" CLASS="TableRowColor">
<TD WIDTH="15%"><B><A HREF="DsUtils.html">DsUtils</A></B></TD>
<TD>&nbsp;</TD>
</TR>
</TABLE>
&nbsp;

<P>
<HR>

<!-- ===== START OF NAVBAR ===== -->
<A NAME="navbar_bottom"><!-- --></A>
<TABLE BORDER="0" WIDTH="100%" CELLPADDING="1" CELLSPACING="0">
<TR>
<TD COLSPAN=3 BGCOLOR="#EEEEFF" CLASS="NavBarCell1">
<A NAME="navbar_bottom_firstrow"><!-- --></A>
<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="3">
```

navExtensionDevKit.html Sample

```

<TR ALIGN="center" VALIGN="top">
<TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../overview-summary.html"><FONT CLA
<TD BGCOLOR="#FFFFFF" CLASS="NavBarCell1Rev"> &nbsp;<FONT CLASS="NavBarFont1Rev"><B>Package</
<TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <FONT CLASS="NavBarFont1">Class</FONT>&nbsp;</T
<TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="package-tree.html"><FONT CLASS="NavBar
<TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../deprecated-list.html"><FONT CLAS
<TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../index-all.html"><FONT CLASS="Nav
<TD BGCOLOR="#EEEEFF" CLASS="NavBarCell1">      <A HREF="../../help-doc.html"><FONT CLASS="NavB
</TR>
</TABLE>
</TD>
<TD ALIGN="right" VALIGN="top" ROWSPAN=3><EM>
</EM>
</TD>
</TR>

<TR>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
&nbsp;<A HREF="../../dbScript/package-summary.html"><B>PREV PACKAGE</B></A>&nbsp;<
&nbsp;<NEXT PACKAGE</FONT></TD>
<TD BGCOLOR="white" CLASS="NavBarCell2"><FONT SIZE="-2">
  <A HREF="../../index.html" TARGET="_top"><B>FRAMES</B></A>   &nbsp;<
&nbsp;<A HREF="package-summary.html" TARGET="_top"><B>NO FRAMES</B></A>   &nbsp;<
&nbsp;<
<SCRIPT>
  <!--
  if(window==top) {
    document.writeln('<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>'
  }
  //-->
</SCRIPT>
<NOSCRIPT>
<A HREF="../../allclasses-noframe.html" TARGET=""><B>All Classes</B></A>
</NOSCRIPT>
</FONT></TD>
</TR>
</TABLE>
<!-- ===== END OF NAVBAR ===== -->

<HR>

</BODY>
</HTML>

```

javadoc-tag Samples

This section contains source code for the following samples.

Samples Included in This Section

jc Samples

jc Samples

This section contains source code for the following samples.

Samples Included in This Section

Dbscr-connection.html Sample

Dbscript.html Sample

Dbscr-connection.html Sample

This topic includes the source code for the Dbscr-connection.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!doctype HTML public "-//W3C//DTD HTML 4.0 Frameset//EN">

<html><!-- InstanceBegin template="/Templates/Tag.dwt" codeOutsideHTMLOutsideIsLocked="false" -->
<head>
<!-- InstanceBeginEditable name="doctitle" -->
<title>@jc:dbscr-connection Annotation</title>
<!-- InstanceEndEditable -->

<!--(Meta)=====-->
<meta http-equiv=Content-Type content="text/html; charset=$CHARSET;">
<meta name=workshop content=WWKSHOP>
<!-- InstanceBeginEditable name="metatags" -->
<!-- In the component metatag, please indicate whether this topic is for Integration, Workshop,
<meta name="component" content>
<!-- In the description metatag, please provide a BRIEF description of the topic contents. -->
<meta name="description" content>
<!-- In the component metatag, please list keywords that will help a user search for this topic
<meta name="keywords" content>

<!-- InstanceEndEditable -->

<!--(Links)=====-->
<!-- InstanceBeginEditable name="head" --><!-- InstanceEndEditable -->
<link href="../../../workshop.css" rel="stylesheet" type="text/css">
<a href="../../../core/index.html" id="index"></a>
<script language="JavaScript" src="../../../core/topicInfo.js"></script>
<script language="JavaScript" src="../../../core/CookieClass.js"></script>
<script language="JavaScript" src="../../../core/displayContent.js"></script>
</head>

<!--(Body)=====-->
<body>
<script language="JavaScript">
displayInFrames();
</script>
<!-- InstanceBeginEditable name="body" -->
<div id="topicTitle">
  <h1 class="Title">@jc:dbscr-connection Annotation</h1>
</div>
<div id="topicText">
  <p>Specifies the database connection parameters for a DBScriptRunner control.</p>
  <h2>Syntax</h2>
  <p class="syntax">@jc:dbscr-connection</p>
  <p class="syntaxindent">data-source-jndi-name=&quot;<span class="syntaxpartname">&lt;jndi-name
  <p class="syntaxindent">batch-size=&quot;<span class="syntaxpartname">&lt;int&gt;&quot;</span>&quot;</p>
```

navExtensionDevKit.html Sample

```
<h2>Attributes</h2>
<p class="attribute"><a name="data-source-jndi-name"></a>data-source-jndi-name</p>
<p class="partdesc">Specifies the JNDI name of the datasource to be used by this DBScriptRunner</p>
<p class="partdesc">Default: <span class="langinline">none</span>.</p>
<p class="attribute"><a name="batch-size"></a>batch-size</p>
<p class="partdesc">The number of INSERT statements the control will batch in one SQL command</p>
<p class="partdesc">Default: <span class="langinline">1</span>.</p>
<h2>Remarks</h2>
<p>The following rules apply to this annotation's use:</p>
</div>
<ul>
  <li>
    <div>
      A single <span class="langinline">@jc:dbscr-connection</span> annotation may appear within a single Javadoc comment block.
    </div>
  </li>
  <li>
    <div>
      Required in front of the interface in a DBScriptRunner control JCX file.
    </div>
  </li>
</ul>
<div>
  <!-- class=content -->
  <p class="relatedtopics">Related Topics</p>
  <p><a href="../../navDBScriptRunner.html">DBScriptRunner Control</a></p>
  <p><a href="dbscript.html">@jc:dbscript Annotation</a></p>
  <!-- class=content -->
</div>
<!-- InstanceEndEditable -->
<script language="JavaScript">
writeTopicInfo();
</script>
</body>
<!-- InstanceEnd --></html>
```

Dbscript.html Sample

This topic includes the source code for the Dbscript.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!doctype HTML public "-//W3C//DTD HTML 4.0 Frameset//EN">

<html><!-- InstanceBegin template="/Templates/Tag.dwt" codeOutsideHTMLOIsLocked="false" -->
<head>
<!-- InstanceBeginEditable name="doctitle" -->
<title>@jc:dbscript Annotation</title>
<!-- InstanceEndEditable -->

<!--(Meta)=====-->
<meta http-equiv=Content-Type content="text/html; charset=$CHARSET;">
<meta name=workshop content=WWKSHOP>
<!-- InstanceBeginEditable name="metatags" -->
<!-- In the component metatag, please indicate whether this topic is for Integration, Workshop,
<meta name="component" content>
<!-- In the description metatag, please provide a BRIEF description of the topic contents. -->
<meta name="description" content>
<!-- In the component metatag, please list keywords that will help a user search for this topic
<meta name="keywords" content>

<!-- InstanceEndEditable -->

<!--(Links)=====-->
<!-- InstanceBeginEditable name="head" --><!-- InstanceEndEditable -->
<link href="../../../workshop.css" rel="stylesheet" type="text/css">
<a href="../../../core/index.html" id="index"></a>
<script language="JavaScript" src="../../../core/topicInfo.js"></script>
<script language="JavaScript" src="../../../core/CookieClass.js"></script>
<script language="JavaScript" src="../../../core/displayContent.js"></script>
</head>

<!--(Body)=====-->
<body>
<script language="JavaScript">
displayInFrames();
</script>
<!-- InstanceBeginEditable name="body" -->
<div id="topicTitle">
  <h1 class="Title">@jc:dbscript Annotation</h1>
</div>
<div id="topicText">
  <p>Annotation on a method of a DBScriptRunner control JCX methods that determine whether the
  <h2>Syntax</h2>

  <p class="syntax">@jc:dbscript</p>
```


navExtensionDevKit.html Sample

```
<p class="syntaxindent">new-data-source-type=&quot;<span class="syntaxpartname">&lt;data-sour
<p class="syntaxindent">new-data-source-name=&quot;<span class="syntaxpartname">&lt;data-sour
<p class="syntaxindent">data-source-url-defaults=&quot;<span class="syntaxpartname">&lt;data-
<p class="syntaxindent">data-source-properties=&quot;<span class="syntaxpartname">&lt;data-so
<p class="syntaxindent">prompt-for-url=&quot;<span class="syntaxpartname">true</span> | <span
<p class="syntaxindent">prompt-for-username=&quot;<span class="syntaxpartname">true</span> |
<p class="syntaxindent">create-notx-datasource=&quot;<span class="syntaxpartname">true</span>
<p class="syntaxindent">wls-admin-user=&quot;<span class="syntaxpartname">&lt;username&gt;</s
<p class="syntaxindent">wls-admin-pwd=&quot;<span class="syntaxpartname">&lt;password&gt;</sp
<p class="syntaxindent">script-file=&quot;<span class="syntaxpartname">&lt;file-path&gt;</spa

<h2>Attributes</h2>

<p class="attribute"><a name="new-data-source-type"></a>new-data-source-type</p>
<p class="partdesc">Type of datasource to create. If present, signals that this method define
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<p class="attribute"><a name="new-data-source-name"></a>new-data-source-name</p>
<p class="partdesc">Name of datasource to create.</p>
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<p class="attribute"><a name="data-source-url-defaults"></a>data-source-url-defaults</p>
<p class="partdesc">URL of datasource to create, if not provided as method parameters.</p>
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<p class="attribute"><a name="data-source-properties"></a>data-source-properties</p>
<p class="partdesc">Properties of datasource to create, if not provided as method parameters.
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<p class="attribute"><a name="prompt-for-url"></a>prompt-for-url</p>
<p class="partdesc">True if parts of URL for datasource are not provided as method parameters
<p class="partdesc">Default: <span class="langinline">>false</span>.</p>

<p class="attribute"><a name="prompt-for-username"></a>prompt-for-username</p>
<p class="partdesc">True if database username and password for datasource are not provided as
<p class="partdesc">Default: <span class="langinline">>false</span>.</p>

<p class="attribute"><a name="create-notx-datasource"></a>create-notx-datasource</p>
<p class="partdesc">Whether to create a second datasource that does not participate in global
<p class="partdesc">Default: <span class="langinline">>false</span>.</p>

<p class="attribute"><a name="wls-admin-user"></a>wls-admin-user</p>
<p class="partdesc">Username of WebLogic Server admin with permission to create datasources.
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<p class="attribute"><a name="wls-admin-pws"></a>wls-admin-pws</p>
<p class="partdesc">Password of WebLogic Server admin with permission to create datasources.
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<p class="attribute"><a name="script-file"></a>script-file</p>
<p class="partdesc">A SQL script file compiled into the jar.</p>
<p class="partdesc">Default: <span class="langinline">none</span>.</p>

<h2>Remarks</h2>
<p>The following rules apply to this annotation's use:</p>
</div>
<ul>
<li>
<div>
A single <span class="langinline">@jc:dbscript</span> annotation may
appear within a single Javadoc comment block.

```

navExtensionDevKit.html Sample

```
</div>
</li>
<li>
  <div>
    Required in front of each method in a DBScriptRunner control JCX file.
  </div>
</li>
</ul>
</ul>
<div>
  <!-- class=content -->
  <p class="relatedtopics">Related Topics</p>
  <p><a href="dbscr-connection.html">@jc:dbscr-connection Annotation</a></p>
  <p><a href="../../navDBScriptRunner.html">DBScriptRunner Control</a></p>
  <!-- class=content -->
</div>
<!-- InstanceEndEditable -->
<script language="JavaScript">
writeTopicInfo();
</script>
</body>
<!-- InstanceEnd --></html>
```

NavDBScriptRunner.html Sample

This topic includes the source code for the NavDBScriptRunner.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!doctype HTML public "-//W3C//DTD HTML 4.0 Frameset//EN">

<html><!-- InstanceBegin template="/Templates/Navigational.dwt" codeOutsideHTMLIsLocked="false" -->
<head>
<!-- InstanceBeginEditable name="doctitle" -->
<title>DBScriptRunner Control</title>
<!-- InstanceEndEditable -->

<!--(Meta)=====-->
<meta http-equiv=Content-Type content="text/html; charset=$CHARSET;">
<meta name=workshop content=WWKSHOP>
<!-- InstanceBeginEditable name="metatags" -->
<!-- In the component metatag, please indicate whether this topic is for Integration, Workshop,
<meta name="component" content="Workshop">
<!-- In the description metatag, please provide a BRIEF description of the topic contents. -->
<meta name="description" content>
<!-- In the component metatag, please list keywords that will help a user search for this topic
<meta name="keywords" content>

<!-- InstanceEndEditable -->

<!--(Links)=====-->
<!-- InstanceBeginEditable name="head" --><!-- InstanceEndEditable -->
<link href="../../../workshop.css" rel="stylesheet" type="text/css">
<a href="../../../core/index.html" id="index"></a>
<script language="JavaScript" src="../../../core/topicInfo.js"></script>
<script language="JavaScript" src="../../../core/CookieClass.js"></script>
<script language="JavaScript" src="../../../core/displayContent.js"></script>
</head>

<!--(Body)=====-->
<body>
<script language="JavaScript">
displayInFrames();
</script>
<!-- InstanceBeginEditable name="body" -->

<div id="topicctitle">
  <h1 class="Title">DBScriptRunner Control</h1>
</div>
<div id="topicctext">
  <p>The topics in this section address the basic concepts you need to understand
    to use the DBScriptRunner Control.</p>
  <h1>Topics Included in This Section</h1>
```

navExtensionDevKit.html Sample

```
<p class=navtitle><a href="ovwDBScriptRunner.html">DBScriptRunner Control Overview</a></p>
<p class=navdesc>How to use the DBScriptRunner control.</p>

<p class=navtitle><a href="conDBScriptRunnerControlWizard.html">DBScriptRunner Control Wiza
<p class=navdesc>How to use the DBScriptRunner control's wizard in the WebLogic Workshop ID

<p class=navtitle><a href="java-class/dbScript/DBScriptRunner.html">DBScriptRunner Interfac
<p class=navdesc>Javadoc for the DBScriptRunner interface.</p>

<p class=navtitle><a href="java-class/dbScript/DBScriptRunnerImpl.html">DBScriptRunnerImpl
<p class=navdesc>Javadoc for the DBScriptRunnerImpl class.</p>

<p class=navtitle><a href="javadoc-tag/jc/dbscr-connection.html">The @jc:dbscr-connection A
<p class=navdesc>Describes the attributes of the DBScriptRunner control's @jc:dbscr-connece

<p class=navtitle><a href="javadoc-tag/jc/dbscript.html">The @jc:dbscript Annotation</a></p>
<p class=navdesc>Describes the attributes of the DBScriptRunner control's @jc:dbscript tag

</div>
<!-- InstanceEndEditable -->
<script language="JavaScript">
writeTopicInfo();
</script>
</body>

<!-- InstanceEnd --></html>
```

OvwDBScriptRunner.html Sample

This topic includes the source code for the OvwDBScriptRunner.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<!doctype HTML public "-//W3C//DTD HTML 4.0 Frameset//EN">

<html><!-- InstanceBegin template="/Templates/Navigational.dwt" codeOutsideHTMLOverview="false" -->
<head>
<!-- InstanceBeginEditable name="doctitle" -->
<title>DBScriptRunner Control Overview</title>
<!-- InstanceEndEditable -->

<!--(Meta)=====-->
<meta http-equiv=Content-Type content="text/html; charset=$CHARSET;">
<meta name=workshop content=WWKSHOP>
<!-- InstanceBeginEditable name="metatags" -->
<!-- In the component metatag, please indicate whether this topic is for Integration, Workshop,
<meta name="component" content="Workshop">
<!-- In the description metatag, please provide a BRIEF description of the topic contents. -->
<meta name="description" content>
<!-- In the component metatag, please list keywords that will help a user search for this topic
<meta name="keywords" content>

<!-- InstanceEndEditable -->

<!--(Links)=====-->
<!-- InstanceBeginEditable name="head" --><!-- InstanceEndEditable -->
<link href="../../workshop.css" rel="stylesheet" type="text/css">
<a href="../../core/index.html" id="index"></a>
<script language="JavaScript" src="../../core/topicInfo.js"></script>
<script language="JavaScript" src="../../core/cookieClass.js"></script>
<script language="JavaScript" src="../../core/displayContent.js"></script>
</head>

<!--(Body)=====-->
<body>
<script language="JavaScript">
displayInFrames();
</script>
<!-- InstanceBeginEditable name="body" -->

<div id="topicTitle">
<h1 class="Title">DBScriptRunner Control Overview</h1>
</div>
<div id="topicText">
<p>A DBScriptRunner control...</p>
<p class="relatedTopics">Related Topics</p>
<p><a href="java-class/dbScript/DBScriptRunner.html">DBScriptRunner Class</a></p>
```

navExtensionDevKit.html Sample

```
<p><a href="java-class/dbScript/DBScriptRunnerImpl.html">DBScriptRunnerImpl Class</a></p>
</div>
<!-- InstanceEndEditable -->
<script language="JavaScript">
writeTopicInfo();
</script>
</body>

<!-- InstanceEnd --></html>
```

Toc.xml Sample

This topic includes the source code for the Toc.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/help/doc/en/partners/I

Sample Source Code

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- CDK TOC -->
<toc-root component="other" xmlns="http://www.bea.com/help/toc.xsd">
  <toc-reference anchor="extensions">
    <toc-node label="DBScriptRunner Example Control" url="navDBScriptRunner.html">
      <toc-node label="DBScriptRunner Control Overview" url="ovwDBScriptRunner.html"/>
      <toc-node label="DBScriptRunner Control Wizard" url="conDBScriptRunnerControlWizard.html"/>
      <toc-node label="DBScriptRunner Class" url="./java-class/dbScript/DBScriptRunner.html"/>
      <toc-node label="DBScriptRunnerImpl Class" url="./java-class/dbScript/DBScriptRunnerImpl.html"/>
      <toc-node label="@jc:dbscr-connection Annotation" url="./javadoc-tag/jc/dbscr-connection.html"/>
      <toc-node label="@jc:dbscript Annotation" url="./javadoc-tag/jc/dbscript.html"/>
    </toc-node>
  </toc-reference>
</toc-root>
```

samples Samples

This section contains source code for the following samples.

