

Döcu Content

IBM DB2, Lotus Notes Domino

Log.nsf Data in DöcuContent.nsf

Part 3.0

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 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:

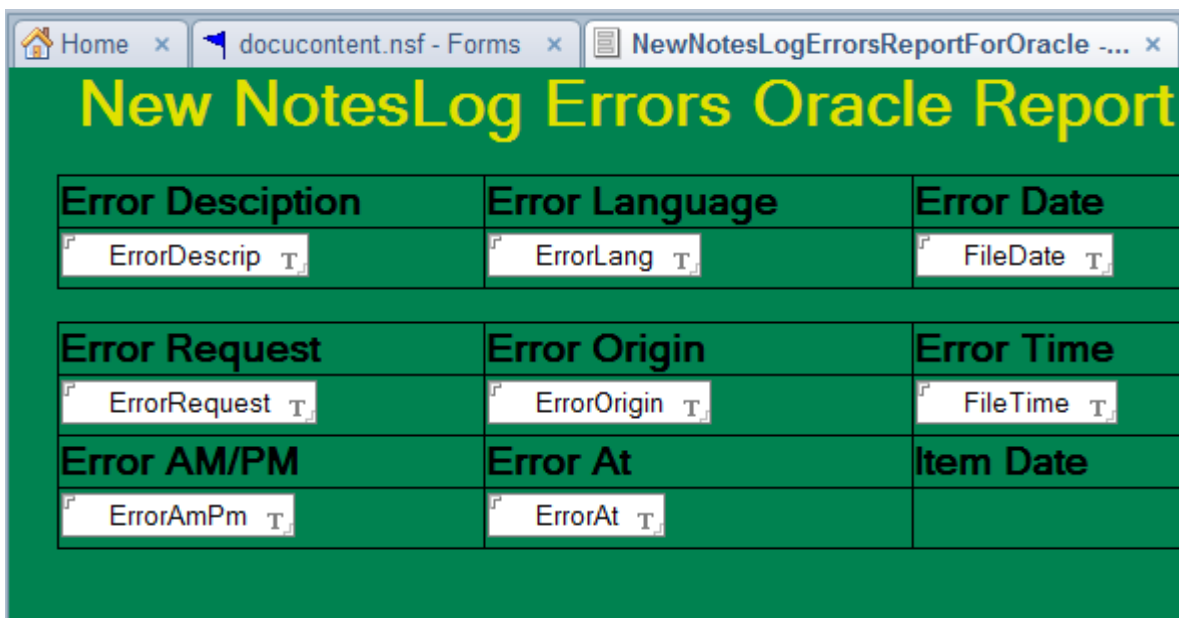
The current consists of Xpages forms, JavaAgent, JavaBean classes, with XML files in URL; wanted to add this in because the XML file creation is its own entity in the whole process. Now, since you've done this before, there's a likelihood that you created Forms and Views in Notes Domino back-end and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

Form: `NewNotesLogErrorsReportForOracle`

Submission to DB2 back-end database, made possible by a JavaBean, JavaAgent, and Xpage form button, with support from back-end Form and View elements

Hints on Creating form fields: In Döcu Content, you can first create a form, by right-clicking on the form, adding fields, simple as that- You probably want to keep a list of these fields, to be used when creating a View element...

