

Döcu Content

IBM DB2, Lotus Notes Domino

Oracle JDeveloper

Xpages, JSF Apps

Part 1

Foreword: Steps included below to create Apps via Oracle JDeveloper on IBM DB2 database, IBM Lotus Notes Domino on NSF back-end are current, however we suggest visiting Oracle and IBM websites for additional information.

System Requirements:

- Windows 7, 8, 10 Operating System Compatible
- Download IBM DB2 Express-C 9.x, 10.x, 11.x (not the Trial Version)
- <http://www.ibm.com/developerworks/downloads/im/db2express/>
- Download IBM Data Studio 4.x
- <http://www.ibm.com/developerworks/downloads/im/data/>
- Download IBM Lotus Notes Domino Designer 8.5.3, 9.x
- <https://www.ibm.com/developerworks/downloads/ls/dominodesigner/>

Disclaimer:

Information contained in this documentation is presented as is, we assume you are familiar with IBM DB2, Lotus Notes Domino Designer and Oracle JDeveloper. Rest assured, if you need additional support be sure to contact us or visit our website: www.dokollolutionsinc.com for Free, ready to use, Step by Step PDF or YouTube Video Tutorials; if you would rather use your favourite search engine for help, we advise you to do so...

Introduction:

Create two sample Apps via IBM Lotus Notes Domino Designer and Oracle JDeveloper, run them side by side to communicate via XML file available in a URL. You will be using code below to help you along (JavaBean, JavaAgent, and Xpages), areas of interest will be highlighted for your convenience.

Step 1

At this point, you should have gone through System requirements above- We suspect you already downloaded software and configured your environments; we are going to skip formal steps. We urge you to consult our documentation for support. Let's begin by setting you up for *success* with the samples in this tutorial, don't want to leave you guessing...

Hint # 1: If you downloaded IBM DB2 Express-C for this tutorial, you should be aware version 9.x that comes with Control Center is deprecated, you will need to download Data Studio- Perhaps you already have that version (9.x), rest assured you can still use it, powerful database application.

Hint # 2: If you downloaded IBM Data Studio 4.x or and earlier version separately for this tutorial, what a treat! However, you should know configuration can be a little getting used to. Please visit our website, Google, or use preferred search engine for additional support.

Hint # 3: No known issues reported with IBM Notes Domino Designer... if by any chance your Java code is faulty and your App is stuck, kill this bad boy via Task Manager, or reboot your system. You may also need to Clean your project in the IDE.

Hint # 4: No real issues reported with Oracle JDeveloper, except that it is helpful to run WebLogic Server prior to launching your Apps. If errors occur in your JSF pages, try to Rebuild the App in question, then you should be good to go, normal stuff-

Alright, enough Spoon-feeding on this front...

What to Expect:

The current consists of two Xpages, one JSF, one JavaBean, one Backing Bean, and one JavaAgent, with an XML file in the middle. Since you've done this before, there's a likelihood that you created a Form and a View in the Notes Domino back-end and are ready to write Java code to interact between IBM and Oracle environments.

Form

Oracle Feeds Form

<input type="text" value="UserName"/>			
Item Number	<input type="text" value="itemNumber"/>	Priority	<input type="text" value="priority"/>
Category	<input type="text" value="Categories"/>	Requestor	<input type="text" value="requestor"/>
AssignedTo	<input type="text" value="assignedTo"/>	Date Created	<input type="text" value="DateCreated"/>
Topic	<input type="text" value="Subject"/>	Status	<input type="text" value="status"/>
UserIP	<input type="text" value="UserIP"/>		

<input type="text" value="PagelD"/>
<input type="text" value="PagelD"/>

<input type="text" value="CancelAccess"/>	<input type="text" value="Reason"/>
<input type="text" value="CancelAccess"/>	<input type="text" value="Reason"/>

View

UserName	PagelD	Priority	Issues Reported	DateCreated
FrenchGirl	http://localhost/docucontent.nsf/xporaclewebfeedsdata.xsp	High	French Gilr does n	2016-08-22-01.06.13
FyooryNick	http://localhost/docucontent.nsf/xporaclewebfeedsdata.xsp	High	Nick can see the A	2016-08-22-01.07.57
HarmourMan	http://localhost/docucontent.nsf/xporaclewebfeedsdata.xsp	Low	Testing for Harmoi	2016-08-22-01.04.49
SymoneBilz	http://localhost/docucontent.nsf/xporaclewebfeedsdata.xsp	High	Testing for Symon	2016-08-22-00.26.30
SymoneBilz	http://localhost/docucontent.nsf/xporaclewebfeedsdata.xsp	Low	Test	2016-08-22-14.05.51

Additional Spoon-feeding forthcoming, why waste your time eh? you've done this before! Read through, then copy and paste the following to your environment

Here is the Xpages Code: What we've done here is we made use of a Cookie running to collect the UserName and UserRole, based on Login data and reused those to post on the Xpages file to limit having to type into inputText controls on the page. We also wanted to post a message where data

collected from the Xpage is compared against back-end values to see if the record being submitted is NOT already in the system, thus to proceed- If all is set in stone, you will be presented with a results page, containing a list of records submitted, including your own.

