

```
// =====  
//  
// Copyright (c) 2006-2015, Talend SA  
//  
// Ce code source a été automatiquement généré par Talend Open Studio for Data Integration  
// / Soumis à la Licence Apache, Version 2.0 (la "Licence") ;  
// votre utilisation de ce fichier doit respecter les termes de la Licence.  
// Vous pouvez obtenir une copie de la Licence sur  
// http://www.apache.org/licenses/LICENSE-2.0  
//  
// Sauf lorsqu'explicitement prévu par la loi en vigueur ou accepté par écrit, le logiciel  
// distribué sous la Licence est distribué "TEL QUEL",  
// SANS GARANTIE OU CONDITION D'AUCUNE SORTE, expresse ou implicite.  
// Consultez la Licence pour connaître la terminologie spécifique régissant les autorisations et  
// les limites prévues par la Licence.
```

```
package local_project.etl_project_process_0_1;
```

```
import routines.Numeric;  
import routines.DataOperation;  
import routines.TalendDataGenerator;  
import routines.TalendStringUtil;  
import routines.TalendString;  
import routines.StringHandling;  
import routines.Relational;  
import routines.TalendDate;  
import routines.Mathematical;  
import routines.system.*;  
import routines.system.api.*;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;
```

```
import java.util.Date;
import java.util.List;
import java.math.BigDecimal;
import java.io.ByteArrayOutputStream;
import java.io.ByteArrayInputStream;
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.ObjectOutputStream;
import java.io.ObjectInputStream;
import java.io.IOException;
import java.util.Comparator;
```

```
@SuppressWarnings("unused")
```

```
/**
```

```
 * Job: etl_project_process Purpose: <br>
```

```
 * Description: <br>
```

```
 *
```

```
 * @author user@talend.com
```

```
 * @version 8.0.1.20211109_1610
```

```
 * @status
```

```
 */
```

```
public class etl_project_process implements TalendJob {
```

```
    protected static void logIgnoredError(String message, Throwable cause) {
```

```
        System.err.println(message);
```

```
        if (cause != null) {
```

```
            cause.printStackTrace();
```

```
        }
```

```
    }
```

```
public final Object obj = new Object();

// for transmitting parameters purpose
private Object valueObject = null;

public Object getValueObject() {
    return this.valueObject;
}

public void setValueObject(Object valueObject) {
    this.valueObject = valueObject;
}

private final static String defaultCharset = java.nio.charset.Charset.defaultCharset().name();

private final static String utf8Charset = "UTF-8";

// contains type for every context property
public class PropertiesWithType extends java.util.Properties {
    private static final long serialVersionUID = 1L;
    private java.util.Map<String, String> propertyTypes = new java.util.HashMap<>();

    public PropertiesWithType(java.util.Properties properties) {
        super(properties);
    }

    public PropertiesWithType() {
        super();
    }
}
```

```

        public void setContextType(String key, String type) {
            propertyTypes.put(key, type);
        }

        public String getContextType(String key) {
            return propertyTypes.get(key);
        }
    }

    // create and load default properties
    private java.util.Properties defaultProps = new java.util.Properties();

    // create application properties with default
    public class ContextProperties extends PropertiesWithType {

        private static final long serialVersionUID = 1L;

        public ContextProperties(java.util.Properties properties) {
            super(properties);
        }

        public ContextProperties() {
            super();
        }

        public void synchronizeContext() {

        }

        // if the stored or passed value is "<TALEND_NULL>" string, it mean null
        public String getStringValue(String key) {

```

```

        String origin_value = this.getProperty(key);

        if
(NULL_VALUE_EXPRESSION_IN_COMMAND_STRING_FOR_CHILD_JOB_ONLY.equals(origin_value)) {

            return null;

        }

        return origin_value;

    }

```

```

    }

```

protected ContextProperties context = new ContextProperties(); // will be instanciated by MS.

```

public ContextProperties getContext() {

    return this.context;

}

```

```

private final String jobVersion = "0.1";

private final String jobName = "etl_project_process";

private final String projectName = "LOCAL_PROJECT";

public Integer errorCode = null;

private String currentComponent = "";

```

```

private final java.util.Map<String, Object> globalMap = new java.util.HashMap<String,
Object>();

```

```

private final static java.util.Map<String, Object> junitGlobalMap = new
java.util.HashMap<String, Object>();

```

```

private final java.util.Map<String, Long> start_Hash = new java.util.HashMap<String, Long>();

private final java.util.Map<String, Long> end_Hash = new java.util.HashMap<String, Long>();

private final java.util.Map<String, Boolean> ok_Hash = new java.util.HashMap<String,
Boolean>();

public final java.util.List<String[]> globalBuffer = new java.util.ArrayList<String[]>();

```

```

// OSGi DataSource

private final static String KEY_DB_DATASOURCES = "KEY_DB_DATASOURCES";

private final static String KEY_DB_DATASOURCES_RAW = "KEY_DB_DATASOURCES_RAW";

public void setDataSources(java.util.Map<String, javax.sql.DataSource> dataSources) {
    java.util.Map<String, routines.system.TalendDataSource> talendDataSources = new
java.util.HashMap<String, routines.system.TalendDataSource>();

    for (java.util.Map.Entry<String, javax.sql.DataSource> dataSourceEntry :
dataSources.entrySet()) {

        talendDataSources.put(dataSourceEntry.getKey(),

                                new
routines.system.TalendDataSource(dataSourceEntry.getValue()));
    }

    globalMap.put(KEY_DB_DATASOURCES, talendDataSources);

    globalMap.put(KEY_DB_DATASOURCES_RAW, new java.util.HashMap<String,
javax.sql.DataSource>(dataSources));
}

public void setDataSourceReferences(List serviceReferences) throws Exception {

    java.util.Map<String, routines.system.TalendDataSource> talendDataSources = new
java.util.HashMap<String, routines.system.TalendDataSource>();

    java.util.Map<String, javax.sql.DataSource> dataSources = new
java.util.HashMap<String, javax.sql.DataSource>();

    for (java.util.Map.Entry<String, javax.sql.DataSource> entry : BundleUtils
        .getServices(serviceReferences,
javax.sql.DataSource.class).entrySet()) {

        dataSources.put(entry.getKey(), entry.getValue());

        talendDataSources.put(entry.getKey(), new
routines.system.TalendDataSource(entry.getValue()));
    }
}

```

```

        globalMap.put(KEY_DB_DATASOURCES, talendDataSources);

        globalMap.put(KEY_DB_DATASOURCES_RAW, new java.util.HashMap<String,
javax.sql.DataSource>(dataSources));
    }

    private final java.io.ByteArrayOutputStream baos = new java.io.ByteArrayOutputStream();

    private final java.io.PrintStream errorMessagePS = new java.io.PrintStream(new
java.io.BufferedOutputStream(baos));

    public String getExceptionStackTrace() {
        if ("failure".equals(this.getStatus())) {
            errorMessagePS.flush();
            return baos.toString();
        }
        return null;
    }

    private Exception exception;

    public Exception getException() {
        if ("failure".equals(this.getStatus())) {
            return this.exception;
        }
        return null;
    }

    private class TalendException extends Exception {

        private static final long serialVersionUID = 1L;

        private java.util.Map<String, Object> globalMap = null;

```

```
private Exception e = null;

private String currentComponent = null;

private String virtualComponentName = null;
```

```
public void setVirtualComponentName(String virtualComponentName) {

    this.virtualComponentName = virtualComponentName;

}
```

```
private TalendException(Exception e, String errorComponent, final
java.util.Map<String, Object> globalMap) {

    this.currentComponent = errorComponent;

    this.globalMap = globalMap;

    this.e = e;

}
```

```
public Exception getException() {

    return this.e;

}
```

```
public String getCurrentComponent() {

    return this.currentComponent;

}
```

```
public String getExceptionCauseMessage(Exception e) {

    Throwable cause = e;

    String message = null;

    int i = 10;

    while (null != cause && 0 < i--) {

        message = cause.getMessage();

        if (null == message) {

            cause = cause.getCause();

        }

    }

}
```



```

        } else {
            break;
        }
    }
    if (null == message) {
        message = e.getClass().getName();
    }
    return message;
}

@Override
public void printStackTrace() {
    if (!(e instanceof TalendException || e instanceof TDieException)) {
        if (virtualComponentName != null &&
currentComponent.indexOf(virtualComponentName + "_" ) == 0) {
            globalMap.put(virtualComponentName +
"_ERROR_MESSAGE", getExceptionCauseMessage(e));
        }
        globalMap.put(currentComponent + "_ERROR_MESSAGE",
getExceptionCauseMessage(e));
        System.err.println("Exception in component " + currentComponent +
" (" + jobName + ")");
    }
    if (!(e instanceof TDieException)) {
        if (e instanceof TalendException) {
            e.printStackTrace();
        } else {
            e.printStackTrace();
            e.printStackTrace(errorMessagePS);
            etl_project_process.this.exception = e;
        }
    }
}

```

```

        if (!(e instanceof TalendException)) {
            try {
                for (java.lang.reflect.Method m :
this.getClass().getEnclosingClass().getMethods()) {
                    if (m.getName().compareTo(currentComponent +
"_error") == 0) {
                        m.invoke(etl_project_process.this, new
Object[] { e, currentComponent, globalMap });
                        break;
                    }
                }
            } catch (Exception e) {
                this.e.printStackTrace();
            }
        }
    }
}

```

```

public void tFileInputDelimited_1_error(Exception exception, String errorComponent,
    final java.util.Map<String, Object> globalMap) throws TalendException {

    end_Hash.put(errorComponent, System.currentTimeMillis());

    status = "failure";

    tFileInputDelimited_1_onSubJobError(exception, errorComponent, globalMap);
}

```

```

        public void tMap_1_error(Exception exception, String errorComponent, final
java.util.Map<String, Object> globalMap)
            throws TalendException {

            end_Hash.put(errorComponent, System.currentTimeMillis());

            status = "failure";

            tFileInputDelimited_1_onSubJobError(exception, errorComponent, globalMap);
        }

        public void tFileOutputDelimited_1_error(Exception exception, String errorComponent,
            final java.util.Map<String, Object> globalMap) throws TalendException {

            end_Hash.put(errorComponent, System.currentTimeMillis());

            status = "failure";

            tFileInputDelimited_1_onSubJobError(exception, errorComponent, globalMap);
        }

        public void tLogRow_2_error(Exception exception, String errorComponent,
            final java.util.Map<String, Object> globalMap) throws TalendException {

            end_Hash.put(errorComponent, System.currentTimeMillis());

            status = "failure";

            tFileInputDelimited_1_onSubJobError(exception, errorComponent, globalMap);
        }

```

```

public void tFileInputJSON_1_error(Exception exception, String errorComponent,
    final java.util.Map<String, Object> globalMap) throws TalendException {

    end_Hash.put(errorComponent, System.currentTimeMillis());

    status = "failure";

    tFileInputJSON_1_onSubJobError(exception, errorComponent, globalMap);
}

```

```

public void tLogRow_1_error(Exception exception, String errorComponent,
    final java.util.Map<String, Object> globalMap) throws TalendException {

    end_Hash.put(errorComponent, System.currentTimeMillis());

    status = "failure";

    tFileInputJSON_1_onSubJobError(exception, errorComponent, globalMap);
}

```

```

public void tFileOutputExcel_1_error(Exception exception, String errorComponent,
    final java.util.Map<String, Object> globalMap) throws TalendException {

    end_Hash.put(errorComponent, System.currentTimeMillis());

    status = "failure";

    tFileInputJSON_1_onSubJobError(exception, errorComponent, globalMap);
}

```

```

public void tAdvancedHash_row4_error(Exception exception, String errorComponent,

```

```

        final java.util.Map<String, Object> globalMap) throws TalendException {

    end_Hash.put(errorComponent, System.currentTimeMillis());

    status = "failure";

    tFileInputJSON_1_onSubJobError(exception, errorComponent, globalMap);
}

    public void tFileInputDelimited_1_onSubJobError(Exception exception, String
errorComponent,

        final java.util.Map<String, Object> globalMap) throws TalendException {

        resumeUtil.addLog("SYSTEM_LOG", "NODE:" + errorComponent, "",
Thread.currentThread().getId() + "", "FATAL", "",
            exception.getMessage(),
ResumeUtil.getExceptionStackTrace(exception), "");

    }

    public void tFileInputJSON_1_onSubJobError(Exception exception, String errorComponent,

        final java.util.Map<String, Object> globalMap) throws TalendException {

        resumeUtil.addLog("SYSTEM_LOG", "NODE:" + errorComponent, "",
Thread.currentThread().getId() + "", "FATAL", "",
            exception.getMessage(),
ResumeUtil.getExceptionStackTrace(exception), "");

    }

    public static class output_salStruct implements
routines.system.IPersistableRow<output_salStruct> {

        final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
new byte[0];

```

```
static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];

protected static final int DEFAULT_HASHCODE = 1;

protected static final int PRIME = 31;

protected int hashCode = DEFAULT_HASHCODE;

public boolean hashCodeDirty = true;


public String loopKey;


public Integer work_year;


public Integer getWork_year() {
    return this.work_year;
}


public String experience_level;


public String getExperience_level() {
    return this.experience_level;
}


public String employment_type;


public String getEmployment_type() {
    return this.employment_type;
}


public String job_title;


public String getJob_title() {
    return this.job_title;
}
```

```
public Integer salary_in_usd;
```

```
public Integer getSalary_in_usd() {  
    return this.salary_in_usd;  
}
```

```
public String employee_residence;
```

```
public String getEmployee_residence() {  
    return this.employee_residence;  
}
```

```
public Integer remote_ratio;
```

```
public Integer getRemote_ratio() {  
    return this.remote_ratio;  
}
```

```
public String company_location;
```

```
public String getCompany_location() {  
    return this.company_location;  
}
```

```
public Character company_size;
```

```
public Character getCompany_size() {  
    return this.company_size;  
}
```

@Override

```
public int hashCode() {  
    if (this.hashCodeDirty) {  
        final int prime = PRIME;  
        int result = DEFAULT_HASHCODE;  
  
        result = prime * result + ((this.work_year == null) ? 0 :  
this.work_year.hashCode());  
  
        this.hashCode = result;  
        this.hashCodeDirty = false;  
    }  
    return this.hashCode;  
}
```

@Override

```
public boolean equals(Object obj) {  
    if (this == obj)  
        return true;  
    if (obj == null)  
        return false;  
    if (getClass() != obj.getClass())  
        return false;  
    final output_salStruct other = (output_salStruct) obj;  
  
    if (this.work_year == null) {  
        if (other.work_year != null)  
            return false;  
  
    } else if (!this.work_year.equals(other.work_year))
```



```

        return false;

    return true;
}

public void copyDataTo(output_salStruct other) {

    other.work_year = this.work_year;
    other.experience_level = this.experience_level;
    other.employment_type = this.employment_type;
    other.job_title = this.job_title;
    other.salary_in_usd = this.salary_in_usd;
    other.employee_residence = this.employee_residence;
    other.remote_ratio = this.remote_ratio;
    other.company_location = this.company_location;
    other.company_size = this.company_size;

}

public void copyKeysDataTo(output_salStruct other) {

    other.work_year = this.work_year;

}

private Integer readInteger(ObjectInputStream dis) throws IOException {
    Integer intReturn;
    int length = 0;
    length = dis.readByte();
    if (length == -1) {
        intReturn = null;
    }
}

```

```

        } else {
            intReturn = dis.readInt();
        }
        return intReturn;
    }

```

```

private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws
IOException {
    Integer intReturn;
    int length = 0;
    length = dis.readByte();
    if (length == -1) {
        intReturn = null;
    } else {
        intReturn = dis.readInt();
    }
    return intReturn;
}

```

```

private void writeInteger(Integer intNum, ObjectOutputStream dos) throws
IOException {
    if (intNum == null) {
        dos.writeByte(-1);
    } else {
        dos.writeByte(0);
        dos.writeInt(intNum);
    }
}

```

```

private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller
marshaller) throws IOException {
    if (intNum == null) {

```

```

        marshaller.writeByte(-1);
    } else {
        marshaller.writeByte(0);
        marshaller.writeInt(intNum);
    }
}

```

```

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;
    int length = 0;
    length = dis.readInt();
    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];
            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];
            }
        }

        dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);
        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }
    return strReturn;
}

```

```

        private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

            String strReturn = null;

            int length = 0;

            length = unmarshaller.readInt();

            if (length == -1) {

                strReturn = null;

            } else {

                if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {

                    if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                        commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

                    } else {

                        commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

                    }

                }

                unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

                strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);

            }

            return strReturn;

        }

        private void writeString(String str, ObjectOutputStream dos) throws IOException {

            if (str == null) {

                dos.writeInt(-1);

            } else {

                byte[] byteArray = str.getBytes(utf8Charset);

                dos.writeInt(byteArray.length);

                dos.write(byteArray);

            }

        }

```

```

        }
    }

    private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {
        if (str == null) {
            marshaller.writeInt(-1);
        } else {
            byte[] byteArray = str.getBytes(utf8Charset);
            marshaller.writeInt(byteArray.length);
            marshaller.write(byteArray);
        }
    }

    public void readData(ObjectInputStream dis) {

        synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
        {

            try {

                int length = 0;

                this.work_year = readInteger(dis);

                this.experience_level = readString(dis);

                this.employment_type = readString(dis);

                this.job_title = readString(dis);

                this.salary_in_usd = readInteger(dis);

```

```

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

}

}

public void readData(org.jboss.marshalling.Unmarshaller dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
{

    try {

        int length = 0;

```

```
        this.work_year = readInteger(dis);

        this.experience_level = readString(dis);

        this.employment_type = readString(dis);

        this.job_title = readString(dis);

        this.salary_in_usd = readInteger(dis);

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

```
}
```

```
public void writeData(ObjectOutputStream dos) {  
    try {  
  
        // Integer  
  
        writeInteger(this.work_year, dos);  
  
        // String  
  
        writeString(this.experience_level, dos);  
  
        // String  
  
        writeString(this.employment_type, dos);  
  
        // String  
  
        writeString(this.job_title, dos);  
  
        // Integer  
  
        writeInteger(this.salary_in_usd, dos);  
  
        // String  
  
        writeString(this.employee_residence, dos);  
  
        // Integer
```



```

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public void writeData(org.jboss.marshalling.Marshaller dos) {
    try {

        // Integer

        writeInteger(this.work_year, dos);

        // String

        writeString(this.experience_level, dos);

```

```
// String
```

```
writeString(this.employment_type, dos);
```

```
// String
```

```
writeString(this.job_title, dos);
```

```
// Integer
```

```
writeInteger(this.salary_in_usd, dos);
```

```
// String
```

```
writeString(this.employee_residence, dos);
```

```
// Integer
```

```
writeInteger(this.remote_ratio, dos);
```

```
// String
```

```
writeString(this.company_location, dos);
```

```
// Character
```

```
if (this.company_size == null) {
```

```
    dos.writeByte(-1);
```

```
} else {
```

```
    dos.writeByte(0);
```

```

        dos.writeChar(this.company_size);
    }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

public String toString() {

    StringBuilder sb = new StringBuilder();
    sb.append(super.toString());
    sb.append("[");
    sb.append("work_year=" + String.valueOf(work_year));
    sb.append(",experience_level=" + experience_level);
    sb.append(",employment_type=" + employment_type);
    sb.append(",job_title=" + job_title);
    sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));
    sb.append(",employee_residence=" + employee_residence);
    sb.append(",remote_ratio=" + String.valueOf(remote_ratio));
    sb.append(",company_location=" + company_location);
    sb.append(",company_size=" + String.valueOf(company_size));
    sb.append("]");

    return sb.toString();
}

/**
 * Compare keys
 */

```

```

public int compareTo(output_salStruct other) {

    int returnValue = -1;

    returnValue = checkNullsAndCompare(this.work_year, other.work_year);
    if (returnValue != 0) {
        return returnValue;
    }

    return returnValue;
}

private int checkNullsAndCompare(Object object1, Object object2) {
    int returnValue = 0;
    if (object1 instanceof Comparable && object2 instanceof Comparable) {
        returnValue = ((Comparable) object1).compareTo(object2);
    } else if (object1 != null && object2 != null) {
        returnValue = compareStrings(object1.toString(), object2.toString());
    } else if (object1 == null && object2 != null) {
        returnValue = 1;
    } else if (object1 != null && object2 == null) {
        returnValue = -1;
    } else {
        returnValue = 0;
    }

    return returnValue;
}

private int compareStrings(String string1, String string2) {
    return string1.compareTo(string2);
}

```