`NewNotesLogErrorsReportForOracle`

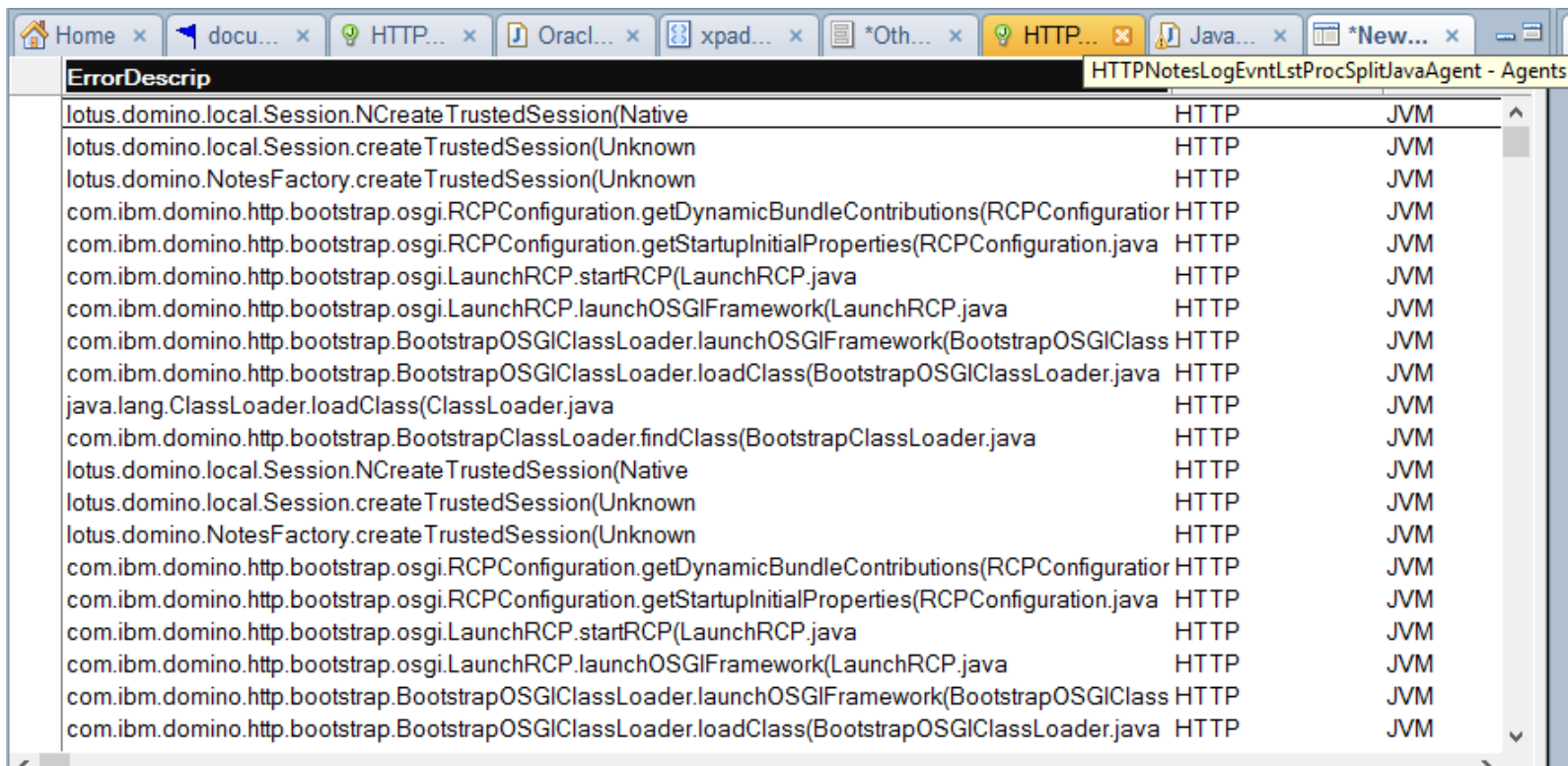


The screenshot shows a web browser window with the following tabs: Home, docucontent.nsf - Forms, and NewNotesLogErrorsReportForOracle ... The main content area has a green background with the title "New NotesLog Errors Oracle Report" in yellow. Below the title is a table with three columns and three rows of input fields:

Error Description	Error Language	Error Date
<input type="text" value="ErrorDescrip"/>	<input type="text" value="ErrorLang"/>	<input type="text" value="FileDate"/>
Error Request	Error Origin	Error Time
<input type="text" value="ErrorRequest"/>	<input type="text" value="ErrorOrigin"/>	<input type="text" value="FileTime"/>
Error AM/PM	Error At	Item Date
<input type="text" value="ErrorAmPm"/>	<input type="text" value="ErrorAt"/>	

View: **NewNotesLogErrorsReportViewOraOut**

View columns showing data already gathered through process in this tutorial...
back-end Data to be added as XML via JavaBean, JavaAgent, and Xpages



