

# Döcu Content

## IBM DB2, Lotus Notes Domino

### Notes Domino XML into DB2 Database

#### Part 3.2

**Foreword:** Steps included below to create Apps via IBM Notes Domino Designer, submit data to NSF and DB2 back-end are current, however we suggest visiting 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... If you need additional support, be sure to contact us or visit our website: [www.dokollolutionsinc.com](http://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:**

Use existing sample Xpages App Döcu Content (docucontent.nsf) via IBM Lotus Notes Domino Designer, run JavaAgent to collect Log.nsf data (external NSF back-end database), parse data into columns and submit to Domino View in back-end- We can then read the View information, create an XML file post to URL to be submitted to DB2 directly from Notes Domino Designer environment, via JavaBean. You will be using Xpages form, with a button to help you along (code and design elements include: JavaBean, JavaAgent, and Xpages), areas of interest are 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 environment; we are going to skip formal steps. We urge you to consult our documentation for support. Let's get you started off right with samples in this tutorial...

**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- However, if you already have that version (9.x), rest assured you can still use it, very powerful database application...

**Hint # 2:** If you downloaded IBM Data Studio 4.x or and earlier version separately for this tutorial, and it is a treat! but, you should be prepared to muscle through it. 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 though, your Java code may be faulty and your App is stuck, you can kill this bad boy via Task Manager, or reboot your system. You may also need to Clean your project in DDE.

**What to Expect:**

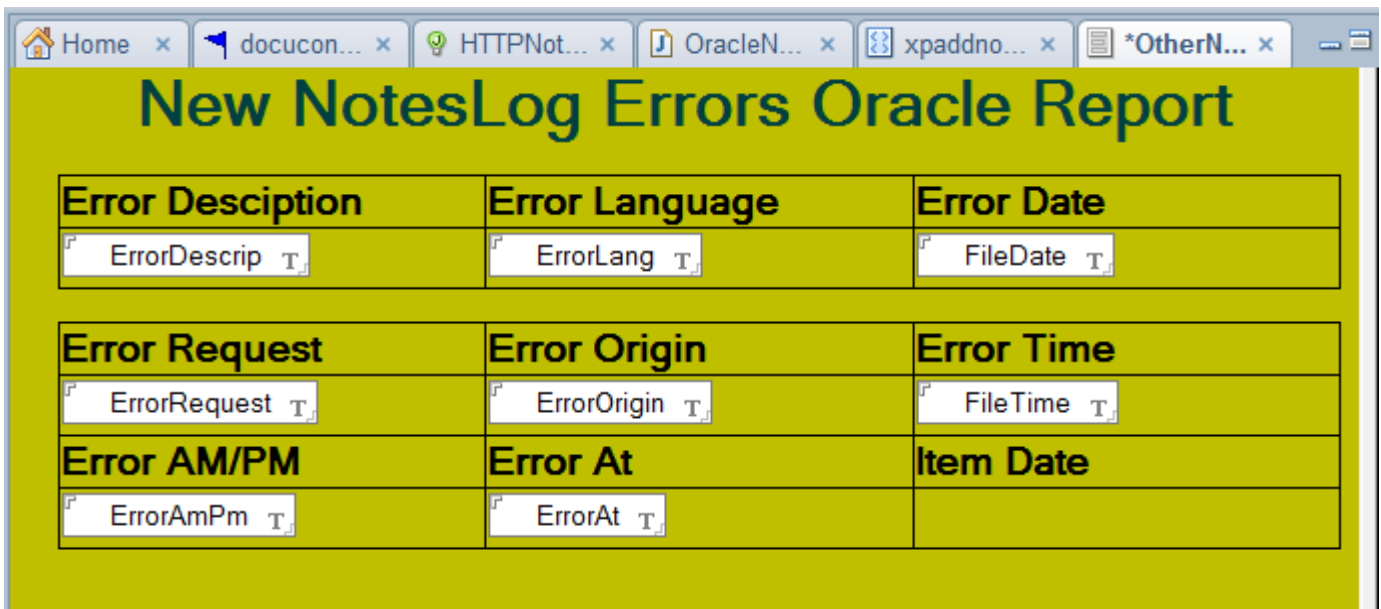
The current consists of Xpages forms, JavaAgent, JavaBean classes, with XML files in URL; wanted to add this in because the XML file creation is its own entity in the whole process. Now, since you've done this before, there's a likelihood that you created Forms and Views in Notes Domino back-end and are ready to modify and run Java code for submission to IBM NSF and DB2 environments.