```
}
```

```
}
```

```
public static class rejectedStruct implements
routines.system.IPersistableRow<rejectedStruct> {

    final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
new byte[0];

    static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];

    protected static final int DEFAULT_HASHCODE = 1;

    protected static final int PRIME = 31;

    protected int hashCode = DEFAULT_HASHCODE;

    public boolean hashCodeDirty = true;


    public String loopKey;


    public Integer work_year;


    public Integer getWork_year() {

        return this.work_year;

    }


    public String experience_level;


    public String getExperience_level() {

        return this.experience_level;

    }


    public String employment_type;


    public String getEmployment_type() {

        return this.employment_type;

    }

}
```

```
}
```

```
public String job_title;
```

```
public String getJob_title() {  
    return this.job_title;  
}
```

```
public Integer salary_in_usd;
```

```
public Integer getSalary_in_usd() {  
    return this.salary_in_usd;  
}
```

```
public String employee_residence;
```

```
public String getEmployee_residence() {  
    return this.employee_residence;  
}
```

```
public Integer remote_ratio;
```

```
public Integer getRemote_ratio() {  
    return this.remote_ratio;  
}
```

```
public String company_location;
```

```
public String getCompany_location() {  
    return this.company_location;  
}
```

```
public Character company_size;
```

```
public Character getCompany_size() {  
    return this.company_size;  
}
```

```
@Override
```

```
public int hashCode() {  
    if (this.hashCodeDirty) {  
        final int prime = PRIME;  
        int result = DEFAULT_HASHCODE;  
  
        result = prime * result + ((this.work_year == null) ? 0 :  
this.work_year.hashCode());  
  
        this.hashCode = result;  
        this.hashCodeDirty = false;  
    }  
    return this.hashCode;  
}
```

```
@Override
```

```
public boolean equals(Object obj) {  
    if (this == obj)  
        return true;  
    if (obj == null)  
        return false;  
    if (getClass() != obj.getClass())  
        return false;  
    final rejectedStruct other = (rejectedStruct) obj;
```

```
        if (this.work_year == null) {
            if (other.work_year != null)
                return false;

        } else if (!this.work_year.equals(other.work_year))

            return false;

        return true;
    }

    public void copyDataTo(rejectedStruct other) {

        other.work_year = this.work_year;
        other.experience_level = this.experience_level;
        other.employment_type = this.employment_type;
        other.job_title = this.job_title;
        other.salary_in_usd = this.salary_in_usd;
        other.employee_residence = this.employee_residence;
        other.remote_ratio = this.remote_ratio;
        other.company_location = this.company_location;
        other.company_size = this.company_size;

    }

    public void copyKeysDataTo(rejectedStruct other) {

        other.work_year = this.work_year;

    }
```



```

private Integer readInteger(ObjectInputStream dis) throws IOException {
    Integer intReturn;
    int length = 0;
    length = dis.readByte();
    if (length == -1) {
        intReturn = null;
    } else {
        intReturn = dis.readInt();
    }
    return intReturn;
}

```

```

private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws
IOException {
    Integer intReturn;
    int length = 0;
    length = dis.readByte();
    if (length == -1) {
        intReturn = null;
    } else {
        intReturn = dis.readInt();
    }
    return intReturn;
}

```

```

private void writeInteger(Integer intNum, ObjectOutputStream dos) throws
IOException {
    if (intNum == null) {
        dos.writeByte(-1);
    } else {
        dos.writeByte(0);
    }
}

```

```

        dos.writeInt(intNum);
    }
}

```

```

private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller
marshaller) throws IOException {
    if (intNum == null) {
        marshaller.writeByte(-1);
    } else {
        marshaller.writeByte(0);
        marshaller.writeInt(intNum);
    }
}

```

```

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;
    int length = 0;
    length = dis.readInt();
    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];
            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];
            }
        }
    }
}

```

```

        dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

    String strReturn = null;

    int length = 0;

    length = unmarshaller.readInt();

    if (length == -1) {

        strReturn = null;

    } else {

        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {

            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

            }

        }

        unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

```

```

private void writeString(String str, ObjectOutputStream dos) throws IOException {
    if (str == null) {
        dos.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        dos.writeInt(byteArray.length);
        dos.write(byteArray);
    }
}

```

```

private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {
    if (str == null) {
        marshaller.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        marshaller.writeInt(byteArray.length);
        marshaller.write(byteArray);
    }
}

```

```

public void readData(ObjectInputStream dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

```

```
        this.experience_level = readString(dis);

        this.employment_type = readString(dis);

        this.job_title = readString(dis);

        this.salary_in_usd = readInteger(dis);

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

}
```

```
public void readData(org.jboss.marshalling.Unmarshaller dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)

    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

            this.experience_level = readString(dis);

            this.employment_type = readString(dis);

            this.job_title = readString(dis);

            this.salary_in_usd = readInteger(dis);

            this.employee_residence = readString(dis);

            this.remote_ratio = readInteger(dis);

            this.company_location = readString(dis);

            length = dis.readByte();
            if (length == -1) {
                this.company_size = null;
            } else {
                this.company_size = dis.readChar();
            }
        }
    }
}
```

```
        } catch (IOException e) {  
            throw new RuntimeException(e);  
        }  
    }  
}
```

```
public void writeData(ObjectOutputStream dos) {  
    try {  
  
        // Integer  
  
        writeInteger(this.work_year, dos);  
  
        // String  
  
        writeString(this.experience_level, dos);  
  
        // String  
  
        writeString(this.employment_type, dos);  
  
        // String  
  
        writeString(this.job_title, dos);  
  
        // Integer
```

```

        writeInteger(this.salary_in_usd, dos);

        // String

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public void writeData(org.jboss.marshalling.Marshaller dos) {
    try {

```



```
// Integer
```

```
writeInteger(this.work_year, dos);
```

```
// String
```

```
writeString(this.experience_level, dos);
```

```
// String
```

```
writeString(this.employment_type, dos);
```

```
// String
```

```
writeString(this.job_title, dos);
```

```
// Integer
```

```
writeInteger(this.salary_in_usd, dos);
```

```
// String
```

```
writeString(this.employee_residence, dos);
```

```
// Integer
```

```
writeInteger(this.remote_ratio, dos);
```

```
// String
```

```

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public String toString() {

    StringBuilder sb = new StringBuilder();
    sb.append(super.toString());
    sb.append("[");
    sb.append("work_year=" + String.valueOf(work_year));
    sb.append(",experience_level=" + experience_level);
    sb.append(",employment_type=" + employment_type);
    sb.append(",job_title=" + job_title);
    sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));
    sb.append(",employee_residence=" + employee_residence);
    sb.append(",remote_ratio=" + String.valueOf(remote_ratio));
    sb.append(",company_location=" + company_location);
    sb.append(",company_size=" + String.valueOf(company_size));

```

```

        sb.append("]");

        return sb.toString();
    }

    /**
     * Compare keys
     */
    public int compareTo(rejectedStruct other) {

        int returnValue = -1;

        returnValue = checkNullsAndCompare(this.work_year, other.work_year);
        if (returnValue != 0) {
            return returnValue;
        }

        return returnValue;
    }

    private int checkNullsAndCompare(Object object1, Object object2) {
        int returnValue = 0;
        if (object1 instanceof Comparable && object2 instanceof Comparable) {
            returnValue = ((Comparable) object1).compareTo(object2);
        } else if (object1 != null && object2 != null) {
            returnValue = compareStrings(object1.toString(), object2.toString());
        } else if (object1 == null && object2 != null) {
            returnValue = 1;
        } else if (object1 != null && object2 == null) {
            returnValue = -1;
        } else {

```

```

        returnValue = 0;
    }

    return returnValue;
}

private int compareStrings(String string1, String string2) {
    return string1.compareTo(string2);
}

}

public static class row3Struct implements routines.system.IPersistableRow<row3Struct> {
    final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
new byte[0];
    static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];

    public Integer work_year;

    public Integer getWork_year() {
        return this.work_year;
    }

    public String experience_level;

    public String getExperience_level() {
        return this.experience_level;
    }

    public String employment_type;

```

```
public String getEmployment_type() {  
    return this.employment_type;  
}  
  
public String job_title;  
  
public String getJob_title() {  
    return this.job_title;  
}  
  
public Integer salary;  
  
public Integer getSalary() {  
    return this.salary;  
}  
  
public String salary_currency;  
  
public String getSalary_currency() {  
    return this.salary_currency;  
}  
  
public Integer salary_in_usd;  
  
public Integer getSalary_in_usd() {  
    return this.salary_in_usd;  
}  
  
public String employee_residence;  
  
public String getEmployee_residence() {
```

```
        return this.employee_residence;
    }
}
```

```
public Integer remote_ratio;
```

```
public Integer getRemote_ratio() {
    return this.remote_ratio;
}
```

```
public String company_location;
```

```
public String getCompany_location() {
    return this.company_location;
}
```

```
public Character company_size;
```

```
public Character getCompany_size() {
    return this.company_size;
}
```

```
private Integer readInteger(ObjectInputStream dis) throws IOException {
    Integer intReturn;
    int length = 0;
    length = dis.readByte();
    if (length == -1) {
        intReturn = null;
    } else {
        intReturn = dis.readInt();
    }
    return intReturn;
}
```

```
}
```

```
private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws  
IOException {  
  
    Integer intReturn;  
  
    int length = 0;  
    length = dis.readByte();  
    if (length == -1) {  
        intReturn = null;  
    } else {  
        intReturn = dis.readInt();  
    }  
    return intReturn;  
}
```

```
private void writeInteger(Integer intNum, ObjectOutputStream dos) throws  
IOException {  
  
    if (intNum == null) {  
        dos.writeByte(-1);  
    } else {  
        dos.writeByte(0);  
        dos.writeInt(intNum);  
    }  
}
```

```
private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller  
marshaller) throws IOException {  
  
    if (intNum == null) {  
        marshaller.writeByte(-1);  
    } else {  
        marshaller.writeByte(0);  
        marshaller.writeInt(intNum);  
    }  
}
```

```

    }
}

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;
    int length = 0;
    length = dis.readInt();
    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

            }

        }

        dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

    String strReturn = null;
    int length = 0;
    length = unmarshaller.readInt();

```



```

        if (length == -1) {
            strReturn = null;
        } else {
            if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
                if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

                } else {

                    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

                }

            }

            unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

            strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
        }

        return strReturn;
    }

```

```

private void writeString(String str, ObjectOutputStream dos) throws IOException {
    if (str == null) {
        dos.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        dos.writeInt(byteArray.length);
        dos.write(byteArray);
    }
}

```

```

private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {

```

```
        if (str == null) {
            marshaller.writeInt(-1);
        } else {
            byte[] byteArray = str.getBytes(utf8Charset);
            marshaller.writeInt(byteArray.length);
            marshaller.write(byteArray);
        }
    }
}
```

```
public void readData(ObjectInputStream dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

            this.experience_level = readString(dis);

            this.employment_type = readString(dis);

            this.job_title = readString(dis);

            this.salary = readInteger(dis);

            this.salary_currency = readString(dis);

            this.salary_in_usd = readInteger(dis);
```

```

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

}

}

public void readData(org.jboss.marshalling.Unmarshaller dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
{

    try {

        int length = 0;

```

```
this.work_year = readInteger(dis);

this.experience_level = readString(dis);

this.employment_type = readString(dis);

this.job_title = readString(dis);

this.salary = readInteger(dis);

this.salary_currency = readString(dis);

this.salary_in_usd = readInteger(dis);

this.employee_residence = readString(dis);

this.remote_ratio = readInteger(dis);

this.company_location = readString(dis);

length = dis.readByte();
if (length == -1) {
    this.company_size = null;
} else {
    this.company_size = dis.readChar();
}

} catch (IOException e) {
    throw new RuntimeException(e);
}
```

```
    }

    }

}

public void writeData(ObjectOutputStream dos) {
    try {

        // Integer

        writeInteger(this.work_year, dos);

        // String

        writeString(this.experience_level, dos);

        // String

        writeString(this.employment_type, dos);

        // String

        writeString(this.job_title, dos);

        // Integer

        writeInteger(this.salary, dos);

        // String
```

```
        writeString(this.salary_currency, dos);

        // Integer

        writeInteger(this.salary_in_usd, dos);

        // String

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

```
}
```

```
public void writeData(org.jboss.marshalling.Marshaller dos) {
```

```
    try {
```

```
        // Integer
```

```
        writeInteger(this.work_year, dos);
```

```
        // String
```

```
        writeString(this.experience_level, dos);
```

```
        // String
```

```
        writeString(this.employment_type, dos);
```

```
        // String
```

```
        writeString(this.job_title, dos);
```

```
        // Integer
```

```
        writeInteger(this.salary, dos);
```

```
        // String
```

```
        writeString(this.salary_currency, dos);
```

```
        // Integer
```

```
        writeInteger(this.salary_in_usd, dos);

        // String

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public String toString() {
```



```

        StringBuilder sb = new StringBuilder();

        sb.append(super.toString());

        sb.append("[");

        sb.append("work_year=" + String.valueOf(work_year));

        sb.append(",experience_level=" + experience_level);

        sb.append(",employment_type=" + employment_type);

        sb.append(",job_title=" + job_title);

        sb.append(",salary=" + String.valueOf(salary));

        sb.append(",salary_currency=" + salary_currency);

        sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));

        sb.append(",employee_residence=" + employee_residence);

        sb.append(",remote_ratio=" + String.valueOf(remote_ratio));

        sb.append(",company_location=" + company_location);

        sb.append(",company_size=" + String.valueOf(company_size));

        sb.append("]");

        return sb.toString();
    }

    /**
     * Compare keys
     */
    public int compareTo(row3Struct other) {

        int returnValue = -1;

        return returnValue;
    }

    private int checkNullsAndCompare(Object object1, Object object2) {
        int returnValue = 0;

```

```

        if (object1 instanceof Comparable && object2 instanceof Comparable) {
            returnValue = ((Comparable) object1).compareTo(object2);
        } else if (object1 != null && object2 != null) {
            returnValue = compareStrings(object1.toString(), object2.toString());
        } else if (object1 == null && object2 != null) {
            returnValue = 1;
        } else if (object1 != null && object2 == null) {
            returnValue = -1;
        } else {
            returnValue = 0;
        }

        return returnValue;
    }

    private int compareStrings(String string1, String string2) {
        return string1.compareTo(string2);
    }
}

```

```

public static class after_tFileInputDelimited_1Struct
    implements
    routines.system.IPersistableRow<after_tFileInputDelimited_1Struct> {
    final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
    new byte[0];

    static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];
    protected static final int DEFAULT_HASHCODE = 1;
    protected static final int PRIME = 31;
    protected int hashCode = DEFAULT_HASHCODE;
    public boolean hashCodeDirty = true;
}

```

```
public String loopKey;
```

```
public Integer work_year;
```

```
public Integer getWork_year() {  
    return this.work_year;  
}
```

```
public String experience_level;
```

```
public String getExperience_level() {  
    return this.experience_level;  
}
```

```
public String employment_type;
```

```
public String getEmployment_type() {  
    return this.employment_type;  
}
```

```
public String job_title;
```

```
public String getJob_title() {  
    return this.job_title;  
}
```

```
public Integer salary;
```

```
public Integer getSalary() {  
    return this.salary;  
}
```

```
public String salary_currency;
```

```
public String getSalary_currency() {  
    return this.salary_currency;  
}
```

```
public Integer salary_in_usd;
```

```
public Integer getSalary_in_usd() {  
    return this.salary_in_usd;  
}
```

```
public String employee_residence;
```

```
public String getEmployee_residence() {  
    return this.employee_residence;  
}
```

```
public Integer remote_ratio;
```

```
public Integer getRemote_ratio() {  
    return this.remote_ratio;  
}
```

```
public String company_location;
```

```
public String getCompany_location() {  
    return this.company_location;  
}
```

```

public Character company_size;

public Character getCompany_size() {
    return this.company_size;
}

@Override
public int hashCode() {
    if (this.hashCodeDirty) {
        final int prime = PRIME;
        int result = DEFAULT_HASHCODE;

        result = prime * result + ((this.work_year == null) ? 0 :
this.work_year.hashCode());

        this.hashCode = result;
        this.hashCodeDirty = false;
    }
    return this.hashCode;
}

@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;

    final after_tFileInputDelimited_1Struct other =
(after_tFileInputDelimited_1Struct) obj;

```

```

        if (this.work_year == null) {
            if (other.work_year != null)
                return false;

        } else if (!this.work_year.equals(other.work_year))

            return false;

        return true;
    }

    public void copyDataTo(after_tFileInputDelimited_1Struct other) {

        other.work_year = this.work_year;
        other.experience_level = this.experience_level;
        other.employment_type = this.employment_type;
        other.job_title = this.job_title;
        other.salary = this.salary;
        other.salary_currency = this.salary_currency;
        other.salary_in_usd = this.salary_in_usd;
        other.employee_residence = this.employee_residence;
        other.remote_ratio = this.remote_ratio;
        other.company_location = this.company_location;
        other.company_size = this.company_size;

    }

    public void copyKeysDataTo(after_tFileInputDelimited_1Struct other) {

        other.work_year = this.work_year;

```

```
}
```

```
private Integer readInteger(ObjectInputStream dis) throws IOException {
```

```
    Integer intReturn;
```

```
    int length = 0;
```

```
    length = dis.readByte();
```

```
    if (length == -1) {
```

```
        intReturn = null;
```

```
    } else {
```

```
        intReturn = dis.readInt();
```

```
    }
```

```
    return intReturn;
```

```
}
```

```
private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws  
IOException {
```

```
    Integer intReturn;
```

```
    int length = 0;
```

```
    length = dis.readByte();
```

```
    if (length == -1) {
```

```
        intReturn = null;
```

```
    } else {
```

```
        intReturn = dis.readInt();
```

```
    }
```

```
    return intReturn;
```

```
}
```

```
private void writeInteger(Integer intNum, ObjectOutputStream dos) throws  
IOException {
```

```
    if (intNum == null) {
```

```
        dos.writeByte(-1);
```

```
    } else {
```

```

        dos.writeByte(0);
        dos.writeInt(intNum);
    }
}

```

```

private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller
marshaller) throws IOException {
    if (intNum == null) {
        marshaller.writeByte(-1);
    } else {
        marshaller.writeByte(0);
        marshaller.writeInt(intNum);
    }
}

```

```

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;
    int length = 0;
    length = dis.readInt();
    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];
            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];
            }
        }
    }
}

```



```

        dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

    String strReturn = null;

    int length = 0;

    length = unmarshaller.readInt();

    if (length == -1) {

        strReturn = null;

    } else {

        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {

            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

            }

        }

        unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