Samples Included in This Section

partners Samples

partners Samples

This section contains source code for the following samples.

Samples Included in This Section

DBScripter Samples

DBScripter Samples

This section contains source code for the following samples.

Samples Included in This Section

DBScripterSample Samples

DBScripterSample Samples

This section contains source code for the following samples.

Samples Included in This Section

ScriptTest Samples

ScriptTest Samples

This section contains source code for the following samples.

Samples Included in This Section

test Samples

test Samples

This section contains source code for the following samples.

Samples Included in This Section

DBScriptTest.jws Sample

DsPointBaseTest.jcx Sample

DBScriptTest.jws Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/samples/partners/DBS

Sample Source Code

```
package test;

import com.bea.control.JwsContext;

public class DBScriptTest implements com.bea.jws.WebService
{
    /**
     * @common:control
     */
    private test.DsPointBaseTest dsPointBaseTest;

    // because "samples" needs to be part of the package name

    public Callback callback;

    /** @common:context */
    JwsContext context;

    public interface Callback
    {
        /**
         * @jws:conversation phase="finish"
         */
        void onScriptDone(java.lang.String arg0);
    }

    /**
     * This method creates two new Data Sources to the existing PointBase database.  These data
     * DsPointBaseTest and DsPointBaseTestNoTx.  The "NoTx" data source sets the flag to not pa
     * in global transactions.
     * <br>
     * Suggested parameters:
     * <br><b>Server: </b>localhost
     * <br><b>Port: </b>9093 (workshop domain)
     * <br><b>Database: </b>workshop (workshop domain)
     * <br><b>User: </b>PBPUBLIC
     * <br><b>Password: </b>PBPUBLIC
     * @common:operation
     */
    public java.lang.String createDsPointBaseTest(java.lang.String server, java.lang.String por
    {
        return dsPointBaseTest.createDsPointBaseTest(server,port,database,user,password);
    }
}
```

```

/**
 * This method runs a script against the data source.
 * The value is assumed to be an absolute file path.
 * @common:operation
 * @jws:conversation phase="start"
 */
public void runFileScript(java.lang.String filePath)
{
    dsPointBaseTest.runFileScript(filePath);
}

public void dsPointBaseTest_onScriptDone(java.lang.String arg0)
{
    callback.onScriptDone(arg0);
}

/**
 * In this method the path is stored as an annotation.
 * It refers to a resource file in the control jar
 * <br>
 * @common:operation
 * @jws:conversation phase="start"
 */
public void runScriptInstnwind()
{
    dsPointBaseTest.runScriptInstnwind();
}
}

```

DsPointBaseTest.jcx Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/ControlDevKit/DBScripter/docs/samples/partners/DBS

Sample Source Code

```
package test;

import com.bea.control.*;
/**
 * @jc:dbscr-connection data-source-jndi-name="DsPointBaseTestNoTx"
 */
public interface DsPointBaseTest extends dbScript.DBScriptRunner, com.bea.control.ControlExtension {
    /**
     * @jc:dbscript new-data-source-type="PointBase"
     *          data-source-url-defaults="//localhost:9093;workshop"
     *          data-source-properties="user=PBPUBLIC;password=PBPUBLIC"
     *          prompt-for-url=true
     *          prompt-for-username=true
     *          wls-admin-user="weblogic"
     *          wls-admin-pwd="weblogic"
     *          new-data-source-name="DsPointBaseTest"
     *          create-notx-data-source=true
     */
    public String createDsPointBaseTest( String server, String port, String database, String user );

    /**
     * @jc:dbscript script-file="/dbScript/scripts/PointBase/instnwind.sql"
     *          @common:message-buffer enable="true"
     */
    public void runScriptInstnwind();

    /**
     *          @common:message-buffer enable="true"
     */
    public void runFileScript(String filePath);

    interface Callback extends dbScript.DBScriptRunner.Callback {
        void onScriptDone(String msg);
    }
}
```


IdeDevKit Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomProject Samples

DragDropSimple Samples

FrameViewSimple Samples

MenuItems Samples

PopupAction Samples

PropertyListener Samples

SearchResults Samples

TocView Samples

ToolbarButton Samples

CustomProject Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

php_template Samples

src Samples

xml Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

CustomProject Samples

CustomProject Samples

This section contains source code for the following samples.

Samples Included in This Section

php_template Samples

This section contains source code for the following samples.

Samples Included in This Section

basic Samples

Template.xml Sample

basic Samples

This section contains source code for the following samples.

Samples Included in This Section

scripts Samples

style Samples

scripts Samples

This section contains source code for the following samples.

Samples Included in This Section

style Samples

This section contains source code for the following samples.

Samples Included in This Section

Template.xml Sample

This topic includes the source code for the Template.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/php_template/

Sample Source Code

```
<template-definition>

  <!-- Web Project: non-diplay base template -->
  <project-template id="php"
    type="urn:com-bea-ide:project.type:php">
    <content type="archive" destination="project" source="basic.zip" />
    <display
      location="importdialog,newdialog"
      label="Php Project"
      description="Creates a new PHP project."
      priority="10"
      categories="Web Design" />
    </project-template>
  </template-definition>
```

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

customProject Samples

customProject Samples

This section contains source code for the following samples.

Samples Included in This Section

PhpDocument.java Sample

PhpDocumentImpl.java Sample

PhpProjectBuildDriver.java Sample

PhpProjectDriver.java Sample

PhpProjectListener.java Sample

PhpProjectRunDriver.java Sample

PhpProjectRunPreferences.java Sample

PhpDocument.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.document.DefaultDocumentHandler;
import com.bea.ide.document.IDocument;
import com.bea.ide.lang.java.JavaDocument;
import com.bea.ide.lang.java.JavaDocumentFactory;

import java.net.URI;

/**
 * A simple handler for a PHP document. A document handler
 * is responsible for relatively high-level functions, such as
 * validating the name of a file that corresponds to the document,
 * providing the file's extension, and so on. In contrast, the document
 * implementation (represented in this sample by PhpDocumentImpl)
 * is responsible for lower-level functions, such as providing
 * views of the document's contents.
 *
 * Note that a "document" (represented by an IDocument instance)
 * is an abstraction that the IDE uses to provide special
 * handling of certain files. For example, a JWS file provides
 * its graphical Design View because its IDocument provides one.
 */
public class PhpDocument extends DefaultDocumentHandler
{
    /**
     * Constructs a new instance of this handler by calling
     * its super with the extension to use for files
     * corresponding to the document type: .php
     */
    public PhpDocument()
    {
        super("php");
    }

    /**
     * Called by the IDE to retrieve the IDocument instance that will
     * be used to represent a PHP document. This method returns an instance
     * of PhpDocumentImpl, which extends TextDocument, which in turn
     * implements IDocument. That instance is responsible for low-level
     * document tasks such as rendering a view of its contents, managing
     * edits, and so on.
     *
     * @param uriFile A URI pointing at the file that corresponds to this
     * document.
     */
}
```

navExtensionDevKit.html Sample

```
* @return A PhpDocumentImpl instance representing the document.
*/
    public IDocument createDocumentObject(Uri uriFile) throws java.io.IOException
    {
        IDocument ret = new PhpDocumentImpl(this.getClass(), uriFile);
        return ret;
    }
}
```

PhpDocumentImpl.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.core.navigation.INavigationPoint;
import com.bea.ide.document.IDocument;
import com.bea.ide.document.IDocumentView;
import com.bea.ide.document.IDocumentViewInfo;
import com.bea.ide.lang.java.JavaDocument;
import com.bea.ide.lang.text.TextDocument;
import com.bea.ide.sourceeditor.EditorSvc;
import com.bea.ide.sourceeditor.driver.ISourceViewDriver;
import java.awt.Container;
import java.io.IOException;
import java.net.URI;
import java.util.ArrayList;
import javax.swing.Icon;

/**
 * A simple document implementation to manage the contents of
 * a PHP document, providing names for views of the document and
 * deferring to TextDocument for all other functionality.
 */
public class PhpDocumentImpl extends TextDocument
{
    /**
     * Constructs an instance of this class with the
     * handler class and a URI to the file that is a physical
     * instance of the document. This merely calls the
     * TextDocument constructor.
     */
    public PhpDocumentImpl(Class handler, URI uriFile) throws IOException
    {
        super(handler, uriFile);
    }

    /**
     * Called by the IDE to retrieve the views of a PHP
     * document. This document implementation supports "two" views,
     * source view and design view, although the two are
     * identical. The class used to define these views is
     * contained within this class.
     */
    public IDocumentViewInfo[] getViewInfo()
    {
        IDocumentViewInfo[] views = new IDocumentViewInfo[2];
        PhpViewInfo viewInfo = new PhpViewInfo(this);
    }
}
```

navExtensionDevKit.html Sample

```
viewInfo.setName("PHP Source View");
views[0] = viewInfo;
viewInfo = new PhpViewInfo(this);
viewInfo.setName("PHP Design View");
views[1] = viewInfo;
return views;
}

/**
 * A simple way to provide a view of PHP files.
 * The getViewInfo method returns instances of this class
 * to provide two view tabs: PHP Source View and
 * PHP Design View. The views are identical, of course,
 * but this code illustrates how you respond to the IDE's
 * request for views of a document.
 */
class PhpViewInfo implements IDocumentViewInfo
{
    TextDocument m_doc;
    String m_name;

    /**
     * Constructs an instance of this view using the
     * document handler class that contains this class.
     *
     * @param doc The document handler to build a
     * view of.
     */
    public PhpViewInfo(TextDocument doc)
    {
        m_doc = doc;
    }

    /**
     * Sets the name of the view as it should
     * appear on the view's tab in the IDE.
     *
     * @param name The name to use for the view.
     */
    public void setName(String name)
    {
        m_name = name;
    }

    /**
     * Called by the IDE to get the name of the view.
     */
    public String getName()
    {
        return m_name;
    }

    /**
     * Called by the IDE to get the icon that should
     * be displayed for this view.
     */
    public Icon getIcon() { return null; }

    /**
     * Called by the IDE to get this view's description.
     */
    public String getDescription() { return "desc for php view...."; }

    /**
     * Called by the IDE to create an instance of this view.
     */
}
```


navExtensionDevKit.html Sample

```
    */  
    public IDocumentView createView()  
    {  
        return EditorSvc.get().createSourceView(m_doc);  
    }  
}  
}
```

PhpProjectBuildDriver.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.core.ResourceSvc;
import com.bea.ide.util.IOUtil;
import com.bea.ide.workspace.IProject;
import com.bea.ide.workspace.project.DefaultBuildDriver;
import java.io.File;
import java.util.regex.Pattern;

/**
 * A simple build driver for PHP projects. The getAntScript method,
 * which is not implemented in class that this class extends,
 * assembles an Ant build.xml file using a template in this
 * projects and values provided primarily by WebLogic Workshop.
 */
public class PhpProjectBuildDriver extends DefaultBuildDriver
{
    /**
     * Constructs a new driver instance with an instance
     * of the PHP project that will use the driver. This
     * implementation merely calls the super's, which
     * uses the IProject to extract information need to
     * create the build.xml file.
     */
    public PhpProjectBuildDriver(IProject project)
    {
        super(project);
    }

    /**
     * Called by the IDE to retrieve default Ant build
     * file contents when the user clicks "Export to Ant file"
     * in the Build panel of the project properties dialog.
     *
     * Note that the build target in the file created merely
     * zips the projects contents and puts the generated ZIP
     * file at the project's root.
     *
     * @return XML that comprises an Ant build file's contents.
     */
    public String getAntScript()
    {
        /**
         * Create the ZIP file's name from the project name.
         * Note that the _project variable is inherited from

```

navExtensionDevKit.html Sample

```
    * DefaultBuildDriver.
    */
    String zipFileName = _project.getName() + ".zip";

    /**
     * Use a utility API to read the contents of the template
     * XML into a string.
     */
    String source = IOUtil.read(
        ResourceSvc.get().getReader("xml/defaultPhpBuild.xml"));

    if (source == null)
        return null;

    /**
     * Call an inherited method to translate the placeholders that
     * correspond to beahome.local.directory, platformhome.local.directory,
     * app.local.directory, and project.local.directory.
     */
    source = expandLocalDirs(source);
    /**
     * Translate the placeholder corresponding to output.filename,
     * replacing it with the name of the project + .zip.
     */
    source = OUTPUT_FILENAME.matcher(source).replaceFirst(zipFileName);

    // Return the template with useful values instead of placeholders.
    return source;
}
}
```

PhpProjectDriver.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.Application;
import com.bea.ide.core.MessageSvc;
import com.bea.ide.filesystem.IFile;
import com.bea.ide.workspace.IProject;
import com.bea.ide.workspace.IWorkspace;
import com.bea.ide.workspace.project.IProjectDriver;

/**
 * A project driver loaded for each project of the PHP project type.
 * A project driver implements IProjectDriver, whose two methods,
 * activate and deactivate, provide a way for you to associate code
 * with the start and end of a project's use in an application.
 *
 * This sample shows a typical use of these methods: adding and removing
 * listeners. By adding a property event listener in the driver's
 * activate method and removing it in the deactivate method, you
 * associate the listener with the project for the duration of
 * the project's use. In this way you can listen for IDE-,
 * application- and document-related events and changes to properties
 * defined by WebLogic Workshop.
 */
public class PhpProjectDriver implements IProjectDriver
{
    IProject m_project = null;
    PhpProjectListener m_listener = null;

    /**
     * Constructs a new instance of this driver using the
     * property with which this instance is associated. A
     * project driver must provide a constructor that takes
     * an IProject instance. If multiple projects of this type
     * are included in the application, then multiple instances
     * of this driver will be created, one for each project.
     *
     * @param project The project with which this driver instance
     * is associated.
     */
    public PhpProjectDriver(IProject project)
    {
        m_project = project;
        /**
         * Create a listener with the project associated with
         * this instance of the driver.
         */
    }
}
```

navExtensionDevKit.html Sample

```
        */
        m_listener = new PhpProjectListener(m_project);
    }

    /**
     * Called by the IDE when a project of this type is activated, such as
     * when a project of this type is discovered to be present in
     * the open application.
     */
    public void activate()
    {
        /**
         * Add a listener that will be used for changes to the PROP_InitLevel
         * property. This property's value changes to track progress of the IDE's
         * startup.
         */
        Application.get().addPropertyChangeListener(Application.PROP_InitLevel, m_listener);
    }

    /**
     * Called by the IDE when a project of this type is deactivated, such as
     * when a project of this type is removed from the open application.
     */
    public void deactivate()
    {
        /**
         * Remove the listener that was added when this project
         * type was activated.
         */
        Application.get().removePropertyChangeListener(m_listener);
    }
}
```

PhpProjectListener.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.Application;
import com.bea.ide.core.MessageSvc;
import com.bea.ide.filesystem.IFile;
import com.bea.ide.util.swing.DialogUtil;
import com.bea.ide.workspace.IProject;
import com.bea.ide.workspace.IWorkspace;
import com.bea.ide.workspace.project.IProjectInfo;
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextArea;

/**
 * A property change listener to listen for IDE startup
 * and display a message about PHP projects in the application.
 *
 * Note that this listener will only be used when the IDE has just
 * finished starting up, and its message will only be displayed if
 * there is at least one PHP project in the application. If you
 * have not yet done so, the easiest way to execute this code
 * is to begin debugging the CustomProject application. That will
 * build the extension and start a new instance of the IDE with
 * the extension added.
 */
public class PhpProjectListener implements PropertyChangeListener
{
    IProject m_phpProject = null;

    public PhpProjectListener (IProject phpProject)
    {
        m_phpProject = phpProject;
    }

    /**
     * Called by the IDE on changes to properties with which this
     * listener is associated. The PhpProjectDriver associates
     * this listener with changes to the Application.PROP_InitLevel
     * property. That property's value changes to track the
     * IDE's startup progress.
     *
     * If the property's new value is Application.INIT_StartupComplete,
     * indicating that the IDE has finished starting up and is ready

```

navExtensionDevKit.html Sample