xporaclewebfeedsdata.xsp

```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core" pageTitle="Bulletin Posting Page"
  xmlns:xc="http://www.ibm.com/xsp/custom" style="background-
color:rgb(180,175,188) ">
```

```
  <xp:this.resources>
    <xp:script src="/docucontentGACodeTest.js" clientSide="true" />
  </xp:this.resources>
```

```
  <xp:this.navigationRules>
    <xp:navigationRule outcome="xsp-success"
      viewId="/xporaclefeedssubmittedview.xsp" />
```

```
    <xp:navigationRule outcome="xsp-failure"
      viewId="/xporaclewebfeedsdata.xsp" />
```

```
  </xp:this.navigationRules>
```

```
  <xp:tabbedPanel id="tabbedPanel1">
    <xp:tabPanel label="Submit Web XML Feeds Record" id="tabPanel1">
      <xp:panel
        style="width:859.0px;text-align:center;font-
size:18pt;height:444.0px">
        <xp:table align="left"
          style="background-
color:rgb(111,103,17);width:85.0%;font-size:9pt">
          <xp:tr style="font-size:10pt">
            <xp:td colspan="4">
              <xp:span style="font-weight:bold">
                PageID
              </xp:span>
              <xp:br style="font-weight:bold"
id="br1"></xp:br>
              <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.pageID}"
                id="pageID" style="font-
size:9pt;width:505.0px" rendered="true" />
              <xp:br id="br2"></xp:br>
            </xp:td>
          </xp:tr>
        </xp:table>
      </xp:panel>
    </xp:tabPanel>
  </xp:tabbedPanel>
```

```

                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right">UserName</xp:td>
                <xp:td>
                    <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.userName}"
                    id="userName" style="font-
size:9pt;width:143.0px" rendered="true"

defaultValue="#{SendOracleXMLFeedsJavaBean.userNameValue}"
                    disabled="true" />
                </xp:td>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right">UserRole</xp:td>
                <xp:td>

                    <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.userRole}"
                    id="userRole" style="font-
size:9pt;width:144.0px" rendered="true"

defaultValue="#{SendOracleXMLFeedsJavaBean.roleNameValue}"
                    disabled="true" />
                </xp:td>

            </xp:tr>
            <xp:tr>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right"></xp:td>
                <xp:td></xp:td>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right"></xp:td>
                <xp:td></xp:td>

            </xp:tr>
            <xp:tr style="font-size:10pt">
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right">DateCreated</xp:td>
                <xp:td>
                    <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.dateCreated}"
                    id="dateCreated" style="font-
size:9pt;width:143.0px" disabled="true" />
                </xp:td>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right">UserIP</xp:td>

```

```

                <xp:td style="margin-left:-24px;font-
size:11pt">
                    <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.userIP}"
                    id="userIP" style="font-
size:9pt;width:144.0px" rendered="true"
                    disabled="true" />

                </xp:td>

            </xp:tr>
            <xp:tr>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right"></xp:td>
                <xp:td></xp:td>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right"></xp:td>
                <xp:td></xp:td>

            </xp:tr>
            <xp:tr style="font-size:10pt">
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right">Priority</xp:td>
                <xp:td>
                    <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.priority}"
                    id="priority" style="font-
size:9pt;width:143.0px" />
                </xp:td>
                <xp:td style="font-weight:bold;margin-
left:24px;text-align:right">ItemNumber</xp:td>
                <xp:td style="margin-left:-24px;font-
size:11pt">
                    <xp:inputText
value="#{SendOracleXMLFeedsJavaBean.itemNumber}"
                    id="itemNumber" style="font-
size:9pt;width:144.0px" rendered="true"
                    disabled="true" />

                </xp:td>

            </xp:tr>
        </xp:tr>
    </xp:tr>

```

```

        <xp:tr>
            <xp:td></xp:td>
            <xp:td></xp:td>
            <xp:td></xp:td>
            <xp:td></xp:td>

        </xp:tr>
        <xp:tr style="font-size:10pt">
            <xp:td style="margin-left:24px;text-align:right;font-weight:bold">
                <xp:div style="margin-left:24px;text-align:right;font-weight:bold">Issues</xp:div>
            </xp:td>
            <xp:td colspan="3">
                <xp:div style="font-size:12pt">
                    <xp:div style="font-size:13pt">
                        <xp:inputTextarea
                            value="#{SendOracleXMLFeedsJavaBean.issuesReported}" id="issuesReported"
                            style="width:411.0px;height:83.0px">
                    </xp:inputTextarea>
                </xp:div>

                <xp:label
                    value="#{SendOracleXMLFeedsJavaBean.message}"
                    id="errMessage1"
                    style="width:298.0px;color:rgb(117,11,11);font-weight:bold;background-color:rgb(255,255,204);text-align:center;font-size:10pt">
                </xp:label>

                <xp:messages layout="table"
                    style="color:red;font-size:9pt;font-weight:bold" id="messages1" />
            </xp:div>

        </xp:td>

    </xp:tr>
    <xp:tr>
        <xp:td></xp:td>
        <xp:td>
            <xp:button value="Submit" id="button2"
                style="font-size:9pt">
                <xp:eventHandler event="onclick"

