JavaBean/LotusScript

Fetch Log.nsf Documents/Create Text File

JavaAgent

System Requirements:
Download Domino Designer **8.5.3** Environment (DDE)
http://www.ibm.com/developerworks/downloads/ls/dominodesigner/

Introduction:
Use JavaBean, JavaAgent, LotusScript and Xpages code samples as guide to search Log.nsf and create a text file to C:/ drive for later purpose. Build a connection to a Lotus Notes Domino Database located on C Drive to get started...

Disclaimer:
Information contained in the following is presented as is. This tutorial assumes you have basic programming knowledge. All tutorials are based on an Eclipse/Eclipse-based software. Should you need to familiarize yourself with a certain Eclipse environment, prior to continuing this tutorial, please stop now and see our Tutorials page...

Copy and Paste Xpages Code

At this point we assume Domino Designer 8.5.3 is downloaded/installed. Copy and paste below code samples to your environment, areas of interest have been highlighted for your convenience.

xpreadfiles.xsp

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

      <xp:button value="Run Agent" id="button2">
            <xp:eventHandler event="onclick" submit="true" refreshMode="complete"
immediate="false" save="true" id="eventHandler2">

                  <xp:this.action><!
[CDATA[#{javascript:RunLogNSFAgentBean.submitEntry()}]]></xp:this.action>
            </xp:eventHandler>
      </xp:button><xp:br></xp:br>
</xp:view>
```

Copy and Paste JavaBean Code

You can also write a new class to run the LotusScript code that will be used to create the text file with new documents from Log.nsf. Copy and paste below code sample to your environment.

RunLogNSFAgentBean.java;

```
package com.dokoll.solutions.inc.iphone;
/**
 * Created: 2012.09.29.12.45.PM
 * Run SearchComStringInLog and SearchHttpStringInLog LotusScript Agents
 */


import javax.faces.context.FacesContext;
import lotus.domino.NotesException;
import lotus.domino.local.Database;

/**
 * @aurthor: Dököll Solutions, Inc.
 * @version 2012.09.29.12.45.PM
 */
public class RunLogNSFAgentBean {

      public void submitEntry() {

            //let's add a try catch here

            try {

                  //get the current database being used
                  Database database= (Database) FacesContext.getCurrentInstance()
                  .getApplication().getVariableResolver()
                  .resolveVariable(FacesContext.getCurrentInstance(), "database");
                  System.out.println("Database Obtained..." + database);
                  //...
                  //Run agent(s)
                  database.getAgent("SearchHttpStringInLog").runOnServer();
                 database.getAgent("SearchComStringInLog").runOnServer();

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

            }

      }
```

Added information

Make sure your LotusScript Agent is set up to report only new documents from the back-end, to avoid load in the programming.

Copy and Paste LotusScript Code

Use this LotusScript code to fetch HTTP errors from the appLog.nsf database, create a CSV file with the data to your C:/ drive.

SearchHttpStringInLog.lss;

```
%REM
App:Docu.nsf
Pro:SearchHttpStringInLog
Now:2012.10.04.12.51.AM
Sys:Dököll Solutions, Inc.
Des:LotusScript Agent created from copy to read Log.nsf database and load results
into text file
   :Agent should be set up to load new documents only, avoid loads in file/servers.
See Agent properties...
%END REM

Option Public

Sub Initialize
      'Declare database variables
      Dim ServerLog As NotesDatabase
      Dim Sess As NotesSession
      Dim ServerName As String
      Dim DBView As NotesView
      Dim DocView As String
      Dim DatabaseName As String
      'Declare fileGrab variables
      Dim RecordsPosition As Long
      Dim RecordsCount As Long
      Dim Filename As String
      Dim RecordsFreeFile As Integer
      'Declare Search variables
      Dim SearchCriteria As String
      Dim LogDoc As NotesDocument
      'Get a Notes session
      Set Sess = New NotesSession
      'Grab the Server Name based ono that session
      ServerName = Sess.CurrentDatabase.Server
      'Add query String... SearchCriteria for specific errors
      Dim srchCriteria (7) As String
      srchCriteria(0)= "java.lang.NullPointerException"
      srchCriteria(1)= ": javax.faces."
      ' the colon is added to ensure we are getting
      ' the right faces error (see applog.nsf info)
      srchCriteria(2)= "Couldn't instantiate"
      srchCriteria(3)= "NotesException: Method is not available"
      srchCriteria(4)= "Couldn't find design note"
      srchCriteria(5)= "Full text operations on database"
      srchCriteria(6)= "Lotus Notes Exception - Invalid or nonexistent document"