```
* for use, this listener will display an informational dialog
* listing the contents of the PHP projects in this application.
* The dialog will be displayed once for each project of the PHP
* type in the application -- each, after all, has its own instance
* of the driver by which this listener was created.
*
* Needless to say, this would be annoying in a real-world
* scenario, but it suggests the potential of a listener.
*
* @param event The event with which this listener is
* associated.
*/
public void propertyChange(PropertyChangeEvent event)
{
    IWorkspace workspace = Application.getWorkspace();

    /**
     * If the received event is "startup complete", put together
     * a message and display it in a dialog.
     */
    if (event.getNewValue().equals(Application.INIT_StartupComplete))
    {
        /**
         * Get an array of projects whose types have an extension.xml
         * containing an <attribute> element whose name attribute value is
         * "supportsPhp". Note that the IProjectInfo interface provides
         * other methods for retrieving the contents of the extension.xml
         * file for the project type.
         */
        IProject[] phpProjects = workspace.getProjects("supportsPhp");

        // A string buffer to contain an assembled message.
        StringBuffer messageBuffer = new StringBuffer();

        // Add the message's introduction.
        messageBuffer.append("Information about a PHP project in this application: \n\n");

        /**
         * Get the IProjectInfo for this project. An instance of this
         * interface contains information stored in the extension.xml
         * for the project type.
         */
        IProjectInfo projectInfo = m_phpProject.getProjectInfo();

        // Get the type's URN.
        String typeId = projectInfo.getTypeId();

        /**
         * Confirm that the project really is a PHP project as
         * specified by the project type URN given in the project
         * type's extension.xml file. This is unnecessary because this
         * listener is only associated with PHP projects by the project
         * driver, but the code shows how you can get the ID.
         */
        if (typeId.equals("urn:com-bea-ide:project.type:php"))
        {
            // Collect the project's name, path, and contents.
            String projectName = m_phpProject.getName();
            String projectPath = m_phpProject.getPath();
            IFile projectIFile = m_phpProject.getIFile();
            IFile[] projectIFiles = projectIFile.listIFiles();
        }
    }
}
```

navExtensionDevKit.html Sample

```
// Put project info into the message.
messageBuffer.append("Name: " + projectName + "\n");
messageBuffer.append("Path: " + projectPath + "\n");
if (projectIFiles.length > 0)
{
    messageBuffer.append("Contains these directories and files:\n");

    // Add to the message the name of each file in the project.
    for (int j = 0; j < projectIFiles.length; j++)
    {
        messageBuffer.append("    " + projectIFiles[j].getName() + "\n");
    }
}
messageBuffer.append("\n");
}

// Show the collected information.
DialogUtil.showInfoDialog(Application.getRootFrame(), messageBuffer.toString());
}
}
```


PhpProjectRunDriver.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.actions.ActionSvc;
import com.bea.ide.debug.DebugSvc;
import com.bea.ide.filesystem.FileSvc;
import com.bea.ide.filesystem.IFile;
import com.bea.ide.ui.browser.BrowserSvc;
import com.bea.ide.util.URIUtil;
import com.bea.ide.util.swing.DialogUtil;
import com.bea.ide.workspace.IProject;
import com.bea.ide.workspace.IWorkspace;
import com.bea.ide.workspace.RunSvc;
import com.bea.ide.workspace.project.IRunDriver;
import com.bea.ide.workspace.project.IRunPreferences;
import java.io.File;
import java.net.MalformedURLException;
import java.net.URI;
import java.net.URISyntaxException;

/**
 * A run driver for a PHP project. PHP project's don't support
 * debugging, so this driver merely launches a browser to
 * display either a) the currently open PHP file, or b) an
 * index.php file if no PHP file is open. If there's no
 * index.php file, the driver prompts the user to create one.
 *
 * A run driver implements IRunDriver, which provides methods
 * that the IDE calls to find out if the project is runnable, to
 * tell the project extension when the user is navigating into
 * or out of the project in the Application window, and to
 * tell the extension when the user has requested to start or
 * stop the project. In other words, many of these methods are
 * designed to give you an opportunity to provide enable or
 * disable Start and Stop buttons, and to execute the project's
 * "running" behavior.
 *
 * To learn more about the "run" lifecycle, set breakpoints
 * in each of the methods and run this project. In the new
 * instance of the IDE that is started, open or create a new
 * PHP project, then open a PHP file in the project and click
 * the Start button.
 */
public class PhpProjectRunDriver implements IRunDriver
{
    // A browser process for displaying a "running" file.
```

navExtensionDevKit.html Sample

```
Process m_browserProcess = null;
// The project requested for running.
IProject m_currentProject = null;

/**
 * Constructs an instance of this driver with the project
 * that the user has asked to run.
 */
public PhpProjectRunDriver(IProject project)
{
    m_currentProject = project;
}

/**
 * Called by the IDE when a project of this type is activated, such as
 * when a project of this type is discovered to be present in
 * the open application.
 *
 * This method is inherited from IProjectDriver. See PhpProjectDriver, an
 * IProjectDriver implementation, for an example.
 */
public void activate()
{}

/**
 * Called by the IDE when a project of this type is deactivated, such as
 * when a project of this type is removed from the open application.
 *
 * This method is inherited from IProjectDriver. See PhpProjectDriver, an
 * IProjectDriver implementation, for an example.
 */
public void deactivate()
{}

/**
 * Called by the IDE when the user is opening a file in a
 * PHP project. This method provides an opportunity to
 * make sure the Start and Stop buttons are what they
 * should be when the user is browsing the project.
 */
public void doEnableChecks()
{
    boolean isRunning = (RunSvc.get().getRunningProject() == m_currentProject);
    if (!isRunning)
    {
        /**
         * This project type doesn't support debugging, so disable that
         * feature when the project has focus. Make sure the other
         * buttons are enabled or disabled as appropriate.
         */
        RunSvc.get().setRunDebugEnabled(Boolean.TRUE);
        RunSvc.get().setRunEnabled(Boolean.TRUE);
        RunSvc.get().setStopEnabled(Boolean.FALSE);
    }
}

/**
 * Called by the IDE to get the debugging preferences. A PHP
 * process doesn't really support debugging,
 */
public IRunPreferences[] getPreferences()
```

navExtensionDevKit.html Sample

```
{
    IRunPreferences[] preferences = new IRunPreferences[1];
    preferences[0] = new PhpProjectRunPreferences();
    return preferences;
}

/**
 * Called by the IDE to discover whether this project
 * type can be run. It can.
 */
public boolean isRunnable()
{
    return true;
}

/**
 * Called by the IDE when a PHP project has received control
 * of the Start, Start with debug, and Stop buttons.
 *
 * Nothing needed here.
 */
public void gainedCurrent()
{}

/**
 * Called by the IDE when a PHP project has lost control
 * of the Start, Start with debug, and Stop buttons.
 *
 * Nothing special needed here.
 */
public void lostCurrent()
{}

/**
 * Called by the IDE when the user saves properties
 * for a PHP project. No preferences to save, so no need to handle
 * this call.
 */
public void onPreferencesSaved()
{}

/**
 * Called by the IDE when the user puts the project
 * into "run" mode while a file is open in the
 * project (as opposed to the project merely having
 * focus). Only PHP files can be run, so if a PHP
 * file isn't currently open, then defer to a
 * project-level request.
 *
 * @param fileUri The file that the user wants to run.
 * @param debugRequested true if the user wants to run with
 * debugging; otherwise false.
 */
public boolean runFile(Uri fileUri, boolean debugRequested)
{
    boolean isRunnable = false;
    if (fileUri.getPath().endsWith(".php"))
    {
        isRunnable = true;
        enterRunState(fileUri);
    } else
    {

```

navExtensionDevKit.html Sample

```

    {
        runProject(false);
    }
    return isRunnable;
}

/**
 * Called by the IDE when the user puts the project
 * into "run" mode -- such as by clicking the Start button.
 * Note that this project type doesn't support debugging,
 * so the <em>debugRequested</em> parameter is ignored.
 *
 * @param debugRequested true if the user wants to run with
 * debugging; otherwise false.
 */
public boolean runProject(boolean debugRequested)
{
    IFile indexIFile = null;
    boolean isRunnable = false;

    String indexFilePath = m_currentProject.getPath() + File.separator + "index.php";
    File indexFile = new File(indexFilePath);
    if (indexFile.exists())
    {
        isRunnable = enterRunState(indexFile.toURI());
    } else
    {
        showNoIndexAlert();
        isRunnable = false;
    }
    return isRunnable;
}

/**
 * Takes the IDE into a "running" state. For PHP projects
 * this merely means that the Start button is disabled, the
 * Stop button is enabled, and a browser is launched to
 * display a PHP file.
 */
private boolean enterRunState(URI fileUri)
{
    boolean isRunning = false;
    try
    {
        m_browserProcess = BrowserSvc.get().invokeBrowser(fileUri.toURL(), true);
        RunSvc.get().setRunEnabled(Boolean.FALSE);
        RunSvc.get().setRunDebugEnabled(Boolean.FALSE);
        RunSvc.get().setStopEnabled(Boolean.TRUE);
        // DebugSvc.get().start(m_currentProject);
        RunSvc.get().setRunningProject(m_currentProject);
        isRunning = true;
        // m_currentProject.getWorkspace().setState(IWorkspace.STATE_IsRunning, _pkg.getStri
    } catch (MalformedURLException mue)
    {
        mue.printStackTrace();
    }
    return isRunning;
}

/**
 * Displays a dialog telling the user that they

```

navExtensionDevKit.html Sample

```
* can't run the project until they've opened
* a PHP file, or there is an index.php file at the
* project's root.
*/
private void showNoIndexAlert()
{
    DialogUtil.showErrorDialog("This file can not be run. Please open a PHP file or " +
        "create an index.php file at the root of the project.");
}

/**
 * Called by the IDE when an application that contains a PHP
 * project is being shut down, immediately before the
 * shutdown occurs.
 */
public void shutDown()
{
    if (m_browserProcess != null)
    {
        m_browserProcess.destroy();
    }
}

/**
 * Called by the IDE when the running project is stopped.
 * Running a PHP project just opens a browser, so this code
 * just closes it.
 */
public boolean stopProject()
{
    RunSvc.get().setStopEnabled(Boolean.FALSE);
    RunSvc.get().setRunEnabled(Boolean.TRUE);
    RunSvc.get().setRunDebugEnabled(Boolean.TRUE);
    if (m_browserProcess != null)
    {
        m_browserProcess.destroy();
    }
    return true;
}
}
```

PhpProjectRunPreferences.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/src/ideExtensions/customPr

Sample Source Code

```
package ideExtensions.customProject;

import com.bea.ide.core.LookAndFeelConstants;
import javax.swing.JPanel;
import com.bea.ide.workspace.project.IRunPreferences;
import java.util.Map;
import java.util.prefs.Preferences;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.UIManager;

/**
 * A preferences class for displaying project type-specific preferences
 * on the Debugging panel of the project preferences dialog. A PHP
 * project has no specific preferences, so this panel merely says so.
 */
public class PhpProjectRunPreferences extends JPanel implements IRunPreferences
{
    public PhpProjectRunPreferences()
    {
        initComponents();
    }

    private void initComponents()
    {
        JLabel lblMessage = new JLabel();
        lblMessage.setText("This is where project-specific debugging " +
            "preferences would go.");
        this.add(lblMessage);
    }

    public String getId()
    {
        return "PHP Run Preferences";
    }

    public JPanel getJPanel()
    {
        return this;
    }

    public String getTabName()
    {
        return "Run Preferences";
    }
}
```

navExtensionDevKit.html Sample

```
public void postLoad(Preferences arg0)
{
}

public void preStore(Preferences arg0)
{
}

public void setAddPropertiesToMap(Map arg0)
{
}

/**
 *
 */
public boolean validateEntries(JDialog dialog)
{
    return true;
}
}
```

xml Samples

This section contains source code for the following samples.

Samples Included in This Section

DefaultPhpBuild.xml Sample

DefaultPhpBuild.xml Sample

This topic includes the source code for the DefaultPhpBuild.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/CustomProject/xml/

Sample Source Code

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- This is a template for generating an Ant build file. The values
preceding with an @ sign are replaced with contextually-
appropriate values by the build driver. The PhpProjectBuildDriver
class extends DefaultBuildDriver, inherited from that class the
functions needed to convert these placeholders to useful values.
See PhpProjectBuildDriver for more information.

The build target of this file merely zip the projects contents
and puts the ZIP file in the project's root directory.
-->
<project name="PHP project" default="build">
  <property environment="env"/>
  <!-- These properties are auto-set to match local values when executed within the IDE.
  <!-- Note that most of the following properties aren't actually used in this file's
  targets. They're here to display values that WebLogic Workshop replaces. -->
  <property name="beahome.local.directory" value="@beahome.local.directory"/>
  <property name="platformhome.local.directory" value="@platformhome.local.directory"/>
  <property name="app.local.directory" value="@app.local.directory"/>
  <property name="project.local.directory" value="@project.local.directory"/>
  <!--end auto properties -->

  <property name="dest.path" value="${project.local.directory}"/>
  <property name="output.filename" value="@output.filename"/>

  <target name="build" description="Zip this project and copy the resulting file to the p
    <delete file="${dest.path}/${output.filename}" failonerror="false"/>
    <mkdir dir="${dest.path}"/>
    <zip destfile="${dest.path}/${output.filename}" basedir="${dest.path}" includes
      <zipfileset dir="${project.local.directory}" includes="*.*"/>
    </zip>
  </target>

  <target name="clean">
    <delete file="${dest.path}/${output.filename}" failonerror="false"/>
  </target>
</project>
```

DragDropSimple Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

dragDropSimple Samples

dragDropSimple Samples

This section contains source code for the following samples.

Samples Included in This Section

`SimpleTree.java` Sample

`SimpleTreeDragDropDriver.java` Sample

`TreePanelView.java` Sample

SimpleTree.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/DragDropSimple/src/ideExtensions/dragDropSimple

Sample Source Code

```
package ideExtensions.dragDropSimple;

import javax.swing.JTree;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeSelectionModel;

/**
 * A Java Swing tree that displays example data for illustrating
 * drag and drop in the WebLogic Workshop IDE. This class builds
 * the tree using instances of DefaultMutableTreeNode, starting
 * with an instance as the tree's root and adding children along the
 * way. One useful aspect of this class is that it supports specifying
 * instances of a "user object" that can live in each node to hold the
 * node's data.
 */
public class SimpleTree extends JTree
{
    // Create the tree's root node.

    static DefaultMutableTreeNode top =
        new DefaultMutableTreeNode("Item List");

    /**
     * Constructs an instance of the tree by starting with the root
     * node and adding child nodes.
     */
    public SimpleTree()
    {
        super(top);
        createNodes(top);

        // Only one node may be selected at a time.
        this.getSelectionModel().setSelectionMode
            (TreeSelectionModel.SINGLE_TREE_SELECTION);
    }

    /**
     * A "user object" class to contain the data for each node in the tree.
     * A new instance of this class is constructed for each data-carrying
     * node in the tree.
     */
    public class ItemInfo
    {
        public String m_itemName;
        public String m_itemDescription;
    }
}
```

navExtensionDevKit.html Sample

```
// Constructs the object with simple data.
public ItemInfo(String item, String description)
{
    m_itemName = item;
    m_itemDescription = description;
}

/**
 * Called to retrieve the data that should be displayed as label
 * for the tree node that this user object instance belongs to.
 *
 * @return The label string.
 */
public String toString()
{
    return m_itemName;
}

/**
 * Gets a string containing the node's label and its description.
 * This method is called to retrieve data for copying to a drop
 * target.
 *
 * @return The data for the node to which this user object
 * belongs.
 */
public String getItemData()
{
    return "Item name: " + m_itemName + "\n" +
        "Item description: " + m_itemDescription;
}
}

/**
 * Builds the portion of the tree beneath the root by adding
 * DefaultMutableTreeNode instances, each with separate
 * instances of the ItemInfo user object. Two subcategories are
 * also added. The nonsensical data represents item names and their
 * descriptions.
 *
 * @param top The tree's top node.
 */
private void createNodes(DefaultMutableTreeNode top)
{
    top.setUserObject(new ItemInfo
        ("All Items", "A list of everything.));

    DefaultMutableTreeNode category = null;
    DefaultMutableTreeNode item = null;

    category = new DefaultMutableTreeNode(new ItemInfo
        ("Category 1",
        "The first set of items.));
    top.add(category);

    item = new DefaultMutableTreeNode(new ItemInfo
        ("flangie",
        "Fully-functional complement to our zinc-full protaic.));
    category.add(item);

    item = new DefaultMutableTreeNode(new ItemInfo
```

navExtensionDevKit.html Sample

```
        ("callioscim",
            "A must for any truly robust melianic hisc, whether " +
            "in or out of frigid environs.");
category.add(item);

category = new DefaultMutableTreeNode(new ItemInfo
    ("Category 2",
        "The second set of items.));
top.add(category);

item = new DefaultMutableTreeNode(new ItemInfo
    ("protaic",
        "Affix this to a flangie to see a manifold increase " +
        "in performant particulates.));
category.add(item);

item = new DefaultMutableTreeNode(new ItemInfo
    ("melianic hisc",
        "Fully supports trending the callioscim to maximum " +
        "post-pattern joining.));
category.add(item);
    }
}
```


SimpleTreeDragDropDriver.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/DragDropSimple/src/ideExtensions/dragDropSimple

Sample Source Code