```

```

submit="true"
refreshMode="complete"
immediate="false" save="true" id="eventHandler2">
    <xp:this.action><!
[CDATA[#{javascript:SendOracleXMLFeedsJavaBean.submitEntry(
) }]]></xp:this.action>
        </xp:eventHandler>
    </xp:button>
</xp:td>
<xp:td></xp:td>
<xp:td style="width:449.0px">
</xp:td>

</xp:tr>
<xp:tr>
    <xp:td></xp:td>
    <xp:td></xp:td>
    <xp:td></xp:td>
    <xp:td></xp:td>
</xp:tr>
</xp:table>

    </xp:panel>
</xp:tabPanel>
</xp:tabbedPanel>

</xp:view>

```

Xpages Submit Design # 1

Home x docuc... x Login... x Send... x WebNe... x Java... x Compa... x News... x xpor... x

Submit Web XML Feeds Record

PageID

UserName UserRole

DateCreated UserIP

Priority ItemNumber

Issues

{errorMessage1} (Error Messages)

Submit

JavaBean code running Xpages File: Copy and paste below JavaBean in your environment to work with the Xpages file submitted earlier and load records into the back-end which can be used to create an XML file. The process is simple, when the Xpages submit button is accessed, two views are searched systematically to collect known values and append the appropriate validation message; of course, areas of interest are highlighted for your convenience... The Bean works hand in hand with a JavaAgent, being posted in a little bit, that creates the actual XML file. The mechanism is that once the XML file is pulled using an External Oracle JDeveloper JSF App via URL, the data is extracted and loaded into an IBM DB2 database.

SendOracleXMLFeedsJavaBean.java

```
/**
 * Created from Original: 2016.08.14.12.33.PM
 * New Oracle XML Feeds data Posting via Xpages form and JavaBean
 * for DocuWebNews App in Oracle JDeveloper
 * Added Info: Domino App Bean works in conjunction with JavaAgent
 * (WebNewsForHTTPUserXMLDataJavaAgent | javaagentdataforwebnewshttp.xml)
 */
package com.dokoll.solutions.inc.oracle.dev;

//...
//Java imports...
import java.io.Serializable;
```

```

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Random;

//...
//faces imports...
import javax.faces.context.FacesContext;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServletRequest;

//domino imports...
import lotus.domino.NotesException;
import lotus.domino.View;
import lotus.domino.ViewEntry;
import lotus.domino.ViewEntryCollection;
import lotus.domino.local.Database;
import lotus.domino.local.Document;

/**
 * @author Dököll Solutions, Inc.
 * @version 2016.08.14.12.33.PM
 */

public class SendOracleXMLFeedsJavaBean implements Serializable {

    static String ISSUE_ACCESS_VIEWNAME =
    "NewIssuesListings";

    static String URL_ACCESS_VIEWNAME =
    "URLSecurityAccessView";
    static String URL_ACCESS_ADD_YESNO = "xporaclewebfeedsdata.xsp";
    static String URL_ACCESS_UPDATE_YESNO = "xporaclewebfeedsdata.xsp";
    static String URL_ACCESS_DELETE_YESNO = "xporaclewebfeedsdata.xsp";
    private String message;
    private String MyEntries;
    private String MyURLEntries;
    private String MyURLAdd;
    private String MyURLUpdate;
    private String MyURLDelete;

    private static final long serialVersionUID = 1L;
    //private String Date_Format = "MM/dd/yy";
    // declare variables, and add static values as test
    //SimpleDateFormat simpledateformat = new SimpleDateFormat(Date_Format);
    //Calendar calendar = Calendar.getInstance();
    //...
    // declare Cookie variables
    public String UserNameValue;
    public String RoleNameValue;

    // Reformat the date to our needs, we need this format
    // (2010.12.30.10.13.AM)
    // @version: 2013.11.03.10.14.PM

    // Reformat the date to our needs, DocuWebNews TimeStamp needs this format
    // (2016-08-14-12-38 PM)
    // @version: 2016-08-14-12.38.00
    private String Doc_Date_Format = "yyyy-MM-dd-HH.mm.ss";

```

```

SimpleDateFormat docdateformat = new SimpleDateFormat (Doc_Date_Format);
Calendar docalendar = Calendar.getInstance();

// ...
// form variables
private String UserRole;
private String ItemNumber;
private String Priority;
private String IssuesReported;
private String DateCreated = docdateformat.format (docalendar.getTime());
//...
// 2011.09.20.11.20.AM
// Randomising to get a ticket number
// TO DO: add the date or the user's ID to the random number.
// Alternatively, one can reformat the date variable as such
// 2012.11.10.8.20 + session ID, or random Number
// arrive at a fancy ticket number; for this exercise the random will do
Random Rand = new Random();

private String UserName;

/**
 * @return the userName
 */
public String getUserName() {
    return UserName;
}

/**
 * @param userName
 *         the userName to set
 */
public void setUserName (String userName) {
    UserName = userName;
}

/**
 * @return the userNameValue
 */
public String getUserNameValue() {
    return UserNameValue;
}

/**
 * @param userNameValue
 *         the userNameValue to set
 */
public void setUserNameValue (String userNameValue) {
    UserNameValue = userNameValue;
}

/**
 * @return the roleNameValue
 */
public String getRoleNameValue() {
    return RoleNameValue;
}

/**