```

```

private void writeString(String str, ObjectOutputStream dos) throws IOException {
    if (str == null) {
        dos.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        dos.writeInt(byteArray.length);
        dos.write(byteArray);
    }
}

```

```

private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {
    if (str == null) {
        marshaller.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        marshaller.writeInt(byteArray.length);
        marshaller.write(byteArray);
    }
}

```

```

public void readData(ObjectInputStream dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

```

```
        this.experience_level = readString(dis);

        this.employment_type = readString(dis);

        this.job_title = readString(dis);

        this.salary = readInteger(dis);

        this.salary_currency = readString(dis);

        this.salary_in_usd = readInteger(dis);

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

```
}
```

```
}
```

```
public void readData(org.jboss.marshalling.Unmarshaller dis) {
```

```
    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)  
{
```

```
    try {
```

```
        int length = 0;
```

```
        this.work_year = readInteger(dis);
```

```
        this.experience_level = readString(dis);
```

```
        this.employment_type = readString(dis);
```

```
        this.job_title = readString(dis);
```

```
        this.salary = readInteger(dis);
```

```
        this.salary_currency = readString(dis);
```

```
        this.salary_in_usd = readInteger(dis);
```

```
        this.employee_residence = readString(dis);
```

```
        this.remote_ratio = readInteger(dis);
```

```

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

}

public void writeData(ObjectOutputStream dos) {

    // Integer

    writeInteger(this.work_year, dos);

    // String

    writeString(this.experience_level, dos);

    // String

```

```
writeString(this.employment_type, dos);
```

```
// String
```

```
writeString(this.job_title, dos);
```

```
// Integer
```

```
writeInteger(this.salary, dos);
```

```
// String
```

```
writeString(this.salary_currency, dos);
```

```
// Integer
```

```
writeInteger(this.salary_in_usd, dos);
```

```
// String
```

```
writeString(this.employee_residence, dos);
```

```
// Integer
```

```
writeInteger(this.remote_ratio, dos);
```

```
// String
```

```
writeString(this.company_location, dos);
```

```
// Character
```

```

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public void writeData(org.jboss.marshalling.Marshaller dos) {
    try {

        // Integer

        writeInteger(this.work_year, dos);

        // String

        writeString(this.experience_level, dos);

        // String

        writeString(this.employment_type, dos);

        // String

```

```
writeString(this.job_title, dos);

// Integer

writeInteger(this.salary, dos);

// String

writeString(this.salary_currency, dos);

// Integer

writeInteger(this.salary_in_usd, dos);

// String

writeString(this.employee_residence, dos);

// Integer

writeInteger(this.remote_ratio, dos);

// String

writeString(this.company_location, dos);

// Character

if (this.company_size == null) {
    dos.writeByte(-1);
} else {
```



```

        dos.writeByte(0);

        dos.writeChar(this.company_size);
    }

} catch (IOException e) {
    throw new RuntimeException(e);
}

}

public String toString() {

    StringBuilder sb = new StringBuilder();
    sb.append(super.toString());
    sb.append("[");
    sb.append("work_year=" + String.valueOf(work_year));
    sb.append(",experience_level=" + experience_level);
    sb.append(",employment_type=" + employment_type);
    sb.append(",job_title=" + job_title);
    sb.append(",salary=" + String.valueOf(salary));
    sb.append(",salary_currency=" + salary_currency);
    sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));
    sb.append(",employee_residence=" + employee_residence);
    sb.append(",remote_ratio=" + String.valueOf(remote_ratio));
    sb.append(",company_location=" + company_location);
    sb.append(",company_size=" + String.valueOf(company_size));
    sb.append("]");

    return sb.toString();
}

```

```

/**
 * Compare keys
 */
public int compareTo(after_tFileInputDelimited_1Struct other) {

    int returnValue = -1;

    returnValue = checkNullsAndCompare(this.work_year, other.work_year);
    if (returnValue != 0) {
        return returnValue;
    }

    return returnValue;
}

private int checkNullsAndCompare(Object object1, Object object2) {
    int returnValue = 0;
    if (object1 instanceof Comparable && object2 instanceof Comparable) {
        returnValue = ((Comparable) object1).compareTo(object2);
    } else if (object1 != null && object2 != null) {
        returnValue = compareStrings(object1.toString(), object2.toString());
    } else if (object1 == null && object2 != null) {
        returnValue = 1;
    } else if (object1 != null && object2 == null) {
        returnValue = -1;
    } else {
        returnValue = 0;
    }

    return returnValue;
}

```

```

        private int compareStrings(String string1, String string2) {
            return string1.compareTo(string2);
        }
    }

    public void tFileInputDelimited_1Process(final java.util.Map<String, Object> globalMap)
throws TalendException {
        globalMap.put("tFileInputDelimited_1_SUBPROCESS_STATE", 0);

        final boolean execStat = this.execStat;

        String iterateld = "";

        String currentComponent = "";

        java.util.Map<String, Object> resourceMap = new java.util.HashMap<String,
Object>();

        try {
            // TDI-39566 avoid throwing an useless Exception
            boolean resumelt = true;

            if (globalResumeTicket == false && resumeEntryMethodName != null) {
                String currentMethodName = new
java.lang.Exception().getStackTrace()[0].getMethodName();
                resumelt = resumeEntryMethodName.equals(currentMethodName);
            }

            if (resumelt || globalResumeTicket) { // start the resume
                globalResumeTicket = true;

                tFileInputJSON_1Process(globalMap);
            }
        }
    }

```

```

row3Struct row3 = new row3Struct();

output_salStruct output_sal = new output_salStruct();

rejectedStruct rejected = new rejectedStruct();


/**
 * [tFileOutputDelimited_1 begin ] start
 */

ok_Hash.put("tFileOutputDelimited_1", false);
start_Hash.put("tFileOutputDelimited_1",
System.currentTimeMillis());

currentComponent = "tFileOutputDelimited_1";

int tos_count_tFileOutputDelimited_1 = 0;

String fileName_tFileOutputDelimited_1 = "";
fileName_tFileOutputDelimited_1 = (new java.io.File(
"C:/Program Files (x86)/TOS_DI-
8.0.1/studio/workspace/ETL_project/sal_outfile.csv"))
.getAbsolutePath().replace("\\", "/");

String fullName_tFileOutputDelimited_1 = null;
String extension_tFileOutputDelimited_1 = null;
String directory_tFileOutputDelimited_1 = null;
if ((fileName_tFileOutputDelimited_1.indexOf("/") != -1)) {
    if (fileName_tFileOutputDelimited_1.lastIndexOf(".") <
fileName_tFileOutputDelimited_1
        .lastIndexOf("/")) {
        fullName_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1;
        extension_tFileOutputDelimited_1 = "";
    } else {

```

```

        fullName_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1.substring(0,

        fileName_tFileOutputDelimited_1.lastIndexOf("."));

        extension_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1

        .substring(fileName_tFileOutputDelimited_1.lastIndexOf("."));

    }

    directory_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1.substring(0,

    fileName_tFileOutputDelimited_1.lastIndexOf("/"));

    } else {

        if (fileName_tFileOutputDelimited_1.lastIndexOf(".") != -1) {

            fullName_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1.substring(0,

            fileName_tFileOutputDelimited_1.lastIndexOf("."));

            extension_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1

            .substring(fileName_tFileOutputDelimited_1.lastIndexOf("."));

        } else {

            fullName_tFileOutputDelimited_1 =
fileName_tFileOutputDelimited_1;

            extension_tFileOutputDelimited_1 = "";

        }

        directory_tFileOutputDelimited_1 = "";

    }

    boolean isFileGenerated_tFileOutputDelimited_1 = true;

    java.io.File filetFileOutputDelimited_1 = new
java.io.File(fileName_tFileOutputDelimited_1);

    globalMap.put("tFileOutputDelimited_1_FILE_NAME",
fileName_tFileOutputDelimited_1);

    if (filetFileOutputDelimited_1.exists()) {

```

```

        throw new RuntimeException("The particular file \" +
filetFileOutputDelimited_1.getAbsolutePath()
                                + "\"" already exist. If you want to overwrite
the file, please uncheck the"
                                + "\"Throw an error if the file already exist\"
option in Advanced settings.");
    }
    int nb_line_tFileOutputDelimited_1 = 0;
    int splitedFileNo_tFileOutputDelimited_1 = 0;
    int currentRow_tFileOutputDelimited_1 = 0;

    final String OUT_DELIM_tFileOutputDelimited_1 = /** Start field
tFileOutputDelimited_1:FIELDSEPARATOR */
                                " , " /** End field
tFileOutputDelimited_1:FIELDSEPARATOR */
                                ;

    final String OUT_DELIM_ROWSEP_tFileOutputDelimited_1 = /**
                                * Start field
                                * tFileOutputDelimited_1:ROWSEPARATOR
                                */
                                "\n" /** End field
tFileOutputDelimited_1:ROWSEPARATOR */
                                ;

    // create directory only if not exists
    if (directory_tFileOutputDelimited_1 != null &&
directory_tFileOutputDelimited_1.trim().length() != 0) {
        java.io.File dir_tFileOutputDelimited_1 = new
java.io.File(directory_tFileOutputDelimited_1);
        if (!dir_tFileOutputDelimited_1.exists()) {
            dir_tFileOutputDelimited_1.mkdirs();

```

```

        }
    }

    // routines.system.Row

    java.io.Writer outtFileOutputDelimited_1 = null;

    java.io.File fileToDelete_tFileOutputDelimited_1 = new
java.io.File(fileName_tFileOutputDelimited_1);

    if (fileToDelete_tFileOutputDelimited_1.exists()) {
        fileToDelete_tFileOutputDelimited_1.delete();
    }

    outtFileOutputDelimited_1 = new java.io.BufferedWriter(new
java.io.OutputStreamWriter(
        new
java.io.FileOutputStream(fileName_tFileOutputDelimited_1, false), "ISO-8859-15"));

    if (filetFileOutputDelimited_1.length() == 0) {
        outtFileOutputDelimited_1.write("work_year");

        outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("experience_level");

        outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("employment_type");

        outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("job_title");

        outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("salary_in_usd");

        outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("employee_residence");

        outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);

```

```

        outtFileOutputDelimited_1.write("remote_ratio");

    outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("company_location");

    outtFileOutputDelimited_1.write(OUT_DELIM_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.write("company_size");

    outtFileOutputDelimited_1.write(OUT_DELIM_ROWSEP_tFileOutputDelimited_1);
        outtFileOutputDelimited_1.flush();
    }

    resourceMap.put("out_tFileOutputDelimited_1",
outtFileOutputDelimited_1);
    resourceMap.put("nb_line_tFileOutputDelimited_1",
nb_line_tFileOutputDelimited_1);

    /**
     * [tFileOutputDelimited_1 begin ] stop
     */

    /**
     * [tLogRow_2 begin ] start
     */

    ok_Hash.put("tLogRow_2", false);
    start_Hash.put("tLogRow_2", System.currentTimeMillis());

    currentComponent = "tLogRow_2";

    int tos_count_tLogRow_2 = 0;

    //////////////////////////////////

```



```

class Util_tLogRow_2 {

    String[] des_top = { ".", ".", "-", "+" };

    String[] des_head = { "|=", "=|", "-", "+" };

    String[] des_bottom = { "", "", "-", "+" };

    String name = "";

    java.util.List<String[]> list = new
java.util.ArrayList<String[]>();

    int[] colLengths = new int[9];

    public void addRow(String[] row) {

        for (int i = 0; i < 9; i++) {
            if (row[i] != null) {
                colLengths[i] =
Math.max(colLengths[i], row[i].length());
            }
        }
        list.add(row);
    }

    public void setTableName(String name) {

        this.name = name;
    }
}

```

```

public StringBuilder format() {

    StringBuilder sb = new StringBuilder();

    sb.append(print(des_top));

    int totals = 0;
    for (int i = 0; i < colLengths.length; i++) {
        totals = totals + colLengths[i];
    }

    // name
    sb.append("|");
    int k = 0;
    for (k = 0; k < (totals + 8 - name.length()) / 2; k++) {
        sb.append(' ');
    }
    sb.append(name);
    for (int i = 0; i < totals + 8 - name.length() - k; i++) {
        sb.append(' ');
    }
    sb.append("|\\n");

    // head and rows
    sb.append(print(des_head));
    for (int i = 0; i < list.size(); i++) {

        String[] row = list.get(i);

        java.util.Formatter formatter = new
java.util.Formatter(new StringBuilder());

```

```
StringBuilder sbformat = new StringBuilder();  
sbformat.append("| %1$-");  
sbformat.append(colLengths[0]);  
sbformat.append("s");
```

```
sbformat.append("| %2$-");  
sbformat.append(colLengths[1]);  
sbformat.append("s");
```

```
sbformat.append("| %3$-");  
sbformat.append(colLengths[2]);  
sbformat.append("s");
```

```
sbformat.append("| %4$-");  
sbformat.append(colLengths[3]);  
sbformat.append("s");
```

```
sbformat.append("| %5$-");  
sbformat.append(colLengths[4]);  
sbformat.append("s");
```

```
sbformat.append("| %6$-");  
sbformat.append(colLengths[5]);  
sbformat.append("s");
```

```
sbformat.append("| %7$-");  
sbformat.append(colLengths[6]);  
sbformat.append("s");
```

```
sbformat.append("| %8$-");
```

```

        sbformat.append(colLengths[7]);
        sbformat.append("s");

        sbformat.append("|%9$-");
        sbformat.append(colLengths[8]);
        sbformat.append("s");

        sbformat.append("|\\n");

        formatter.format(sbformat.toString(),
(Object[]) row);

        sb.append(formatter.toString());
        if (i == 0)
            sb.append(print(des_head)); // print
the head

    }

    // end
    sb.append(print(des_bottom));
    return sb;
}

private StringBuilder print(String[] fillChars) {
    StringBuilder sb = new StringBuilder();
    // first column
    sb.append(fillChars[0]);
    for (int i = 0; i < colLengths[0] - fillChars[0].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);

```

```

1; i++) {
    for (int i = 0; i < colLengths[1] - fillChars[3].length() +
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[2] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[3] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[4] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[5] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[6] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[7] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }

```

```

    }

    sb.append(fillChars[3]);

    // last column
    for (int i = 0; i < colLengths[8] - fillChars[1].length() +
1; i++) {

        sb.append(fillChars[2]);

    }

    sb.append(fillChars[1]);

    sb.append("\n");

    return sb;

}

public boolean isEmpty() {

    if (list.size() > 1)

        return false;

    return true;

}

}

Util_tLogRow_2 util_tLogRow_2 = new Util_tLogRow_2();

util_tLogRow_2.setTableName("tLogRow_2");

util_tLogRow_2.addRow(new String[] { "work_year",
"experience_level", "employment_type", "job_title",

        "salary_in_usd", "employee_residence",
"remote_ratio", "company_location", "company_size", });

StringBuilder strBuffer_tLogRow_2 = null;

int nb_line_tLogRow_2 = 0;

//////////

/**

 * [tLogRow_2 begin ] stop

 */

```

```

/**
 * [tMap_1 begin ] start
 */

ok_Hash.put("tMap_1", false);
start_Hash.put("tMap_1", System.currentTimeMillis());

currentComponent = "tMap_1";

int tos_count_tMap_1 = 0;

// #####

// # Lookup's keys initialization

        org.talend.designer.components.lookup.memory.AdvancedMemoryLookup<row4Struct>
tHash_Lookup_row4 =
(org.talend.designer.components.lookup.memory.AdvancedMemoryLookup<row4Struct>)
((org.talend.designer.components.lookup.memory.AdvancedMemoryLookup<row4Struct>)
globalMap

                                .get("tHash_Lookup_row4"));

row4Struct row4HashKey = new row4Struct();
row4Struct row4Default = new row4Struct();

// #####

// #####

// # Vars initialization

        class Var__tMap_1__Struct {

        }

        Var__tMap_1__Struct Var__tMap_1 = new Var__tMap_1__Struct();

// #####

```

```

// #####
// # Outputs initialization

        output_salStruct output_sal_tmp = new output_salStruct();
        rejectedStruct rejected_tmp = new rejectedStruct();

// #####

        /**
        * [tMap_1 begin ] stop
        */

        /**
        * [tFileInputDelimited_1 begin ] start
        */

        ok_Hash.put("tFileInputDelimited_1", false);
        start_Hash.put("tFileInputDelimited_1", System.currentTimeMillis());

        currentComponent = "tFileInputDelimited_1";

        int tos_count_tFileInputDelimited_1 = 0;

        final routines.system.RowState rowstate_tFileInputDelimited_1 =
new routines.system.RowState();

        int nb_line_tFileInputDelimited_1 = 0;
        org.talend.fileprocess.FileInputDelimited fid_tFileInputDelimited_1 =
null;

        int limit_tFileInputDelimited_1 = -1;
        try {

```



```

        Object filename_tFileInputDelimited_1 = "C:/Program Files
(x86)/TOS_DI-8.0.1/studio/workspace/ETL_project/salaries.csv";

        if (filename_tFileInputDelimited_1 instanceof
java.io.InputStream) {

                                int footer_value_tFileInputDelimited_1 = 0,
random_value_tFileInputDelimited_1 = -1;

                                if (footer_value_tFileInputDelimited_1 > 0 ||
random_value_tFileInputDelimited_1 > 0) {

                                        throw new java.lang.Exception(

                                                "When the input source is a
stream,footer and random shouldn't be bigger than 0.");

                                        }

                                }

                                try {

                                        fid_tFileInputDelimited_1 = new
org.talend.fileprocess.FileInputDelimited(

                                                "C:/Program Files (x86)/TOS_DI-
8.0.1/studio/workspace/ETL_project/salaries.csv",

                                                "US-ASCII", ",", "\n", false, 1, 0,
limit_tFileInputDelimited_1, -1, false);

                                } catch (java.lang.Exception e) {

                                        globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE", e.getMessage());

                                        System.err.println(e.getMessage());

                                }

                                while (fid_tFileInputDelimited_1 != null &&
fid_tFileInputDelimited_1.nextRecord()) {

                                        rowstate_tFileInputDelimited_1.reset();

                                        row3 = null;

```

```

        boolean whetherReject_tFileInputDelimited_1 =
false;

        row3 = new row3Struct();

        try {

            int

columnIndexWithD_tFileInputDelimited_1 = 0;

            String temp = "";

            columnIndexWithD_tFileInputDelimited_1 =
0;

            temp =
fid_tFileInputDelimited_1.get(columnIndexWithD_tFileInputDelimited_1);

            if (temp.length() > 0) {

                try {

                    row3.work_year =
ParserUtils.parseTo_Integer(temp);

                } catch (java.lang.Exception

ex_tFileInputDelimited_1) {

                    globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE",

                    ex_tFileInputDelimited_1.getMessage());

                    rowstate_tFileInputDelimited_1.setException(new RuntimeException(String.format(

                                                                    "Couldn't
parse value for column '%s' in '%s', value is '%s'. Details: %s",

                                                                    "work_year", "row3", temp, ex_tFileInputDelimited_1),

```

```

ex_tFileInputDelimited_1));

                                }

                                } else {

                                row3.work_year = null;

                                }

                                columnIndexWithD_tFileInputDelimited_1 =
1;

                                row3.experience_level =
fid_tFileInputDelimited_1

                                .get(columnIndexWithD_tFileInputDelimited_1);

                                columnIndexWithD_tFileInputDelimited_1 =
2;

                                row3.employment_type =
fid_tFileInputDelimited_1

                                .get(columnIndexWithD_tFileInputDelimited_1);

                                columnIndexWithD_tFileInputDelimited_1 =
3;