```
package ideExtensions.dragDropSimple;

import com.bea.ide.core.MessageSvc;
import com.bea.ide.core.ResourceSvc;
import com.bea.ide.core.datatransfer.DefaultDragDropDriver;
import ideExtensions.dragDropSimple.SimpleTree.ItemInfo;

import java.awt.Component;
import java.awt.Image;
import java.awt.Point;
import java.awt.datatransfer.StringSelection;
import java.awt.datatransfer.Transferable;
import javax.swing.tree.DefaultMutableTreeNode;
import javax.swing.tree.TreeSelectionModel;

/**
 * A drag/drop driver that supports copying data from the simple
 * tree to a document source. This drag/drop driver is registered by
 * the TreePanelView as the tree's driver for drag/drop operations.
 * This driver knows how to extract data from the tree that will be
 * copied to the drop source.
 */
public class SimpleTreeDragDropDriver extends DefaultDragDropDriver
{
    /**
     * Called by the IDE to get the IDragSourceInfo instance through which
     * the IDE can retrieve the data to copy from the drag source to
     * the drop target.
     *
     * @param component The drag source.
     * @param point The point representing the start of the drag.
     * @return The drag source info containing transfer data.
     */
    public IDragSourceInfo getDragInfo(Component component, Point point)
    {
        IDragSourceInfo dragInfo = null;
        try
        {
            /**
             * If there's anything selected in the tree, construct a draginfo
             * from the selected node. Note that this tree does not support
             * multiple selections; see SimpleTree for more information.
             */
            SimpleTree simpleTree = (SimpleTree) component;
            TreeSelectionModel selectionModel = simpleTree.getSelectionModel();
        }
    }
}
```

navExtensionDevKit.html Sample

```

        DefaultMutableTreeNode node =
            (DefaultMutableTreeNode) (selectionModel.getSelectionPath().getLastPathComponent());
        dragInfo = new TreeViewDragInfo(node);
        return dragInfo;
    } catch (Exception e)
    {
        MessageSvc.get().displayError("Error while dragging: " +
            e.getMessage(), 1);
        e.printStackTrace();
        return null;
    }
}

/**
 * A drag info class to support copying data from the simple tree
 * to a document.
 */
class TreeViewDragInfo implements IDragSourceInfo
{
    private String m_infoData = null;
    StringBuffer m_infoBuffer = new StringBuffer();

    /**
     * Constructs an instance of this class with a node from the simple
     * tree, extracting data from the node for copying to the drop
     * target.
     *
     * @param selection The selected tree node.
     */
    public TreeViewDragInfo(DefaultMutableTreeNode selection)
    {
        // Retrieve the object that contains node data.
        Object itemInfo = selection.getUserObject();
        /**
         * If the node is a leaf (that is, if it has no children) then simply
         * get its data out as a string for copying to the drop target.
         */
        if (selection.isLeaf())
        {
            m_infoData = ((ItemInfo) itemInfo).getItemData();
        }
        /**
         * If the node is not a leaf (it has children) then collect the
         * data from each of its children for copying to the drop target.
         */
        else
        {
            DefaultMutableTreeNode node = (DefaultMutableTreeNode) selection;
            ItemInfo itemUserObject = (ItemInfo) node.getUserObject();
            String infoData = itemUserObject.getItemData();
            m_infoBuffer.append(infoData + "\n\n");

            int count = selection.getChildCount();
            for (int i = 0; i < count; i++)
            {
                node = (DefaultMutableTreeNode) selection.getChildAt(i);
                extractInfoFromNode(node);
            }
            m_infoData = m_infoBuffer.toString();
        }
    }
}

```

navExtensionDevKit.html Sample

```
/**
 * Extracts info from a child node, calling itself for any children
 * the child has.
 */
private void extractInfoFromNode(DefaultMutableTreeNode node)
{
    ItemInfo itemUserObject = (ItemInfo) node.getUserObject();
    String infoData = itemUserObject.getItemData();
    m_infoBuffer.append(infoData + "\n\n");
    for (int i = 0; i < node.getChildCount(); i++)
    {
        extractInfoFromNode((DefaultMutableTreeNode)node.getChildAt(i));
    }
}

/**
 * Called by the IDE to retrieve the image that should be displayed
 * with the pointer during dragging.
 *
 * @return The image to display.
 */
public Image getDragImage()
{
    return ResourceSvc.get().getImage("images/txt.gif");
}

/**
 * Called by the IDE to retrieve the icon image's offset from the
 * mouse pointer.
 *
 * @return The offset point.
 */
public Point getIconOffset()
{
    return new Point(0, 0);
}

/**
 * Called by the IDE to retrieve the Transferable instance
 * that contains data to be copied (transferred) from the
 * drag source to the drop target. This method uses the string
 * extracted from the selected simple tree node to construct
 * an instance of StringSelection, which implements Transferable.
 *
 * @return The Transferable instance.
 */
public Transferable getTransferable()
{
    Transferable dropData = new StringSelection(m_infoData);
    return dropData;
}
}
```

TreePanelView.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/DragDropSimple/src/ideExtensions/dragDropSimple

Sample Source Code

```
package ideExtensions.dragDropSimple;

import com.bea.ide.core.datatransfer.DataTransferSvc;

import java.awt.Component;
import javax.swing.JPanel;
import javax.swing.JTree;
import javax.swing.JScrollPane;
import java.awt.GridLayout;

/**
 * A frame view that displays a tree with example data for dragging
 * and dropping. This view registers the tree as a participant in
 * drag/drop operations using the data transfer service.
 */
public class TreePanelView extends JPanel
    implements com.bea.ide.ui.frame.IFrameView
{
    /**
     * Called by the IDE to retrieve the user interface that should be
     * displayed in the frame.
     *
     * @param viewId The id, if any, for the frame specified in the frame's
     * extension.xml file.
     * @return The user interface to display within the frame's
     * boundaries.
     */
    public Component getView(String viewId)
    {
        return this;
    }

    /**
     * Called by the IDE to discover whether this frame is available
     * for display; only called if the frame's extension.xml includes
     * the askavailable attribute set to "true".
     *
     * @return <code>true</code> if the frame should be available;
     * <code>false</code> if it should not be.
     */
    public boolean isAvailable()
    {
        return true;
    }
}
```

navExtensionDevKit.html Sample

```
/**
 * Constructs this view panel by creating a new instance of the
 * SimpleTree and placing it within a scrolling pane. This code
 * also registers the tree contained by this panel as supporting
 * drag/drop actions, specifying the drag/drop driver to use.
 */
public TreePanelView()
{
    super(new GridLayout(1, 0));
    SimpleTree tree = new SimpleTree();
    JScrollPane treePane = new JScrollPane(tree);
    this.add(treePane);

    /**
     * Register a drag/drop driver for the tree, specifying that
     * this is an action to copy data from source to target (rather
     * than, say, a move action) and that a drag image should be
     * used.
     */
    DataTransferSvc.get().registerDnDSupport(tree,
        new SimpleTreeDragDropDriver(),
        javax.swing.TransferHandler.COPY,
        true);
}
}
```

FrameViewSimple Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

src Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

FrameViewSimple Samples

FrameViewSimple Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

frameViewSimple Samples

frameViewSimple Samples

This section contains source code for the following samples.

Samples Included in This Section

FrameView.java Sample

FrameView.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/FrameViewSimple/src/ideExtensions/frame

Sample Source Code

```
package ideExtensions.frameViewSimple;

import com.bea.ide.ui.frame.FrameSvc;
import com.bea.ide.ui.frame.IFrameView;
import java.awt.Component;
import javax.swing.JLabel;

/**
 * A frame to display a simple message. This class extends JLabel, a
 * Java Swing user interface component, returning an instance of itself
 * when the IDE requests its UI. It also implements IFrameView, an
 * interface to support this class' role as a frame in the WebLogic Workshop
 * IDE.
 */
public class FrameView extends JLabel implements com.bea.ide.ui.frame.IFrameView
{
    /**
     * Constructs a new instance of this class with a simple message to
     * display in the frame.
     */
    public FrameView()
    {
        // Call the JLabel class' constructor.
        super("hi mom. yo dad. peace sis.");
    }

    /**
     * Called by the IDE to retrieve the user interface component to
     * display in the frame.
     *
     * @param id The value of the id attribute specified for this <frame-view>
     * in the extension.xml; <code>null</code> if the attribute was not
     * specified.
     * @return The UI component to display in the frame.
     */
    public Component getView(String id)
    {
        // Return a pointer to an instance of this class, which extends JLabel.
        return this;
    }

    /**
     * Called by the IDE to determine whether this frame is "available" --
     * in other words, whether the frame may be displayed in the UI.
     */
}
```

navExtensionDevKit.html Sample

```
* This method will only be called if the extension.xml specifies the
* askavailable attribute as "true" in the <frame-view> element.
*
* @return <code>true</code> if the frame is available for display;
* <code>false</code> if it is not.
*/
    public boolean isAvailable()
    {
        return true;
    }
}
```

MenuItems Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

src Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

MenuItems Samples

MenuItems Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

menuItems Samples

menuItems Samples

This section contains source code for the following samples.

Samples Included in This Section

AddFavoriteAction.java Sample

DeleteFavoriteAction.java Sample

FavoritesGenerator.java Sample

LaunchBrowserAction.java Sample

LaunchBrowserAction1.java Sample

LaunchBrowserAction2.java Sample

LaunchBrowserAction3.java Sample

AddFavoriteAction.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

import com.bea.ide.Application;
import com.bea.ide.actions.ActionSvc;
import com.bea.ide.actions.DefaultAction;
import com.bea.ide.core.PreferencesSvc;

import java.awt.event.ActionEvent;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 * An action behind an "Add Favorite" menu command. The command
 * adds a "favorite" URL menu command to a "Favorites"
 * menu in the IDE. This class extends DefaultAction, which in turn
 * implements IAction. Action extensions such as menu commands and
 * toolbar buttons implement the IAction interface's actionPerformed
 * method to provide logic for the action. The IDE calls actionPerformed
 * when the command is selected by the IDE's user.
 *
 * The extension.xml file provided in the META-INF folder of the MenuItems
 * sample specifies this class as the handler for the "Add Favorite" action.
 */
public class AddFavoriteAction extends DefaultAction
{
    /**
     * Adds a "favorite" URL to the "Favorites" menu provided by the
     * MenuItems sample. This method is called by the IDE when the user
     * clicks the "Favorites -> Add Favorite" menu.
     *
     * @param e The menu click event that occurred in the IDE.
     */
    public void actionPerformed(ActionEvent e)
    {
        /**
         * Start creating the user interface for a dialog to prompt the
         * user for a URL to add. The panel contains the text box, then is
         * added to a dialog below.
         */
        JPanel panel = new JPanel();
        JTextField textField = new JTextField(25);
        textField.setText("http://");
        textField.setCaretPosition(7);
    }
}
```

navExtensionDevKit.html Sample

```
JLabel label = new JLabel("Yo. Enter a favorite web link: ");
panel.add(label);
panel.add(textField);

/**
 * Display a simple dialog that includes the panel created above. If
 * the user clicks OK (rather than Cancel), then proceed with the
 * following.
 */
if (JOptionPane.showConfirmDialog(Application.getRootFrame(),
    panel,
    "Add Favorite",
    JOptionPane.OK_CANCEL_OPTION,
    JOptionPane.PLAIN_MESSAGE,
    null) == JOptionPane.OK_OPTION)
{
    /**
     * Use the preferences service to get any favorites already saved
     * to $USER_HOME/.workshop.pref.
     */
    PreferencesSvc.IPreferencePkg prefs =
        PreferencesSvc.get().getUserPreferences(Application.get(),
            Application.class).getPackageForNode("favorites");

    /**
     * Favorites are stored in .workshop.pref as <option> elements
     * whose name attribute is a value such as "favorites1" and value
     * attribute is the URL. Find the end of the list and add a new entry
     * with the text in the dialog's text box.
     */
    int favNum = 0;
    while (prefs.get("favorites" + (++favNum)) != null) ;
    prefs.put("favorites" + favNum, textField.getText());

    /**
     * Use the action service to refresh the menu with the complete list
     * using the FavoritesGenerator class included in this extension.
     * See the code in that class for details.
     */
    ActionSvc.get().refreshGenerator(FavoritesGenerator.class.getName());
}
}
```

DeleteFavoriteAction.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

import com.bea.ide.Application;
import com.bea.ide.actions.ActionSvc;
import com.bea.ide.actions.DefaultAction;

import java.awt.event.ActionEvent;
import java.util.prefs.Preferences;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

/**
 * An action behind a "Delete Favorite" menu command. See the comments
 * in AddFavoriteAction for more information on the action API.
 */
public class DeleteFavoriteAction extends DefaultAction
{
    /**
     * Deletes an item from the "favorites" menu. This method is
     * called by the IDE when the user clicks the "Delete Favorite"
     * menu command.
     *
     * The extension.xml specifies this class as the logic for the "Delete
     * Favorites" menu command.
     *
     * @param e The menu click event that occurred in the IDE.
     */
    public void actionPerformed(ActionEvent e)
    {
        // Start creation of a panel for display in a JOptionPane.
        JPanel panel = new JPanel();
        JLabel label = new JLabel("Yo. Which link gets dissed?");
        JComboBox box = new JComboBox();
        box.setEditable(false);

        // Grab the node that contains all the user prefs for favorite links.
        Preferences prefs =
            Application.get().userNodeForPackage(Application.class).node("favorites");

        // Populate the JComboBox with the individual links
        int favnum = 0;
        while (prefs.get("favorites" + (++favnum), null) != null)
        {
```

navExtensionDevKit.html Sample

```
        box.addItem(prefs.get("favorites" + favnum, null));
    }

    // Finish creation of the panel for display in the following JOptionPane.
    panel.add(label);
    panel.add(box);

    if (JOptionPane.showConfirmDialog(Application.getRootFrame(),
        panel,
        "Remove Favorite",
        JOptionPane.OK_CANCEL_OPTION,
        JOptionPane.PLAIN_MESSAGE,
        null) == JOptionPane.OK_OPTION)
    {
        // Remove the selected link from the node.
        int index = box.getSelectedIndex() + 1;
        prefs.remove("favorites" + index);

        // Shift any remaining links up one in the indexing.
        while (prefs.get("favorites" + (++index), null) != null)
        {
            String value = prefs.get("favorites" + index, null);
            prefs.remove("favorites" + index);
            prefs.put("favorites" + (index - 1), value);
        }

        // Refresh the menu generator to remove the deleted entry.
        ActionSvc.get().refreshGenerator(FavoritesGenerator.class.getName());
    }
}
```

FavoritesGenerator.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

import com.bea.ide.Application;
import com.bea.ide.actions.ActionSvc;
import com.bea.ide.actions.IActionContainer;
import com.bea.ide.actions.IActionProxy;
import com.bea.ide.actions.IGenerator;
import com.bea.ide.core.PreferencesSvc;

/**
 * A generator class to generate "favorites" menu items. A generator
 * class implements IGenerator to create menu items dynamically.
 * This is in contrast to the menu commands created statically in
 * an extension.xml file.
 *
 * The extension.xml file specifies this class as the generator to use
 * for the "favorites" menu.
 */
public class FavoritesGenerator implements IGenerator
{
    /**
     * Called by the IDE to discover whether this generator's actions
     * should be visible. Some implementation may want to check
     * the IDE's context to determine the value to return.
     *
     * @return <code>true</code> to make the action visible; otherwise,
     * <code>false</code>.
     */
    public boolean isVisible()
    {
        return true;
    }

    /**
     * Populates the menu items from the entries stored in the "favorites"
     * node in the $USER_HOME/.workshop.pref file. The <em>menu</em>
     * received here is specified in this extension's extension.xml. There,
     * an <action-group> element's generator attribute assigns this class
     * as the generator to use for a submenu of the "favorites" menu.
     *
     * @param menu The menu that should be populated.
     * @return <code>true</code> if items were added; <code>false</code>
     * if none were added.
     */
    public boolean populate(IActionContainer menu)
```


navExtensionDevKit.html Sample

```
{
    /**
     * The preferences service provides easy access to the .workshop.prf
     * file so that extensions can use it to store settings.
     */
    PreferencesSvc.IPreferencePkg prefs =
        PreferencesSvc.get().getUserPreferences(Application.get(),
            Application.class).getPackageForNode("favorites");

    /**
     * Iterate through the stored "favorites" preferences, adding each
     * as a menu command.
     */
    int favNum = 0;
    String url;
    while ((url = prefs.get("favorites" + (++favNum))) != null)
    {
        // Create a new UI object to represent "favorites" menu command.
        IActionProxy ap = new LaunchBrowserAction().getProxy();
        // Assign a label and URL to the menu UI.
        ap.putValue(IActionProxy.PROP_Label, url);
        // Add the new menu to menu received from the IDE.
        menu.add(ap);
    }

    /**
     * Create a "Delete Favorite" menu command and enable it if
     * there are any favorites.
     */
    IActionProxy toggle =
        ActionSvc.get().getAction>DeleteFavoriteAction.class.getName());
    if (favNum == 1)
    {
        toggle.setEnabled(false);
    } else
    {
        toggle.setEnabled(true);
    }

    return true;
}
}
```

LaunchBrowserAction.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

import com.bea.ide.ui.*;
import com.bea.ide.actions.DefaultAction;
import com.bea.ide.actions.IActionProxy;
import com.bea.ide.ui.browser.BrowserSvc;
import com.bea.ide.core.MessageSvc;

import java.awt.event.ActionEvent;
import java.net.URL;