      'initialize variable for viewName
      DocView = "MiscEvents"
      'initialize variable for Logdb Name
      DatabaseName = "applog.nsf"

      'TO DO: Revive this Integer(Count) for testing purposes
      'initialize variable for Count
      RecordsCount = 0
      '[add RecordCounts here To uniquely identify values]
```

```
    'Set up name for the text file being created
    FileName = ServerName & "LOGNSFMULTHTTPERRORSTYPE"

    'Set up placeHolder for the text file being created
    'FileName = "C:\temp\TXT_DATA\" & Filename &".txt"

    'Set up placeHolder for the text file being created
    FileName = "C:\temp\CSV_DATA\" & Filename &".csv"

    'Initialize freeFile placeholder
    RecordsFreeFile = FreeFile()

    'use placeholder to load data/documents in File
    Open FileName For Output As #RecordsFreeFile

    'yank HTTP JVM values from this database
    Set ServerLog = New NotesDatabase(ServerName, DatabaseName)

    'HTTP JVM list of items are contained in this view
    Set DBView = ServerLog.GetView(DocView)
    'TO DO: Add an entry here to see if the document is actually new
    'Still set up this database in 'Agent Properties' to grab values from
applog.nsf
    'this should help avoid creating a truly large text file (enhances
performance)
    'Further, query the back-end (applog.nsf) with a dateRange, in which case you
may need to index
    'the applog.nsf database so it is searchable

    'Grab first document
    Set LogDoc = DBView.GetFirstDocument

    'run through this list (EventList) until last row in the document is reached
    While Not LogDoc Is Nothing
        'TO DO: Add a dateRange here to minimize the load in your textFile
        'Alternatively, you can DateDiffer against LogDoc.Created, get results
based on number of days
        'you want returned to thr program to creat the textFile

        'pick any logItems created from EventList
        If LogDoc.Created Then
            'read records sequentially from eventList
            ForAll LogRecords In LogDoc.EventList

                ForAll errDescrip In srchCriteria
                    'search the value SearchCriteria through the records
returned from eventList
                    RecordsPosition = InStr(LogRecords, errDescrip)
                    'if found HTTP is found, fill in RecordsFile
                    If RecordsPosition > 0 Then
                        'Print the records into the text (RecordsFile)
file
                        Print #RecordsFreeFile,LogRecords '[add
RecordCounts here to uniquely identify values]
                        'increment the RecordsCount and loop through
until end of the textfile (newly created)
                        RecordsCount = RecordsCount + 1
                    End If
```

```
                        End ForAll

                  End ForAll
            End If
            'Get the remaining docs sequentially from eventList, if any
            Set LogDoc = DBView.GetNextDocument(LogDoc)
      Wend
      'close the file
      Close #RecordsFreeFile
      'TO DO: Revive this messageBox for testing purposes
      MessageBox "Number of Records Searched " & RecordsCount & " based on Search
Criteria"
End Sub
```

Copy and Paste LotusScript Code

Use this LotusScript code to fetch other error types from the appLog.nsf database, create a CSV file with the data on your C:/ drive. Why are we doing this? If you look at the error log from appLog.nsf (copy of log.nsf) you will notice that some of the error lines are not aligned properly column-wise, so the query tries to split lines and account for additional spaces, hidden chars, and so on... thus text is not where they should be. In addition to this, each error comes with additional info that is thrown onto new lines. Needless to say, the code can be combined to do both tasks, error types/lists.

SearchComStringInLog.lss;

```
%REM
App:Docu.nsf
Pro:SearchComStringInLog
Now:2012.9.08.2.02.AM
Sys:Dököll Solutions, Inc.
Des:LotusScript Agent created from copy to read Log.nsf database and load results
into text file
   :Agent should be set up to load new documents only, avoid loads in file/servers.
See Agent properties...
%END REM