                                row3.job_title =
fid_tFileInputDelimited_1.get(columnIndexWithD_tFileInputDelimited_1);

                                columnIndexWithD_tFileInputDelimited_1 =
4;

```

```

                                temp =
fid_tFileInputDelimited_1.get(columnIndexWithD_tFileInputDelimited_1);
                                if (temp.length() > 0) {

                                try {

                                row3.salary =

ParserUtils.parseTo_Integer(temp);

                                } catch (java.lang.Exception
ex_tFileInputDelimited_1) {

                                globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE",

                                ex_tFileInputDelimited_1.getMessage());

                                rowstate_tFileInputDelimited_1.setException(new RuntimeException(String.format(
                                                                    "Couldn't
parse value for column '%s' in '%s', value is '%s'. Details: %s",
                                                                    "salary",
"row3", temp, ex_tFileInputDelimited_1),

                                ex_tFileInputDelimited_1));

                                }

                                } else {

                                row3.salary = null;

                                }

                                columnIndexWithD_tFileInputDelimited_1 =

5;

```

```

row3.salary_currency =
fid_tFileInputDelimited_1

        .get(columnIndexWithD_tFileInputDelimited_1);

columnIndexWithD_tFileInputDelimited_1 =
6;

temp =
fid_tFileInputDelimited_1.get(columnIndexWithD_tFileInputDelimited_1);
        if (temp.length() > 0) {

                try {

                        row3.salary_in_usd =
ParserUtils.parseTo_Integer(temp);

                } catch (java.lang.Exception
ex_tFileInputDelimited_1) {

                        globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE",

                                ex_tFileInputDelimited_1.getMessage());

                                rowstate_tFileInputDelimited_1.setException(new RuntimeException(String.format(
                                                                "Couldn't
parse value for column '%s' in '%s', value is '%s'. Details: %s",

                                "salary_in_usd", "row3", temp, ex_tFileInputDelimited_1),

                                ex_tFileInputDelimited_1));

                                }

                                } else {

```

```

row3.salary_in_usd = null;

    }

columnIndexWithD_tFileInputDelimited_1 =
7;

row3.employee_residence =
fid_tFileInputDelimited_1

.get(columnIndexWithD_tFileInputDelimited_1);

columnIndexWithD_tFileInputDelimited_1 =
8;

temp =
fid_tFileInputDelimited_1.get(columnIndexWithD_tFileInputDelimited_1);
if (temp.length() > 0) {

    try {

row3.remote_ratio =
ParserUtils.parseTo_Integer(temp);

    } catch (java.lang.Exception
ex_tFileInputDelimited_1) {

    globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE",

ex_tFileInputDelimited_1.getMessage());

rowstate_tFileInputDelimited_1.setException(new RuntimeException(String.format(
                                                                    "Couldn't
parse value for column '%s' in '%s', value is '%s'. Details: %s",

```

```

        "remote_ratio", "row3", temp, ex_tFileInputDelimited_1),

        ex_tFileInputDelimited_1));

    }

    } else {

        row3.remote_ratio = null;

    }

    columnIndexWithD_tFileInputDelimited_1 =

9;

    row3.company_location =

fid_tFileInputDelimited_1

        .get(columnIndexWithD_tFileInputDelimited_1);

    columnIndexWithD_tFileInputDelimited_1 =

10;

    temp =

fid_tFileInputDelimited_1.get(columnIndexWithD_tFileInputDelimited_1);

    if (temp.length() > 0) {

        try {

            row3.company_size =

ParserUtils.parseTo_Character(temp);

        } catch (java.lang.Exception

ex_tFileInputDelimited_1) {

```

```

globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE",

ex_tFileInputDelimited_1.getMessage());

rowstate_tFileInputDelimited_1.setException(new RuntimeException(String.format(
                                                                    "Couldn't
parse value for column '%s' in '%s', value is '%s'. Details: %s",

"company_size", "row3", temp, ex_tFileInputDelimited_1),

ex_tFileInputDelimited_1));

    }

    } else {

        row3.company_size = null;

    }

    if
(rowstate_tFileInputDelimited_1.getException() != null) {

        throw
rowstate_tFileInputDelimited_1.getException();

    }

    } catch (java.lang.Exception e) {

globalMap.put("tFileInputDelimited_1_ERROR_MESSAGE", e.getMessage());

        whetherReject_tFileInputDelimited_1 = true;

        System.err.println(e.getMessage());

        row3 = null;

```



```
}
```

```
/**
```

```
 * [tFileInputDelimited_1 begin ] stop
```

```
 */
```

```
/**
```

```
 * [tFileInputDelimited_1 main ] start
```

```
 */
```

```
currentComponent = "tFileInputDelimited_1";
```

```
tos_count_tFileInputDelimited_1++;
```

```
/**
```

```
 * [tFileInputDelimited_1 main ] stop
```

```
 */
```

```
/**
```

```
 * [tFileInputDelimited_1 process_data_begin ] start
```

```
 */
```

```
currentComponent = "tFileInputDelimited_1";
```

```
/**
```

```
 * [tFileInputDelimited_1 process_data_begin ] stop
```

```
 */
```

```
// Start of branch "row3"
```

```
if (row3 != null) {
```

```
    /**
```

```

* [tMap_1 main ] start
*/

currentComponent = "tMap_1";

boolean
hasCasePrimitiveKeyWithNull_tMap_1 = false;

// #####
// # Input tables (lookups)
boolean rejectedInnerJoin_tMap_1 = false;
boolean mainRowRejected_tMap_1 = false;

////////////////////////////////////
// Starting Lookup Table "row4"
////////////////////////////////////

boolean forceLooprow4 = false;

row4Struct row4ObjectFromLookup = null;

if (!rejectedInnerJoin_tMap_1) { //
G_TM_M_020

    hasCasePrimitiveKeyWithNull_tMap_1 = false;

    row4HashKey.work_year =
row3.work_year;

    row4HashKey.experience_level =
row3.experience_level;

```

row3.employment_type;	row4HashKey.employment_type =
row3.job_title;	row4HashKey.job_title =
	row4HashKey.salary = row3.salary;
row3.salary_currency;	row4HashKey.salary_currency =
row3.salary_in_usd;	row4HashKey.salary_in_usd =
row3.employee_residence;	row4HashKey.employee_residence =
row3.remote_ratio;	row4HashKey.remote_ratio =
row3.company_location;	row4HashKey.company_location =
row3.company_size;	row4HashKey.company_size =
	row4HashKey.hashCodeDirty = true;
tHash_Lookup_row4.lookup(row4HashKey);	
// G_TM_M_090	if (!tHash_Lookup_row4.hasNext()) {

```

true;

rejectedInnerJoin_tMap_1 =

} // G_TM_M_090

} // G_TM_M_020

if (tHash_Lookup_row4 != null &&
tHash_Lookup_row4.getCount(row4HashKey) > 1) { // G 071

// System.out.println("WARNING:
UNIQUE MATCH is configured for the lookup 'row4'

// and it contains more one result

from keys : row4.work_year = "" +

// row4HashKey.work_year + "",

row4.experience_level = "" +

// row4HashKey.experience_level +

"", row4.employment_type = "" +

// row4HashKey.employment_type +

"", row4.job_title = "" + row4HashKey.job_title

// + "", row4.salary = "" +

row4HashKey.salary + "", row4.salary_currency = "" +

// row4HashKey.salary_currency + "",

row4.salary_in_usd = "" +

// row4HashKey.salary_in_usd + "",

row4.employee_residence = "" +

//

row4HashKey.employee_residence + "", row4.remote_ratio = "" +

// row4HashKey.remote_ratio + "",

row4.company_location = "" +

// row4HashKey.company_location +

"", row4.company_size = "" +

// row4HashKey.company_size + "");

} // G 071

row4Struct row4 = null;

```

```

row4Struct fromLookup_row4 = null;
row4 = row4Default;

tHash_Lookup_row4.hasNext() { // G 099

    if (tHash_Lookup_row4 != null &&

        fromLookup_row4 =

tHash_Lookup_row4.next();

    } // G 099

    if (fromLookup_row4 != null) {
        row4 = fromLookup_row4;
    }

    // #####
    { // start of Var scope

        //

#####

        // # Vars tables

Var__tMap_1__Struct Var =

Var__tMap_1;// #####

        //

#####

        // # Output tables

        output_sal = null;
        rejected = null;

        if (!rejectedInnerJoin_tMap_1) {

```

```

// # Output table : 'output_sal'

output_sal_tmp.work_year =
row3.work_year;

output_sal_tmp.experience_level = row3.experience_level;

output_sal_tmp.employment_type = row3.employment_type;

output_sal_tmp.job_title =
row3.job_title;

output_sal_tmp.salary_in_usd = row3.salary_in_usd;

output_sal_tmp.employee_residence = row3.employee_residence;

output_sal_tmp.remote_ratio = row3.remote_ratio;

output_sal_tmp.company_location = row3.company_location;

output_sal_tmp.company_size = row3.company_size;

output_sal =
output_sal_tmp;

} // closing inner join bracket (1)

// ##### START REJECTS #####

// # Output reject table : 'rejected'
// # Filter conditions

if (rejectedInnerJoin_tMap_1) {

rejected_tmp.work_year =
row3.work_year;

rejected_tmp.experience_level = row3.experience_level;

rejected_tmp.employment_type = row3.employment_type;

rejected_tmp.job_title =
row3.job_title;

```

```

rejected_tmp.salary_in_usd
= row3.salary_in_usd;

rejected_tmp.employee_residence = row3.employee_residence;

rejected_tmp.remote_ratio
= row3.remote_ratio;

rejected_tmp.company_location = row3.company_location;

rejected_tmp.company_size
= row3.company_size;

rejected = rejected_tmp;
} // closing filter/reject

// #####

} // end of Var scope

rejectedInnerJoin_tMap_1 = false;

tos_count_tMap_1++;

/**
 * [tMap_1 main ] stop
 */

/**
 * [tMap_1 process_data_begin ] start
 */

currentComponent = "tMap_1";

/**
 * [tMap_1 process_data_begin ] stop
 */

```

```
// Start of branch "output_sal"
```

```
if (output_sal != null) {
```

```
/**
```

```
 * [tFileOutputDelimited_1 main ]
```

```
start
```

```
*/
```

```
currentComponent =
```

```
"tFileOutputDelimited_1";
```

```
StringBuilder
```

```
sb_tFileOutputDelimited_1 = new StringBuilder();
```

```
if (output_sal.work_year != null) {
```

```
    sb_tFileOutputDelimited_1.append(output_sal.work_year);
```

```
    }
```

```
    sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
```

```
    if (output_sal.experience_level !=
```

```
null) {
```

```
        sb_tFileOutputDelimited_1.append(output_sal.experience_level);
```

```
    }
```

```
    sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
```

```
    if (output_sal.employment_type !=
```

```
null) {
```

```
        sb_tFileOutputDelimited_1.append(output_sal.employment_type);
```

```
    }
```

```
    sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
```

```
    if (output_sal.job_title != null) {
```



```

sb_tFileOutputDelimited_1.append(output_sal.job_title);
                                }

sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
                                if (output_sal.salary_in_usd != null) {

sb_tFileOutputDelimited_1.append(output_sal.salary_in_usd);
                                }

sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
                                if (output_sal.employee_residence
!= null) {

sb_tFileOutputDelimited_1.append(output_sal.employee_residence);
                                }

sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
                                if (output_sal.remote_ratio != null) {

sb_tFileOutputDelimited_1.append(output_sal.remote_ratio);
                                }

sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
                                if (output_sal.company_location !=
null) {

sb_tFileOutputDelimited_1.append(output_sal.company_location);
                                }

sb_tFileOutputDelimited_1.append(OUT_DELIM_tFileOutputDelimited_1);
                                if (output_sal.company_size != null) {

sb_tFileOutputDelimited_1.append(output_sal.company_size);
                                }

```

```
sb_tFileOutputDelimited_1.append(OUT_DELIM_ROWSEP_tFileOutputDelimited_1);
```

```
nb_line_tFileOutputDelimited_1++;
```

```
resourceMap.put("nb_line_tFileOutputDelimited_1", nb_line_tFileOutputDelimited_1);
```

```
outtFileOutputDelimited_1.write(sb_tFileOutputDelimited_1.toString());
```

```
tos_count_tFileOutputDelimited_1++;
```

```
stop
```

```
process_data_begin ] start
```

```
"tFileOutputDelimited_1";
```

```
process_data_begin ] stop
```

```
process_data_end ] start
```

```
/**
```

```
 * [tFileOutputDelimited_1 main ]
```

```
 */
```

```
/**
```

```
 * [tFileOutputDelimited_1
```

```
 */
```

```
currentComponent =
```

```
/**
```

```
 * [tFileOutputDelimited_1
```

```
 */
```

```
/**
```

```
 * [tFileOutputDelimited_1
```

```
"tFileOutputDelimited_1";
```

```
process_data_end ] stop
```

```
// Start of branch "rejected"
```

```
////////////////////
```

```
String[9];
```

```
String.valueOf(rejected.work_year);
```

```
*/
```

```
currentComponent =
```

```
/**
```

```
* [tFileOutputDelimited_1
```

```
*/
```

```
} // End of branch "output_sal"
```

```
if (rejected != null) {
```

```
/**
```

```
* [tLogRow_2 main ] start
```

```
*/
```

```
currentComponent = "tLogRow_2";
```

```
String[] row_tLogRow_2 = new
```

```
if (rejected.work_year != null) { //
```

```
row_tLogRow_2[0] =
```

```
} //
```

```
{ //

String.valueOf(rejected.experience_level);

null) { //

String.valueOf(rejected.employment_type);

String.valueOf(rejected.job_title);

String.valueOf(rejected.salary_in_usd);

null) { //

String.valueOf(rejected.employee_residence);
```

```
if (rejected.experience_level != null)

    row_tLogRow_2[1] =

} //

if (rejected.employment_type !=

    row_tLogRow_2[2] =

} //

if (rejected.job_title != null) { //

    row_tLogRow_2[3] =

} //

if (rejected.salary_in_usd != null) { //

    row_tLogRow_2[4] =

} //

if (rejected.employee_residence !=

    row_tLogRow_2[5] =

} //

if (rejected.remote_ratio != null) { //
```

```

String.valueOf(rejected.remote_ratio);

row_tLogRow_2[6] =

} //

if (rejected.company_location !=

null) { //

row_tLogRow_2[7] =

String.valueOf(rejected.company_location);

} //

if (rejected.company_size != null) { //

row_tLogRow_2[8] =

String.valueOf(rejected.company_size);

} //

util_tLogRow_2.addRow(row_tLogRow_2);

nb_line_tLogRow_2++;

/////

/////

////////////////////

tos_count_tLogRow_2++;

/**
 * [tLogRow_2 main ] stop
 */

```

start

```
/**  
 * [tLogRow_2 process_data_begin ]  
  
 */
```

```
currentComponent = "tLogRow_2";
```

stop

```
/**  
 * [tLogRow_2 process_data_begin ]  
  
 */
```

start

```
/**  
 * [tLogRow_2 process_data_end ]  
  
 */
```

```
currentComponent = "tLogRow_2";
```

stop

```
/**  
 * [tLogRow_2 process_data_end ]  
  
 */
```

```
} // End of branch "rejected"
```

```
/**  
 * [tMap_1 process_data_end ] start  
 */
```

```
currentComponent = "tMap_1";
```

```

        /**
        * [tMap_1 process_data_end ] stop
        */

    } // End of branch "row3"

    /**
    * [tFileInputDelimited_1 process_data_end ] start
    */

    currentComponent = "tFileInputDelimited_1";

    /**
    * [tFileInputDelimited_1 process_data_end ] stop
    */

    /**
    * [tFileInputDelimited_1 end ] start
    */

    currentComponent = "tFileInputDelimited_1";

    }

    } finally {

        if (!((Object) ("C:/Program Files (x86)/TOS_DI-
8.0.1/studio/workspace/ETL_project/salaries.csv") instanceof java.io.InputStream)) {

            if (fid_tFileInputDelimited_1 != null) {

                fid_tFileInputDelimited_1.close();

            }

        }

        if (fid_tFileInputDelimited_1 != null) {

```

```

                                globalMap.put("tFileInputDelimited_1_NB_LINE",
fid_tFileInputDelimited_1.getRowNumber());

                                }
                                }

                                ok_Hash.put("tFileInputDelimited_1", true);
                                end_Hash.put("tFileInputDelimited_1", System.currentTimeMillis());

                                /**
                                * [tFileInputDelimited_1 end ] stop
                                */

                                /**
                                * [tMap_1 end ] start
                                */

                                currentComponent = "tMap_1";

                                // #####
                                // # Lookup hashes releasing
                                if (tHash_Lookup_row4 != null) {
                                        tHash_Lookup_row4.endGet();
                                }
                                globalMap.remove("tHash_Lookup_row4");

                                // #####

                                ok_Hash.put("tMap_1", true);
                                end_Hash.put("tMap_1", System.currentTimeMillis());

```



```

/**
 * [tMap_1 end ] stop
 */

/**
 * [tFileOutputDelimited_1 end ] start
 */

currentComponent = "tFileOutputDelimited_1";

if (outtFileOutputDelimited_1 != null) {
    outtFileOutputDelimited_1.flush();
    outtFileOutputDelimited_1.close();
}

globalMap.put("tFileOutputDelimited_1_NB_LINE",
nb_line_tFileOutputDelimited_1);
globalMap.put("tFileOutputDelimited_1_FILE_NAME",
fileName_tFileOutputDelimited_1);

resourceMap.put("finish_tFileOutputDelimited_1", true);

ok_Hash.put("tFileOutputDelimited_1", true);
end_Hash.put("tFileOutputDelimited_1",
System.currentTimeMillis());

/**
 * [tFileOutputDelimited_1 end ] stop
 */

/**
 * [tLogRow_2 end ] start

```

```

*/

currentComponent = "tLogRow_2";

/////

java.io.PrintStream consoleOut_tLogRow_2 = null;
if (globalMap.get("tLogRow_CONSOLE") != null) {
    consoleOut_tLogRow_2 = (java.io.PrintStream)
globalMap.get("tLogRow_CONSOLE");
} else {
    consoleOut_tLogRow_2 = new java.io.PrintStream(new
java.io.BufferedOutputStream(System.out));
    globalMap.put("tLogRow_CONSOLE",
consoleOut_tLogRow_2);
}

consoleOut_tLogRow_2.println(util_tLogRow_2.format().toString());
consoleOut_tLogRow_2.flush();

/////

globalMap.put("tLogRow_2_NB_LINE", nb_line_tLogRow_2);

////////////////////

ok_Hash.put("tLogRow_2", true);
end_Hash.put("tLogRow_2", System.currentTimeMillis());

/**
 * [tLogRow_2 end ] stop
 */

} // end the resume

```

```

    } catch (java.lang.Exception e) {

        TalendException te = new TalendException(e, currentComponent,
globalMap);

        throw te;
    } catch (java.lang.Error error) {

        throw error;
    } finally {

        // free memory for "tMap_1"
        globalMap.remove("tHash_Lookup_row4");

        try {

            /**
             * [tFileInputDelimited_1 finally ] start
             */

            currentComponent = "tFileInputDelimited_1";

            /**
             * [tFileInputDelimited_1 finally ] stop
             */

            /**
             * [tMap_1 finally ] start
             */

```

```
currentComponent = "tMap_1";
```

```
/**
```

```
 * [tMap_1 finally ] stop
```

```
 */
```

```
/**
```

```
 * [tFileOutputDelimited_1 finally ] start
```

```
 */
```

```
currentComponent = "tFileOutputDelimited_1";
```

```
if (resourceMap.get("finish_tFileOutputDelimited_1") == null) {
```

resourceMap