/**
 * An action to launch a browser with the URL for a "favorite"
 * in the "Favorites" menu. Along with LaunchBrowserAction1,
 * LaunchBrowserAction2, and LaunchBrowserAction3, this class
 * is specified by the extension.xml as the handler for one of the
 * predefined favorites. See the class FavoritesGenerator for
 * code that dynamically generates favorites based on user
 * preferences.
 */
public class LaunchBrowserAction extends DefaultAction
{
    /**
     * Launches a browser whose URL is the value specified in the
     * menu's label.
     *
     * @param e The menu click event that occurred in the IDE.
     */
    public void actionPerformed(ActionEvent e)
    {
        IActionProxy proxy = getProxy();
        try
        {
            // Get the value of the URL off the action's label attribute in the extension.xml
            URL url = new URL((String) proxy.getValue(IActionProxy.PROP_Label));
            // Use the browser service to display a browser with the URL.
            BrowserSvc.get().invokeBrowser(url, true);
        } catch (Exception ex)
        {
            // Display a debugging message in the IDE's Output window.
            MessageSvc.get().debugLog("Error while displaying favorite: "
                + ex.getLocalizedMessage());
        }
    }
}
```

LaunchBrowserAction1.java Sample

This topic includes the source code for the LaunchBrowserAction1.java Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

/**
 * An action to launch a browser with the URL for a "favorite"
 * in the "Favorites" menu. See LaunchBrowserAction for details.
 */
public class LaunchBrowserAction1 extends LaunchBrowserAction
{
}
```

LaunchBrowserAction2.java Sample

This topic includes the source code for the LaunchBrowserAction2.java Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

/**
 * An action to launch a browser with the URL for a "favorite"
 * in the "Favorites" menu. See LaunchBrowserAction for details.
 */
public class LaunchBrowserAction2 extends LaunchBrowserAction
{
}
```

LaunchBrowserAction3.java Sample

This topic includes the source code for the LaunchBrowserAction3.java Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/MenuItems/src/ideExtensions/menuItems/

Sample Source Code

```
package ideExtensions.menuItems;

/**
 * An action to launch a browser with the URL for a "favorite"
 * in the "Favorites" menu. See LaunchBrowserAction for details.
 */
public class LaunchBrowserAction3 extends LaunchBrowserAction
{
}
```

PopupAction Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

lib Samples

src Samples

strings Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

PopupAction Samples

PopupAction Samples

This section contains source code for the following samples.

Samples Included in This Section

lib Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

popupAction Samples

popupAction Samples

This section contains source code for the following samples.

Samples Included in This Section

FTPPrefsPanel.java Sample

FTPSettings.java Sample

UploadAction.java Sample

FTPPrefsPanel.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/PopupAction/src/ideExtensions/popupAction

Sample Source Code

```
package ideExtensions.popupAction;

import com.bea.ide.core.MessageSvc;
import com.bea.ide.core.ResourceSvc;
import com.bea.ide.swing.TitledTopBorder;
import com.bea.ide.workspace.IProject;
import com.bea.ide.workspace.IProjectPropertyPanel;
import com.bea.wlw.runtime.core.util.CryptUtil;

import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.net.UnknownHostException;
import java.util.prefs.Preferences;
import javax.swing.JDialog;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;

/**
 * A properties panel through which users of the PopupAction extension
 * can set properties for the extension's FTP behavior; these properties
 * include the host name, port, user name, and password that should be
 * used for FTP operations.
 *
 * This class extends JPanel, a Java Swing component that's useful for
 * containing other components for display in a dialog.
 */
public class FTPPrefsPanel extends JPanel implements IProjectPropertyPanel
{
    // Get the package of resource strings that will be used in the UI.
    static ResourceSvc.IResourcePkg s_pkg =
        ResourceSvc.get().getResourcePackage(FTPPrefsPanel.class, "ftp");

    // Variables for user interface components.
    private JPanel m_ftpSettingsPanel;
    private JLabel m_hostNameLabel;
    private JTextField m_hostName;
    private JLabel m_portLabel;
    private JTextField m_port;
    private JLabel m_userNameLabel;
    private JTextField m_userName;
    private JLabel m_passwordLabel;
    private JPasswordField m_password;
```

navExtensionDevKit.html Sample

```
private JLabel m_remoteDirectoryLabel;
private JTextField m_remoteDirectory;
private JPanel m_filler;
private JLabel m_remoteDirectoryTip;
private JLabel m_portDefaultTip;

/**
 * A variable to represent the project for which the user is currently
 * setting properties. This value will be set to the value received
 * from the IDE through the setProject method.
 */
private IProject m_project;

/**
 * Constructs a new instance of this class, calling a method
 * that builds the user interface from Java Swing components.
 */
public FTPPrefsPanel()
{
    super();
    initComponents();
}

/**
 * Retrieves properties from stored preferences and loads them into
 * the FTP properties panel for display.
 */
public void loadProperties()
{
    Preferences prefs = m_project.systemNodeForPackage(FTPPrefsPanel.class);
    m_hostName.setText(prefs.get(FTPSettings.HOSTNAME, ""));
    m_port.setText(prefs.get(FTPSettings.PORT, ""));
    m_remoteDirectory.setText(prefs.get(FTPSettings.REMOTE_DIRECTORY, ""));
    m_userName.setText(prefs.get(FTPSettings.USERNAME, ""));
    String password = prefs.get(FTPSettings.PASSWORD, null);
    if (password != null)
    {
        try
        {
            password = CryptUtil.get().deobfuscate(password);
            m_password.setText(password);
        } catch (Exception e)
        {
            MessageSvc.get().debugLog("Exception while attempting to decrypt proxy password");
        }
    }
}

/**
 * Stores property values entered in the FTP properties panel.
 */
public void storeProperties()
{
    // Retrieve the FTP preferences.
    Preferences prefs = m_project.systemNodeForPackage(FTPPrefsPanel.class);

    /**
     * Get the hostname property value. If the value is not equal to the
     * value already set, then set the new value.
     */
    String hostName = m_hostName.getText();
```

navExtensionDevKit.html Sample

```
if (!hostName.equals(prefs.get(FTPSettings.HOSTNAME, null)))
{
    prefs.put(FTPSettings.HOSTNAME, hostName);
}

/**
 * Get the port property value. If the value is not equal to the
 * value already set, then set the new value.
 */
String port = m_port.getText();
if (!port.equals(prefs.get(FTPSettings.PORT, null)))
{
    prefs.put(FTPSettings.PORT, port);
}

/**
 * Get the remote directory property value. If the value is not equal to the
 * value already set, then set the new value.
 */
String remoteDirectory = m_remoteDirectory.getText();
if (!remoteDirectory.equals(prefs.get(FTPSettings.REMOTE_DIRECTORY, null)))
{
    prefs.put(FTPSettings.REMOTE_DIRECTORY, remoteDirectory);
}

/**
 * Get the username property value. If the value is not equal to the
 * value already set, then set the new value.
 */
String userName = m_userName.getText();
if (!userName.equals(prefs.get(FTPSettings.USERNAME, null)))
{
    prefs.put(FTPSettings.USERNAME, userName);
}

/**
 * Retrieve the new password set in the properties panel;
 * retrieve the old password from preferences.
 */
String password = m_password.getText();
String oldPassword = prefs.get(FTPSettings.PASSWORD, null);
/**
 * If the old password isn't null, then attempt to decrypt it for
 * comparison with the new password.
 */
if (oldPassword != null)
{
    try
    {
        oldPassword = CryptUtil.get().deobfuscate(oldPassword);
    } catch (Exception e)
    {
        MessageSvc.get().debugLog("Exception while attempting to decrypt proxy password");
        oldPassword = null;
    }
}
/**
 * If the new password is not the same as the old password, then
 * set the new password the preferences.
 */
if (!password.equals(oldPassword))
```

navExtensionDevKit.html Sample

```

    {
        try
        {
            prefs.put(FTPSettings.PASSWORD, CryptUtil.get().obfuscate(password));
        } catch (Exception e)
        {
            MessageSvc.get().debugLog("Exception while attempting to encrypt proxy password");
        }
    }
}

/**
 * Called by the IDE to discover whether property values entered by
 * the user are valid before the entries are saved and the dialog is
 * closed. This method will be called after the user clicks OK in the
 * dialog, and before the IDE calls your storeProperties implementation..
 * This method provides an opportunity for you to check the new
 * values and return <code>true</code> if they're valid for use as
 * property values.
 *
 * The IDE will call your implementation of the storeProperties method
 * if you return <code>true</code> from validateEntries.
 *
 * @param dialog The dialog that contains this panel.
 * @return <code>true</code> if the entries are valid; <code>false</code>
 * if they are not.
 */
public boolean validateEntries(JDialog dialog)
{
    // Validate the host name by ensuring it exists.
    String hostName = m_hostName.getText();
    try
    {
        if (hostName == null || hostName.equals(""))
        {
            throw new UnknownHostException();
        }
    } catch (UnknownHostException uhe)
    {
        MessageSvc.get().displayError(s_pkg.getString("hostnameError"), MessageSvc.LEVEL_ERROR);
        return false;
    }

    // Validate the port by ensuring that it exists as an int.
    String port = m_port.getText();
    if (!port.equals(""))
    {
        try
        {
            int value = Integer.parseInt(port);
        } catch (NumberFormatException nfe)
        {
            MessageSvc.get().displayError(s_pkg.getString("portError"), MessageSvc.LEVEL_ERROR);
            return false;
        }
    }
    return true;
}

/**
 * Called by the IDE after the user clicks the Cancel button. If this

```


navExtensionDevKit.html Sample

```
* panel had acquired resources, this would be a good place to
* release them.
*/
public void cancel()
{
    // Not implemented
}

/**
 * Called by the IDE to inform this panel as to which project the
 * user is currently setting properties for. It is necessary to know
 * this information in order to retrieve and display properties for
 * the correct project.
 *
 * @param project The project for which the user is setting
 * properties.
 */
public void setProject(IProject project)
{
    m_project = project;
}

/**
 * Builds the user interface for the properties panel by arranging
 * Java Swing components into a "gridbag" layout.
 */
public void initComponents()
{
    setLayout(new GridBagLayout());

    m_ftpSettingsPanel = new JPanel();

    // The label and text box for the hostname property value.
    m_hostNameLabel = new JLabel(s_pkg.getString("host"));
    m_hostName = new JTextField();

    // The label and text box for the port property value.
    m_portLabel = new JLabel(s_pkg.getString("port"));
    m_port = new JTextField();

    // The label and text box for the username property value.
    m_userNameLabel = new JLabel(s_pkg.getString("username"));
    m_userName = new JTextField();

    // The label and text box for the password property value.
    m_passwordLabel = new JLabel(s_pkg.getString("password"));
    m_password = new JPasswordField();

    // The label and text box for the remote directory property value.
    m_remoteDirectoryLabel = new JLabel(s_pkg.getString("remoteDirectory"));
    m_remoteDirectory = new JTextField();

    /**
     * The panel that holds the input components. This is nested inside
     * the panel that the IDE puts into the Properties dialog.
     */
    m_ftpSettingsPanel.setBorder(new TitledTopBorder(s_pkg.getString("ftpSettings")));
    m_ftpSettingsPanel.setLayout(new GridBagLayout());

    // The descriptive text for the remote directory and port boxes.
    m_remoteDirectoryTip = new JLabel(s_pkg.getString("remoteDirectoryTip"));
```

navExtensionDevKit.html Sample

```
m_portDefaultTip = new JLabel(s_pkg.getString("portDefaultTip"));
m_filler = new JPanel();

/**
 * Use GridBagConstraints instances to hold the components
 * that make up the user interface. Each of the following sections
 * creates an instance, sets layout characteristics for the
 * instance, then adds a component to the panel that contains it
 * using the instance as it has been set up. The added component
 * is positioned in the panel according to the characteristics
 * set for the constraints instance.
 */
GridBagConstraints gbc = new GridBagConstraints();
gbc.gridwidth = 1;
gbc.weightx = 0.0;
gbc.insets = new Insets(2, 2, 2, 6);
gbc.anchor = GridBagConstraints.WEST;
gbc.fill = GridBagConstraints.HORIZONTAL;
m_ftpSettingsPanel.add(m_hostNameLabel, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
m_ftpSettingsPanel.add(m_hostName, gbc);

gbc.gridwidth = 1;
gbc.weightx = 0.0;
m_ftpSettingsPanel.add(m_portLabel, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
m_ftpSettingsPanel.add(m_port, gbc);

GridBagConstraints tipGBC = new GridBagConstraints();
tipGBC.gridwidth = GridBagConstraints.REMAINDER;
tipGBC.weightx = 1.0;
tipGBC.insets = new Insets(2, 10, 2, 6);
tipGBC.anchor = GridBagConstraints.WEST;
tipGBC.fill = GridBagConstraints.HORIZONTAL;
m_ftpSettingsPanel.add(m_portDefaultTip, tipGBC);

gbc.gridwidth = 1;
gbc.weightx = 0.0;
m_ftpSettingsPanel.add(m_remoteDirectoryLabel, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
m_ftpSettingsPanel.add(m_remoteDirectory, gbc);

tipGBC = new GridBagConstraints();
tipGBC.gridwidth = GridBagConstraints.REMAINDER;
tipGBC.weightx = 1.0;
tipGBC.insets = new Insets(2, 10, 2, 6);
tipGBC.anchor = GridBagConstraints.WEST;
tipGBC.fill = GridBagConstraints.HORIZONTAL;
m_ftpSettingsPanel.add(m_remoteDirectoryTip, tipGBC);

GridBagConstraints fillerGBC = new GridBagConstraints();
fillerGBC.gridwidth = GridBagConstraints.REMAINDER;
fillerGBC.anchor = GridBagConstraints.WEST;
fillerGBC.fill = GridBagConstraints.HORIZONTAL;
fillerGBC.insets = new Insets(8, 0, 8, 0);
fillerGBC.weightx = 1.0;
fillerGBC.weighty = 1.0;
m_ftpSettingsPanel.add(m_filler, fillerGBC);
```

```

gbc.gridwidth = 1;
gbc.weightx = 0.0;
m_ftpSettingsPanel.add(m_userNameLabel, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
m_ftpSettingsPanel.add(m_userName, gbc);

gbc.gridwidth = 1;
gbc.weightx = 0.0;
m_ftpSettingsPanel.add(m_passwordLabel, gbc);
gbc.gridwidth = GridBagConstraints.REMAINDER;
gbc.weightx = 1.0;
m_ftpSettingsPanel.add(m_password, gbc);

GridBagConstraints topGbc = new GridBagConstraints();
topGbc.gridwidth = GridBagConstraints.REMAINDER;
topGbc.anchor = GridBagConstraints.WEST;
topGbc.fill = GridBagConstraints.HORIZONTAL;
topGbc.insets = new Insets(4, 1, 8, 1);
topGbc.weightx = 1.0;
topGbc.weighty = 1.0;
add(m_ftpSettingsPanel, topGbc);
add(new JPanel(), topGbc);
    }
}

```

FTPSettings.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/PopupAction/src/ideExtensions/popupAction

Sample Source Code

```
package ideExtensions.popupAction;

/**
 * A class of constants to specify the names used for properties in the
 * preferences file.
 */
public class FTPSettings
{
    public static final String HOSTNAME = "hostname";
    public static final String PORT = "port";
    public static final String REMOTE_DIRECTORY = "remote.directory";
    public static final String USERNAME = "username";
    public static final String PASSWORD = "password";

    public static final int DEFAULT_PORT = 22;
}
```

UploadAction.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/PopupAction/src/ideExtensions/popupAction

Sample Source Code

```
package ideExtensions.popupAction;

import com.bea.ide.Application;
import com.bea.ide.actions.DefaultAction;
import com.bea.ide.actions.IActionProxy;
import com.bea.ide.actions.IPopupAction;
import com.bea.ide.actions.IPopupContext;
import com.bea.ide.core.MessageSvc;
import com.bea.ide.core.ResourceSvc;
import com.bea.ide.core.asyncTask.AsyncTaskSvc;
import com.bea.ide.core.asyncTask.IAsyncTask;
import com.bea.ide.filesystem.FileSvc;
import com.bea.ide.filesystem.IFile;
import com.bea.ide.ui.output.OutputMessage;
import com.bea.ide.ui.output.OutputSvc;
import com.bea.ide.util.URIUtil;
import com.bea.ide.workspace.IProject;
import com.bea.ide.workspace.IWorkspace;
import com.bea.ide.workspace.IWorkspaceEventContext;
import com.bea.wlw.runtime.core.util.CryptUtil;

import java.awt.Color;
import java.awt.event.ActionEvent;
import java.io.File;
import java.io.FileInputStream;
import java.net.URI;
import java.net.UnknownHostException;
import java.util.Enumeration;
import java.util.Hashtable;
import java.util.Stack;
import java.util.prefs.Preferences;

import org.apache.commons.net.ftp.FTPClient;
import org.apache.commons.net.ftp.FTPReply;

/**
 * An action behind a popup menu that uploads the currently selected
 * file or folder via FTP.
 *
 * From DefaultAction, this class gets
 */
public class UploadAction extends DefaultAction implements IPopupAction
{
    static ResourceSvc.IResourcePkg s_pkg =
        ResourceSvc.get().getResourcePackage(FTPPrefsPanel.class, "ftp");
```