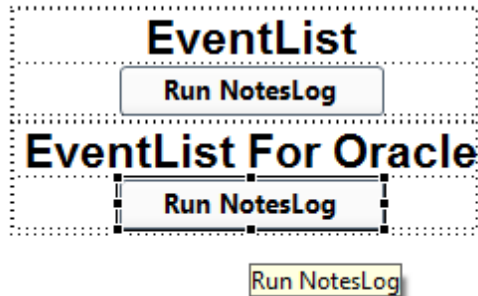
ErrorDescrip	HTTP	JVM
lotus.domino.local.Session.NCreateTrustedSession(Native	HTTP	JVM
lotus.domino.local.Session.createTrustedSession(Unknown	HTTP	JVM
lotus.domino.NotesFactory.createTrustedSession(Unknown	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getDynamicBundleContributions(RCPConfiguratio	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getStartupInitialProperties(RCPConfiguration.java	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(BootstrapOSGIClass	HTTP	JVM
com.ibm.domino.http.bootstrap.BootstrapOSGIClassLoader.loadClass(BootstrapOSGIClassLoader.java	HTTP	JVM
java.lang.ClassLoader.loadClass(ClassLoader.java	HTTP	JVM
com.ibm.domino.http.bootstrap.BootstrapClassLoader.findClass(BootstrapClassLoader.java	HTTP	JVM
lotus.domino.local.Session.NCreateTrustedSession(Native	HTTP	JVM
lotus.domino.local.Session.createTrustedSession(Unknown	HTTP	JVM
lotus.domino.NotesFactory.createTrustedSession(Unknown	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getDynamicBundleContributions(RCPConfiguratio	HTTP	JVM
com.ibm.domino.http.bootstrap.osgi.RCPConfiguration.getStartupInitialProperties(RCPConfiguration.java	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(BootstrapOSGIClass	HTTP	JVM
com.ibm.domino.http.bootstrap.BootstrapOSGIClassLoader.loadClass(BootstrapOSGIClassLoader.java	HTTP	JVM

Here is Xpages Code, with button: What we've done here is we created and Xpage ([xprunnoteslogevents.xsp](#)) and added a button to run a JavaBean that then launches a JavaAgent to grab Log.nsf errors generated by Döcu Content when run in Browser or during development. An XML file is later created to a URL that gets picked up by internal and external applications.

xpaddnoteslogerrxmltodb2.xsp

Xpages Submit Design # 1

NotesLog Data Example



```
<?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>Notes Data Example</title>
        </head>
        <xp:br></xp:br><xp:div style="text-align:center">
            <xp:span style="font-weight:bold;font-size:28pt">NotesLog Data Example
            </xp:span></xp:div><xp:div style="text-align:center;font-
size:9pt"></xp:div><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">
                <td
                    style="font-weight:bold;font-size:12pt;text-
align:center"></td>
                <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">EventList</xp:span></xp:div><xp:div style="text-align:center;font-
size:13pt"><xp:button value="Run NotesLog" 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:RunNotesLogEventListJavaBean.doRunNotesLogErrorEvents () }]]></xp
:this.action>
</xp:eventHandler>
</xp:button></xp:div></td>
</tr>
<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">EventList For
Oracle</xp:span></xp:div><xp:div style="text-align:center;font-
size:13pt"><xp:button value="Run NotesLog" id="button1" style="font-
weight:bold;font-size:10pt">
<xp:eventHandler
event="onclick" submit="true" refreshMode="complete" immediate="false" save="true"
id="eventHandler1">
<xp:this.action><![
CDATA[#{javascript:RunNotesLogEventListJavaBean.doRunNotesLo
gEventsForOracle () }]]></xp:this.action>
</xp:eventHandler>
</xp:button></xp:div></td>
</tr>
</table>
</xp:form>
</body>
</html>
</xp:view>

```

JavaBean code running JavaAgent, via Xpages button: Copy and paste below JavaBean in your environment to work with the Xpages file submitted earlier and load records into the back-end as Notes Documents, which can be used to create an XML file. The process is simple, when the Xpages submit button is accessed, search begins to collect errors from Log.nsf (external NSF database), areas of interest are highlighted for your convenience...

RunNotesLogEventListJavaBean.java

```

package com.dokoll.solutions.inc.search;

import javax.faces.context.FacesContext;
import lotus.domino.NotesException;
import lotus.domino.Session;
import lotus.domino.local.Database;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import com.ibm.xsp.model.domino.DominoUtils;

public class RunNotesLogEventListJavaBean {

    public void doRunNotesLogErrorEvents() throws NotesException{
        // set up the loggers
        Log log = LogFactory.getLog(RunNotesLogEventListJavaBean.class);
        Log debug = LogFactory.getLog("DEBUG");

        // tell what they are
        System.out.println("log For NotesLog Errors " +
log.getClass().getName());
        System.out.println("debug For NotesLog Errors " +
debug.getClass().getName());

        log.info("For NotesLog Errors info message");
        debug.debug("For NotesLog Errors debug message");

        log.info("For NotesLog Errors");
        debug.debug("For NotesLog Errors debug message");

        //get the current database being used
        Database database= (Database) FacesContext.getCurrentInstance()
        .getApplication().getVariableResolver()
        .resolveVariable(FacesContext.getCurrentInstance(), "database");
        System.out.println("docucontent Database Obtained..." + database);

database.getAgent("NotesLogErrorsEventListProcessingJavaAgent").runOnServer();
    }