```
    java.io.Writer outtFileOutputDelimited_1 = (java.io.Writer)
```

```
        .get("out_tFileOutputDelimited_1");
```

```
    if (outtFileOutputDelimited_1 != null) {
```

```
        outtFileOutputDelimited_1.flush();
```

```
        outtFileOutputDelimited_1.close();
```

```
    }
```

```
}
```

```
/**
```

```
 * [tFileOutputDelimited_1 finally ] stop
```

```
 */
```

```
/**
```

```
 * [tLogRow_2 finally ] start
```

```
 */
```

```

        currentComponent = "tLogRow_2";

        /**
         * [tLogRow_2 finally ] stop
         */

    } catch (java.lang.Exception e) {
        // ignore
    } catch (java.lang.Error error) {
        // ignore
    }
    resourceMap = null;
}

globalMap.put("tFileInputDelimited_1_SUBPROCESS_STATE", 1);
}

```

```

public static class row4Struct implements
routines.system.IPersistableComparableLookupRow<row4Struct> {

    final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
new byte[0];

    static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];

    protected static final int DEFAULT_HASHCODE = 1;

    protected static final int PRIME = 31;

    protected int hashCode = DEFAULT_HASHCODE;

    public boolean hashCodeDirty = true;

    public String loopKey;

    public Integer work_year;

```

```
public Integer getWork_year() {  
    return this.work_year;  
}
```

```
public String experience_level;
```

```
public String getExperience_level() {  
    return this.experience_level;  
}
```

```
public String employment_type;
```

```
public String getEmployment_type() {  
    return this.employment_type;  
}
```

```
public String job_title;
```

```
public String getJob_title() {  
    return this.job_title;  
}
```

```
public Integer salary;
```

```
public Integer getSalary() {  
    return this.salary;  
}
```

```
public String salary_currency;
```

```
public String getSalary_currency() {
```

```
        return this.salary_currency;
    }

    public Integer salary_in_usd;

    public Integer getSalary_in_usd() {
        return this.salary_in_usd;
    }

    public String employee_residence;

    public String getEmployee_residence() {
        return this.employee_residence;
    }

    public Integer remote_ratio;

    public Integer getRemote_ratio() {
        return this.remote_ratio;
    }

    public String company_location;

    public String getCompany_location() {
        return this.company_location;
    }

    public Character company_size;

    public Character getCompany_size() {
        return this.company_size;
    }
```

```
}
```

```
@Override
```

```
public int hashCode() {
```

```
    if (this.hashCodeDirty) {
```

```
        final int prime = PRIME;
```

```
        int result = DEFAULT_HASHCODE;
```

```
        result = prime * result + ((this.work_year == null) ? 0 :  
this.work_year.hashCode());
```

```
        result = prime * result + ((this.experience_level == null) ? 0 :  
this.experience_level.hashCode());
```

```
        result = prime * result + ((this.employment_type == null) ? 0 :  
this.employment_type.hashCode());
```

```
        result = prime * result + ((this.job_title == null) ? 0 :  
this.job_title.hashCode());
```

```
        result = prime * result + ((this.salary == null) ? 0 :  
this.salary.hashCode());
```

```
        result = prime * result + ((this.salary_currency == null) ? 0 :  
this.salary_currency.hashCode());
```

```
        result = prime * result + ((this.salary_in_usd == null) ? 0 :  
this.salary_in_usd.hashCode());
```

```
        result = prime * result + ((this.employee_residence == null) ? 0 :  
this.employee_residence.hashCode());
```

```
        result = prime * result + ((this.remote_ratio == null) ? 0 :  
this.remote_ratio.hashCode());
```



```
        result = prime * result + ((this.company_location == null) ? 0 :
this.company_location.hashCode());
```

```
        result = prime * result + ((this.company_size == null) ? 0 :
this.company_size.hashCode());
```

```
        this.hashCode = result;
        this.hashCodeDirty = false;
    }
    return this.hashCode;
}
```

```
@Override
```

```
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    final row4Struct other = (row4Struct) obj;

    if (this.work_year == null) {
        if (other.work_year != null)
            return false;

    } else if (!this.work_year.equals(other.work_year))

        return false;

    if (this.experience_level == null) {
```

```
        if (other.experience_level != null)
            return false;

    } else if (!this.experience_level.equals(other.experience_level))

        return false;

    if (this.employment_type == null) {
        if (other.employment_type != null)
            return false;

    } else if (!this.employment_type.equals(other.employment_type))

        return false;

    if (this.job_title == null) {
        if (other.job_title != null)
            return false;

    } else if (!this.job_title.equals(other.job_title))

        return false;

    if (this.salary == null) {
        if (other.salary != null)
            return false;

    } else if (!this.salary.equals(other.salary))

        return false;
```

```
if (this.salary_currency == null) {
    if (other.salary_currency != null)
        return false;

} else if (!this.salary_currency.equals(other.salary_currency))

    return false;

if (this.salary_in_usd == null) {
    if (other.salary_in_usd != null)
        return false;

} else if (!this.salary_in_usd.equals(other.salary_in_usd))

    return false;

if (this.employee_residence == null) {
    if (other.employee_residence != null)
        return false;

} else if (!this.employee_residence.equals(other.employee_residence))

    return false;

if (this.remote_ratio == null) {
    if (other.remote_ratio != null)
        return false;

} else if (!this.remote_ratio.equals(other.remote_ratio))

    return false;
```

```
        if (this.company_location == null) {
            if (other.company_location != null)
                return false;

        } else if (!this.company_location.equals(other.company_location))

            return false;

        if (this.company_size == null) {
            if (other.company_size != null)
                return false;

        } else if (!this.company_size.equals(other.company_size))

            return false;

        return true;
    }
}
```

```
public void copyDataTo(row4Struct other) {

    other.work_year = this.work_year;
    other.experience_level = this.experience_level;
    other.employment_type = this.employment_type;
    other.job_title = this.job_title;
    other.salary = this.salary;
    other.salary_currency = this.salary_currency;
    other.salary_in_usd = this.salary_in_usd;
    other.employee_residence = this.employee_residence;
    other.remote_ratio = this.remote_ratio;
```

```
        other.company_location = this.company_location;
        other.company_size = this.company_size;

    }

    public void copyKeysDataTo(row4Struct other) {

        other.work_year = this.work_year;
        other.experience_level = this.experience_level;
        other.employment_type = this.employment_type;
        other.job_title = this.job_title;
        other.salary = this.salary;
        other.salary_currency = this.salary_currency;
        other.salary_in_usd = this.salary_in_usd;
        other.employee_residence = this.employee_residence;
        other.remote_ratio = this.remote_ratio;
        other.company_location = this.company_location;
        other.company_size = this.company_size;

    }
```

```
    private Integer readInteger(ObjectInputStream dis) throws IOException {
        Integer intReturn;
        int length = 0;
        length = dis.readByte();
        if (length == -1) {
            intReturn = null;
        } else {
            intReturn = dis.readInt();
        }
        return intReturn;
    }
```

```
}
```

```
private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws  
IOException {  
  
    Integer intReturn;  
  
    int length = 0;  
    length = dis.readByte();  
    if (length == -1) {  
        intReturn = null;  
    } else {  
        intReturn = dis.readInt();  
    }  
    return intReturn;  
}
```

```
private void writeInteger(Integer intNum, ObjectOutputStream dos) throws  
IOException {  
  
    if (intNum == null) {  
        dos.writeByte(-1);  
    } else {  
        dos.writeByte(0);  
        dos.writeInt(intNum);  
    }  
}
```

```
private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller  
marshaller) throws IOException {  
  
    if (intNum == null) {  
        marshaller.writeByte(-1);  
    } else {  
        marshaller.writeByte(0);  
        marshaller.writeInt(intNum);  
    }  
}
```

```

    }
}

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;
    int length = 0;
    length = dis.readInt();
    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

            }

        }

        dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

    String strReturn = null;
    int length = 0;
    length = unmarshaller.readInt();

```

```

        if (length == -1) {
            strReturn = null;
        } else {
            if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
                if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

                } else {

                    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

                }

            }

            unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

            strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
        }

        return strReturn;
    }

```

```

private void writeString(String str, ObjectOutputStream dos) throws IOException {
    if (str == null) {
        dos.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        dos.writeInt(byteArray.length);
        dos.write(byteArray);
    }
}

```

```

private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {

```



```
        if (str == null) {
            marshaller.writeInt(-1);
        } else {
            byte[] byteArray = str.getBytes(utf8Charset);
            marshaller.writeInt(byteArray.length);
            marshaller.write(byteArray);
        }
    }
}
```

```
public void readKeysData(ObjectInputStream dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

            this.experience_level = readString(dis);

            this.employment_type = readString(dis);

            this.job_title = readString(dis);

            this.salary = readInteger(dis);

            this.salary_currency = readString(dis);

            this.salary_in_usd = readInteger(dis);
```

```
this.employee_residence = readString(dis);
```

```
this.remote_ratio = readInteger(dis);
```

```
this.company_location = readString(dis);
```

```
length = dis.readByte();
```

```
if (length == -1) {
```

```
    this.company_size = null;
```

```
} else {
```

```
    this.company_size = dis.readChar();
```

```
}
```

```
} catch (IOException e) {
```

```
    throw new RuntimeException(e);
```

```
}
```

```
}
```

```
}
```

```
public void readKeysData(org.jboss.marshalling.Unmarshaller dis) {
```

```
{  
    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
```

```
    try {
```

```
        int length = 0;
```

```
this.work_year = readInteger(dis);

this.experience_level = readString(dis);

this.employment_type = readString(dis);

this.job_title = readString(dis);

this.salary = readInteger(dis);

this.salary_currency = readString(dis);

this.salary_in_usd = readInteger(dis);

this.employee_residence = readString(dis);

this.remote_ratio = readInteger(dis);

this.company_location = readString(dis);

length = dis.readByte();
if (length == -1) {
    this.company_size = null;
} else {
    this.company_size = dis.readChar();
}

} catch (IOException e) {
    throw new RuntimeException(e);
}
```

```
    }

    }

}

public void writeKeysData(ObjectOutputStream dos) {
    try {

        // Integer

        writeInteger(this.work_year, dos);

        // String

        writeString(this.experience_level, dos);

        // String

        writeString(this.employment_type, dos);

        // String

        writeString(this.job_title, dos);

        // Integer

        writeInteger(this.salary, dos);

        // String
```

```
        writeString(this.salary_currency, dos);

        // Integer

        writeInteger(this.salary_in_usd, dos);

        // String

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

```
}
```

```
public void writeKeysData(org.jboss.marshalling.Marshaller dos) {
```

```
    try {
```

```
        // Integer
```

```
        writeInteger(this.work_year, dos);
```

```
        // String
```

```
        writeString(this.experience_level, dos);
```

```
        // String
```

```
        writeString(this.employment_type, dos);
```

```
        // String
```

```
        writeString(this.job_title, dos);
```

```
        // Integer
```

```
        writeInteger(this.salary, dos);
```

```
        // String
```

```
        writeString(this.salary_currency, dos);
```

```
        // Integer
```

```

        writeInteger(this.salary_in_usd, dos);

        // String

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

/**
 * Fill Values data by reading ObjectInputStream.

```

```

*/
public void readValuesData(DataInputStream dis, ObjectInputStream ois) {
    try {

        int length = 0;

    }

    finally {
    }

}

public void readValuesData(DataInputStream dis, org.jboss.marshalling.Unmarshaller
objectIn) {
    try {
        int length = 0;

    }

    finally {
    }

}

/**
 * Return a byte array which represents Values data.
 */
public void writeValuesData(DataOutputStream dos, ObjectOutputStream oos) {
    try {

```



```

        } finally {
        }

    }

    public void writeValuesData(DataOutputStream dos, org.jboss.marshalling.Marshaller
objectOut) {

        try {

        } finally {
        }

    }

    public boolean supportMarshaller() {

        return true;

    }

    public String toString() {

        StringBuilder sb = new StringBuilder();
        sb.append(super.toString());
        sb.append("[");
        sb.append("work_year=" + String.valueOf(work_year));
        sb.append(",experience_level=" + experience_level);
        sb.append(",employment_type=" + employment_type);
        sb.append(",job_title=" + job_title);
        sb.append(",salary=" + String.valueOf(salary));
        sb.append(",salary_currency=" + salary_currency);
        sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));
        sb.append(",employee_residence=" + employee_residence);
        sb.append(",remote_ratio=" + String.valueOf(remote_ratio));
    }

```

```

        sb.append(",company_location=" + company_location);
        sb.append(",company_size=" + String.valueOf(company_size));
        sb.append("]");

        return sb.toString();
    }

    /**
     * Compare keys
     */
    public int compareTo(row4Struct other) {

        int returnValue = -1;

        returnValue = checkNullsAndCompare(this.work_year, other.work_year);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.experience_level,
other.experience_level);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.employment_type,
other.employment_type);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.job_title, other.job_title);

```

```
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.salary, other.salary);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.salary_currency,
other.salary_currency);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.salary_in_usd,
other.salary_in_usd);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.employee_residence,
other.employee_residence);
        if (returnValue != 0) {
            return returnValue;
        }

        returnValue = checkNullsAndCompare(this.remote_ratio,
other.remote_ratio);
        if (returnValue != 0) {
            return returnValue;
        }
```

```
        returnValue = checkNullsAndCompare(this.company_location,  
other.company_location);
```

```
        if (returnValue != 0) {  
            return returnValue;  
        }
```

```
        returnValue = checkNullsAndCompare(this.company_size,  
other.company_size);
```

```
        if (returnValue != 0) {  
            return returnValue;  
        }
```

```
        return returnValue;  
    }
```

```
private int checkNullsAndCompare(Object object1, Object object2) {
```

```
    int returnValue = 0;
```

```
    if (object1 instanceof Comparable && object2 instanceof Comparable) {
```

```
        returnValue = ((Comparable) object1).compareTo(object2);
```

```
    } else if (object1 != null && object2 != null) {
```

```
        returnValue = compareStrings(object1.toString(), object2.toString());
```

```
    } else if (object1 == null && object2 != null) {
```

```
        returnValue = 1;
```

```
    } else if (object1 != null && object2 == null) {
```

```
        returnValue = -1;
```

```
    } else {
```

```
        returnValue = 0;
```

```
    }
```

```
    return returnValue;
```

```
}
```

```

        private int compareStrings(String string1, String string2) {
            return string1.compareTo(string2);
        }
    }

    public static class row2Struct implements routines.system.IPersistableRow<row2Struct> {
        final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
new byte[0];
        static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];

        public Integer work_year;

        public Integer getWork_year() {
            return this.work_year;
        }

        public String experience_level;

        public String getExperience_level() {
            return this.experience_level;
        }

        public String employment_type;

        public String getEmployment_type() {
            return this.employment_type;
        }

        public String job_title;
    }

```

```
public String getJob_title() {  
    return this.job_title;  
}
```

```
public Integer salary;
```

```
public Integer getSalary() {  
    return this.salary;  
}
```

```
public String salary_currency;
```

```
public String getSalary_currency() {  
    return this.salary_currency;  
}
```

```
public Integer salary_in_usd;
```

```
public Integer getSalary_in_usd() {  
    return this.salary_in_usd;  
}
```

```
public String employee_residence;
```

```
public String getEmployee_residence() {  
    return this.employee_residence;  
}
```

```
public Integer remote_ratio;
```

```
public Integer getRemote_ratio() {  
    return this.remote_ratio;  
}
```

```
public String company_location;
```

```
public String getCompany_location() {  
    return this.company_location;  
}
```

```
public Character company_size;
```

```
public Character getCompany_size() {  
    return this.company_size;  
}
```

```
private Integer readInteger(ObjectInputStream dis) throws IOException {  
    Integer intReturn;  
    int length = 0;  
    length = dis.readByte();  
    if (length == -1) {  
        intReturn = null;  
    } else {  
        intReturn = dis.readInt();  
    }  
    return intReturn;  
}
```

```
private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws  
IOException {  
    Integer intReturn;
```

```

        int length = 0;

        length = dis.readByte();

        if (length == -1) {
            intReturn = null;
        } else {
            intReturn = dis.readInt();
        }

        return intReturn;
    }

```

```

private void writeInteger(Integer intNum, ObjectOutputStream dos) throws
IOException {
    if (intNum == null) {
        dos.writeByte(-1);
    } else {
        dos.writeByte(0);
        dos.writeInt(intNum);
    }
}

```

```

private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller
marshaller) throws IOException {
    if (intNum == null) {
        marshaller.writeByte(-1);
    } else {
        marshaller.writeByte(0);
        marshaller.writeInt(intNum);
    }
}

```

```

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;

```



```

        int length = 0;

        length = dis.readInt();

        if (length == -1) {
            strReturn = null;
        } else {
            if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
                if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];
                } else {

                    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];
                }
            }

            dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

            strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
        }

        return strReturn;
    }

```

```

private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

```

```

    String strReturn = null;

    int length = 0;

    length = unmarshaller.readInt();

    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {

```

```

        if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

        } else {

    commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

        }

    }

    unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);

    }

    return strReturn;

}

```

```

private void writeString(String str, ObjectOutputStream dos) throws IOException {

    if (str == null) {

        dos.writeInt(-1);

    } else {

        byte[] byteArray = str.getBytes(utf8Charset);

        dos.writeInt(byteArray.length);

        dos.write(byteArray);

    }

}

```

```

private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {

    if (str == null) {

        marshaller.writeInt(-1);

    } else {

        byte[] byteArray = str.getBytes(utf8Charset);

```

```
        marshaller.writeInt(byteArray.length);
        marshaller.write(byteArray);
    }
}
```

```
public void readData(ObjectInputStream dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

            this.experience_level = readString(dis);

            this.employment_type = readString(dis);

            this.job_title = readString(dis);

            this.salary = readInteger(dis);

            this.salary_currency = readString(dis);

            this.salary_in_usd = readInteger(dis);

            this.employee_residence = readString(dis);

            this.remote_ratio = readInteger(dis);
```

```

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

}

}

public void readData(org.jboss.marshalling.Unmarshaller dis) {

    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
    {

        try {

            int length = 0;

            this.work_year = readInteger(dis);

            this.experience_level = readString(dis);

```

```
        this.employment_type = readString(dis);

        this.job_title = readString(dis);

        this.salary = readInteger(dis);

        this.salary_currency = readString(dis);

        this.salary_in_usd = readInteger(dis);

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}
```

```
}
```

```
public void writeData(ObjectOutputStream dos) {
```

```
    try {
```

```
        // Integer
```

```
        writeInteger(this.work_year, dos);
```

```
        // String
```

```
        writeString(this.experience_level, dos);
```

```
        // String
```

```
        writeString(this.employment_type, dos);
```

```
        // String
```

```
        writeString(this.job_title, dos);
```

```
        // Integer
```

```
        writeInteger(this.salary, dos);
```

```
        // String
```

```
        writeString(this.salary_currency, dos);
```

```
        // Integer
```

```

        writeInteger(this.salary_in_usd, dos);

        // String

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public void writeData(org.jboss.marshalling.Marshaller dos) {
    try {

```

```
// Integer
```

```
writeInteger(this.work_year, dos);
```

```
// String
```

```
writeString(this.experience_level, dos);
```

```
// String
```

```
writeString(this.employment_type, dos);
```

```
// String
```

```
writeString(this.job_title, dos);
```

```
// Integer
```

```
writeInteger(this.salary, dos);
```

```
// String
```

```
writeString(this.salary_currency, dos);
```

```
// Integer
```

```
writeInteger(this.salary_in_usd, dos);
```

```
// String
```



```

        writeString(this.employee_residence, dos);

        // Integer

        writeInteger(this.remote_ratio, dos);

        // String

        writeString(this.company_location, dos);

        // Character

        if (this.company_size == null) {
            dos.writeByte(-1);
        } else {
            dos.writeByte(0);
            dos.writeChar(this.company_size);
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }
}

public String toString() {

    StringBuilder sb = new StringBuilder();
    sb.append(super.toString());
    sb.append("[");
    sb.append("work_year=" + String.valueOf(work_year));