```

// A flag to indicate whether it's valid to go ahead and upload.
private boolean m_isValidUploadContext;
// An IFile representing the project from which files will be uploaded.
private IFile m_projectDirectory;
/**
 * A project within which there are selected files; used to test for
 * selections across projects.
 */
private IProject m_project;
/**
 * A flag to indicate whether one of the items selected to upload
 * is a project directory. A project will need special handling.
 */
private boolean m_projectSelected;
/**
 * A flag to indicate, whether the list items selected for upload
 * spans multiple projects.
 */
private boolean m_activeProjectChanged;

// A flag to indicate whether this action is currently trying to upload.
private boolean m_uploadInProgress;
/**
 * A flag to indicate that the popup is figuring out whether (and
 * what) to display.
 */
private boolean m_prepareInProgress;

// An FTPClient instance to give this action FTP abilities.
private FTPClient m_ftp;

// The root directory as specified in project FTP properties.
private String m_rootDir;

private Hashtable m_LCAncestors;

// A window to display FTP status.
private OutputSvc.IOutputWindow m_win;
// The window's description.
private OutputSvc.OutputWindowDescription m_desc;

private static final Color BLACK = new Color(55, 47, 47);
private static final Color RED = new Color(223, 45, 45);

public UploadAction()
{
    // initialize the output window.
    m_desc = new OutputSvc.OutputWindowDescription();
    m_desc.title = "FTP Output";
    m_desc.destination = "FTP";
    m_win = OutputSvc.get().getWindow(m_desc, false, false);
    m_activeProjectChanged = false;
    m_uploadInProgress = false;
    m_prepareInProgress = false;
}

/**
 * Called by the IDE to inform the popup that it is about to be requested
 * for display. The <em>pc</em> parameter includes information
 * about the popup's potential position in the IDE.

```

navExtensionDevKit.html Sample

```
*
* This method retrieves information
* needs to handle the case where multiple items in the application
* tree are selected
*
* given the set of nodes that are selected in the application tree,
* determines whether the "upload" menu item should be visible.
*/
public void prepare(IPopupContext pc)
{
    // If there's already some FTPing going on, don't show this command.
    if (m_uploadInProgress || m_prepareInProgress)
    {
        getProxy().setEnabled(false);
        getProxy().putValue(IActionProxy.PROP_Visible, Boolean.valueOf(true));
        getProxy().putValue(IActionProxy.PROP_Label, "Upload");
        return;
    }
    // Set flags to initial values.
    m_prepareInProgress = true;
    m_isValidUploadContext = true;
    m_project = null;
    m_projectSelected = false;

    /**
     * Get a list of all the projects in this application, along with the
     * files in each project. Add each file to an IFile array. An IFile
     * provides many methods useful to handling files through an
     * extension.
     */
    IWorkspace workspace =
        (IWorkspace) Application.get().getProperty(Application.PROP_ActiveWorkspace);
    IProject[] projects = workspace.getProjects();
    IFile[] projectIFiles = new IFile[projects.length];
    for (int i = 0; i < projectIFiles.length; i++)
    {
        projectIFiles[i] = projects[i].getIFile();
    }

    /**
     * This section verifies that the files selected in the Application
     * window are part of the same project; FTP actions are not
     * permitted for multiple files crossing project boundaries.
     */
    IWorkspaceEventContext ctx = (IWorkspaceEventContext) pc;
    // Get the URIs of the selected nodes in the Application window.
    URI[] uris = ctx.getURIs();

    /**
     *
     */
    for (int i = 0; i < uris.length; i++)
    {
        URI uri = uris[i];
        /**
         * Find out if an entire project is selected. If it is, this action
         * will later need to add all of the project's file to the list of those
         * to upload.
         */
        if (uri.getScheme().equals("project"))
        {

```

navExtensionDevKit.html Sample

```

/**
 * FileSvc.I.getIFile() expects URIs whose scheme component is "file",
 * so adjust "project" URIs accordingly.
 */
uri = URIUtil.changeScheme(uri, "file");
if (uri == null)
{
    MessageSvc.get().displayError(s_pkg.getString("uriChangeSchemeError"), Mess
    m_prepareInProgress = false;
    return;
}
m_projectSelected = true;
}

/**
 * The item selected in the window must be a regular file, folder, or a
 * project (as noted above). In other words, the scheme must be
 * "file," rather than "library", "workspace", or "resource."
 */
if (!uri.getScheme().equals("file"))
{
    m_isValidUploadContext = false;
    break;
}

// Get an IFile for the item at the URI.
IFile file = FileSvc.get().getIFile(uri);
/**
 * A flag to indicate whether the current file's parent project
 * has been found.
 */
boolean foundProject = false;

while (file != null)
{
    /**
     * Loop through the list of projects in this application, checking
     * (essentially) to see whether there are multiple projects
     * represented in the list of files selected in the Application
     * window.
     */
    for (int pIndex = 0; pIndex < projectIFiles.length; pIndex++)
    {
        IFile projectIFile = projectIFiles[pIndex];
        /**
         * If the current file in the list of selected files is the same
         * as the current IFile representing a project in the application,
         * then find out if the current project IFile is also the same as the
         * the project IFile established for testing. If a different project
         * IFile, then the selection list spans multiple projects, and
         * that's bad because this action doesn't support that.
         */
        if (file.compareTo(projectIFile) == 0)
        {
            /**
             * If, among the files selected for upload, a test project hasn't
             * been determined yet, then set the test project to the
             * project represented by the current projectIFile.
             */
            if (m_project == null)
            {

```


navExtensionDevKit.html Sample

```
/**
 * If the current projectIFile is not the same as the IFile
 * established for testing, then a change of projects
 * has occurred.
 */
if (projectIFile != m_projectDirectory)
{
    m_activeProjectChanged = true;
} else
{
    m_activeProjectChanged = false;
}
/**
 * Set the test project directory and test project to the
 * current directory and project. If these don't change
 * during the course of this loop, then they'll be used
 * for the FTP upload. If they do change, then this action
 * is unsupported because it means the selection list
 * spans projects.
 */
m_projectDirectory = projectIFile;
m_project = projects[pIndex];
/**
 * If the test project directory is not the same as the current
 * project IFile, then this action is unsupported because
 * multiple projects are represented in the list of files to
 * upload.
 */
} else if (m_projectDirectory.compareTo(projectIFile) != 0)
{
    m_isValidUploadContext = false;
}
foundProject = true;
break;
    }
}

/**
 * If a project has been found, exit this loop. Otherwise,
 * set the file to check to the current file's parent.
 */
if (foundProject)
{
    break;
}
file = file.getParentIFile();
}

/**
 * For some reason, the upload action can't be done. Stop
 * looping through the list of selected files.
 */
if (!m_isValidUploadContext)
{
    break;
}
}
/**
 * If no project is represented by the list of selected files, then
 * set the "valid" flag to false. There's nothing to upload.
 */
```

navExtensionDevKit.html Sample

```

if (m_project == null)
{
    m_isValidUploadContext = false;
}

/**
 * If the items selected are valid for uploading, then enable display
 * of this popup and set display characteristics. Otherwise,
 * don't allow display.
 */
if (m_isValidUploadContext)
{
    getProxy().setEnabled(true);
    getProxy().putValue(IActionProxy.PROP_Visible, Boolean.valueOf(true));
    getProxy().putValue(IActionProxy.PROP_Label, "Upload");
} else
{
    getProxy().setEnabled(false);
    getProxy().putValue(IActionProxy.PROP_Visible, Boolean.valueOf(false));
}
// All done now, so indicate that preparation is finished.
m_prepareInProgress = false;
}

/**
 * A background task for performing the FTP operation.
 */
private class FTPTask implements IAsyncTask
{
    // Constructs an instance of this class.
    FTPTask()
    {}

    // No task to run.
    public void cleanup()
    {}

    // No task to run.
    public void interrupt()
    {}

    // No task to run.
    public void runForeground()
    {}

    /**
     * Called by the IDE for code to be run in the background.
     */
    public void runBackground()
    {
        // If uploading is already happening, then don't do it now.
        if (m_uploadInProgress || m_prepareInProgress)
        {
            return;
        }
        // If the upload action wasn't in progress, it is now.
        m_uploadInProgress = true;

        // Initialize the status window.
        m_win = OutputSvc.get().getWindow(m_desc, true, true);
        m_win.addMessage(new OutputMessage("", BLACK, null));
    }
}

```

navExtensionDevKit.html Sample

```

m_win.addMessage(new OutputMessage("~~~~~"));
m_win.addMessage(new OutputMessage("Current project: " +
    m_projectDirectory.getName(), BLACK, null));

// Store all the nodes in a hashtable as IFiles.
m_LCAncestors = new Hashtable();
/**
 * If the project itself is selected, put all its direct descendants
 * on the list.
 */
if (m_projectSelected)
{
    IFile[] childIFiles = m_projectDirectory.listFiles();
    for (int i = 0; i < childIFiles.length; i++)
    {
        m_LCAncestors.put(childIFiles[i], Boolean.TRUE);
    }
} else
{
    /**
     * Use an object representing the current application tree's
     * selections to get URIs for the selected items.
     */
    IWorkspaceEventContext ctx =
        (IWorkspaceEventContext) Application.get().getCookie(IWorkspaceEventCon
    URI[] uris = ctx.getURIs();
    /**
     * Get an IFile for each of the selected items, then put each into
     * a HashTable to
     */
    for (int i = 0; i < uris.length; i++)
    {
        URI uri = uris[i];
        IFile file = FileSvc.get().getIFile(uri);
        m_LCAncestors.put(file, Boolean.TRUE);
    }

    /**
     * For each child node, if one of the other nodes is an ancestor,
     * remove the child node from the list of least common ancestors.
     */
    for (int i = 0; i < uris.length; i++)
    {
        URI childURI = uris[i];
        IFile childIFile = FileSvc.get().getIFile(childURI);
        boolean ancestorFound = false;

        for (int j = 0; j < uris.length; j++)
        {
            URI ancestorURI = uris[j];
            IFile ancestorIFile = FileSvc.get().getIFile(ancestorURI);

            if (childIFile.compareTo(ancestorIFile) == 0)
            {
                continue;
            }

            IFile parentIFile = childIFile.getParentIFile();
            while (parentIFile != null && parentIFile.compareTo(m_projectDirectory)
            {
                if (parentIFile.compareTo(ancestorIFile) == 0)

```

navExtensionDevKit.html Sample

```

        {
            if (m_LCAncestors.containsKey(childIFile))
            {
                m_LCAncestors.remove(childIFile);
            }
            ancestorFound = true;
            break;
        }

        parentIFile = parentIFile.getParentIFile();
    }

    if (ancestorFound)
    {
        break;
    }
}

}

// Get the FTP preferences set by the user.
Preferences prefs = m_project.systemNodeForPackage(FTPPrefsPanel.class);

/**
 * If there's no FTP connection yet, attempt to create one using
 * FTP preferences.
 */
if (m_ftp == null || !m_ftp.isConnected() || m_activeProjectChanged)
{
    String hostname = "";
    try
    {
        m_ftp = new FTPClient();
        hostname = prefs.get(FTPSettings.HOSTNAME, "");
        hostname = hostname.trim();
        if (hostname.equals(""))
        {
            m_win.addMessage(new OutputMessage("You must specify a hostname for FTP",
            m_uploadInProgress = false;
            return;
        }
        m_ftp.connect(hostname);
        int reply = m_ftp.getReplyCode();
        if (!(FTPReply.isPositiveCompletion(reply)))
        {
            m_win.addMessage(new OutputMessage(hostname + ": refused connection.",
            m_ftp.disconnect();
            m_uploadInProgress = false;
            return;
        }
        m_win.addMessage(new OutputMessage("Connected to " + hostname, BLACK, null));

        // Get the FTP password and decrypt it.
        String password = prefs.get(FTPSettings.PASSWORD, null);
        if (password != null)
        {
            try
            {
                password = CryptUtil.get().deobfuscate(password);
            } catch (Exception ex)
            {

```

navExtensionDevKit.html Sample

```

        MessageSvc.get().debugLog("Exception while attempting to decrypt pr
        password = "";
    }
}

// Get the preferred FTP port number and set it for file transfers.
String port = prefs.get(FTPSettings.PORT, null);
int portNumber;
try
{
    portNumber = Integer.parseInt(port);
} catch (Exception ex)
{
    portNumber = FTPSettings.DEFAULT_PORT;
}
if (m_ftp.getDefaultPort() != portNumber)
{
    m_ftp.setDefaultPort(portNumber);
}

/**
 * Get the preferred username and attempt to log in using
 * it and the preferred password. Publish a message to
 * the status window.
 */
String username = prefs.get(FTPSettings.USERNAME, "");
username = username.trim();
if (!m_ftp.login(username, password))
{
    m_win.addMessage(new OutputMessage("Login failed for user: \"" + username + "\""));
    m_ftp.disconnect();
    m_uploadInProgress = false;
    return;
}
m_win.addMessage(new OutputMessage("Login successful for user: \"" + username + "\""));

/**
 * Get the preferred root directory and change to that directory
 * on the server.
 */
m_rootDir = prefs.get(FTPSettings.REMOTE_DIRECTORY, "");
m_rootDir = m_rootDir.trim();
if (m_rootDir.equals(""))
{
    m_win.addMessage(new OutputMessage("You must specify a directory where to upload"));
    m_uploadInProgress = false;
    return;
}
if (!m_ftp.changeWorkingDirectory(m_rootDir))
{
    if (!m_ftp.makeDirectory(m_rootDir))
    {
        m_win.addMessage(new OutputMessage("Failure changing to directory: " + m_rootDir));
        m_uploadInProgress = false;
        return;
    }
}
m_win.addMessage(new OutputMessage("Remote directory: " + m_rootDir, BLACK, null));
} catch (UnknownHostException ex)
{
    m_win.addMessage(new OutputMessage("\"" + hostname + "\" is an unknown host"));
}

```

navExtensionDevKit.html Sample

```

        m_uploadInProgress = false;
        return;
    } catch (Exception ex)
    {
        ex.printStackTrace();
        try
        {
            m_ftp.disconnect();
        } catch (Exception exc)
        {
            exc.printStackTrace();
        }
        m_uploadInProgress = false;
        return;
    }
}

// Ensure that the root directory exists.
try
{
    if (!m_ftp.changeWorkingDirectory(m_rootDir))
    {
        if (!m_ftp.makeDirectory(m_rootDir))
        {
            m_win.addMessage(new OutputMessage("Failure changing to directory: " +
            m_uploadInProgress = false;
            return;
        }
    }
} catch (Exception ex)
{
    ex.printStackTrace();
    try
    {
        m_ftp.disconnect();
    } catch (Exception exc)
    {
        exc.printStackTrace();
    }
    m_uploadInProgress = false;
    return;
}

/**
 * For each node to transfer, create its parents on the remote server
 * if necessary, then upload the node and its children.
 */
Enumeration enum = m_LCAncestors.keys();
while (enum.hasMoreElements())
{
    IFile file = (IFile) (enum.nextElement());
    String cwd = m_createParents(file, m_rootDir);
    m_uploadFile(cwd, file);
}
// Publish a completion message to the FTP status window.
m_win.addMessage(new OutputMessage("DONE with FTP task.", BLACK, null));
// Signal FTPing is done now.
m_uploadInProgress = false;
}
}

```

navExtensionDevKit.html Sample

```
/**
 * Called by the IDE to execute the action associated with this popup.
 * This method adds to the IDE a background task to perform the
 * file upload.
 *
 * @param e The IDE event that occurred.
 */
public void actionPerformed(ActionEvent e)
{
    AsyncTaskSvc.get().addTask(new FTPTask());
}

/**
 * Creates the subdirectory path from a file to the parent project on
 * the remote server. Returns the string that maps to the file's parent
 * directory. This method is called by the background task in FTPTask
 * in preparation for uploading.
 *
 * @param file The file to upload.
 * @param cwd The remote root directory specified in project properties.
 */
private String m_createParents(IFile file, String cwd)
{
    /**
     * If the IFile is really a project root, then just return the
     * root directory; this function is really only intended to be
     * called with a file inside a project root.
     */
    if (file.compareTo(m_projectDirectory) == 0)
    {
        return cwd;
    }

    try
    {
        Stack subdirs = new Stack();

        // Push the file and its parents onto a stack.
        while ((file = file.getParentIFile()) != null && file.compareTo(m_projectDirectory)
        {
            subdirs.push(file);
        }

        /**
         * For each of the items in the stack, make an IFile and
         * append its name to a path. Then set the current FTP working
         * directory to the one specified by the path.
         */
        while (!subdirs.empty())
        {
            IFile dir = (IFile) subdirs.pop();
            cwd += "/" + dir.getName();
            if (!m_ftp.changeWorkingDirectory(cwd))
            {
                if (!m_ftp.makeDirectory(cwd))
                {
                    break;
                }
            }
        }
    } catch (Exception ex)
```

```

    {
        ex.printStackTrace();
    }

    return cwd;
}

/**
 * Uploads the specified <em>file</em> to the root directory specified
 * by <em>cwd</em>.
 *
 * @param cwd The root directory to upload to.
 * @param file The file to upload.
 */
private void m_uploadFile(String cwd, IFile file)
{
    try
    {
        /**
         * If this is a file, not a directory, then prepend the path to the
         * root and upload it. Publish a status message to the status window.
         */
        if (file.isFile())
        {
            File f = new File(file.getURI());
            FileInputStream stream = new FileInputStream(f);
            String path = cwd + "/" + file.getName();
            boolean ret = m_ftp.storeFile(path, stream);
            if (ret)
            {
                m_win.addMessage(new OutputMessage(path + ": upload successful.", BLACK, null));
            } else
            {
                m_win.addMessage(new OutputMessage(path + ": upload failed.", RED, null));
            }
            return;
        }