Option Public

Sub Initialize
      'Declare database variables
      Dim ServerLog As NotesDatabase
      Dim Sess As NotesSession
      Dim ServerName As String
      Dim DBView As NotesView
      Dim DocView As String
      Dim DatabaseName As String
      'Declare fileGrab variables
      Dim RecordsPosition As Long
      Dim RecordsCount As Long
      Dim Filename As String
      Dim RecordsFreeFile As Integer
      'Declare Search variables
```

```
        Dim SearchCriteria As String
        Dim LogDoc As NotesDocument

        'Get a Notes session
        Set Sess = New NotesSession
        'Grab the Server Name based on session
        ServerName = Sess.CurrentDatabase.Server
        'Declare and initialize Array (query items)
        Dim srchCriteria (5) As String
        srchCriteria(0)= "at com.ibm."
        srchCriteria(1)= "at com.sun."
        srchCriteria(2)= "at javax.faces."
        srchCriteria(3)= "at java.security."
        srchCriteria(4)= "at lotus.domino."
        srchCriteria(5)= "at java.lang."

        'Declare and initialize variable for viewName
        DocView = "MiscEvents"
        'Declare and initialize variable for Logdb Name
        DatabaseName = "applog.nsf"
        'TO DO: Revive this Integer(Count) for testing purposes
        RecordsCount = 0
        'Set up name for the text file being created
        FileName = ServerName & "LOGNSFMULTCOMERRORSTYPE"
        'Set up placeHolder for the text file being created
        FileName = "C:\temp\CSV_DATA\" & Filename &".csv"




        'Initialize freeFile placeholder
        RecordsFreeFile = FreeFile()
        'use placeholder, load data in FileName
        Open FileName For Output As #RecordsFreeFile
        '...
        Set ServerLog = New NotesDatabase(ServerName, DatabaseName)
        '...
        Set DBView = ServerLog.GetView(DocView)
        'TO DO: Add an entry here to see if the document is actually new
        'Still set up this database 'Agent Properties' from applog.nsf
        'this should help avoid creating a truly large text file (enhances
performance)
        'Further, query the back-end (applog.nsf) with a dateRange, in which case you
may need to index
        'the applog.nsf database so it is searchable
        'Grab the first document
        Set LogDoc = DBView.GetFirstDocument
        'run through this list (EventList) until last cursor in the documents list
        While Not LogDoc Is Nothing

                'TO DO: Add a dateRange here to minimize the load in your textFile
                'Alternatively, you can DateDiffer against LogDoc.Created
                'and count days from LogDoc.Created to search entries on
                'pick any logItems created from EventList
                If LogDoc.Created Then


                        'read records sequentially from eventList
                        ForAll LogRecords In LogDoc.EventList
                        ForAll errDescrip In srchCriteria
                        'search the value SearchCriteria through the records
returned from eventList
                              RecordsPosition = InStr(LogRecords, errDescrip)
                              'if found HTTP is found, fill in RecordsFile
                              If RecordsPosition > 0 Then
                              'Print the records into the text (RecordsFile) file
                                    Print #RecordsFreeFile,LogRecords '[add
RecordCounts here to uniquely identify values]
                              'increment the RecordsCount and loop through until
end of the textfile (newly created)
                                    RecordsCount = RecordsCount + 1
                              End If


                        End ForAll

                  End ForAll
            End If
            'Get the remaining docs sequentially from eventList, if any
            Set LogDoc = DBView.GetNextDocument(LogDoc)

      Wend
      'close the file
      Close #RecordsFreeFile
      'TO DO: Revive this messageBox for testing purposes
      MessageBox "Number of Records Searched " & RecordsCount & " based on Search
Criteria"

End Sub
```

WARNING:

Restrict only to new documents, or write additional code to get a dateRange and return a comfortable number of documents, to create a smaller text file-size.

Conclusion:

You can convert the LotusScript to a JavaAgent for scalability purposes, if you prefer. We hope to have a way to compare Log.nsf CSV data files against troubleshooting logs, to save developers the work of searching for information/code already present.

Questions, comments, please post a brief message on our Contact form on the main site.

Thank you for coming...