```

```

    public void doRunNotesLogEventsForOracle() throws
NotesException{

        Session session = DominoUtils.getCurrentSession(FacesContext
        .getCurrentInstance());
        Database database = (Database)
session.getDatabase(session.getServerName(), "log.nsf");
        // set up the loggers
        Log log = LogFactory.getLog(RunNotesLogEventListJavaBean.class);
        Log debug = LogFactory.getLog("DEBUG");

        // tell what they are
        System.out.println("log For NotesLog Errors " +

```

```

log.getClass().getName());
    System.out.println("debug For NotesLog Errors " +
debug.getClass().getName());

    log.info("doRunNotesLogEventsForOracle | For NotesLog Errors info
message");
    debug.debug("For NotesLog Errors debug message");

    log.info("For NotesLog Errors");
    debug.debug("For NotesLog Errors debug message");

    //get the current database being used
    database= (Database) FacesContext.getCurrentInstance()
    .getApplication().getVariableResolver()
    .resolveVariable(FacesContext.getCurrentInstance(), "database");
    System.out.println("docucontent Database Obtained..." + database);

database.getAgent("HTTPNotesLogEvntLstProcSplitJavaAgent").runOnServ
er();
    }

}

```

JavaAgent code grabbing Log.nsf Notes Documents: Copy and paste below JavaAgent in your environment to work with the Xpages file and JavaBean submitted earlier and retrieve records from Log.nsf back-end, split the data into chunks to fill in back-end View **NewNotesLogErrorsReportViewOraOut** in docucontent.nsf.

JavaAgent.java

```

/**
 * Created from Original: 2016.09.05.7.13.PM
 * HTTPNotesLogEvntLstProcSplitJavaAgent | javaagentnoteslogerrhttp.xml
 * NotesLog items to be converted to TXT/CSV/XML/DB2 data for Oracle JDeveloper
 DocuWebNews
 * TODO: Before running code, be sure to remove System.outs, meant for Testing...
 * After creating Records in DB2, be sure to remove access for Anonymous User
 */
import java.util.Enumeration;
import java.util.Vector;

import lotus.domino.AgentBase;
import lotus.domino.Database;
import lotus.domino.DocumentCollection;
import lotus.domino.Session;

```

```

import lotus.domino.local.Document;

/**
 * @author Dököll Solutions, Inc.
 * @version 2016.09.05.7.13.PM
 *
 */
public class JavaAgent extends AgentBase {

    //TODO: Create TXT, CSV, and XML files with the results, feed to Xpages
    public void NotesMain() {

        try {
            //Grab session, yank database out of that
            Session session = getSession();
            System.out.println("FOUND_SESSION " + session);
            //...

            Database logDatabase = session.getDatabase("", "log.nsf");
            System.out.println("FOUND_DATABASE " + session);
            //Create a date based on session, we want today's date
            //DateTime dtToday = session.createDateTime("Today");

            String dtToday = "08/29/2012";
            System.out.println("FOUND_DATE " + dtToday);

            //Convert this date to String, endsWith or startsWith can't search therefore
            String ConvertDateToString = dtToday.toString();
            System.out.println("FOUND_DATE_CONVERSION " + ConvertDateToString);
            //Fetch EventList through Forms, by pass the MiscEvents view
            //TODO: Grab EventList View if you need specific dates
            DocumentCollection docColl = logDatabase.search("Form = 'Events'");
            Document logDoc = (Document) docColl.getFirstDocument();
            //Enter Loop, we are looking at only todays docs/Events
            System.out.println("BEGIN LOOP...");
            while (logDoc != null) {
                //DEBUG ONLY>>
                //...
                //System.out.println("INSIDE LOOP...");
                // Grab EventList items
                //System.out.println("Grabbing EventList items...");
                Vector itemsList = logDoc.getItemValue("EventList");
                //DEBUG ONLY>>
                //...
                //System.out.println("INSIDE VECTOR...");
                //Run through Events line by line
                for (Enumeration values = itemsList.elements(); values
                    .hasMoreElements();) {
                    String eventFetch = (String) values.nextElement();
                    //DEBUG ONLY>>
                    //...
                    //System.out.println("INSIDE eventFetch...");
                    //grab only the Events that match the search criteria, date and text
                    //TODO: Retrieve multiple error types and language
                    if (eventFetch.startsWith(ConvertDateToString) &&
                        eventFetch.endsWith(")") || eventFetch.matches("com.")){

                        //TODO: Perhaps begin conversion to XML and so on here...