```

```

    * @param roleNameValue
    *         the roleNameValue to set
    */
    public void setRoleNameValue(String roleNameValue) {
        RoleNameValue = roleNameValue;
    }

    /**
     * @return the pageID
     */
    public String getPageID() {
        return PageID;
    }

    /**
     * @param pageID
     *         the pageID to set
     */
    public void setPageID(String pageID) {
        PageID = pageID;
    }

    /**
     * @return the message
     */
    public String getMessage() {
        return message;
    }

    /**
     * @param message
     *         the message to set
     */
    public void setMessage(String message) {
        this.message = message;
    }

    /**
     * @return the myEntries
     */
    public String getMyEntries() {
        return MyEntries;
    }

    /**
     * @param myEntries
     *         the myEntries to set
     */
    public void setMyEntries(String myEntries) {
        MyEntries = myEntries;
    }

    // 2011.09.20.12.23.PM
    // Grabbing User IPs
    // TO DO: Get User's real IP, this item gets the external IP
    HttpServletRequest httpServletRequest = (HttpServletRequest) FacesContext
        .getCurrentInstance().getExternalContext().getRequest();
    String UserIP = httpServletRequest.getRemoteAddr();

```

```

HttpServletRequest reqURL = (HttpServletRequest) FacesContext
    .getCurrentInstance().getExternalContext().getRequest();
String PageID = reqURL.getRequestURL().toString();

/**
 * @return the userIP
 */
public String getUserIP() {
    return UserIP;
}

/**
 * @param userIP
 *         the userIP to set
 */
public void setUserIP(String userIP) {
    UserIP = userIP;
}

/**
 * @return the userRole
 */
public String getUserRole() {
    return UserRole;
}

/**
 * @param userRole
 *         the userRole to set
 */
public void setUserRole(String userRole) {
    UserRole = userRole;
}

/**
 * @return the itemNumber
 */
public String getItemNumber() {
    return ItemNumber;
}

/**
 * @param itemNumber
 *         the itemNumber to set
 */
public void setItemNumber(String itemNumber) {
    ItemNumber = itemNumber;
}

/**
 * @return the priority
 */
public String getPriority() {
    return Priority;
}

/**
 * @param priority
 *         the priority to set

```

```

    */
    public void setPriority(String priority) {
        Priority = priority;
    }

    /**
     * @return the issuesReported
     */
    public String getIssuesReported() {
        return IssuesReported;
    }

    /**
     * @param issuesReported
     *         the issuesReported to set
     */
    public void setIssuesReported(String issuesReported) {
        IssuesReported = issuesReported;
    }

    /**
     * @return the dateCreated
     */
    public String getDateCreated() {
        return DateCreated;
    }

    /**
     * @param dateCreated
     *         the dateCreated to set
     */
    public void setDateCreated(String dateCreated) {
        DateCreated = dateCreated;
    }

    /**
     * @return the myURLEntries
     */
    public String getMyURLEntries() {
        return MyURLEntries;
    }

    /**
     * @param myURLEntries the myURLEntries to set
     */
    public void setMyURLEntries(String myURLEntries) {
        MyURLEntries = myURLEntries;
    }

    /**
     * @return the myURLAdd
     */
    public String getMyURLAdd() {
        return MyURLAdd;
    }

    /**
     * @param myURLAdd the myURLAdd to set
     */

```

```

public void setMyURLAdd(String myURLAdd) {
    MyURLAdd = myURLAdd;
}

/**
 * @return the myURLUpdate
 */
public String getMyURLUpdate() {
    return MyURLUpdate;
}

/**
 * @param myURLUpdate the myURLUpdate to set
 */
public void setMyURLUpdate(String myURLUpdate) {
    MyURLUpdate = myURLUpdate;
}

/**
 * @return the myURLDelete
 */
public String getMyURLDelete() {
    return MyURLDelete;
}

/**
 * @param myURLDelete the myURLDelete to set
 */
public void setMyURLDelete(String myURLDelete) {
    MyURLDelete = myURLDelete;
}

public SendOracleXMLFeedsJavaBean() {

    //...
    //TODO: Run query to grab Add, Update, Delete values and Show/Hide
    buttons controls at will
    //based on User access to those controls, see URLSecurityAccessView for
    more information
    //private String MyURLAdd;
    //private String MyURLUpdate;
    //private String MyURLDelete;
    //...
    // get userCookies
    FacesContext facesContext = FacesContext.getCurrentInstance();
    String cookieName = null;
    Cookie cookie[] = ((HttpServletRequest) facesContext
        .getExternalContext().getRequest()).getCookies();
    if (cookie != null && cookie.length > 0) {
        for (int i = 0; i < cookie.length; i++) {
            cookieName = cookie[i].getName();
            if (cookieName.equals("cookieKeyUser")) {
                UserNameValue = cookie[i].getValue();
                System.out
                    .println("SendOracleXMLFeedsJavaBean!
Found this UserNameValue Cookie..."
                        + UserNameValue);
            }
        }
    }
}