```

```

        sb.append(",experience_level=" + experience_level);
        sb.append(",employment_type=" + employment_type);
        sb.append(",job_title=" + job_title);
        sb.append(",salary=" + String.valueOf(salary));
        sb.append(",salary_currency=" + salary_currency);
        sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));
        sb.append(",employee_residence=" + employee_residence);
        sb.append(",remote_ratio=" + String.valueOf(remote_ratio));
        sb.append(",company_location=" + company_location);
        sb.append(",company_size=" + String.valueOf(company_size));
        sb.append("]");

        return sb.toString();
    }

    /**
     * Compare keys
     */
    public int compareTo(row2Struct other) {

        int returnValue = -1;

        return returnValue;
    }

    private int checkNullsAndCompare(Object object1, Object object2) {
        int returnValue = 0;
        if (object1 instanceof Comparable && object2 instanceof Comparable) {
            returnValue = ((Comparable) object1).compareTo(object2);
        } else if (object1 != null && object2 != null) {
            returnValue = compareStrings(object1.toString(), object2.toString());
        }
    }

```

```

    } else if (object1 == null && object2 != null) {
        returnValue = 1;
    } else if (object1 != null && object2 == null) {
        returnValue = -1;
    } else {
        returnValue = 0;
    }

```

```

    return returnValue;
}

```

```

private int compareStrings(String string1, String string2) {
    return string1.compareTo(string2);
}

```

```

}

```

```

public static class row1Struct implements routines.system.IPersistableRow<row1Struct> {
    final static byte[] commonByteArrayLock_LOCAL_PROJECT_etl_project_process =
new byte[0];
    static byte[] commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[0];

    public Integer work_year;

    public Integer getWork_year() {
        return this.work_year;
    }

    public String experience_level;

    public String getExperience_level() {

```

```
        return this.experience_level;
    }

    public String employment_type;

    public String getEmployment_type() {
        return this.employment_type;
    }

    public String job_title;

    public String getJob_title() {
        return this.job_title;
    }

    public Integer salary;

    public Integer getSalary() {
        return this.salary;
    }

    public String salary_currency;

    public String getSalary_currency() {
        return this.salary_currency;
    }

    public Integer salary_in_usd;

    public Integer getSalary_in_usd() {
        return this.salary_in_usd;
    }
}
```

```
}
```

```
public String employee_residence;
```

```
public String getEmployee_residence() {  
    return this.employee_residence;  
}
```

```
public Integer remote_ratio;
```

```
public Integer getRemote_ratio() {  
    return this.remote_ratio;  
}
```

```
public String company_location;
```

```
public String getCompany_location() {  
    return this.company_location;  
}
```

```
public Character company_size;
```

```
public Character getCompany_size() {  
    return this.company_size;  
}
```

```
private Integer readInteger(ObjectInputStream dis) throws IOException {  
    Integer intReturn;  
    int length = 0;  
    length = dis.readByte();  
    if (length == -1) {
```

```

        intReturn = null;
    } else {
        intReturn = dis.readInt();
    }
    return intReturn;
}

```

```

private Integer readInteger(org.jboss.marshalling.Unmarshaller dis) throws
IOException {
    Integer intReturn;
    int length = 0;
    length = dis.readByte();
    if (length == -1) {
        intReturn = null;
    } else {
        intReturn = dis.readInt();
    }
    return intReturn;
}

```

```

private void writeInteger(Integer intNum, ObjectOutputStream dos) throws
IOException {
    if (intNum == null) {
        dos.writeByte(-1);
    } else {
        dos.writeByte(0);
        dos.writeInt(intNum);
    }
}

```

```

private void writeInteger(Integer intNum, org.jboss.marshalling.Marshaller
marshaller) throws IOException {

```

```

        if (intNum == null) {
            marshaller.writeByte(-1);
        } else {
            marshaller.writeByte(0);
            marshaller.writeInt(intNum);
        }
    }
}

```

```

private String readString(ObjectInputStream dis) throws IOException {
    String strReturn = null;
    int length = 0;
    length = dis.readInt();
    if (length == -1) {
        strReturn = null;
    } else {
        if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {
            if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];
            } else {

                commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];
            }
        }

        dis.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

        strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);
    }

    return strReturn;
}

```

```

        private String readString(org.jboss.marshalling.Unmarshaller unmarshaller) throws
IOException {

            String strReturn = null;

            int length = 0;

            length = unmarshaller.readInt();

            if (length == -1) {

                strReturn = null;

            } else {

                if (length >
commonByteArray_LOCAL_PROJECT_etl_project_process.length) {

                    if (length < 1024 &&
commonByteArray_LOCAL_PROJECT_etl_project_process.length == 0) {

                        commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[1024];

                    } else {

                        commonByteArray_LOCAL_PROJECT_etl_project_process = new byte[2 * length];

                    }

                }

                unmarshaller.readFully(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length);

                strReturn = new
String(commonByteArray_LOCAL_PROJECT_etl_project_process, 0, length, utf8Charset);

            }

            return strReturn;

        }

        private void writeString(String str, ObjectOutputStream dos) throws IOException {

            if (str == null) {

                dos.writeInt(-1);

            } else {

                byte[] byteArray = str.getBytes(utf8Charset);

                dos.writeInt(byteArray.length);

            }

        }

```



```
        dos.write(byteArray);
    }
}
```

```
private void writeString(String str, org.jboss.marshalling.Marshaller marshaller)
throws IOException {
```

```
    if (str == null) {
        marshaller.writeInt(-1);
    } else {
        byte[] byteArray = str.getBytes(utf8Charset);
        marshaller.writeInt(byteArray.length);
        marshaller.write(byteArray);
    }
}
```

```
public void readData(ObjectInputStream dis) {
```

```
    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)
{
```

```
    try {
```

```
        int length = 0;
```

```
        this.work_year = readInteger(dis);
```

```
        this.experience_level = readString(dis);
```

```
        this.employment_type = readString(dis);
```

```
        this.job_title = readString(dis);
```

```

        this.salary = readInteger(dis);

        this.salary_currency = readString(dis);

        this.salary_in_usd = readInteger(dis);

        this.employee_residence = readString(dis);

        this.remote_ratio = readInteger(dis);

        this.company_location = readString(dis);

        length = dis.readByte();
        if (length == -1) {
            this.company_size = null;
        } else {
            this.company_size = dis.readChar();
        }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

}

public void readData(org.jboss.marshalling.Unmarshaller dis) {

```

```
{  
    synchronized (commonByteArrayLock_LOCAL_PROJECT_etl_project_process)  
  
        try {  
  
            int length = 0;  
  
            this.work_year = readInteger(dis);  
  
            this.experience_level = readString(dis);  
  
            this.employment_type = readString(dis);  
  
            this.job_title = readString(dis);  
  
            this.salary = readInteger(dis);  
  
            this.salary_currency = readString(dis);  
  
            this.salary_in_usd = readInteger(dis);  
  
            this.employee_residence = readString(dis);  
  
            this.remote_ratio = readInteger(dis);  
  
            this.company_location = readString(dis);  
  
            length = dis.readByte();  
            if (length == -1) {  
                this.company_size = null;  
            } else {
```

```

        this.company_size = dis.readChar();
    }

} catch (IOException e) {
    throw new RuntimeException(e);
}

}

}

public void writeData(ObjectOutputStream dos) {
    try {

        // Integer

        writeInteger(this.work_year, dos);

        // String

        writeString(this.experience_level, dos);

        // String

        writeString(this.employment_type, dos);

        // String

        writeString(this.job_title, dos);
    }
}

```

```
// Integer
```

```
writeInteger(this.salary, dos);
```

```
// String
```

```
writeString(this.salary_currency, dos);
```

```
// Integer
```

```
writeInteger(this.salary_in_usd, dos);
```

```
// String
```

```
writeString(this.employee_residence, dos);
```

```
// Integer
```

```
writeInteger(this.remote_ratio, dos);
```

```
// String
```

```
writeString(this.company_location, dos);
```

```
// Character
```

```
if (this.company_size == null) {
```

```
    dos.writeByte(-1);
```

```
} else {
```

```
    dos.writeByte(0);
```

```
    dos.writeChar(this.company_size);
```

```
    }

    } catch (IOException e) {
        throw new RuntimeException(e);
    }

}

public void writeData(org.jboss.marshalling.Marshaller dos) {
    try {

        // Integer

        writeInteger(this.work_year, dos);

        // String

        writeString(this.experience_level, dos);

        // String

        writeString(this.employment_type, dos);

        // String

        writeString(this.job_title, dos);

        // Integer

        writeInteger(this.salary, dos);
```

```
// String

writeString(this.salary_currency, dos);

// Integer

writeInteger(this.salary_in_usd, dos);

// String

writeString(this.employee_residence, dos);

// Integer

writeInteger(this.remote_ratio, dos);

// String

writeString(this.company_location, dos);

// Character

if (this.company_size == null) {
    dos.writeByte(-1);
} else {
    dos.writeByte(0);
    dos.writeChar(this.company_size);
}

} catch (IOException e) {
    throw new RuntimeException(e);
}
```

```

    }

}

public String toString() {

    StringBuilder sb = new StringBuilder();
    sb.append(super.toString());
    sb.append("[");
    sb.append("work_year=" + String.valueOf(work_year));
    sb.append(",experience_level=" + experience_level);
    sb.append(",employment_type=" + employment_type);
    sb.append(",job_title=" + job_title);
    sb.append(",salary=" + String.valueOf(salary));
    sb.append(",salary_currency=" + salary_currency);
    sb.append(",salary_in_usd=" + String.valueOf(salary_in_usd));
    sb.append(",employee_residence=" + employee_residence);
    sb.append(",remote_ratio=" + String.valueOf(remote_ratio));
    sb.append(",company_location=" + company_location);
    sb.append(",company_size=" + String.valueOf(company_size));
    sb.append("]");

    return sb.toString();
}

/**
 * Compare keys
 */
public int compareTo(row1Struct other) {

    int returnValue = -1;

```



```
        return returnValue;
    }
}
```

```
private int checkNullsAndCompare(Object object1, Object object2) {
    int returnValue = 0;
    if (object1 instanceof Comparable && object2 instanceof Comparable) {
        returnValue = ((Comparable) object1).compareTo(object2);
    } else if (object1 != null && object2 != null) {
        returnValue = compareStrings(object1.toString(), object2.toString());
    } else if (object1 == null && object2 != null) {
        returnValue = 1;
    } else if (object1 != null && object2 == null) {
        returnValue = -1;
    } else {
        returnValue = 0;
    }

    return returnValue;
}

}
```

```
private int compareStrings(String string1, String string2) {
    return string1.compareTo(string2);
}

}
```

```
public void tFileInputJSON_1Process(final java.util.Map<String, Object> globalMap) throws
TalendException {
    globalMap.put("tFileInputJSON_1_SUBPROCESS_STATE", 0);
}
```

```

final boolean execStat = this.execStat;

String iterateld = "";

String currentComponent = "";

java.util.Map<String, Object> resourceMap = new java.util.HashMap<String,
Object>();

try {
    // TDI-39566 avoid throwing an useless Exception
    boolean resumelt = true;
    if (globalResumeTicket == false && resumeEntryMethodName != null) {
        String currentMethodName = new
java.lang.Exception().getStackTrace()[0].getMethodName();
        resumelt = resumeEntryMethodName.equals(currentMethodName);
    }
    if (resumelt || globalResumeTicket) { // start the resume
        globalResumeTicket = true;

        row1Struct row1 = new row1Struct();
        row1Struct row2 = row1;
        row1Struct row4 = row1;

        /**
         * [tAdvancedHash_row4 begin ] start
         */

        ok_Hash.put("tAdvancedHash_row4", false);
        start_Hash.put("tAdvancedHash_row4", System.currentTimeMillis());

        currentComponent = "tAdvancedHash_row4";
    }
}

```

```

        int tos_count_tAdvancedHash_row4 = 0;

        // connection name:row4
        // source node:tFileOutputExcel_1 - inputs:(row2)
outputs:(row4,row4) | target

        // node:tAdvancedHash_row4 - inputs:(row4) outputs:()
        // linked node: tMap_1 - inputs:(row3,row4)
outputs:(output_sal,rejected)


        org.talend.designer.components.lookup.common.ICommonLookup.MATCHING_MODE
matchingModeEnum_row4 =
org.talend.designer.components.lookup.common.ICommonLookup.MATCHING_MODE.UNIQUE_MAT
CH;


        org.talend.designer.components.lookup.memory.AdvancedMemoryLookup<row4Struct>
tHash_Lookup_row4 = org.talend.designer.components.lookup.memory.AdvancedMemoryLookup

.<row4Struct>getLookup(matchingModeEnum_row4);


        globalMap.put("tHash_Lookup_row4", tHash_Lookup_row4);


        /**
         * [tAdvancedHash_row4 begin ] stop
         */

        /**
         * [tFileOutputExcel_1 begin ] start
         */

        ok_Hash.put("tFileOutputExcel_1", false);
        start_Hash.put("tFileOutputExcel_1", System.currentTimeMillis());

```

```

currentComponent = "tFileOutputExcel_1";

int tos_count_tFileOutputExcel_1 = 0;

int columnIndex_tFileOutputExcel_1 = 0;
boolean headerIsInserted_tFileOutputExcel_1 = false;

int nb_line_tFileOutputExcel_1 = 0;

String fileName_tFileOutputExcel_1 = "C:/Program Files
(x86)/TOS_DI-8.0.1/studio/workspace/ETL_project/salaries_outfile.xls";

java.io.File file_tFileOutputExcel_1 = new
java.io.File(fileName_tFileOutputExcel_1);

boolean isFileGenerated_tFileOutputExcel_1 = true;

//create directory only if not exists

java.io.File parentFile_tFileOutputExcel_1 =
file_tFileOutputExcel_1.getParentFile();

if (parentFile_tFileOutputExcel_1 != null &&
!parentFile_tFileOutputExcel_1.exists()) {

    parentFile_tFileOutputExcel_1.mkdirs();

}

jxl.write.WritableWorkbook writeableWorkbook_tFileOutputExcel_1
= null;

jxl.write.WritableSheet writableSheet_tFileOutputExcel_1 = null;

jxl.WorkbookSettings workbookSettings_tFileOutputExcel_1 = new
jxl.WorkbookSettings();

workbookSettings_tFileOutputExcel_1.setEncoding("ISO-8859-15");

writeableWorkbook_tFileOutputExcel_1 = new
jxl.write.biff.WritableWorkbookImpl(

```

```

        new java.io.BufferedOutputStream(new
java.io.FileOutputStream(fileName_tFileOutputExcel_1)),
        true, workbookSettings_tFileOutputExcel_1);

        writableSheet_tFileOutputExcel_1 =
writeableWorkbook_tFileOutputExcel_1.getSheet("Sheet1");
        if (writableSheet_tFileOutputExcel_1 == null) {
            writableSheet_tFileOutputExcel_1 =
writeableWorkbook_tFileOutputExcel_1.createSheet("Sheet1",

writeableWorkbook_tFileOutputExcel_1.getNumberOfSheets());
        }

        // modif start
        int startRowNum_tFileOutputExcel_1 =
writableSheet_tFileOutputExcel_1.getRows();
        // modif end

        int[] fitWidth_tFileOutputExcel_1 = new int[11];
        for (int i_tFileOutputExcel_1 = 0; i_tFileOutputExcel_1 < 11;
i_tFileOutputExcel_1++) {
            int fitCellViewSize_tFileOutputExcel_1 =
writableSheet_tFileOutputExcel_1

.getColumnView(i_tFileOutputExcel_1).getSize();

            fitWidth_tFileOutputExcel_1[i_tFileOutputExcel_1] =
fitCellViewSize_tFileOutputExcel_1 / 256;
            if (fitCellViewSize_tFileOutputExcel_1 % 256 != 0) {
                fitWidth_tFileOutputExcel_1[i_tFileOutputExcel_1]
+= 1;
            }
        }

        if (startRowNum_tFileOutputExcel_1 == 0) {
            // modif end

```

```

// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(0,
nb_line_tFileOutputExcel_1, "work_year"));
// modif end
fitWidth_tFileOutputExcel_1[0] =
fitWidth_tFileOutputExcel_1[0] > 9 ? fitWidth_tFileOutputExcel_1[0]
: 9;
// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(1,
nb_line_tFileOutputExcel_1, "experience_level"));
// modif end
fitWidth_tFileOutputExcel_1[1] =
fitWidth_tFileOutputExcel_1[1] > 16
? fitWidth_tFileOutputExcel_1[1]
: 16;
// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(2,
nb_line_tFileOutputExcel_1, "employment_type"));
// modif end
fitWidth_tFileOutputExcel_1[2] =
fitWidth_tFileOutputExcel_1[2] > 15
? fitWidth_tFileOutputExcel_1[2]
: 15;
// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(3,
nb_line_tFileOutputExcel_1, "job_title"));
// modif end
fitWidth_tFileOutputExcel_1[3] =
fitWidth_tFileOutputExcel_1[3] > 9 ? fitWidth_tFileOutputExcel_1[3]
: 9;

```

```

// modif start
writableSheet_tFileOutputExcel_1
.nb_line_tFileOutputExcel_1, "salary"));
        .addCell(new jxl.write.Label(4,

// modif end
        fitWidth_tFileOutputExcel_1[4] =
fitWidth_tFileOutputExcel_1[4] > 6 ? fitWidth_tFileOutputExcel_1[4]
        : 6;

// modif start
writableSheet_tFileOutputExcel_1
.nb_line_tFileOutputExcel_1, "salary_currency"));
        .addCell(new jxl.write.Label(5,

// modif end
        fitWidth_tFileOutputExcel_1[5] =
fitWidth_tFileOutputExcel_1[5] > 15
        ? fitWidth_tFileOutputExcel_1[5]
        : 15;

// modif start
writableSheet_tFileOutputExcel_1
.nb_line_tFileOutputExcel_1, "salary_in_usd"));
        .addCell(new jxl.write.Label(6,

// modif end
        fitWidth_tFileOutputExcel_1[6] =
fitWidth_tFileOutputExcel_1[6] > 13
        ? fitWidth_tFileOutputExcel_1[6]
        : 13;

// modif start
writableSheet_tFileOutputExcel_1
.nb_line_tFileOutputExcel_1, "employee_residence"));
        .addCell(new jxl.write.Label(7,

// modif end
        fitWidth_tFileOutputExcel_1[7] =
fitWidth_tFileOutputExcel_1[7] > 18
        ? fitWidth_tFileOutputExcel_1[7]

```

```

: 18;

// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(8,
nb_line_tFileOutputExcel_1, "remote_ratio"));

// modif end
fitWidth_tFileOutputExcel_1[8] =
fitWidth_tFileOutputExcel_1[8] > 12
? fitWidth_tFileOutputExcel_1[8]
: 12;

// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(9,
nb_line_tFileOutputExcel_1, "company_location"));

// modif end
fitWidth_tFileOutputExcel_1[9] =
fitWidth_tFileOutputExcel_1[9] > 16
? fitWidth_tFileOutputExcel_1[9]
: 16;

// modif start
writableSheet_tFileOutputExcel_1
.addCell(new jxl.write.Label(10,
nb_line_tFileOutputExcel_1, "company_size"));

// modif end
fitWidth_tFileOutputExcel_1[10] =
fitWidth_tFileOutputExcel_1[10] > 12
? fitWidth_tFileOutputExcel_1[10]
: 12;

nb_line_tFileOutputExcel_1++;
headerIsInserted_tFileOutputExcel_1 = true;
}

/**