```



```

//Adding to Notes View
//...
//DEBUG ONLY>>
//...
//System.out.println("INSIDE indexOf...");
String strEventBitsReplace = eventFetch.toString();
//Write this block better, or add on one line
    strEventBitsReplace =
strEventBitsReplace.replaceAll(":", ",");
    strEventBitsReplace =
strEventBitsReplace.replaceAll(" ", ",");
    strEventBitsReplace =
strEventBitsReplace.replaceAll("[()]", ",");
    strEventBitsReplace =
strEventBitsReplace.replaceAll("[ ]", ",WebHost");
//DEBUG ONLY>>
//System.out.println("INSIDE REPLACE...");
//...
//Split items and load to awaiting array variable...
String[] strColSplit = strEventBitsReplace.split(",");
//plug array values into local String variables
String strDateTime = strColSplit[0].trim(); //DateAndTime
String strHour = strColSplit[1].trim(); //Hour
String strMinute = strColSplit[2].trim(); //Minute
String strSeconds = strColSplit[3].trim(); //Seconds
String strAmPm = strColSplit[4].trim(); //AmPm
String strErrorSpace = strColSplit[5].trim(); //ErrorSpace
String strErrorRequest = strColSplit[6].trim(); //ErrorRequest
String strErrorOrig = strColSplit[7].trim(); // ErrorOrig
//Troubleshoot this one, but do not add to NSF back-end
String strErrorDummy = strColSplit[8].trim(); //Dummy
String strErrorAt = strColSplit[9].trim(); //ErrorAt
String strErrorDescrip = strColSplit[10].trim(); //ErrorDescrip
String strErrorLang = strColSplit[11].trim(); //ErrorLang
//Convert/format time data for FileTime column
String ConvertStringToString = strHour + ":" + strMinute + ":" +
strSeconds;
//FOR DEBUG ONLY:
//use below sparingly, you are re-populating Log.nsf
//...
//System.out.println("FOUND_LOG " + ConvertStringToString + " " +
strAmPm + " " + strErrorSpace + " " + strErrorLang+ " " + strErrorRequest + " " + "
" + strErrorOrig + " " + strErrorSpace + " " + strErrorAt + " " + strErrorDescrip);
//...
//these documents will be available in docucontent.nsf
Database currDatabase = session.getDatabase("",
"docucontent.nsf");
System.out.println("FOUND_CURR_DATABASE " + session);
//Create a date based on session, we want today's date
//DateTime dtToday = session.createDateTime("Today");
int docount = 0;

//declare and initialise document variable
//create document(s)
Document submitDocument = (Document) currDatabase.createDocument();
//load to form 'OtherNewNotesLogErrorsReportForOracle' in your back-end

```

```

submitDocument.replaceItemValue("Form",
"NewNotesLogErrorsReportForOracle");
submitDocument.replaceItemValue("FileDate", strDateTime);
submitDocument.replaceItemValue("FileTime", ConvertStringToString);
submitDocument.replaceItemValue("ErrorAmPm", strAmPm);
submitDocument.replaceItemValue("ErrorRequest", strErrorRequest);
submitDocument.replaceItemValue("ErrorDescrip", strErrorDescrip);
submitDocument.replaceItemValue("ErrorLang", strErrorLang);
submitDocument.replaceItemValue("ErrorAt", strErrorAt);
submitDocument.replaceItemValue("ErrorOrigin", strErrorOrig);

//save this document, otherwise it is not recorded in the back-end
submitDocument.save();
//increment counter
docount += 1;
System.out.println("FOUND_COUNT " + docount);

}

}
//grab additional log items, as necessary...
logDoc = (Document) docColl.getNextDocument();
}

}
catch (Exception e) {
e.printStackTrace();
}
} //end of program...
}

```

DB2 Database: ORA_LOGFEEDS_DB (Screenshot # 1): IBM DB2 database showing results from XML, accessed through URL, provided by Lotus Notes Domino JavaAgent...