```

```

        if (cookieName.equals("cookieKeyRole")) {
            RoleNameValue = cookie[i].getValue();
            System.out
                .println("SendOracleXMLFeedsJavaBean!
Found this RoleNameValue Cookie..."
                + RoleNameValue);
        } else
            System.out.println("...");

        // TODO: Add this method to JSFUtil class,
        // also delete the cookies when logging out
    }
}

// button code
public String submitEntry() {
    try {
        // get the current database being used
        Database database = (Database) FacesContext.getCurrentInstance()
            .getApplication().getVariableResolver().resolveVariab
le(
            FacesContext.getCurrentInstance(),
            "database");

        // Find the First view in question
        View IssueView =
database.getView(ISSUE_ACCESS_VIEWNAME);
        // Find the Second view in question
        View URLView =
database.getView(URL_ACCESS_VIEWNAME);
        System.out
            .println("SendOracleXMLFeedsJavaBean.java |
submitEntry Connected to "
            + IssueView);

        System.out
            .println("SendOracleXMLFeedsJavaBean.java | submitEntry Connected
to "
            + URLView);

        // determine database state for searching
        if (database.isFTIndexed())
            database.updateFTIndex(false);
        else
            database.updateFTIndex(true);

        //...
        //Search Issues by UserName from Issues View
String SearchIssueText =

```



```
UserName.toString();
```

```
//...  
//Search URL data by UserName from URLSecurityAccess view View
```

```
String SearchURLText =
```

```
UserName.toString();
```

```
// perform search via ViewEntryCollection
```

```
ViewEntryCollection IssueVec =
```

```
IssueView.getAllEntriesByKey(SearchIssueText.trim(), true);
```

```
ViewEntryCollection URLVec =
```

```
URLView.getAllEntriesByKey(SearchURLText.trim(), false);
```

```
// Note, count can be 0 here, thus Message(s) should load...
```

```
System.out.println("SendOracleXMLFeedsJavaBean.java |
```

```
ViewEntryCollection Counts Obtained...");
```

```
//System.out.println("SendOracleXMLFeedsJavaBean.java |
```

```
ViewEntries for Issues View"
```

```
// + IssueVec.getCount() + " Account(s)");
```

```
System.out.println("SendOracleXMLFeedsJavaBean.java | ViewEntries
```

```
for URL Access View"
```

```
+ URLVec.getCount() + " Account(s)");
```

```
System.out
```

```
.println("SendOracleXMLFeedsJavaBean.java |
```

```
ViewEntryCollection completed successfully...");
```

```
ViewEntry tmpURLEntry = null;
```

```
ViewEntry tmpIssueEntry = null;
```

```
ViewEntry issueEntry = IssueVec.getFirstEntry();
```

```
ViewEntry URLEntry = URLVec.getFirstEntry();
```

```
// ensure UserName field/column returned is not null (record not  
found) from NSF back-end,
```

```
// if so, load text/err Message(s) to user
```

```
if (issueEntry == null || URLEntry
```

```
== null) {
```

```
// TODO: Do something else with the current component
```

```
// example, combine with built-in items
```

```
// ...
```

```
MyEntries = "Error: User Data Invalid";
```

```
// load to custom errMessage label control
```

```
message = MyEntries.toString();
```

```
} else {
```

```
// if record is found, do not permit user to submit, this  
time load a 'record exists' message
```

```
// add value needed for pairing with session, and so on...
```

```
MyEntries =
```

```
URLEntry.getDocument().getItemValueString("UserName");
```

```
MyURLEntries =
```

```
issueEntry.getDocument().getItemValueString("userID");
```

```
if
```

```
MyEntries.equalsIgnoreCase(MyURLEntri
```

```

es)) {
    // ...
    MyEntries = "Error: UserName Already Entered";
    // load to custom errMsgage label control
    message = MyEntries.toString();
}

// TODO: load friendlier debug message to log
// in which case, inputText or Label should
// load 'Existing Record/UserName message to form
// ...
// otherwise check to see if it equals the value in Cookie, to
again
// Post a message and NOT continue

else {
    //do submit a value that isn't already in the NSF back-end
    (in NewIssuesListings view)
    if (!
MyEntries.equalsIgnoreCase(MyURLEntri
es)) {
    //...
    //Run the appropriate method if above is true...
    submitXMLRecord();
    //...
    //Get Next Values to compare up against, as record
being submitted
    tmpIssueEntry = IssueVec.getNextEntry();
    //...
    //check against Security view, as record being
submitted
    tmpURLEntry = URLVec.getNextEntry();
    //...
    //be Green, clean it up...
    issueEntry.recycle();
    URLEntry.recycle();
    //...
    issueEntry = tmpIssueEntry;
    URLEntry = tmpURLEntry;
    // throwable initialised if there is an error, either
in the field or the query

    //...
    //load the list page here, if successful
    return "xsp-success";
    // return "/xpredir.xsp";
} else {
    //...
    //keep user on same page so that error message can be