```



```

* [tFileOutputExcel_1 begin ] stop
*/

/**
* [tLogRow_1 begin ] start
*/

ok_Hash.put("tLogRow_1", false);
start_Hash.put("tLogRow_1", System.currentTimeMillis());

currentComponent = "tLogRow_1";

int tos_count_tLogRow_1 = 0;

////////////////////////////////////

class Util_tLogRow_1 {

    String[] des_top = { ".", ".", "-", "+" };

    String[] des_head = { "|=", "=|", "-", "+" };

    String[] des_bottom = { "", "", "-", "+" };

    String name = "";

    java.util.List<String[]> list = new
java.util.ArrayList<String[]>();

    int[] colLengths = new int[11];

```

```

public void addRow(String[] row) {

    for (int i = 0; i < 11; i++) {
        if (row[i] != null) {
            colLengths[i] =
Math.max(colLengths[i], row[i].length());
        }
    }
    list.add(row);
}

public void setTableName(String name) {

    this.name = name;
}

public StringBuilder format() {

    StringBuilder sb = new StringBuilder();

    sb.append(print(des_top));

    int totals = 0;
    for (int i = 0; i < colLengths.length; i++) {
        totals = totals + colLengths[i];
    }

    // name
    sb.append(" |");
    int k = 0;
    for (k = 0; k < (totals + 10 - name.length()) / 2; k++) {

```

```

        sb.append(' ');
    }
    sb.append(name);
    for (int i = 0; i < totals + 10 - name.length() - k; i++) {
        sb.append(' ');
    }
    sb.append("\n");

```

```

// head and rows
sb.append(print(des_head));
for (int i = 0; i < list.size(); i++) {

```

```

    String[] row = list.get(i);

```

```

        java.util.Formatter formatter = new
java.util.Formatter(new StringBuilder());

```

```

        StringBuilder sbformat = new StringBuilder();
        sbformat.append("|%1$-");
        sbformat.append(colLengths[0]);
        sbformat.append("s");

```

```

        sbformat.append("|%2$-");
        sbformat.append(colLengths[1]);
        sbformat.append("s");

```

```

        sbformat.append("|%3$-");
        sbformat.append(colLengths[2]);
        sbformat.append("s");

```

```

        sbformat.append("|%4$-");

```

```
sbformat.append(colLengths[3]);  
sbformat.append("s");
```

```
sbformat.append("| %5$-");  
sbformat.append(colLengths[4]);  
sbformat.append("s");
```

```
sbformat.append("| %6$-");  
sbformat.append(colLengths[5]);  
sbformat.append("s");
```

```
sbformat.append("| %7$-");  
sbformat.append(colLengths[6]);  
sbformat.append("s");
```

```
sbformat.append("| %8$-");  
sbformat.append(colLengths[7]);  
sbformat.append("s");
```

```
sbformat.append("| %9$-");  
sbformat.append(colLengths[8]);  
sbformat.append("s");
```

```
sbformat.append("| %10$-");  
sbformat.append(colLengths[9]);  
sbformat.append("s");
```

```
sbformat.append("| %11$-");  
sbformat.append(colLengths[10]);  
sbformat.append("s");
```

(Object[]) row);

the head

```
sbformat.append("\n");
```

```
formatter.format(sbformat.toString(),
```

```
sb.append(formatter.toString());
```

```
if (i == 0)
```

```
sb.append(print(des_head)); // print
```

```
}
```

```
// end
```

```
sb.append(print(des_bottom));
```

```
return sb;
```

```
}
```

```
private StringBuilder print(String[] fillChars) {
```

```
StringBuilder sb = new StringBuilder();
```

```
// first column
```

```
sb.append(fillChars[0]);
```

```
for (int i = 0; i < colLengths[0] - fillChars[0].length() +
```

1; i++) {

```
sb.append(fillChars[2]);
```

```
}
```

```
sb.append(fillChars[3]);
```

```
for (int i = 0; i < colLengths[1] - fillChars[3].length() +
```

1; i++) {

```
sb.append(fillChars[2]);
```

```
}
```

```
sb.append(fillChars[3]);
```

```
for (int i = 0; i < colLengths[2] - fillChars[3].length() +
```

1; i++) {

```

        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[3] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[4] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[5] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[6] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[7] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);
    for (int i = 0; i < colLengths[8] - fillChars[3].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);

```

```

1; i++) {
    for (int i = 0; i < colLengths[9] - fillChars[3].length() +
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[3]);

    // last column
    for (int i = 0; i < colLengths[10] - fillChars[1].length() +
1; i++) {
        sb.append(fillChars[2]);
    }
    sb.append(fillChars[1]);
    sb.append("\n");
    return sb;
}

public boolean isEmpty() {
    if (list.size() > 1)
        return false;
    return true;
}

}

Util_tLogRow_1 util_tLogRow_1 = new Util_tLogRow_1();
util_tLogRow_1.setTableName("console1");
util_tLogRow_1.addRow(new String[] { "work_year",
"experience_level", "employment_type", "job_title",
"salary", "salary_currency", "salary_in_usd",
"employee_residence", "remote_ratio",
"company_location", "company_size", });
StringBuilder strBuffer_tLogRow_1 = null;
int nb_line_tLogRow_1 = 0;

//////////

```

```

/**
 * [tLogRow_1 begin ] stop
 */

/**
 * [tFileInputJSON_1 begin ] start
 */

ok_Hash.put("tFileInputJSON_1", false);
start_Hash.put("tFileInputJSON_1", System.currentTimeMillis());

currentComponent = "tFileInputJSON_1";

int tos_count_tFileInputJSON_1 = 0;

class JsonPathCache_tFileInputJSON_1 {
    final java.util.Map<String, com.jayway.jsonpath.JsonPath>
jsonPathString2compiledJsonPath = new java.util.HashMap<String, com.jayway.jsonpath.JsonPath>();

    public com.jayway.jsonpath.JsonPath
getCompiledJsonPath(String jsonPath) {
        if
(jsonPathString2compiledJsonPath.containsKey(jsonPath)) {
            return
(jsonPathString2compiledJsonPath.get(jsonPath);
        } else {
            com.jayway.jsonpath.JsonPath
compiledLoopPath = com.jayway.jsonpath.JsonPath
                .compile(jsonPath);

            jsonPathString2compiledJsonPath.put(jsonPath, compiledLoopPath);

            return compiledLoopPath;
        }
    }
}

```



```

        }
    }
}

int nb_line_tFileInputJSON_1 = 0;

JsonPathCache_tFileInputJSON_1 jsonPathCache_tFileInputJSON_1 =
new JsonPathCache_tFileInputJSON_1();

String loopPath_tFileInputJSON_1 = "$[*]";
java.util.List<Object> resultset_tFileInputJSON_1 = new
java.util.ArrayList<Object>();

java.io.InputStream is_tFileInputJSON_1 = null;
com.jayway.jsonpath.ParseContext parseContext_tFileInputJSON_1 =
com.jayway.jsonpath.JsonPath

    .using(com.jayway.jsonpath.Configuration.defaultConfiguration());

Object filenameOrStream_tFileInputJSON_1 = null;
try {
    filenameOrStream_tFileInputJSON_1 = "C:/Program Files
(x86)/TOS_DI-8.0.1/studio/workspace/ETL_project/salaries.json";
} catch (java.lang.Exception e_tFileInputJSON_1) {
    globalMap.put("tFileInputJSON_1_ERROR_MESSAGE",
e_tFileInputJSON_1.getMessage());

    globalMap.put("tFileInputJSON_1_ERROR_MESSAGE",
e_tFileInputJSON_1.getMessage());

    System.err.println(e_tFileInputJSON_1.getMessage());
}

com.jayway.jsonpath.ReadContext document_tFileInputJSON_1 =
null;

try {

```

```

        if (filenameOrStream_tFileInputJSON_1 instanceof
java.io.InputStream) {

            is_tFileInputJSON_1 = (java.io.InputStream)
filenameOrStream_tFileInputJSON_1;

        } else {

            is_tFileInputJSON_1 = new
java.io.FileInputStream((String) filenameOrStream_tFileInputJSON_1);

        }

        document_tFileInputJSON_1 =
parseContext_tFileInputJSON_1.parse(is_tFileInputJSON_1, "UTF-8");

        com.jayway.jsonpath.JsonPath
compiledLoopPath_tFileInputJSON_1 = jsonPathCache_tFileInputJSON_1

        .getCompiledJsonPath(loopPath_tFileInputJSON_1);

        Object result_tFileInputJSON_1 =
document_tFileInputJSON_1.read(compiledLoopPath_tFileInputJSON_1,

            net.minidev.json.JSONObject.class);

        if (result_tFileInputJSON_1 instanceof
net.minidev.json.JSONArray) {

            resultset_tFileInputJSON_1 =
(net.minidev.json.JSONArray) result_tFileInputJSON_1;

        } else {

            resultset_tFileInputJSON_1.add(result_tFileInputJSON_1);

        }

    } catch (java.lang.Exception e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE",
e_tFileInputJSON_1.getMessage());

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE",
e_tFileInputJSON_1.getMessage());

        System.err.println(e_tFileInputJSON_1.getMessage());

    } finally {

```

```

        if (is_tFileInputJSON_1 != null) {
            is_tFileInputJSON_1.close();
        }
    }

    String jsonPath_tFileInputJSON_1 = null;
    com.jayway.jsonpath.JsonPath compiledJsonPath_tFileInputJSON_1 =
null;

    Object value_tFileInputJSON_1 = null;
    Object root_tFileInputJSON_1 = null;
    for (Object row_tFileInputJSON_1 : resultset_tFileInputJSON_1) {
        nb_line_tFileInputJSON_1++;
        row1 = null;
        boolean whetherReject_tFileInputJSON_1 = false;
        row1 = new row1Struct();

        try {
            jsonPath_tFileInputJSON_1 = "work_year";
            compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

            .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

            try {

                if
(jsonPath_tFileInputJSON_1.startsWith("$")) {

                    if (root_tFileInputJSON_1 == null) {
                        root_tFileInputJSON_1 =
document_tFileInputJSON_1

                        .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

```

```

        }

        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

        } else {

            value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

        }

        if (value_tFileInputJSON_1 != null &&
!value_tFileInputJSON_1.toString().isEmpty()) {

            row1.work_year =
ParserUtils.parseTo_Integer(value_tFileInputJSON_1.toString());

        } else {

            row1.work_year =

                null;

        }

    } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

        row1.work_year =

            null;

    }

    jsonPath_tFileInputJSON_1 = "experience_level";

    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

        .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

```

```

        if (root_tFileInputJSON_1 == null) {
            root_tFileInputJSON_1 =
document_tFileInputJSON_1

        .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));
        }
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);
    } else {
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);
    }
    row1.experience_level =
value_tFileInputJSON_1 == null ?

        null :

value_tFileInputJSON_1.toString();
    } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());
        row1.experience_level =

            null;

    }
    jsonPath_tFileInputJSON_1 = "employment_type";
    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

        .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

```

```

        if (root_tFileInputJSON_1 == null) {
            root_tFileInputJSON_1 =
document_tFileInputJSON_1

        .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));
        }
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);
    } else {
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);
    }
    row1.employment_type =
value_tFileInputJSON_1 == null ?

        null :

value_tFileInputJSON_1.toString();
    } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());
        row1.employment_type =

            null;

    }
    jsonPath_tFileInputJSON_1 = "job_title";
    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

        .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

```

```

        if (root_tFileInputJSON_1 == null) {
            root_tFileInputJSON_1 =
document_tFileInputJSON_1

        .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));
        }
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);
    } else {
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);
    }
    row1.job_title = value_tFileInputJSON_1 ==
null ?

        null :

value_tFileInputJSON_1.toString();

    } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());
        row1.job_title =

        null;

    }
    jsonPath_tFileInputJSON_1 = "salary";
    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

    .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

```

```

        if (root_tFileInputJSON_1 == null) {
            root_tFileInputJSON_1 =
document_tFileInputJSON_1

        .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));
        }
        value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);
        } else {
            value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);
        }
        if (value_tFileInputJSON_1 != null &&
!value_tFileInputJSON_1.toString().isEmpty()) {
            row1.salary =
ParserUtils.parseTo_Integer(value_tFileInputJSON_1.toString());
        } else {
            row1.salary =

            null;

        }
    } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());
        row1.salary =

        null;

    }

    jsonPath_tFileInputJSON_1 = "salary_currency";
    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

    .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

```



```

        try {

            if
(jsonPath_tFileInputJSON_1.startsWith("$")) {

                if (root_tFileInputJSON_1 == null) {

                    root_tFileInputJSON_1 =
document_tFileInputJSON_1

                    .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

                }

                value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

            } else {

                value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

            }

            row1.salary_currency =
value_tFileInputJSON_1 == null ?

                null :

value_tFileInputJSON_1.toString();

        } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

            globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

            row1.salary_currency =

                null;

        }

        jsonPath_tFileInputJSON_1 = "salary_in_usd";
        compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

        .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

```

```

        try {

            if
(jsonPath_tFileInputJSON_1.startsWith("$")) {

                if (root_tFileInputJSON_1 == null) {

                    root_tFileInputJSON_1 =

document_tFileInputJSON_1

                    .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

                }

                value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

            } else {

                value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

            }

            if (value_tFileInputJSON_1 != null &&
!value_tFileInputJSON_1.toString().isEmpty()) {

                row1.salary_in_usd =
ParserUtils.parseTo_Integer(value_tFileInputJSON_1.toString());

            } else {

                row1.salary_in_usd =

                    null;

            }

        } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

            globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

            row1.salary_in_usd =

                null;

        }

        jsonPath_tFileInputJSON_1 = "employee_residence";

```

```

compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

.getCompiledJsonPath(jsonPath_tFileInputJSON_1);

try {

    if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

        if (root_tFileInputJSON_1 == null) {

            root_tFileInputJSON_1 =

document_tFileInputJSON_1

.read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

        }

        value_tFileInputJSON_1 =

compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

    } else {

        value_tFileInputJSON_1 =

compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

    }

    row1.employee_residence =

value_tFileInputJSON_1 == null ?

null :

value_tFileInputJSON_1.toString();

} catch

(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

    globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

    row1.employee_residence =

null;

}

jsonPath_tFileInputJSON_1 = "remote_ratio";

```

```

compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

    .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

            if (root_tFileInputJSON_1 == null) {

                root_tFileInputJSON_1 =

document_tFileInputJSON_1

                .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

            }

            value_tFileInputJSON_1 =

compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

        } else {

            value_tFileInputJSON_1 =

compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

        }

        if (value_tFileInputJSON_1 != null &&

!value_tFileInputJSON_1.toString().isEmpty()) {

            row1.remote_ratio =

ParserUtils.parseTo_Integer(value_tFileInputJSON_1.toString());

        } else {

            row1.remote_ratio =

            null;

        }

    } catch

(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

        row1.remote_ratio =

```

```

null;
    }
    jsonPath_tFileInputJSON_1 = "company_location";
    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

    .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if

(jsonPath_tFileInputJSON_1.startsWith("$")) {

            if (root_tFileInputJSON_1 == null) {

                root_tFileInputJSON_1 =

document_tFileInputJSON_1

                .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

            }

            value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

        } else {

            value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

        }

        row1.company_location =

value_tFileInputJSON_1 == null ?

        null :

value_tFileInputJSON_1.toString();

    } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

        row1.company_location =

```

```

null;
    }
    jsonPath_tFileInputJSON_1 = "company_size";
    compiledJsonPath_tFileInputJSON_1 =
jsonPathCache_tFileInputJSON_1

    .getCompiledJsonPath(jsonPath_tFileInputJSON_1);

    try {

        if
(jsonPath_tFileInputJSON_1.startsWith("$")) {

            if (root_tFileInputJSON_1 == null) {

                root_tFileInputJSON_1 =

document_tFileInputJSON_1

                .read(jsonPathCache_tFileInputJSON_1.getCompiledJsonPath("$"));

            }

            value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(root_tFileInputJSON_1);

        } else {

            value_tFileInputJSON_1 =
compiledJsonPath_tFileInputJSON_1.read(row_tFileInputJSON_1);

        }

        if (value_tFileInputJSON_1 != null &&
!value_tFileInputJSON_1.toString().isEmpty()) {

            row1.company_size =
ParserUtils.parseTo_Character(value_tFileInputJSON_1.toString());

        } else {

            row1.company_size =

            null;

        }
    }

```

```

        } catch
(com.jayway.jsonpath.PathNotFoundException e_tFileInputJSON_1) {

    globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

        row1.company_size =

                                null;

        }

    } catch (java.lang.Exception e_tFileInputJSON_1) {

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

        whetherReject_tFileInputJSON_1 = true;

        System.err.println(e_tFileInputJSON_1.getMessage());

        row1 = null;

        globalMap.put("tFileInputJSON_1_ERROR_MESSAGE", e_tFileInputJSON_1.getMessage());

        }

//}

/**
 * [tFileInputJSON_1 begin ] stop
 */

/**
 * [tFileInputJSON_1 main ] start
 */

currentComponent = "tFileInputJSON_1";

tos_count_tFileInputJSON_1++;

/**

```

```

* [tFileInputJSON_1 main ] stop
*/

/**
* [tFileInputJSON_1 process_data_begin ] start
*/

currentComponent = "tFileInputJSON_1";

/**
* [tFileInputJSON_1 process_data_begin ] stop
*/

// Start of branch "row1"

if (row1 != null) {

    /**
    * [tLogRow_1 main ] start
    */

    currentComponent = "tLogRow_1";

    //////////////////////////////////////

    String[] row_tLogRow_1 = new String[11];

    if (row1.work_year != null) { //
        row_tLogRow_1[0] =
String.valueOf(row1.work_year);

    } //

```



```

String.valueOf(row1.experience_level);

        if (row1.experience_level != null) { //
            row_tLogRow_1[1] =

        } //

        if (row1.employment_type != null) { //
            row_tLogRow_1[2] =

String.valueOf(row1.employment_type);

        } //

        if (row1.job_title != null) { //
            row_tLogRow_1[3] =

String.valueOf(row1.job_title);

        } //

        if (row1.salary != null) { //
            row_tLogRow_1[4] =

String.valueOf(row1.salary);

        } //

        if (row1.salary_currency != null) { //
            row_tLogRow_1[5] =

String.valueOf(row1.salary_currency);

        } //

        if (row1.salary_in_usd != null) { //
            row_tLogRow_1[6] =

String.valueOf(row1.salary_in_usd);

```

```

    } //

    if (row1.employee_residence != null) { //
        row_tLogRow_1[7] =
String.valueOf(row1.employee_residence);

    } //

    if (row1.remote_ratio != null) { //
        row_tLogRow_1[8] =
String.valueOf(row1.remote_ratio);

    } //

    if (row1.company_location != null) { //
        row_tLogRow_1[9] =
String.valueOf(row1.company_location);

    } //

    if (row1.company_size != null) { //
        row_tLogRow_1[10] =
String.valueOf(row1.company_size);

    } //

    util_tLogRow_1.addRow(row_tLogRow_1);
    nb_line_tLogRow_1++;

/////

/////

```

////////////////////////////////

row2 = row1;

tos\_count\_tLogRow\_1++;