**Before you Begin:** You should have gone through Tutorial Version 3.1 prior to seeing the following... our efforts continue here while we submit data obtained through Döcu Content back-end View to DB2...

**Form :**

**OtherNewNotesLogErrorsReportForOracle**

works with View being posted after to supply JavaAgent data as XML into DB2 database



The screenshot shows a web browser window with several tabs. The active tab is titled '\*OtherN...'. The main content area has a yellow background and a title 'New NotesLog Errors Oracle Report' in large green font. Below the title are three tables of input fields, each with a header row and a data row. The first table has columns 'Error Description', 'Error Language', and 'Error Date'. The second table has columns 'Error Request', 'Error Origin', and 'Error Time'. The third table has columns 'Error AM/PM', 'Error At', and 'Item Date'. Each input field has a small 'T' icon next to it, indicating a text field.

Error Description	Error Language	Error Date
<input type="text" value="ErrorDescrip T"/>	<input type="text" value="ErrorLang T"/>	<input type="text" value="FileDate T"/>

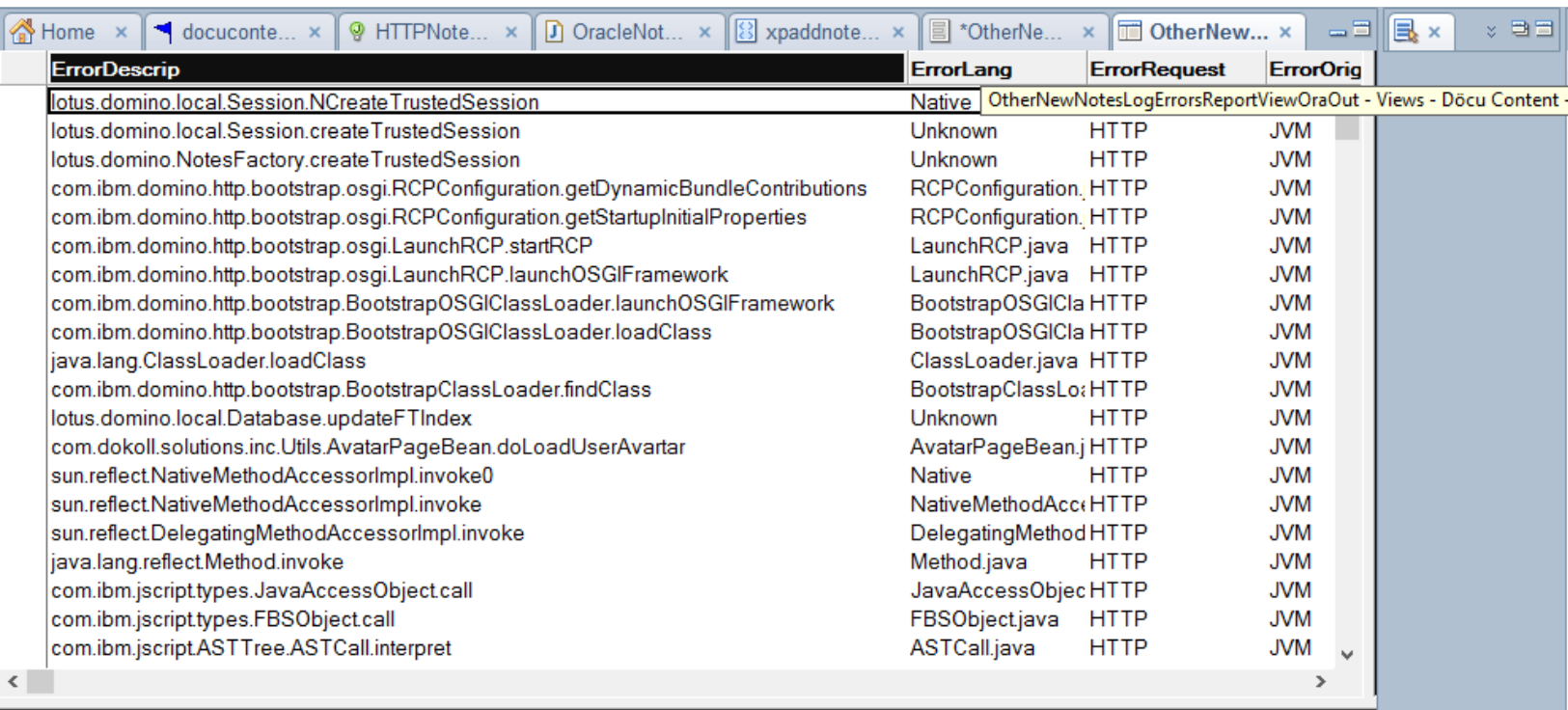
Error Request	Error Origin	Error Time
<input type="text" value="ErrorRequest T"/>	<input type="text" value="ErrorOrigin T"/>	<input type="text" value="FileTime T"/>