```

shown

```
        return "xsp-failure";
    }
}

}

}

} catch (NotesException e) {
    System.out.println(e.id + " " + e.text);
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
}
return UserName;
}

//...
//See above method, runs if true (meaning that record(s) not found in
NewIssuesListings view

public void submitXMLRecord () {

    try {
        // get the current database being used
        Database database = (Database) FacesContext.getCurrentInstance()
            .getApplication().getVariableResolver().resolveVariable(
                FacesContext.getCurrentInstance(), "database");
        // instantiate the document create call
        Document submitDocument = (Document) database
            .createDocument();
        // submit using Company form (Alternatively, one can add the Alias of
that form)

        submitDocument.appendItemValue("form", "Company");
        // using appendItemValue to insert in the fields needed
        // ...
        submitDocument.appendItemValue("UserName", UserName);
        submitDocument.appendItemValue("UserRole", UserRole);
        submitDocument.appendItemValue("itemNumber", ItemNumber);
        submitDocument.appendItemValue("Priority", Priority);
        submitDocument.appendItemValue("issues", IssuesReported);
        submitDocument.appendItemValue("DateCreated", DateCreated);
        submitDocument.appendItemValue("userIP", UserIP);
        // no need to show this on the Xpage
        submitDocument.appendItemValue("pageID", PageID);

        // saves the data, based on above fields to form for XML file
        submitDocument.save();

        System.out.println("inserting rows to For XML Feeds...");

        // cleans up the system
        submitDocument.recycle();
    }
}
```

```

    } catch (NotesException e) {
        //print this error to the server/log
        e.printStackTrace();
    }

}

}

```

Xpages List file, after data Submission: Nothing fancy here, the current is made available via IBM Notes Domino Designer 'Drag and Drop' feature... We will skip highlighting areas of interest at this point.

xporaclefeedssubmittedview.xsp

```

<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">

    <xp:this.data>
        <xp:dominoView var="view1"
viewName="NewIssuesListingsAlias"></xp:dominoView>
    </xp:this.data>
    <xp:span style="font-weight:bold;font-size:16pt">Oracle</xp:span>
    <xp:span style="font-weight:bold;font-size:16pt"> Feeds</xp:span>
    <xp:span style="font-weight:bold;font-size:16pt"> For</xp:span>
    <xp:span style="font-weight:bold;font-size:16pt"> XML</xp:span>
    <xp:span style="font-weight:bold;font-size:16pt"> Data</xp:span>
    <xp:br style="font-weight:bold;font-size:16pt"></xp:br>
    <xp:br></xp:br>
    <xp:viewPanel value="#{view1}" id="viewPanel1">
        <xp:this.facets>
            <xp:pager partialRefresh="true" layout="Previous Group Next"
                xp:key="headerPager" id="pager1">
            </xp:pager>
        </xp:this.facets>
        <xp:viewColumn columnName="UserName" id="viewColumn1">
            <xp:this.facets>
                <xp:viewColumnHeader value="UserName" xp:key="header"
                    id="viewColumnHeader1" style="font-weight:bold;font-
size:12pt">
                </xp:viewColumnHeader>
            </xp:this.facets>
        </xp:viewColumn>
        <xp:viewColumn columnName="PageID" id="viewColumn2">
            <xp:this.facets>
                <xp:viewColumnHeader value="PageID" xp:key="header"
                    id="viewColumnHeader2" style="font-weight:bold;font-
size:12pt">
                </xp:viewColumnHeader>
            </xp:this.facets>
        </xp:viewColumn>
    </xp:viewPanel>

```

```

<xp:viewColumn columnName="priority" id="viewColumn3">
  <xp:this.facets>
    <xp:viewColumnHeader value="Priority" xp:key="header"
      id="viewColumnHeader3" style="font-weight:bold;font-
size:12pt">
      </xp:viewColumnHeader>
    </xp:this.facets>
  </xp:viewColumn>
<xp:viewColumn columnName="issues" id="viewColumn4">
  <xp:this.facets>
    <xp:viewColumnHeader value="IssuesReported"
      xp:key="header" id="viewColumnHeader4" style="font-
weight:bold;font-size:12pt">
      </xp:viewColumnHeader>
    </xp:this.facets>
  </xp:viewColumn>
<xp:viewColumn columnName="DateCreated" id="viewColumn5">
  <xp:this.facets>
    <xp:viewColumnHeader value="DateCreated" xp:key="header"
      id="viewColumnHeader5" style="font-weight:bold;font-
size:12pt">
      </xp:viewColumnHeader>
    </xp:this.facets>
  </xp:viewColumn>
<xp:viewColumn columnName="UserIP" id="viewColumn6">
  <xp:this.facets>
    <xp:viewColumnHeader value="UserIP" xp:key="header"
      id="viewColumnHeader6" style="font-weight:bold;font-
size:12pt">
      </xp:viewColumnHeader>
    </xp:this.facets>
  </xp:viewColumn>
<xp:viewColumn columnName="UserRole" id="viewColumn7">
  <xp:this.facets>
    <xp:viewColumnHeader value="UserRole" xp:key="header"
      id="viewColumnHeader7" style="font-weight:bold;font-
size:12pt">
      </xp:viewColumnHeader>
    </xp:this.facets>
  </xp:viewColumn>
</xp:viewPanel>
</xp:view>

```

Xpages List Design # 2

Oracle Feeds For XML Data

Previous 1|2|3|4|5 Next

UserName	PageID	Priority	IssuesReported	DateCreated	UserIP	UserRole
UserName	PageID	priority	issues	DateCreated	UserIP	UserRole

JavaAgent performs XML creation magic: Read the TODO portion of the code, some helpful information there... the process stays dormant until JDeveloper JSF App picks up the XML file via URL and attempt to grab data out of it, in turn, submitting to DB2 database. Have a good look, areas of interest are also highlighted here, for your convenience. Code basically just reads view data and creates the XML tags, the .xml extension is added through the JavaAgent's properties, will add a quick screenshot of that for a closer look.