        /**
         * If the file is a directory, make a corresponding directory on
         * the receiving server.
         */
        cwd += "/" + file.getName();
        if (!m_ftp.changeWorkingDirectory(cwd))
        {
            if (!m_ftp.makeDirectory(cwd))
            {
                return;
            }
        }

        // Upload each of the files in the current directory.
        IFile[] childIFiles = file.listFiles();
        for (int i = 0; i < childIFiles.length; i++)
        {
            m_uploadFile(cwd, childIFiles[i]);
        }
    } catch (Exception ex)
    {
        ex.printStackTrace();
    }
}

```



```
}    }
```

strings Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

PopupAction Samples

PopupAction Samples

This section contains source code for the following samples.

Samples Included in This Section

PropertyListener Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

src Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

PropertyListener Samples

PropertyListener Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

propertyListener Samples

propertyListener Samples

This section contains source code for the following samples.

Samples Included in This Section

PropertyListenerAction.java Sample

PropertyListenerAction.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/PropertyListener/src/ideExtensions/property

Sample Source Code

```
package ideExtensions.propertyListener;

import com.bea.ide.Application;
import com.bea.ide.actions.DefaultAction;
import com.bea.ide.core.LookAndFeelConstants;
import com.bea.ide.document.IDocument;
import com.bea.ide.ui.IURISelectionContext;
import com.bea.ide.ui.output.OutputMessage;
import com.bea.ide.ui.output.OutputSvc;
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.beans.PropertyChangeEvent;
import java.beans.PropertyChangeListener;
import java.net.URI;
import javax.swing.UIManager;

/**
 * An action class that listens for changes to properties defined by WebLogic
 * Workshop.
 */
public class PropertyListenerAction extends DefaultAction
{
    // A window for displaying messages.
    private OutputSvc.IOutputWindow m_win;
    // Descriptive information about the window.
    private OutputSvc.OutputWindowDescription m_desc;

    /**
     * Constructs an instance of this action class, initializing
     * an output window for messages and adding listeners for property
     * changes.
     */
    public PropertyListenerAction()
    {
        // Initialize the output window.
        m_desc = new OutputSvc.OutputWindowDescription();
        m_desc.title = "PropertyListener";
        m_desc.destination = "PropertyListener";
        m_win = OutputSvc.get().getWindow(m_desc, true, false);

        // Add a listener that will receive an event if a file is edited or saved.
        Application.get().addPropertyChangeListener(Application.EVENT_DocumentDirtyChange, m_ev
        // Add a listener that will receive an event if the URI of the selection changes.
        Application.get().addPropertyChangeListener(Application.PROP_FocusedURIs, m_eventListen
    }
}
```

navExtensionDevKit.html Sample

```

/**
 * Called by the IDE when the Properties -> Property Listener
 * menu command is clicked.
 *
 * @param e The menu command click event.
 */
public void actionPerformed(ActionEvent e)
{
    // Display a window for messages.
    m_win = OutputSvc.get().getWindow(m_desc, true, false);
}

/**
 * Creates a PropertyChangeListener instance whose propertyChange method
 * handles changes to URI focus and document dirty state.
 */
private PropertyChangeListener m_eventListener = new PropertyChangeListener()
{
    /**
     * Called by the IDE when an event occurs for which this listener is
     * registered.
     *
     * @param e The change event that occurred.
     */
    public void propertyChange(PropertyChangeEvent e)
    {
        String propertyName = e.getPropertyName();
        Object oldValueObj = e.getOldValue();
        Object newValueObj = e.getNewValue();

        /**
         * If the property change event is due to a change in selection context
         * (such as a change in which file is selected in the Application
         * window), then print a message describing what is selected.
         */
        if (propertyName.equals(Application.PROP_FocusedURIs))
        {
            if (newValueObj != null && newValueObj instanceof IURISelectionContext)
            {
                IURISelectionContext c = (IURISelectionContext)newValueObj;
                // Get the URIs for selected items.
                URI[] uris = c.getFileURIs();
                if (uris.length > 0)
                {
                    m_win.addMessage("The following files are selected in the Application t
                    // For each URI, print the URI to the window.
                    for (int i = 0; i < uris.length; i++)
                    {
                        // Display a message describing the current selected context.
                        m_win.addMessage(new OutputMessage(getAppRelativePath(uris[i]),
                            UIManager.getColor(LookAndFeelConstants.ALERT_COLOR), null));
                    }
                    m_win.addMessage("");
                }
            }
        }
        /**
         * If the property change event is due to the open document getting edited
         * (ie, becoming "dirty") or saved, then print a message describing
         * what happened.

```

navExtensionDevKit.html Sample

```

    */
    if (propertyName.equals(Application.EVENT_DocumentDirtyChange))
    {
        if(newValueObj != null && newValueObj instanceof IDocument)
        {
            IDocument doc = (IDocument)newValueObj;
            // Find out if the document has been dirtied (true) or saved (false).
            Boolean dirty =(Boolean)doc.getProperty(IDocument.PROP_DocumentDirty);
            // Display a message about the document's "dirty state".
            m_win.addMessage("The following document was just " +
                (dirty.booleanValue() ? "dirtied" : "saved") + ":");
            // Display the document's URI.
            m_win.addMessage(new OutputMessage(getAppRelativePath(doc.getURI()),
                UIManager.getColor(LookAndFeelConstants.ALERT_COLOR), null));
            m_win.addMessage("");
        }
    }
}

};
/**
 * Gets the path of <em>uri</em> relative to the
 * application root. This assumes that the URI corresponds
 * to a file.
 *
 * @param uri A URI corresponding to the file whose path
 * should be retrieved.
 * @return The relative path.
 */
private String getAppRelativePath(URI uri)
{
    if(Application.getWorkspace() == null) { return ""; }
    URI appURI = Application.getWorkspace().getDirectory().getAbsoluteURI();
    return "$APP_HOME/" + uri.toString().substring(appURI.toString().length());
}
}

```

SearchResults Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

SearchResults Samples

SearchResults Samples

This section contains source code for the following samples.

Samples Included in This Section

TocView Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

src Samples

This section contains source code for the following samples.

Samples Included in This Section

toView Samples

tocView Samples

This section contains source code for the following samples.

Samples Included in This Section

ToolBarButton Samples

This section contains source code for the following samples.

Samples Included in This Section

IdeExtensions Samples

src Samples

IdeExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

ToolBarButton Samples

ToolBarButton Samples

This section contains source code for the following samples.

Samples Included in This Section

src Samples

This section contains source code for the following samples.

Samples Included in This Section

ideExtensions Samples

ideExtensions Samples

This section contains source code for the following samples.

Samples Included in This Section

toolbarButton Samples

toolbarButton Samples

This section contains source code for the following samples.

Samples Included in This Section

LaunchBrowser.java Sample

LaunchBrowser.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/IdeDevKit/ToolBarButton/src/ideExtensions/toolbarBu

Sample Source Code

```
package ideExtensions.toolbarButton;

import com.bea.ide.actions.DefaultAction;
import com.bea.ide.ui.browser.BrowserSvc;
import com.bea.ide.ui.frame.FrameSvc;
import com.bea.ide.core.MessageSvc;

import java.awt.event.ActionEvent;
import java.net.URL;

/**
 * An action behind a toolbar button. The button simply launches
 * a browser with the URL in the code below. This class extends DefaultAction,
 * which in turn implements IAction. Action extensions such as menu
 * commands and toolbar buttons implement the IAction interface's actionPerformed
 * method to provide logic for the action. The IDE calls actionPerformed
 * when the command is selected by the IDE's user.
 *
 * The extension.xml file provided in the META-INF folder of the ToolBarButton
 * sample specifies this class as the handler for the button action.
 */
public class LaunchBrowser extends DefaultAction
{
    /**
     * Launches a browser to the URL given in code. This method is
     * called by the IDE when the user clicks the "Favorites" button.
     *
     * @param e The menu click event that occurred in the IDE.
     */
    public void actionPerformed(ActionEvent e)
    {
        try
        {
            BrowserSvc.get().invokeBrowser(new URL("http://dev2dev.bea.com"), true);
        }
        catch(Exception ex)
        {
            // Display a debugging message in the IDE's Output window.
            MessageSvc.get().debugLog("Error while displaying favorite: "
                + ex.getLocalizedMessage());
        }
    }
}
```

TaglibExtDevKit Samples

This section contains source code for the following samples.

Samples Included in This Section

TagHandlers Samples

TaglibInstall Samples

TaglibWebProject Samples

TldxHandlers Samples

TagHandlers Samples

This section contains source code for the following samples.

Samples Included in This Section

TDK Samples

TDK Samples

This section contains source code for the following samples.

Samples Included in This Section

Barcode.java Sample

Barcode.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TagHandlers/TDK/

Sample Source Code

```
package TDK;

import java.util.ArrayList;
import javax.servlet.jsp.JspException;
import javax.servlet.jsp.PageContext;
import javax.servlet.jsp.tagext.Tag;
import javax.servlet.jsp.tagext.BodyTagSupport;

public class Barcode extends BodyTagSupport
{
    protected String encodingType = null;
    public void setEncodingType(String type)
    {
        encodingType = type;
    }
    public String getEncodingType()
    {
        return encodingType;
    }

    protected String value = null;
    public void setValue(String value)
    {
        this.value = value;
    }
    public String getValue()
    {
        return value;
    }

    public int doStartTag() throws JspException
    {
        return EVAL_BODY_TAG;
    }

    public int doEndTag() throws JspException
    {
        StringBuffer html = new StringBuffer();
        html.append("<div>");
        html.append("SUCCESS: value to encode=" + value);
        html.append("</div>");
        try
        {
            pageContext.getOut().print(html.toString());
        }
    }
}
```

navExtensionDevKit.html Sample

```
    }  
    catch(Exception e) { e.printStackTrace(); }  
    return EVAL_PAGE;  
  }  
}
```

TaglibInstall Samples

This section contains source code for the following samples.

Samples Included in This Section

TDK Samples

template Samples

TDK Samples

This section contains source code for the following samples.

Samples Included in This Section

TemplateProcessor.java Sample

TemplateProcessor.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TaglibInstall/TDK/

Sample Source Code

```
package TDK;

import com.bea.ide.workspace.project.IProjectTemplate;
import com.bea.ide.workspace.project.IProjectTemplateProcessor;
import com.bea.ide.workspace.project.web.IWebProjectDriver;
import com.bea.wlw.config.j2ee.WebAppDocument.WebApp;
import com.bea.wlw.config.j2ee.WebAppDocument.WebApp.Taglib;
import com.bea.xbean.tool.XMLBean;

public class TemplateProcessor implements IProjectTemplateProcessor
{
    private IProjectTemplate _template;
    private IContext _context;

    public TemplateProcessor (IProjectTemplate template, IContext c)
    {
        _template = template;
        _context = c;
    }

    public boolean load()
    {
        _context.loadAllContent(_template);
        IWebProjectDriver driver = (IWebProjectDriver)_context.getProject().getDriver(IWebProjectDriver.class);
        WebApp bean = (WebApp)driver.getWebXmlBean();

        // UNDONE -- extract the taglibs from template dynamically.
        String taglibLocation = "/WEB-INF/tdk-tags.tld";
        String taglibUri = "tdk-tags.tld";

        // check if this guy is already defined in web.xml
        Taglib[] taglibs = bean.getTaglibArray();
        for(int i = 0; i < taglibs.length; i++)
        {
            if(taglibs[i].getTaglibLocation().equals(taglibLocation) &&
                taglibs[i].getTaglibUri().equals(taglibUri))
            {
                return true;
            }
        }

        // add a new <taglib> entry to web.xml using xmlbeans.
        Taglib taglib = bean.addNewTaglib();
        taglib.setTaglibLocation(taglibLocation);
        taglib.setTaglibUri(taglibUri);
    }
}
```

navExtensionDevKit.html Sample

```
        bean.setTaglibArray(taglibs.length, taglib);
        driver.setWebXmlBean(bean, true);
        return true;
    }

    public String check()
    {
        return null;
    }
}
```

template Samples

This section contains source code for the following samples.

Samples Included in This Section

Template.xml Sample

Template.xml Sample

This topic includes the source code for the Template.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TaglibInstall/template/

Sample Source Code

```
<template-definition>

  <project-template id="TDK_install"
    type="urn:com-bea-ide:project.type:WebApp"
    extends="base"
    processor="TDK.TemplateProcessor" >

    <display
      location="contextmenu"
      label=".: TDK Taglibs :."
      description="Installs the tdk taglibs." />

    <content type="archive" destination="project" source="tdk-resources.zip" overwrite="true" />
  </project-template>

</template-definition>
```

TaglibWebProject Samples

This section contains source code for the following samples.

Samples Included in This Section

Controller.jspf Sample

Demo.jsp Sample

Error.jsp Sample

Index.jsp Sample

Controller.jpf Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TaglibWebProject/

Sample Source Code

```
import com.bea.wlw.netui.pageflow.PageFlowController;
import com.bea.wlw.netui.pageflow.Forward;

/**
 * This is the default controller for a blank web application.
 *
 * @jpf:controller
 * @jpf:view-properties view-properties::
 * <!-- This data is auto-generated. Hand-editing this section is not recommended. -->
 * <view-properties>
 *     <pageflow-object id="action:begin.do">
 *         <property name="x" value="60"/>
 *         <property name="y" value="80"/>
 *     </pageflow-object>
 *     <pageflow-object id="page:index.jsp">
 *         <property name="x" value="220"/>
 *         <property name="y" value="80"/>
 *     </pageflow-object>
 *     <pageflow-object id="page:error.jsp">
 *         <property name="x" value="220"/>
 *         <property name="y" value="160"/>
 *     </pageflow-object>
 *     <pageflow-object id="forward:path#index#index.jsp#@action:begin.do@">
 *         <property name="elbowsY" value="72,72,72,72"/>
 *         <property name="elbowsX" value="96,140,140,184"/>
 *         <property name="toPort" value="West_1"/>
 *         <property name="fromPort" value="East_1"/>
 *         <property name="label" value="index"/>
 *     </pageflow-object>
 *     <pageflow-object id="pageflow:/Controller.jpf"/>
 * </view-properties>::
 */
public class Controller extends PageFlowController
{
    /**
     * @jpf:action
     * @jpf:forward name="index" path="index.jsp"
     */
    protected Forward begin()
    {
        return new Forward("index");
    }
}
```

Demo.jsp Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TaglibWebProject/

Sample Source Code

```
<!--Generated by WebLogic Workshop-->
<%@ 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"%>
<%@ taglib prefix="tdk" uri="tdk-tags.tld"%>

<netui:html>
  <head>
    <title>
      Demo Taglib Page
    </title>
  </head>
  <body>
    <p>
      <tdk:barcode encodingType="code39" value="vogue"></tdk:barcode>
    </p>
  </body>
</netui:html>
```


Error.jsp Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TaglibWebProject/

Sample Source Code

```
<!--Generated by WebLogic Workshop-->
<%@ page language="java" contentType="text/html; charset=UTF-8" isErrorPage="true"%>
<%@ 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>Error</title>
  </head>
  <body>
    <p>
      An error has occurred:
    </p>
    <blockquote>
      <netui:label value="{request.errorMessage}" defaultValue="" />
      <br/>
      <netui:exceptions showMessage="true" />
    </blockquote>
  </body>
</netui:html>

<!-- Some browsers will not display this page unless the response status code is 200. -->
<% response.setStatus(200); %>
```

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/ExtensionDevKit/TaglibExtDevKit/TaglibWebProject/

Sample Source Code

```
<!--Generated by WebLogic Workshop-->
<%@ 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>
      New Web Application Page
    </p>
  </body>
</netui:html>
```

TidxHandlers Samples

This section contains source code for the following samples.

Samples Included in This Section

help Samples

TDK Samples

help Samples

This section contains source code for the following samples.

Samples Included in This Section

en Samples

en Samples

This section contains source code for the following samples.

Samples Included in This Section

tldx Samples

tldx Samples

This section contains source code for the following samples.

Samples Included in This Section

Index.html Sample

Overview.html Sample

Reference.html Sample

taglib Samples

Toc.xml Sample

Index.html Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/help/en/tldx/

Sample Source Code

```
index.html content goes here...
```

Overview.html Sample

This topic includes the source code for the Overview.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/help/en/tldx/

Sample Source Code

overview page placeholder.

Reference.html Sample

This topic includes the source code for the Reference.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/help/en/tldx/

Sample Source Code

reference page placeholder.

taglib Samples

This section contains source code for the following samples.

Samples Included in This Section

www.bea.com Samples

www.bea.com Samples

This section contains source code for the following samples.

Samples Included in This Section

tdk Samples

tdk Samples

This section contains source code for the following samples.

Samples Included in This Section

tdk-tags-1.0 Samples

tdk-tags-1.0 Samples

This section contains source code for the following samples.

Samples Included in This Section

Barcode.html Sample

Barcode.html Sample

This topic includes the source code for the Barcode.html Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/help/en/tldx/taglib/ww

Sample Source Code

```
<html>
<body>
barcode.<br />
barcode.barcode.<br />
barcode.barcode.barcode.<br />
barcode.barcode.barcode.barcode.<br />
barcode.barcode.barcode.barcode.barcode.<br />
barcode.barcode.barcode.barcode.barcode.<br />
barcode.barcode.barcode.<br />
barcode.barcode.<br />
</body>
</html>
```

Toc.xml Sample

This topic includes the source code for the Toc.xml Sample.

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/help/en/tldx/

Sample Source Code

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<toc-root component="taglibs" xmlns="http://www.bea.com/help/toc.xsd">
  <toc-reference anchor="extensions">
    <toc-node label="tdk taglib docs" url="./index.html" >
      <toc-node label="Overview" url="./overview.html" />
      <toc-node label="Reference" url="./reference.html" >
        <toc-node label="custom barcode tag" url="./taglib/www.bea.com/
      </toc-node>
    </toc-node>
  </toc-reference>
</toc-root>
```

TDK Samples

This section contains source code for the following samples.