Error AM/PM	Error At	Item Date
<input type="text" value="ErrorAmPm T"/>	<input type="text" value="ErrorAt T"/>	

View:

## OtherNewNotesLogErrorsReportViewOraOut

records from this view obtained by JavaAgent will provide a URL with XML for data extract to DB2, see properties section of the JavaAgent for additional configuration info...

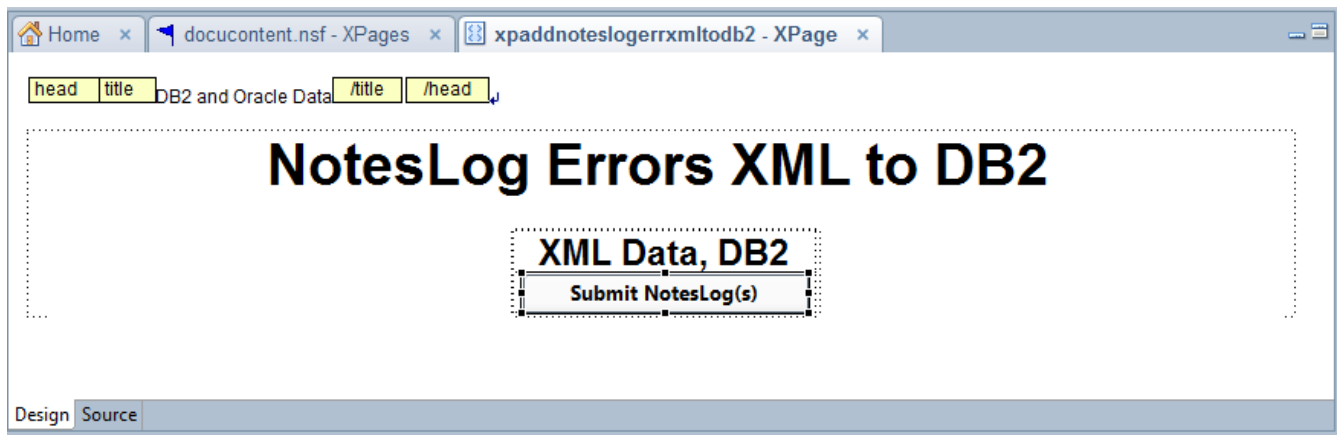


ErrorDescrip	ErrorLang	ErrorRequest	ErrorOrig
lotus.domino.local.Session.NCreateTrustedSession	Native	OtherNewNotesLogErrorsReportViewOraOut - Views - Docu Content -	
lotus.domino.local.Session.createTrustedSession	Unknown	HTTP	JVM
lotus.domino.NotesFactory.createTrustedSession	Unknown	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getDynamicBundleContributions	RCPConfiguration,	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getStartupInitialProperties	RCPConfiguration,	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.LaunchRCP.startRCP	LaunchRCP.java	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.LaunchRCP.launchOSGiFramework	LaunchRCP.java	HTTP	JVM
com.ibm.domino.http.bootstrap.BootstrapOSGiClassLoader.launchOSGiFramework	BootstrapOSGiCla	HTTP	JVM
com.ibm.domino.http.bootstrap.BootstrapOSGiClassLoader.loadClass	BootstrapOSGiCla	HTTP	JVM
java.lang.ClassLoader.loadClass	ClassLoader.java	HTTP	JVM
com.ibm.domino.http.bootstrap.BootstrapClassLoader.findClass	BootstrapClassLo:	HTTP	JVM
lotus.domino.local.Database.updateFTIndex	Unknown	HTTP	JVM
com.dokoll.solutions.inc.Utils.AvatarPageBean.doLoadUserAvatar	AvatarPageBean.j	HTTP	JVM
sun.reflect.NativeMethodAccessorImpl.invoke0	Native	HTTP	JVM
sun.reflect.NativeMethodAccessorImpl.invoke	NativeMethodAcc:	HTTP	JVM
sun.reflect.DelegatingMethodAccessorImpl.invoke	DelegatingMethod	HTTP	JVM
java.lang.reflect.Method.invoke	Method.java	HTTP	JVM
com.ibm.jscrip.types.JavaAccessObject.call	JavaAccessObjec	HTTP	JVM
com.ibm.jscrip.types.FBSObject.call	FBSObject.java	HTTP	JVM
com.ibm.jscrip.ASTTree.ASTCall.interpret	ASTCall.java	HTTP	JVM

Here is Xpages Code, again with a button: What we've done here is we created and Xpage ([xpaddnoteslogerrxmltodb2.xsp](#)) and added a button to run a JavaBean code that pull in XML file through URL then submit the records to DB2 database.