ORA_LOGFEEDS_DB

Control Center - DB2COPY1

Control Center Selected Edit View Tools Help

Object View Command Editor 1 X

Commands Query Results Access Plan

Edits to these results are performed as positioned UPDATES and DELETES. Use the Tools Settings notebook to change the form of editing.

RECID	ERRORDESCRPT	ERRORLANG	ERRORDATE	ERRORTIME	ERRORAMPM
0	com.ibm.designer.runtime.domino.adapter.ComponentModule\$AdapterInvoker.invokeServlet	ComponentModule.java:847	08/29/2012	11:49:43	PM
1	com.ibm.designer.runtime.domino.adapter.ComponentModule\$AdapterInvoker.invokeServlet	ComponentModule.java:847	08/29/2012	11:49:43	PM
2	com.ibm.jspicript.JSEExpression.interpretExpression	JSEExpression.java:410	08/29/2012	11:49:43	PM
3	com.ibm.jspicript.JSEExpression.interpretExpression	JSEExpression.java:410	08/29/2012	11:49:43	PM
4	com.ibm.jspicript.JSEExpression.evaluateValue	JSEExpression.java:251	08/29/2012	11:49:43	PM
5	com.ibm.xsp.javascripct.JavaScriptInterpreter.interpret	JavaScriptInterpreter.java:221	08/29/2012	11:49:43	PM
6	javax.faces.component.UIComponentBase.isRendered	UIComponentBase.java:451	08/29/2012	11:49:43	PM
7	javax.faces.component.UIComponentBase.fillShadowedFlags	UIComponentBase.java:880	08/29/2012	11:49:43	PM
8	javax.faces.component.UIComponentBase.encodeBegin	UIComponentBase.java:948	08/29/2012	11:49:43	PM
9	com.ibm.xsp.util.FacesUtil.renderComponent	FacesUtil.java:840	08/29/2012	11:49:43	PM
10	com.ibm.xsp.util.FacesUtil.renderChildren	FacesUtil.java:871	08/29/2012	11:49:43	PM
11	com.ibm.xsp.renderkit.html_extended.HtmlBasicRenderer.encodeChildren	HtmlBasicRenderer.java:206	08/29/2012	11:49:43	PM
12	com.ibm.xsp.renderkit.ReadOnlyAdapterRenderer.encodeChildren	ReadOnlyAdapterRenderer.java:162	08/29/2012	11:49:43	PM
13	javax.faces.component.UIComponentBase.encodeChildren	UIComponentBase.java:979	08/29/2012	11:49:43	PM
14	com.sun.faces.lifecycle.LifecycleImpl.phase	LifecycleImpl.java:210	08/29/2012	11:49:43	PM
15	com.ibm.xsp.controller.FacesControllerImpl.render	FacesControllerImpl.java:264	08/29/2012	11:49:43	PM
16	com.ibm.xsp.webapp.FacesServlet.serviceView	FacesServlet.java:248	08/29/2012	11:49:43	PM
17	com.ibm.xsp.webapp.FacesServletEx.serviceView	FacesServletEx.java:200	08/29/2012	11:49:43	PM
18	com.ibm.xsp.webapp.FacesServlet.service	FacesServlet.java:160	08/29/2012	11:49:43	PM
19	com.ibm.xsp.webapp.FacesServletEx.service	FacesServletEx.java:137	08/29/2012	11:49:43	PM
20	com.ibm.xsp.webapp.DesignerFacesServlet.service	DesignerFacesServlet.java:103	08/29/2012	11:49:43	PM
21	com.ibm.designer.runtime.domino.adapter.ComponentModule.invokeServlet	ComponentModule.java:576	08/29/2012	11:49:43	PM
22	com.ibm.domino.xsp.module.nsf.NSFComponentModule.invokeServlet	NSFComponentModule.java:1267	08/29/2012	11:49:43	PM
23	com.ibm.designer.runtime.domino.adapter.ComponentModule\$AdapterInvoker.invokeServlet	ComponentModule.java:847	08/29/2012	11:49:43	PM
24	com.ibm.designer.runtime.domino.adapter.ComponentModule\$ServletInvoker.doService	ComponentModule.java:796	08/29/2012	11:49:43	PM

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/UCSimDTpK0oe7OrPsYOE4nw>

Dököll Solutions, Inc.

<http://www.dokollsolutionsinc.com>

version: 2016.09.24.2.22.PM