Samples Included in This Section

BarcodeEncodings.java Sample

PaletteGenerator.java Sample

Renderer.java Sample

BarcodeEncodings.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/TDK/

Sample Source Code

```
package TDK;

import java.util.ArrayList;

public class BarcodeEncodings
{
    static class Slice implements java.io.Serializable
    {
        public int relStartX;
        public int relStartY;
        public int width;
        public int height;
        public String color;

        public Slice() { }
        public Slice(int startX, int startY, int w, int h, String c)
        {
            relStartX = startX;
            relStartY = startY;
            width = w;
            height = h;
            color = c;
        }
    }

    static class tokenEncoding implements java.io.Serializable
    {
        public static final char EMPTY_CHAR = '#';

        public char token;
        public ArrayList slices;

        public tokenEncoding(char c)
        {
            token = c;
            slices = new ArrayList();
        }

        public void add(Slice s)
        {
            slices.add(s);
        }
    }
}
```

navExtensionDevKit.html Sample

```
// DEPRECATED. doesn't seem to be marshalled quite right.
public static class BarcodeException extends Throwable
{
    String _message;
    public BarcodeException(String m)
    {
        _message = m;
    }
    public String getError()
    {
        return _message;
    }
}

public interface Encoding extends java.io.Serializable
{
    public static final String BLACK = "#000000";
    public static final String WHITE = "#FFFFFF";

    public abstract ArrayList encode();
    public boolean verifyInput(String src) throws Exception;
}

public static class code39Encoding implements Encoding
{
    public String src;
    public ArrayList encoding;

    public static final int WIDE = 0;
    public static final int NORMAL = 1;
    public static final int NORMAL_BAR_WIDTH = 1;
    public static final int WIDE_BAR_WIDTH = 3;
    public static final int DEFAULT_HEIGHT = 60;

    public code39Encoding(String s)
    {
        src = s;
        encoding = new ArrayList();
    }

    public boolean verifyInput(String src) throws Exception
    {
        if(src == null)
        {
            throw new Exception("generateCode39(): expected non-empty input.");
        }
        return true;
    }

    public ArrayList encode()
    {
        src = src.toUpperCase();
        for(int i = 0; i < src.length(); i++)
        {
            char c = src.charAt(i);
            tokenEncoding slices = new tokenEncoding(c);

            switch(c)
            {
                case 'A':
```

```

slices.add(processBlackSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;

case 'B':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;

case 'C':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'D':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;

case 'E':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));

```

navExtensionDevKit.html Sample

```
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'F':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'G':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;

case 'H':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'I':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'J':

slices.add(processBlackSlice(NORMAL));
```

```

slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'K':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
break;

case 'L':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
break;

case 'M':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
break;

case 'N':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
break;

```

```

case 'O':

    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(NORMAL));
    break;

case 'P':

    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(NORMAL));
    break;

case 'Q':

    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(WIDE));
    break;

case 'R':

    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(NORMAL));
    break;

case 'S':

    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));

```

```

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
break;

case 'T':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
break;

case 'U':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;

case 'V':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;

case 'W':

slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;

case 'X':

slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));

```

navExtensionDevKit.html Sample

```
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
break;
```

```
case 'Y':
```

```
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;
```

```
case 'Z':
```

```
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
break;
```

```
case '0':
```

```
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
break;
```

```
case '1':
```

```
slices.add(processBlackSlice(WIDE));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(WIDE));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(NORMAL));
slices.add(processWhiteSlice(NORMAL));
slices.add(processBlackSlice(WIDE));
```



```

break;

case '2':

    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    break;

case '3':

    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    break;

case '4':

    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    break;

case '5':

    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(NORMAL));
    break;

case '6':

    slices.add(processBlackSlice(NORMAL));
    slices.add(processWhiteSlice(NORMAL));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(WIDE));
    slices.add(processBlackSlice(WIDE));
    slices.add(processWhiteSlice(NORMAL));

```

```

        slices.add(processBlackSlice(NORMAL));
        slices.add(processWhiteSlice(NORMAL));
        slices.add(processBlackSlice(NORMAL));
        break;

        case '7':

            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(WIDE));
            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(WIDE));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(WIDE));
            break;

        case '8':

            slices.add(processBlackSlice(WIDE));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(WIDE));
            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(WIDE));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(NORMAL));
            break;

        case '9':

            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(WIDE));
            slices.add(processWhiteSlice(WIDE));
            slices.add(processBlackSlice(NORMAL));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(WIDE));
            slices.add(processWhiteSlice(NORMAL));
            slices.add(processBlackSlice(NORMAL));
            break;
    }

    encoding.add(slices);

} // end for

return encoding;
}

public Slice processBlackSlice(int type)
{
    return processBlackSlice(type, DEFAULT_HEIGHT);
}

public Slice processBlackSlice(int type, int height)
{
    if(type == NORMAL)

```

navExtensionDevKit.html Sample

```
{
    return new Slice(0, 0, NORMAL_BAR_WIDTH, height, BLACK);
}
else if(type == WIDE)
{
    return new Slice(0, 0, WIDE_BAR_WIDTH, height, BLACK);
} else
{
    return new Slice(0, 0, NORMAL_BAR_WIDTH, height, WHITE);
}
}

public Slice processWhiteSlice(int type)
{
    return processWhiteSlice(type, DEFAULT_HEIGHT);
}

public Slice processWhiteSlice(int type, int height)
{
    if(type == NORMAL)
    {
        return new Slice(0, 0, NORMAL_BAR_WIDTH, height, WHITE);
    }
    else if(type == WIDE)
    {
        return new Slice(0, 0, WIDE_BAR_WIDTH, height, WHITE);
    }
    else
    {
        return new Slice(0, 0, NORMAL_BAR_WIDTH, height, WHITE);
    }
}
}

public static class UPCEncoding implements Encoding
{
    public String src;
    public ArrayList encoding;

    public static final int DEFAULT_START_X = 0;
    public static final int DEFAULT_START_Y = 0;
    public static final int DEFAULT_HEIGHT = 60;
    public static final int ENCODING_MID = 5;
    public static final int ENCODING_END = 10;

    public UPCEncoding(String s)
    {
        src = s;
        encoding = new ArrayList();
    }

    public boolean verifyInput(String src) throws Exception
    {
        if(src == null)
        {
            throw new Exception("generateUPC(): expected non-empty input.");
        }

        // pre-process and verify that this is a sequence of digits.
        if(src.length() != 10)
        {

```

navExtensionDevKit.html Sample

```
        throw new Exception("generateUPC(): expected input containing 10 digits.");
    }

    for(int i = 0; i < src.length(); i++)
    {
        char c = src.charAt(i);
        if(!Character.isDigit(c))
        {
            throw new Exception("generateUPC(): expected input containing all digits.");
        }
    }

    return true;
}

public ArrayList encode()
{
    tokenEncoding slices;

    // logic for first five digits.
    for(int i = 0; i < ENCODING_MID && i < src.length(); i++)
    {
        char c = src.charAt(i);
        slices = new tokenEncoding(c);
        switch(c)
        {
            case '0':
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                break;
            case '1':
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                break;
            case '2':
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                break;
            case '3':
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                break;
            case '4':
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
                slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
                slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        }
    }
}
```

```

        break;
    case '5':
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        break;
    case '6':
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        break;
    case '7':
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        break;
    case '8':
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        break;
    case '9':
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        break;
    default:
        break;
}

encoding.add(slices);
}

// insert middle bars.
slices = new tokenEncoding(tokenEncoding.EMPTY_CHAR);

slices.add(processBlackSlice(DEFAULT_START_X, -10, 1, 80));
slices.add(processWhiteSlice(DEFAULT_START_X, -10, 1, 80));
slices.add(processBlackSlice(DEFAULT_START_X, -10, 1, 80));
slices.add(processWhiteSlice(DEFAULT_START_X, -10, 1, 80));

encoding.add(slices);

// switch to right-parity.

// logic for last five digits.
for(int i = ENCODING_MID; i < ENCODING_END && i < src.length(); i++)
{
    char c = src.charAt(i);
    slices = new tokenEncoding(c);
    switch(c)

```

```

{
    case '0':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        break;
    case '1':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        break;
    case '2':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        break;
    case '3':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 5));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        break;
    case '4':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));

        break;
    case '5':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        break;
    case '6':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 5));
        break;
    case '7':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
        slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
        break;
    case '8':

        slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));

```

navExtensionDevKit.html Sample

```
slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 4));
break;
case '9':

    slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
    slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 2));
    slices.add(processBlackSlice(DEFAULT_START_X, DEFAULT_START_Y, 1));
    slices.add(processWhiteSlice(DEFAULT_START_X, DEFAULT_START_Y, 3));
    break;
default:
    break;
}

encoding.add(slices);
}

return encoding;
}

public Slice processBlackSlice(int relStartX, int relStartY, int width)
{
    return processBlackSlice(relStartX, relStartY, width, DEFAULT_HEIGHT);
}

public Slice processBlackSlice(int relStartX, int relStartY, int width, int height)
{
    return new Slice(relStartX, relStartY, width, height, BLACK);
}

public Slice processWhiteSlice(int relStartX, int relStartY, int width)
{
    return processWhiteSlice(relStartX, relStartY, width, DEFAULT_HEIGHT);
}

public Slice processWhiteSlice(int relStartX, int relStartY, int width, int height)
{
    return new Slice(relStartX, relStartY, width, height, WHITE);
}

public static ArrayList generateCode39(String src) throws Exception
{
    code39Encoding encoding = new code39Encoding(src);
    encoding.verifyInput(src);
    ArrayList arr = encoding.encode();
    return arr;
}

/*
public ArrayList generateUPC(String src) throws Exception
{
    UPCEncoding encoding = new UPCEncoding(src);
    encoding.verifyInput(src);
    return encoding.encode();
}
*/
}
```

PaletteGenerator.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/TDK/

Sample Source Code

```
package TDK;

import com.bea.ide.Application;
import com.bea.ide.jspdesigner.InsertWizard;
import com.bea.ide.jspdesigner.PaletteItem;
import com.bea.ide.lang.jsp.IDOMDocument;
import com.bea.ide.lang.jsp.IDOMNode;
import com.bea.ide.lang.jsp.IDOMTextNode;
import com.bea.ide.lang.jsp.IJspDocument;
import com.bea.ide.lang.jsp.IJspAction;
import com.bea.ide.lang.jsp.IJspTaglibDirective;
import java.awt.Frame;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ComponentAdapter;
import java.awt.event.ComponentEvent;
import java.beans.PropertyChangeEvent;
import java.util.ArrayList;
import javax.servlet.jsp.tagext.TagInfo;
import javax.swing.AbstractAction;
import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextArea;
import java.beans.PropertyChangeListener;
import javax.servlet.jsp.tagext.TagLibraryInfo;
import javax.swing.JLabel;
import javax.swing.JTextField;

public class PaletteGenerator extends com.bea.ide.jspdesigner.PaletteGenerator
{
    public String getLabel()
    {
        return ".:TDK tags:.";
    }

    public int getPriority()
    {
        return 10;
    }
}
```


navExtensionDevKit.html Sample

```
class DemoPaletteItem extends PaletteItem
{
    private String _label;
    private IDOMDocument _doc;
    private TagLibraryInfo _tli;

    public DemoPaletteItem(String label, IDOMDocument doc, TagLibraryInfo tli)
    {
        _label = label;
        _doc = doc;
        _tli = tli;
    }
    public String getLabel() { return ":@" + _label + ":@"; }
    public IDOMDocument getInsertDocument() { return _doc; }

    // attach a custom wizard to the action of dragging the "barcode" tag onto jsp designer
    public InsertWizard getInsertWizard()
    {
        if(_label.equals("barcode"))
        {
            return new DemoBarcodeInsertWizard(_tli);
        }
        return null;
    }
}

class DemoBarcodeInsertWizard extends InsertWizard
{
    private JDialog _dialog;
    private TagLibraryInfo _tli;

    public DemoBarcodeInsertWizard(TagLibraryInfo tli)
    {
        _tli = tli;
    }

    public IDOMDocument invokeWizard(IDOMNode nodeParent, IDOMNode nodeRefChild)
    {
        IDOMDocument docFrag = nodeParent.getOwnerDocument().createCompatibleDocument();
        IJspAction el = null;

        JPanel panel = new JPanel();
        JTextField textField = new JTextField(15);
        JLabel label = new JLabel("Yo. Enter snippet to encode in code39 format: ");
        panel.add(label);
        panel.add(textField);
        if(JOptionPane.showConfirmDialog(Application.getRootFrame(), panel, "Demo Insert Wi
        {
            // insert a tag into the document with the name <tdk:barcode>
            el = (IJspAction) docFrag.createElement(IJspAction.class, _tli.getShortName() +

            // we only support code39 right now.
            el.setAttribute("encodingType", "code39");
            el.setAttribute("value", textField.getText());

            docFrag.appendChild(el);
            return docFrag;
        }

        return null;
    }
}
```

```
        }  
    }  
  
    public PaletteItem[] getPaletteItems()  
    {  
        TagInfo[] tagInfoArray = getTagLibraryInfo().getTags();  
        PaletteItem ret[] = new PaletteItem[tagInfoArray.length];  
        for(int i = 0; i < tagInfoArray.length; i++)  
        {  
            TagInfo tagInfo = tagInfoArray[i];  
            ret[i] = new DemoPaletteItem(tagInfo.getTagName(), null, getTagLibraryInfo());  
        }  
  
        return ret;  
    }  
}
```

Renderer.java Sample

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

Sample Location

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

BEA_HOME/weblogic81/samples/workshop/ExtensionDevKit/TaglibExtDevKit/TldxHandlers/TDK/

Sample Source Code

```
package TDK;

import TDK.BarcodeEncodings.Slice;
import TDK.BarcodeEncodings.tokenEncoding;
import com.bea.ide.jspdesigner.IHTMLWriter;
import com.bea.ide.lang.jsp.IDOMNode;
import com.bea.ide.lang.jsp.IDOMTextNode;
import com.bea.ide.lang.jsp.IJspAction;
import java.util.ArrayList;
import java.util.ListIterator;

public class Renderer extends com.bea.ide.jspdesigner.Renderer
{
    /**
     * creates the design time html-representation of the barcode tag.
     */
    public void writeTag(IHTMLWriter writer, IDOMNode node)
    {
        IJspAction el = (IJspAction) node;
        Object attr = el.getAttribute("encodingType");
        String encodingType = attr != null ? attr.toString() : "";
        attr = el.getAttribute("value");
        String value = attr != null ? attr.toString() : "";

        StringBuffer html = new StringBuffer();
        html.append("<br /><br />");

        if(encodingType.equals("code39"))
        {
            try
            {
                html.append("<table cellpadding=\"0\" cellspacing=\"0\" border=\"0\">\n");
                html.append("<tr>\n");

                ArrayList enc = BarcodeEncodings.generateCode39(value);
                ListIterator encodingIter = enc.listIterator();
                while(encodingIter.hasNext())
                {
                    // <!-- begin token TABLE -->
                    html.append("<td>\n");
                    html.append("<table cellpadding=\"0\" cellspacing=\"0\" border=\"0\">\n");
                    html.append("<tr>\n");
                    tokenEncoding tEncoding = (tokenEncoding) encodingIter.next();
                    ArrayList slices = tEncoding.slices;
                    ListIterator slicesIter = slices.listIterator();

```

navExtensionDevKit.html Sample

```
int count = 0;
while(slicesIter.hasNext())
{
    Slice slice = (Slice)slicesIter.next();
    count++;

    html.append("<td width=\"" + slice.width + "px\"");
    html.append("<table cellpadding=\"" + 0 + "\" cellspacing=");
    html.append("<tr>\n");
    html.append("<td width=\"" + slice.width + "px\"");
    html.append("</tr>\n");
    html.append("</table>\n");
    html.append("</td>\n");

} // end inner while.

html.append("</tr>\n");
html.append("<tr>\n");
html.append("<td align=\"center\" style=\"font:9px Verdana\"");
html.append(tEncoding.token + "\n");
html.append("</td>\n");
html.append("</tr>\n");
html.append("</table>\n");
html.append("</td>\n");
// <!-- end token TABLE -->

} // end outer while.

html.append("</tr>\n");
html.append("</table>\n");
}
catch(Exception e) { e.printStackTrace(); }
}

html.append("<br /><br />");
writer.write(html.toString());
}
```