## xpaddnoteslogerrxmltodb2.xsp

## Xpages Submit Design # 2



```

<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">
  <xp:this.resources>
    <xp:script src="/docucontentGACodeTest.js" clientSide="true" />
  </xp:this.resources>
  <html>
    <head>
      <title>DB2 and Oracle Data</title>
    </head>
    <xp:br></xp:br>
    <xp:div style="text-align:center">
      <xp:span style="font-weight:bold;font-size:28pt">
        NotesLog Errors XML to DB2
      </xp:span>
      <body style="font-weight:bold;font-size:10pt;text-
align:center"><xp:form id="formId" style="font-weight:bold;font-size:11pt;text-
align:center">

```

```

      <table style="font-weight:bold;font-size:10pt;text-
align:center">

```

```

      <tr style="font-weight:bold;font-
size:11pt;text-align:center">

```

```

      <td style="font-weight:bold;font-
size:12pt;text-align:center">

```

```

      <xp:div style="text-
align:center;font-size:13pt">

```

```

      <xp:span style="font-
weight:bold;font-size:17pt">

```

```

      </xp:span>
    </xp:div>
    <xp:div style="text-
align:center;font-size:13pt">

```

```

      <xp:button value="Submit
NotesLog(s)" id="button4" style="font-weight:bold;font-size:10pt">

```

```

      <xp:eventHandler

```

```
event="onclick" submit="true" refreshMode="complete" immediate="false" save="true"
id="eventHandler4">
```

```
<xp:this.action><!
```

```
[CDATA[#{javascript:WebNewsNotesLogErrorJavaBean.doSu
```

```
bmitNotesLogErr ( ) }]]></xp:this.action>
```

```
</xp:eventHandler>
```

```
</xp:button>
```

```
</xp:div>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
</xp:form>
```

```
</body></xp:div></html>
```

```
</xp:view>
```

**JavaBean code submitting Notes Documents to DB2:** Copy and paste below JavaBean in your environment to work with the Xpages file submitted earlier and load records into the back-end database. The process is simple, when the Xpages submit button is pushed, View search begins promptly with the JavaBean calling the JavaAgent to run collect Data from the back-end View and gives it to the JavaBean to perform the process of submitting to DB2, areas of interest, here as well, are highlighted for your convenience...