WebNewsForHTTPUserXMLDataJavaAgent | [javaagentdataforwebnewshttp.xml](#)

```
/**
 * Created: 2011.04.15.11.54.AM
 * WebNewsForHTTPUserXMLDataJavaAgent | javaagentdataforwebnewshttp.xml
 * XML data for Xpage to be read by JavaBean
 */

//load imports
import lotus.domino.*;
import java.io.PrintWriter;

/**
 * @author Dököll Solutions, Inc.
 * @version 2011.04.15.11.54.AM
 *
 */
```

```

// begin class
public class JavaAgent extends AgentBase {

    // open method, this actually runs the whole program
    public void NotesMain() {

        // let's add a try catch here, to grab errors near the end
        try {
            // open our session...
            Session session = getSession();
            // load info console for debugging purposes
            System.out.println("WebNewsForHTTPUserXMLDataJavaAgent
session..."
                + session);

            // load agentContext
            AgentContext agentContext = session.getAgentContext();
            // load info console for debugging purposes
            System.out
                .println("WebNewsForHTTPUserXMLDataJavaAgent
agentContext..."
                    + agentContext);

            // get the server name from the current database
            Database cdb = agentContext.getCurrentDatabase();
            // load info console for debugging purposes
            System.out.println("database connection to..." + cdb);
            // grab current serverName
            String ServerName = cdb.getServer();
            // load info console for debugging purposes
            System.out.println("database ServerName..." + ServerName);
            // Access the Database this agent is running in

            Database db = session.getDatabase(ServerName, "docucontent.nsf");
            // db.open();
            // load info console for debugging purposes
            System.out.println("Connection to database..." + db
                + " established");

            // ...
            // Option # 1
            // TODO: Query 'NewIssuesListings' and URLSecurityAccessView and
IBM DB2 prior to creating XML and compare records
            // to fend off existing ones that may have been already
thrown to XML, grabbed by external JDeveloper App
            // Note: doing so will help avoid a 'getNextException' error for
records already submitted to DB2
            // For this exercise, we are going to Query DB2 from
JDeveloper to ensure records being received
            // from Notes Domino are not in the DB2 database
            // Should probably do this both in this Agent and
BackingBean on JDeveloper eventually (perhaps next version)
            // ...
            // Option # 2
            // ALTERNATIVE: Delete records after from NewIssuesListings view
prior to creating XML in URL; however, if you want to keep
            // a record of what has been submitted as a Log on
entries, use the first option
            // 2016.08.22.3.52.PM

```

```

// ...

View pview =
db.getView("NewIssuesListings");
// load info console for debugging purposes
System.out.println("We've got a view..." + pview);
// Create Variables to hold the Documents and get the first
document
Document doc;
Document ndoc;
doc = pview.getFirstDocument();
// Create a Java PrintWriter object variable to save coding time
and
// size
PrintWriter pw =
getAgentOutput();
System.out
.println("XML to URL:
http://localhost/docucontent.nsf/ commenced...");
System.out
.println("XML Named:
/javaagentdataforwebnewshttp.xml...");
// Set the content type
pw.println("Content-type:text/xml");
pw.println("");
// Write out the XML Opening Tags
pw.println("<?xml version='1.0' ?>");
pw.println("<sitesissues>");
while (doc != null) { // Write the value in the
UserName field to
// the output stream
pw.println("<person>");

pw.println("<username>");
pw.println(doc.getItemValueString("UserName"));
pw.println("</username>");

pw.println("<pageid>");
pw.println(doc.getItemValueString("PageID"));
pw.println("</pageid>");

pw.println("<priority>");
pw.println(doc.getItemValueString("Priority"));
pw.println("</priority>");

pw.println("<usernotes>");
pw.println(doc.getItemValueString("issues"));
pw.println("</usernotes>");

pw.println("<timestamp>");
pw.println(doc.getItemValueString("DateCreated"));
pw.println("</timestamp>");

pw.println("</person>");

