

Döcu Content

IBM DB2, Lotus Notes Domino

Log.nsf Docs | DöcuContent.nsf XML

Part 3.1

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.

Before you Begin: Since you are here, you most likely went through the previous Tutorial (Version 3.0)- The attempt below is to grab similar records but for 2016 Year...

Added Info:

We did write additional code to get 2016 Errors from Log.nsf DB, we wanted those in a different view, thus OtherNewNotesLogErrorsReportForOracle form and OtherNewNotesLogErrorsReportViewOraOut view had to be created-

JavaAgent performs XML creation magic: Read the TODO portion of the code, some helpful information there... the process stays dormant until JDeveloper ADF/JSF App picks up the XML file via URL and attempts to grab data out of it, in turn, submitting to DB2 database. Have a good look at this, 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.

[HTTPNotesLogEvntLstProcSplitJavaAgent | javaagentdataforwebnewshttp.xml](#)

JavaAgent for Parsing Log.nsf Records into readable Documents

```
/**
 * 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 = "09/10/2016";
            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 Events 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",
"OtherNewNotesLogErrorsReportForOracle");
    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...
}

```

JavaAgent providing XML data, obtained as Notes Documents: Copy and paste below JavaAgent in your environment to work with the Xpages file submitted earlier and make preparations for loads of XML data to funnel through to DB2 back-end. The process is simple, when the Xpages submit button is pressed, a view is searched systematically to collect all values that will be used to create below XML file, areas of interest are highlighted for your convenience...

OracleNotesLogFeedJavaAgent.java

JavaAgent creating XML File from View Data (Post XML, to be read by JavaBean)

Code to create XML file from existing data- This is to demonstrate that XML file creation with Back-end View data and keep on StandBy to be pulled in by JavaBean to perform submit to DB2 database. Visit our website: www.dokollolutionsinc.com for additional info on how we arrived a XML file with Log.nsf records- You can also search Döcu Content t find the Java classes used to create the XML file in question.

```

/**
 * Created from Original: 2016.09.07.4.51.AM
 * HTTPNotesLogEvtLstProcSplitXMLJavaAgent | javaagentoranotesfeedsdatahttp.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.io.PrintWriter;

import lotus.domino.AgentBase;
import lotus.domino.AgentContext;
import lotus.domino.Database;
import lotus.domino.Session;
import lotus.domino.View;
import lotus.domino.local.Document;

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

    // TODO: Create TXT, CSV, and XML files with the results, feed to Xpages
    // 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("HTTPNotesLogEvtLstProcSplitXMLJavaAgent |
javaagentoranotesfeedsdatahttp.xml session..."
                + session);

            // load agentContext
            AgentContext agentContext = session.getAgentContext();
            // load info console for debugging purposes
            System.out
                .println("HTTPNotesLogEvtLstProcSplitXMLJavaAgent |
javaagentoranotesfeedsdatahttp.xml 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("HTTPNotesLogEvtLstProcSplitXMLJavaAgent |
javaagentoranotesfeedsdatahttp.xml Connection to database..." + db
        + " established");

        View pview =
db.getView("NewNotesLogErrorsReportViewOraOut");
        // 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 = (Document) 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:
/javaagentoranotesfeedsdatahttp.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("<noteslogerrors>");
        while (doc != null) { // Write the value in the UserName field to
            // the output stream

                pw.println("<errorreading>");

                pw.println("<errordescrrip>");

                pw.println(doc.getItemValueString("ErrorDescrrip"));
                pw.println("</errordescrrip>");

                pw.println("<errorlang>");

                pw.println(doc.getItemValueString("ErrorLang"));
                pw.println("</errorlang>");

                pw.println("<filedate>");

                pw.println(doc.getItemValueString("FileDate"));
                pw.println("</filedate>");

```

```

        pw.println("<filetime>");

pw.println(doc.getItemValueString("FileTime"));
        pw.println("</filetime>");

        pw.println("<errorampm>");

pw.println(doc.getItemValueString("ErrorAmPm"));
        pw.println("</errorampm>");

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

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

    pw.println("</noteslogerrors>");
    System.out
        .println("XML Data Created with
HTTPNotesLogEvtLstProcSplitXMLJavaAgent | javaagentoranotesfeedsdatahttp.xml");

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

```

JavaAgent Properties

- GoogleAddRowJavaAger
- GoogleAddSpreadsheets
- GoogleRetrieveSpreadsh
- GoogleRetrieveUniqueW
- GoogleRetrieveUniqueW
- GoogleRetrieveUniqueW
- GoogleRetrieveUniqueW
- GoogleRetrieveWorkshee
- GoogleRetrieveWorkshee
- GoogleSpreadsheetsGoo
- HTTPCSVDataJavaAgent
- HTTPDDECSVDataDevJa
- HTTPHTMLDashDataSou
- HTTPHTMLDataJavaAge
- HTTPHTMLEditJavaAger
- HTTPjQueryReportToolJ
- HTTPNotesLogEvtLstPr
- HTTPNotesLogEvtLstPr

HTTPNotesLogEvtLstProcSplitXMLJavaAgent - Agents - Döcu Content - \\Local\docucontent.nsf
 in \\local\docucontent.nsf

Agent Contents

New Java Class Import Export Compile All

Src
 (default package)
 OracleNotesLogFeedJavaAgent.java
 Archive
 Res

Editing the Agent

This Java Agent tab contains the contents of the Java agent. Here, you can create or import files, and select files to edit.

To edit a file, double-click it in the list; the file opens for editing in a separate tab.

Keep this Java Agent tab open as you edit the files, then save and close it when you finish editing.

Properties Events Problems (0 errors, 2 warnings, 0 ot...)

Outline
 An outline is not available.

- Basics
- Security
- Document Selection

Name: HTTPNotesLogEvtLstProcSplitXMLJa
 Alias: javaagentoranotesfeedsdatahttp.xml
 Comment: testing NotesLog grab to be dissected

Java
 Base class: OracleNotesLogFeedJavaAgent.class
 Compile Java code with debugging information

WebNewsForHTTPUserXMLDataJavaAgent
in \\local\docucontent.nsf

Agent Contents

New Java Class Import Export Compile All

Src
(default package)
JavaAgent.java
Archive
Res

Editing the Agent

This Java Agent tab contains the contents of the Java agent. Here, you can create or import files, and select files to edit.

To edit a file, double-click it in the list; the file opens for editing in a separate tab.

Keep this Java Agent tab open as you edit the files, then save and close it when you finish editing.

Properties Events Problems (0 items)

Basics
Security
Document Selection

Name: WebNewsForHTTPUserXMLDataJavaA
Alias: javaagentdataforwebnewshttp.xml
Comment: this item is for Xpages forms, data will

Java
Base class: JavaAgent.class
 Compile Java code with debugging information

Runtime
Trigger: On event On schedule
Action menu selection
Edit settings...
Target: All documents in database

Options
 Shared Private
 Store search in search bar menu
 Store highlights in document
 Run in background client thread
 Profile this agent

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.dokollsolutionsinc.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/UCSimDTpK0e7OrPsYOE4nw>

Dököll Solutions, Inc.

<http://www.dokollsolutionsinc.com>

version: 2016.09.24.2.22.PM