## WebNewsNotesLogErrorJavaBean.java

```
/**
 * Create from Copy: 2016.09.10.1.38.PM
 * Notes Log Errors to Feed Oracle DocuWebNews App via DB2
 */
package com.dokoll.solutions.inc.oracle.dev;

import java.sql.Connection;
import java.sql.PreparedStatement;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;

import org.w3c.dom.Document;
```

```

import org.w3c.dom.Element;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;

import com.dokoll.solutions.inc.Utills.DB2Connector;

/**
 * @author Dokoll Solutions, Inc.
 * @version: 2016.09.10.1.38.AM
 *
 */

public class WebNewsNotesLogErrorJavaBean {

    public void doSubmitNotesLogErr () {

        //TO DO: Send ip and other info to back-end if page firing
        //this button code

        String columnName = "";
        String dataOutput = "";

        // entering try catch
        try {
            // set up document read/build
            DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
            DocumentBuilder db = dbf.newDocumentBuilder();
            Document doc =
db.parse("http://localhost/docucontent.nsf/javaagentoranotesfe
edsdatahttp.xml");
            //Document doc = db.parse("http://localhost:7101/DocuWebNews-
ViewController-context-root/faces/javaagentnoteslogwebnewshttp.xml");
            // normalize text representation
            doc.getDocumentElement().normalize();
            System.out.println("Root element of the doc is " +
doc.getDocumentElement().getNodeName());
            // ...
            // Get count of values per tags
            NodeList listOfIssues = doc.getElementsByTagName("errorreading");
            // perform count of documents retrieved
            int totalIssues = listOfIssues.getLength();
            System.out.println("Total number of Errors : " + totalIssues);
            // loop through issues and fill nodes
            for (int s = 0; s < listOfIssues.getLength(); s++) {
                // ...
                // Set up issuesNode to gather data into elements
                Node logErrNode = listOfIssues.item(s);
                if (logErrNode.getNodeType() == Node.ELEMENT_NODE) {
                    // ...
                    Element logErrElement = (Element) logErrNode;
                    // ...
                    // load to respective elements
                    NodeList logErrDescriptList =

```

```

logErrElement.getElementsByTagName("errordescr");
    Element logErrDescriptElement = (Element)
logErrDescriptList.item(0);
    // ...
    // feed values outbound via each element
    NodeList textLogErrDescriptList =
logErrDescriptElement.getChildNodes();
    System.out.println("errordescr : " +
(textLogErrDescriptList.item(0)).getNodeValue().trim());
    // ...
    // load to respective elements
    NodeList logErrLangList =
logErrElement.getElementsByTagName("errorlang");
    Element logErrLangElement = (Element) logErrLangList.item(0);

    // ...
    // feed values outbound via each element
    NodeList textLogErrLangList =
logErrLangElement.getChildNodes();
    System.out.println("errorlang : " +
(textLogErrLangList.item(0)).getNodeValue().trim());

    // ...
    // load to respective elements
    NodeList logErrFileDateList =
logErrElement.getElementsByTagName("filedate");
    Element logErrFileDateElement = (Element)
logErrFileDateList.item(0);

    // ...
    // feed values outbound via each element
    NodeList textLogErrFileDateList =
logErrFileDateElement.getChildNodes();
    System.out.println("filedate : " +
(textLogErrFileDateList.item(0)).getNodeValue().trim());

    // ...
    // load to respective elements
    NodeList logErrFileTimeList =
logErrElement.getElementsByTagName("filetime");
    Element logErrFileTimeElement = (Element)
logErrFileTimeList.item(0);

    // ...
    // feed values outbound via each element
    NodeList textLogErrFileTimeList =
logErrFileTimeElement.getChildNodes();
    System.out.println("filetime : " +
(textLogErrFileTimeList.item(0)).getNodeValue().trim());

    // ...
    // load to respective elements
    NodeList logErrAmPmList =
logErrElement.getElementsByTagName("errorampm");
    Element logErrAmPmElement = (Element) logErrAmPmList.item(0);