```

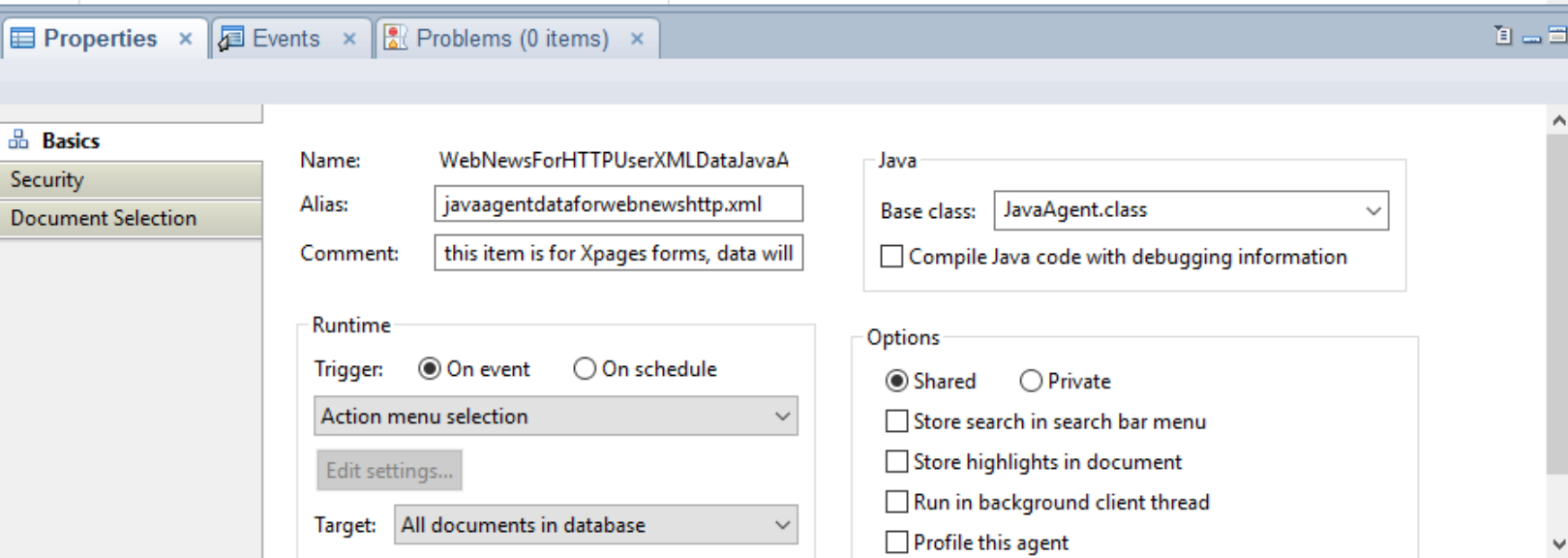
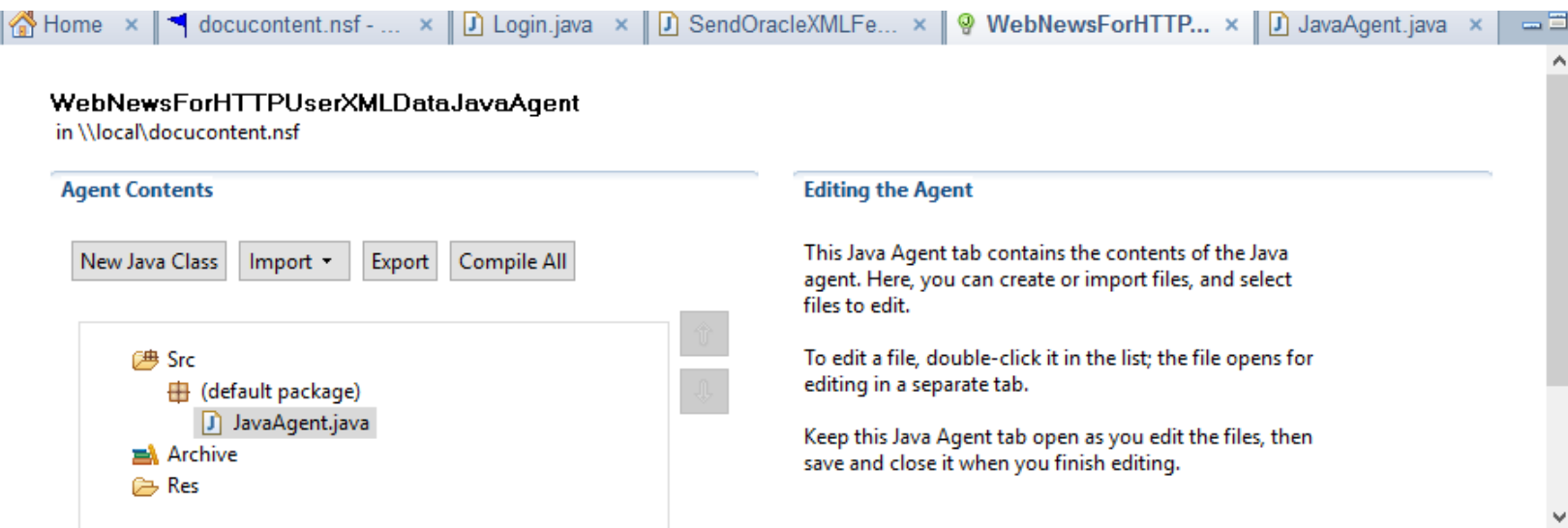


```
        // Get the next document in the view
        ndoc =
pview.getNextDocument (doc) ;
        // recycle the doc object to preserve memory
        doc.recycle();
        //...
        doc = ndoc;
    }

    pw.println("</sitesissues>");
    System.out
        .println("XML Data Created with
WebNewsForHTTPUserXMLDataJavaAgent");

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

JavaAgent Properties



Conclusion:

You are now able to run an Xpages App using JavaBean and JavaAgent to create an XML file which can be pulled via URL by an external JDeveloper App, submitting records from that XML to DB2 database. Please see Part 2 of this tutorial for Oracle JDeveloper App samples

For all Questions and comments, please add a Quick note to our Contact form, or visit our social media networks

Contact

<http://www.dokollolutionsinc.com/apptrendscontactemail.php>

Facebook

<https://www.facebook.com/Dököll-Solutions-Inc-233555900032117/>

Google+

<https://plus.google.com/u/0/+DököllSolutions/posts>

Twitter

<https://twitter.com/DokollSolutions>

YouTube

<https://www.youtube.com/channel/UCSimDTpK0oe7OrPsYOE4nww>

Dököll Solutions, Inc.

version: 2016.08.22.8.53.PM