

# JavaAgent

Fetch CSV File Data/Create New CSV File(s)

# JavaAgent

## System Requirements:

Download Domino Designer **8.5.3** Environment (DDE)

<http://www.ibm.com/developerworks/downloads/ls/dominodesigner/>

## Introduction:

Use this JavaAgent code sample as guide to search CSV File for Log.nsf documents added prior and create a CSV file to C:/ drive for later purpose. Build a connection to a Lotus Notes Domino Database located on C Drive ton 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 JavaAgent Code

At this point we assume Domino Designer 8.5.3 is downloaded/installed. You are running this JavaAgent class as a Java Program to run against a Text file and create a CSV file that can later be compared against another CSV file to match errors and furnish a report. Copy and paste below code samples to your environment, areas of interest have been highlighted for your convenience.

## ComDataGrab.java;

```
/**
 * Copyright 2012 Dököll Solutions, Inc.
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 *
 *
 * @AppName: Docu.nsf
 * Program: ComDataGrab.java
```

```

* Created from Copy: 2012.28.6.00.AM
*
*/
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Random;
import java.util.StringTokenizer;

/**
 * @author Dököll Solutions, Inc.
 * @version 2012.28.6.00.AM
 *
 * @param args
 * @throws FileNotFoundException
 */
public class ComDataGrab {

    public static void main(String[] args) throws FileNotFoundException {
        // TODO Auto-generated method stub
        BufferedWriter writer = null;
        //Grab file created by LotusScript Agent
(SearchHttpStringInLog:2012.09.08.2.02.AM)
        //fileName="C:/temp/TXT_DATA/LOGNSFMULTERRORSTYPE.csv";
        String fileName="C:/temp/CSV_DATA/LOGNSFMULTERRORSTYPE.csv";

        //Write the results of the file into this path as a new CSV
        writer = new BufferedWriter(new OutputStreamWriter(new
FileOutputStream("C:/temp/CSV_DATA/BULK_CSV_COM_DATA_SPLIT.csv")));
        //entering Try Catch

        try {

            BufferedReader br = new BufferedReader( new FileReader(fileName));
            //Set up tokenPositions for file in question, based on spaces in fileLine
            StringTokenizer stringTokenizer = null;
            //declare and initialize counters for both the number of tokens found
            //and the number of lines retrieved
            int lineNumber = 0, tokenNumber = 0;

            //Create ColumNames for the CSV file
            System.out.println("ComDataGrab Creating columns, please wait...");

            writer.append("FileDate");
            writer.append(", ");
            writer.append("FileTime");
            writer.append(", ");
            writer.append("ErrorAmPm");
            writer.append(", ");
            writer.append("ErrorFrag");

```

```

writer.append(", ");
writer.append("ErrorRequest");
writer.append(", ");
writer.append("ErrorOrigin");
writer.append(", ");
writer.append("ErrorDummy");
writer.append(", ");
writer.append("ErrorAt");
writer.append(", ");
writer.append("ErrorDescrip");
writer.append(", ");
writer.append("ErrorLang");
writer.append(", ");
writer.append("ErrorNum");
//go to next line,prepare to load records t ocolumns...
writer.append("\r\n");
//Connect ColumNames to records
System.out.println("ComDataGrab Columns created, adding records to
columns...");
//run through file and read line by line
while ( (fileName = br.readLine()) != null)
{
//...
if(lineNumber++ == 0)
continue;
stringTokenizer = new StringTokenizer(fileName);

//set up array of String values to add to columns
String[] myString = null;

//run though until no more spaces found in fileLine
while (stringTokenizer.hasMoreTokens())
{
//load myString[x] in conjunction to each space (column)

lineNumber++;
myString = fileName.split("[\\s\\]");

//write the array into myString variable
writer.write(myString[0].trim() + ", " + myString[1].trim() + ", " +
myString[2].trim() + ", " + myString[3].trim() + ", " + myString[4].trim() + ", " +
myString[5].trim() + ", " + myString[6].trim()+", " + myString[7].trim()+", " +
myString[8].trim()+", " + myString[9].trim()+", " + lineNumber+"\n");

//increment tokenNumber
tokenNumber++;
//TO DO: Revive this console item for testing purposes
stringTokenizer.nextToken(); //
}

//clear tokenNumber
tokenNumber = 0;

```

```
    }  
    //flush/close writer  
    writer.flush();  
    writer.close();  
  
    System.out.print("ComDataGrab Agent Inserted Data to CSV file...");  
  
    } catch (FileNotFoundException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    } catch (IOException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }  
    }  
}
```

### **Conclusion:**

Compare results of the newly created CSV file against another CSV or Text file containing similarly-generated error-List, and make the errors search-able as SQL queries.

Questions, comments, please post a brief message on our [Contact](#) form on the main site.

Thank you for coming...