```

```

        // ...
        // feed values outbound via each element
        NodeList textLogErrAmPmList =
logErrAmPmElement.getChildNodes();
        System.out.println("errorampm : " +
(textLogErrAmPmList.item(0)).getNodeValue().trim());

        // load results to ColumnNames bound for CSV file
        String LogErrDescrip =
((textLogErrDescriptList.item(0)).getNodeValue().trim());

        String LogErrLang =
((textLogErrLangList.item(0)).getNodeValue().trim());

        // load results to ColumnNames bound for CSV file
        String LogErrDate =
((textLogErrFileDateList.item(0)).getNodeValue().trim());

        // load results to ColumnNames bound for CSV file
        String LogErrTime =
((textLogErrFileTimeList.item(0)).getNodeValue().trim());

        // load results to ColumnNames bound for CSV file
        String LogErrAmPm =
((textLogErrAmPmList.item(0)).getNodeValue().trim());
        System.out.println("Notes Domino Message: Debugging purposes:
");

        // pre-arrange column values for CSV file
        columnName =
            "LogErrDescrip" + "," + "LogErrLang" + "," + "LogErrDate" +
            "," + "LogErrTime" + "," + "LogErrAmPm" +
            "\n"; // ...
        // load values to columns set up
        dataOutput =
            dataOutput + LogErrDescrip + "," + LogErrLang + "," +
LogErrDate + "," + LogErrTime +
            "," + LogErrAmPm + "\n";

        // add debug messages
        // TODO: Remove in Prod
        System.out.println(columnName + dataOutput);

        //setup variable(s) to grab values from XML, add to DB2
        String strFileDescript =
LogErrDescrip.toString().trim();
        String strFileLang =
LogErrLang.toString().trim();
        String strFileDate =
LogErrDate.toString().trim();
        String strFileTime =
LogErrTime.toString().trim();

```



```

        String strFileAmPm =
LogErrAmPm.toString().trim();

        //TODO: Prepare a Statement here to Search DB2 prior to
Submitting records to the DB
        // if this is not done, you will see a getNextException
        //establish a connection
        Connection connection = DB2Connector.getConnection();

        System.out.println("Entering Insert query...");
        //prepare INSERT
        PreparedStatement prep =
        connection.prepareStatement("insert into
DB2ADMIN.ORA_LOGFEEDS_DB (ErrorDescript, ErrorLang,
ErrorDate, ErrorTime, ErrorAmPm) values (?,?,?,?,?)");

        //Added Data for report
        //2016.08.30.1.26.AM
                prep.setString(1, strFileDescript);
                prep.setString(2, strFileLang);
                prep.setString(3, strFileDate);
                prep.setString(4, strFileTime);
                prep.setString(5, strFileAmPm);

        prep.addBatch();

        System.out.println("Values added...");
        connection.setAutoCommit(false);
        prep.executeBatch();

        connection.setAutoCommit(true);
        //...
        connection.close();
    }

}

} catch (SAXParseException err) {
    System.out.println("*** Parsing error" + ", line " + err.getLineNumber()
+ ", uri " + err.getSystemId());
    System.out.println(" " + err.getMessage());

} catch (SAXException e) {
    Exception x = e.getException();
    ((x == null) ? e : x).printStackTrace();

} catch (Throwable t) {
    t.printStackTrace();
}

} // end of program

}

```

**DB2 Database:** ORA\_LOGFEEDS\_DB (Screenshot # 2): IBM DB2 database values obtained via XML from Lotus Notes Domino Designer App URL... Info that can be used to assist Oracle JDeveloper Apps to collect XML data into ADF/Faces App, submit to DB2 back-end...

## ORA\_LOGFEEDS\_DB

RECID	ERRORDESCRPT	ERRORLANG	ERRORDATE	ERRORTIME	ERRORAMP
316	lotus.domino.local.Session.NCreateTrustedSession	Native	09/10/2016	09:25:21	AM
317	lotus.domino.local.Session.createTrustedSession	Unknown	09/10/2016	09:25:21	AM
318	lotus.domino.NotesFactory.createTrustedSession	Unknown	09/10/2016	09:25:21	AM
319	com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getDynamicBundleContributions	RCPConfiguration.java	09/10/2016	09:25:21	AM
320	com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getStartupInitialProperties	RCPConfiguration.java	09/10/2016	09:25:21	AM
321	com.ibm.domino.http.bootstrap.osgi.LaunchRCP.startRCP	LaunchRCP.java	09/10/2016	09:25:21	AM
322	com.ibm.domino.http.bootstrap.osgi.LaunchRCP.launchOSGIFramework	LaunchRCP.java	09/10/2016	09:25:21	AM
323	com.ibm.domino.http.bootstrap.BootstrapOSGIClassLoader.launchOSGIFramework	BootstrapOSGIClassLoader.java	09/10/2016	09:25:21	AM
324	com.ibm.domino.http.bootstrap.BootstrapOSGIClassLoader.loadClass	BootstrapOSGIClassLoader.java	09/10/2016	09:25:21	AM
325	java.lang.ClassLoader.loadClass	ClassLoader.java	09/10/2016	09:25:21	AM
326	com.ibm.domino.http.bootstrap.BootstrapClassLoader.findClass	BootstrapClassLoader.java	09/10/2016	09:25:21	AM
327	lotus.domino.local.Database.updateFTIndex	Unknown	09/10/2016	07:05:48	PM
328	com.dokol.solutions.inc.Utilis.AvatarPageBean.doLoadUserAvatar	AvatarPageBean.java	09/10/2016	07:05:48	PM
329	sun.reflect.NativeMethodAccessorImpl.invoke0	Native	09/10/2016	07:05:48	PM
330	sun.reflect.NativeMethodAccessorImpl.invoke	NativeMethodAccessorImpl.java	09/10/2016	07:05:48	PM
331	sun.reflect.DelegatingMethodAccessorImpl.invoke	DelegatingMethodAccessorImpl.java	09/10/2016	07:05:48	PM
332	java.lang.reflect.Method.invoke	Method.java	09/10/2016	07:05:48	PM
333	com.ibm.jscript.types.JavaAccessObject.call	JavaAccessObject.java	09/10/2016	07:05:48	PM
334	com.ibm.jscript.types.FBSObject.call	FBSObject.java	09/10/2016	07:05:48	PM
335	com.ibm.jscript.ASTTree.ASTCall.interpret	ASTCall.java	09/10/2016	07:05:48	PM
336	com.ibm.jscript.ASTTree.ASTProgram.interpret	ASTProgram.java	09/10/2016	07:05:48	PM
337	com.ibm.jscript.ASTTree.ASTProgram.interpretEx	ASTProgram.java	09/10/2016	07:05:48	PM
338	com.ibm.jscript.JSEExpression._interpretExpression	JSEExpression.java	09/10/2016	07:05:48	PM
339	com.ibm.jscript.JSEExpression.access\$1	JSEExpression.java	09/10/2016	07:05:48	PM
340	com.ibm.jscript.JSEExpression\$2.run	JSEExpression.java	09/10/2016	07:05:48	PM
341	java.security.AccessController.doPrivileged	AccessController.java	09/10/2016	07:05:48	PM
342	com.ibm.jscript.JSEExpression.interpretExpression	JSEExpression.java	09/10/2016	07:05:48	PM

## Bonus info

XSP code added to show records submitted to View in browser

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: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 #3

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

**See Next Tutorial**

### Conclusion:

You are now able to run an Xpages App using JavaBean and JavaAgent to create an XML file that submits to NSF and DB2 back-end databases, accessed via URL to be made available to external JDeveloper App, do stay tuned, see Tutorial Part 4, Next...

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/UCSImDTpK0oe7QrPsYOE4nw>

-----  
**Dököll Solutions, Inc.**

<http://www.dokollolutionsinc.com>

version: 2016.09.24.2.22.PM