/\*\*

\* [tLogRow\_1 main ] stop

\*/

/\*\*

\* [tLogRow\_1 process\_data\_begin ] start

\*/

currentComponent = "tLogRow\_1";

/\*\*

\* [tLogRow\_1 process\_data\_begin ] stop

\*/

/\*\*

\* [tFileOutputExcel\_1 main ] start

\*/

currentComponent = "tFileOutputExcel\_1";

if (row2.work\_year != null) {

//modif start

```

columnIndex_tFileOutputExcel_1 = 0;

jxl.write.WritableCell
cell_0_tFileOutputExcel_1 = new jxl.write.Number(

    columnIndex_tFileOutputExcel_1,

    startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

row2.work_year);

//modif start

// If we keep the cell format from the
existing cell in sheet

//modif ends

writableSheet_tFileOutputExcel_1.addCell(cell_0_tFileOutputExcel_1);

int currentWith_0_tFileOutputExcel_1 =
String
    .valueOf(((jxl.write.Number)
cell_0_tFileOutputExcel_1).getValue()).trim().length();

currentWith_0_tFileOutputExcel_1 =
currentWith_0_tFileOutputExcel_1 > 10 ? 10
    :
currentWith_0_tFileOutputExcel_1;

fitWidth_tFileOutputExcel_1[0] =
fitWidth_tFileOutputExcel_1[0] > currentWith_0_tFileOutputExcel_1
    ?
fitWidth_tFileOutputExcel_1[0]
    :
currentWith_0_tFileOutputExcel_1 + 2;

}

if (row2.experience_level != null) {

```

```
//modif start
```

```
columnIndex_tFileOutputExcel_1 = 1;
```

```
jxl.write.WritableCell
```

```
cell_1_tFileOutputExcel_1 = new jxl.write.Label(
```

```
columnIndex_tFileOutputExcel_1,
```

```
startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,
```

```
//modif end
```

```
row2.experience_level);
```

```
//modif start
```

```
// If we keep the cell format from the
```

```
existing cell in sheet
```

```
//modif ends
```

```
writableSheet_tFileOutputExcel_1.addCell(cell_1_tFileOutputExcel_1);
```

```
int currentWith_1_tFileOutputExcel_1 =
```

```
cell_1_tFileOutputExcel_1.getContents().trim()
```

```
.length();
```

```
fitWidth_tFileOutputExcel_1[1] =
```

```
fitWidth_tFileOutputExcel_1[1] > currentWith_1_tFileOutputExcel_1
```

```
?
```

```
fitWidth_tFileOutputExcel_1[1]
```

```
:
```

```
currentWith_1_tFileOutputExcel_1 + 2;
```

```
}
```

```
if (row2.employment_type != null) {
```

```

//modif start

                                                                    columnIndex_tFileOutputExcel_1 = 2;

                                                                    jxl.write.WritableCell
cell_2_tFileOutputExcel_1 = new jxl.write.Label(

                                                                    columnIndex_tFileOutputExcel_1,

                                                                    startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

                                                                    row2.employment_type);

//modif start

                                                                    // If we keep the cell format from the
existing cell in sheet

//modif ends

                                                                    writableSheet_tFileOutputExcel_1.addCell(cell_2_tFileOutputExcel_1);

                                                                    int currentWith_2_tFileOutputExcel_1 =
cell_2_tFileOutputExcel_1.getContents().trim()

                                                                    .length();

                                                                    fitWidth_tFileOutputExcel_1[2] =
fitWidth_tFileOutputExcel_1[2] > currentWith_2_tFileOutputExcel_1

                                                                    ?

fitWidth_tFileOutputExcel_1[2]

                                                                    :

currentWith_2_tFileOutputExcel_1 + 2;

                                                                    }

                                                                    if (row2.job_title != null) {

//modif start

```

```

columnIndex_tFileOutputExcel_1 = 3;

jxl.write.WritableCell
cell_3_tFileOutputExcel_1 = new jxl.write.Label(

    columnIndex_tFileOutputExcel_1,

    startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

    row2.job_title);

//modif start

    // If we keep the cell format from the
existing cell in sheet

//modif ends

writableSheet_tFileOutputExcel_1.addCell(cell_3_tFileOutputExcel_1);

    int currentWith_3_tFileOutputExcel_1 =
cell_3_tFileOutputExcel_1.getContents().trim()

        .length();

    fitWidth_tFileOutputExcel_1[3] =
fitWidth_tFileOutputExcel_1[3] > currentWith_3_tFileOutputExcel_1

        ?

fitWidth_tFileOutputExcel_1[3]

        :

currentWith_3_tFileOutputExcel_1 + 2;

    }

    if (row2.salary != null) {

//modif start

```

```

columnIndex_tFileOutputExcel_1 = 4;

jxl.write.WritableCell
cell_4_tFileOutputExcel_1 = new jxl.write.Number(

    columnIndex_tFileOutputExcel_1,

    startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

    row2.salary);

//modif start

    // If we keep the cell format from the
existing cell in sheet

//modif ends

writableSheet_tFileOutputExcel_1.addCell(cell_4_tFileOutputExcel_1);

int currentWith_4_tFileOutputExcel_1 =
String

    .valueOf(((jxl.write.Number)
cell_4_tFileOutputExcel_1).getValue()).trim().length();

currentWith_4_tFileOutputExcel_1 =
currentWith_4_tFileOutputExcel_1 > 10 ? 10

    :

currentWith_4_tFileOutputExcel_1;

fitWidth_tFileOutputExcel_1[4] =
fitWidth_tFileOutputExcel_1[4] > currentWith_4_tFileOutputExcel_1

    ?

fitWidth_tFileOutputExcel_1[4]

    :

currentWith_4_tFileOutputExcel_1 + 2;

    }

    if (row2.salary_currency != null) {

```



```
//modif start
```

```
columnIndex_tFileOutputExcel_1 = 5;
```

```
jxl.write.WritableCell
```

```
cell_5_tFileOutputExcel_1 = new jxl.write.Label(
```

```
columnIndex_tFileOutputExcel_1,
```

```
startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,
```

```
//modif end
```

```
row2.salary_currency));
```

```
//modif start
```

```
// If we keep the cell format from the
```

```
existing cell in sheet
```

```
//modif ends
```

```
writableSheet_tFileOutputExcel_1.addCell(cell_5_tFileOutputExcel_1);
```

```
int currentWith_5_tFileOutputExcel_1 =
```

```
cell_5_tFileOutputExcel_1.getContents().trim()
```

```
.length();
```

```
fitWidth_tFileOutputExcel_1[5] =
```

```
fitWidth_tFileOutputExcel_1[5] > currentWith_5_tFileOutputExcel_1
```

```
?
```

```
fitWidth_tFileOutputExcel_1[5]
```

```
:
```

```
currentWith_5_tFileOutputExcel_1 + 2;
```

```
}
```

```
if (row2.salary_in_usd != null) {
```

```

columnIndex_tFileOutputExcel_1 = 6;

jxl.write.WritableCell

cell_6_tFileOutputExcel_1 = new jxl.write.Number(

columnIndex_tFileOutputExcel_1,

startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

row2.salary_in_usd);

//modif start

// If we keep the cell format from the

existing cell in sheet

//modif ends

writableSheet_tFileOutputExcel_1.addCell(cell_6_tFileOutputExcel_1);

int currentWith_6_tFileOutputExcel_1

String

.valueOf(((jxl.write.Nu

cell_6_tFileOutputExcel_1).getValue()).trim()).length());

currentWith_6_tFileOutputExcel_1 =

currentWith_6_tFileOutputExcel_1 > 10 ? 10

:

currentWith_6_tFileOutputExcel_1;

fitWidth_tFileOutputExcel_1[6] =

fitWidth_tFileOutputExcel_1[6] > currentWith_6_tFileOutputExcel_1

?

fitWidth_tFileOutputExcel_1[6]

:

currentWith_6_tFileOutputExcel_1 + 2;

}

```

```

        if (row2.employee_residence != null) {

//modif start

                                columnIndex_tFileOutputExcel_1 = 7;

                                jxl.write.WritableCell
cell_7_tFileOutputExcel_1 = new jxl.write.Label(

                                columnIndex_tFileOutputExcel_1,

                                startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

                                row2.employee_residence);

//modif start

                                // If we keep the cell format from the
existing cell in sheet

//modif ends

                                writableSheet_tFileOutputExcel_1.addCell(cell_7_tFileOutputExcel_1);

                                int currentWith_7_tFileOutputExcel_1 =
cell_7_tFileOutputExcel_1.getContents().trim()

                                .length();

                                fitWidth_tFileOutputExcel_1[7] =
fitWidth_tFileOutputExcel_1[7] > currentWith_7_tFileOutputExcel_1

                                ?

fitWidth_tFileOutputExcel_1[7]

                                :

currentWith_7_tFileOutputExcel_1 + 2;

                                }

```

```

        if (row2.remote_ratio != null) {

//modif start


                columnIndex_tFileOutputExcel_1 = 8;

                jxl.write.WritableCell

cell_8_tFileOutputExcel_1 = new jxl.write.Number(

                columnIndex_tFileOutputExcel_1,

                startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end


                row2.remote_ratio);

//modif start


                // If we keep the cell format from the
existing cell in sheet


//modif ends


writableSheet_tFileOutputExcel_1.addCell(cell_8_tFileOutputExcel_1);

                int currentWith_8_tFileOutputExcel_1 =

String

                .valueOf(((jxl.write.Number)

cell_8_tFileOutputExcel_1).getValue()).trim().length();

                currentWith_8_tFileOutputExcel_1 =

currentWith_8_tFileOutputExcel_1 > 10 ? 10

                :

currentWith_8_tFileOutputExcel_1;

                fitWidth_tFileOutputExcel_1[8] =

fitWidth_tFileOutputExcel_1[8] > currentWith_8_tFileOutputExcel_1

                ?

fitWidth_tFileOutputExcel_1[8]

```

```

:
currentWith_8_tFileOutputExcel_1 + 2;

    }

    if (row2.company_location != null) {

//modif start

        columnIndex_tFileOutputExcel_1 = 9;

        jxl.write.WritableCell
cell_9_tFileOutputExcel_1 = new jxl.write.Label(

        columnIndex_tFileOutputExcel_1,

        startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

        row2.company_location);

//modif start

        // If we keep the cell format from the
existing cell in sheet

//modif ends

        writableSheet_tFileOutputExcel_1.addCell(cell_9_tFileOutputExcel_1);

        int currentWith_9_tFileOutputExcel_1 =
cell_9_tFileOutputExcel_1.getContents().trim()

        .length();

        fitWidth_tFileOutputExcel_1[9] =
fitWidth_tFileOutputExcel_1[9] > currentWith_9_tFileOutputExcel_1

        ?

fitWidth_tFileOutputExcel_1[9]

        :

currentWith_9_tFileOutputExcel_1 + 2;

```

```

    }

    if (row2.company_size != null) {

//modif start

        columnIndex_tFileOutputExcel_1 = 10;

        jxl.write.WritableCell
cell_10_tFileOutputExcel_1 = new jxl.write.Label(

        columnIndex_tFileOutputExcel_1,

        startRowNum_tFileOutputExcel_1 + nb_line_tFileOutputExcel_1,

//modif end

        String.valueOf(row2.company_size));
//modif start

        // If we keep the cell format from the
existing cell in sheet

//modif ends

        writableSheet_tFileOutputExcel_1.addCell(cell_10_tFileOutputExcel_1);

        int currentWith_10_tFileOutputExcel_1 =
cell_10_tFileOutputExcel_1.getContents().trim()

        .length();

        fitWidth_tFileOutputExcel_1[10] =
fitWidth_tFileOutputExcel_1[10] > currentWith_10_tFileOutputExcel_1

        ?

fitWidth_tFileOutputExcel_1[10]

        :

currentWith_10_tFileOutputExcel_1 + 2;

    }

```

```
nb_line_tFileOutputExcel_1++;
```

```
row4 = row2;
```

```
tos_count_tFileOutputExcel_1++;
```

```
/**
```

```
 * [tFileOutputExcel_1 main ] stop
```

```
 */
```

```
/**
```

```
 * [tFileOutputExcel_1 process_data_begin ] start
```

```
 */
```

```
currentComponent = "tFileOutputExcel_1";
```

```
/**
```

```
 * [tFileOutputExcel_1 process_data_begin ] stop
```

```
 */
```

```
/**
```

```
 * [tAdvancedHash_row4 main ] start
```

```
 */
```

```
currentComponent = "tAdvancedHash_row4";
```

```
row4Struct row4_HashRow = new row4Struct();
```

```
row4_HashRow.work_year = row4.work_year;
```

row4.experience_level;	row4_HashRow.experience_level =
row4.employment_type;	row4_HashRow.employment_type =
	row4_HashRow.job_title = row4.job_title;
	row4_HashRow.salary = row4.salary;
row4.salary_currency;	row4_HashRow.salary_currency =
	row4_HashRow.salary_in_usd = row4.salary_in_usd;
row4.employee_residence;	row4_HashRow.employee_residence =
	row4_HashRow.remote_ratio = row4.remote_ratio;
row4.company_location;	row4_HashRow.company_location =
	row4_HashRow.company_size = row4.company_size;
	tHash_Lookup_row4.put(row4_HashRow);
	tos_count_tAdvancedHash_row4++;
	/**
	* [tAdvancedHash_row4 main ] stop
	*/



```
/**  
 * [tAdvancedHash_row4 process_data_begin ] start  
 */
```

```
currentComponent = "tAdvancedHash_row4";
```

```
/**  
 * [tAdvancedHash_row4 process_data_begin ] stop  
 */
```

```
/**  
 * [tAdvancedHash_row4 process_data_end ] start  
 */
```

```
currentComponent = "tAdvancedHash_row4";
```

```
/**  
 * [tAdvancedHash_row4 process_data_end ] stop  
 */
```

```
/**  
 * [tFileOutputExcel_1 process_data_end ] start  
 */
```

```
currentComponent = "tFileOutputExcel_1";
```

```
/**  
 * [tFileOutputExcel_1 process_data_end ] stop  
 */
```

```
/**
```

```

        * [tLogRow_1 process_data_end ] start
        */

        currentComponent = "tLogRow_1";

        /**
        * [tLogRow_1 process_data_end ] stop
        */

    } // End of branch "row1"

    /**
    * [tFileInputJSON_1 process_data_end ] start
    */

    currentComponent = "tFileInputJSON_1";

    /**
    * [tFileInputJSON_1 process_data_end ] stop
    */

    /**
    * [tFileInputJSON_1 end ] start
    */

    currentComponent = "tFileInputJSON_1";

}

globalMap.put("tFileInputJSON_1_NB_LINE",
nb_line_tFileInputJSON_1);

```

```
ok_Hash.put("tFileInputJSON_1", true);
end_Hash.put("tFileInputJSON_1", System.currentTimeMillis());
```

```
/**
 * [tFileInputJSON_1 end ] stop
 */
```

```
/**
 * [tLogRow_1 end ] start
 */
```

```
currentComponent = "tLogRow_1";
```

```
/////
```

```
java.io.PrintStream consoleOut_tLogRow_1 = null;
if (globalMap.get("tLogRow_CONSOLE") != null) {
    consoleOut_tLogRow_1 = (java.io.PrintStream)
globalMap.get("tLogRow_CONSOLE");
} else {
    consoleOut_tLogRow_1 = new java.io.PrintStream(new
java.io.BufferedOutputStream(System.out));
    globalMap.put("tLogRow_CONSOLE",
consoleOut_tLogRow_1);
}

consoleOut_tLogRow_1.println(util_tLogRow_1.format().toString());
consoleOut_tLogRow_1.flush();
```

```
/////
```

```
globalMap.put("tLogRow_1_NB_LINE", nb_line_tLogRow_1);
```

```
////////////////////
```

```

ok_Hash.put("tLogRow_1", true);
end_Hash.put("tLogRow_1", System.currentTimeMillis());

/**
 * [tLogRow_1 end ] stop
 */

/**
 * [tFileOutputExcel_1 end ] start
 */

currentComponent = "tFileOutputExcel_1";

writeableWorkbook_tFileOutputExcel_1.write();
writeableWorkbook_tFileOutputExcel_1.close();
if (headerIsInserted_tFileOutputExcel_1 &&
nb_line_tFileOutputExcel_1 > 0) {
    nb_line_tFileOutputExcel_1 = nb_line_tFileOutputExcel_1 -
1;
}
globalMap.put("tFileOutputExcel_1_NB_LINE",
nb_line_tFileOutputExcel_1);

ok_Hash.put("tFileOutputExcel_1", true);
end_Hash.put("tFileOutputExcel_1", System.currentTimeMillis());

/**
 * [tFileOutputExcel_1 end ] stop
 */

/**

```

```

        * [tAdvancedHash_row4 end ] start
        */

        currentComponent = "tAdvancedHash_row4";

        tHash_Lookup_row4.endPut();

        ok_Hash.put("tAdvancedHash_row4", true);
        end_Hash.put("tAdvancedHash_row4", System.currentTimeMillis());

        /**
        * [tAdvancedHash_row4 end ] stop
        */

    } // end the resume

} catch (java.lang.Exception e) {

    TalendException te = new TalendException(e, currentComponent,
globalMap);

    throw te;
} catch (java.lang.Error error) {

    throw error;
} finally {

    try {

        /**
        * [tFileInputJSON_1 finally ] start

```

```
*/
```

```
currentComponent = "tFileInputJSON_1";
```

```
/**
```

```
 * [tFileInputJSON_1 finally ] stop
```

```
*/
```

```
/**
```

```
 * [tLogRow_1 finally ] start
```

```
*/
```

```
currentComponent = "tLogRow_1";
```

```
/**
```

```
 * [tLogRow_1 finally ] stop
```

```
*/
```

```
/**
```

```
 * [tFileOutputExcel_1 finally ] start
```

```
*/
```

```
currentComponent = "tFileOutputExcel_1";
```

```
/**
```

```
 * [tFileOutputExcel_1 finally ] stop
```

```
*/
```

```
/**
```

```
 * [tAdvancedHash_row4 finally ] start
```

```
*/
```

```

        currentComponent = "tAdvancedHash_row4";

        /**
         * [tAdvancedHash_row4 finally ] stop
         */

    } catch (java.lang.Exception e) {
        // ignore
    } catch (java.lang.Error error) {
        // ignore
    }
    resourceMap = null;
}

    globalMap.put("tFileInputJSON_1_SUBPROCESS_STATE", 1);
}

public String resuming_logs_dir_path = null;
public String resuming_checkpoint_path = null;
public String parent_part_launcher = null;
private String resumeEntryMethodName = null;
private boolean globalResumeTicket = false;

public boolean watch = false;
// portStats is null, it means don't execute the statistics
public Integer portStats = null;
public int portTraces = 4334;
public String clientHost;
public String defaultClientHost = "localhost";
public String contextStr = "Default";

```

```

public boolean isDefaultContext = true;

public String pid = "0";

public String rootPid = null;

public String fatherPid = null;

public String fatherNode = null;

public long startTime = 0;

public boolean isChildJob = false;

public String log4jLevel = "";


private boolean enableLogStash;


private boolean execStat = true;


private ThreadLocal<java.util.Map<String, String>> threadLocal = new
ThreadLocal<java.util.Map<String, String>>() {

    protected java.util.Map<String, String> initialValue() {

        java.util.Map<String, String> threadRunResultMap = new
java.util.HashMap<String, String>();

        threadRunResultMap.put("errorCode", null);

        threadRunResultMap.put("status", "");

        return threadRunResultMap;

    };

};


protected PropertiesWithType context_param = new PropertiesWithType();

public java.util.Map<String, Object> parentContextMap = new java.util.HashMap<String,
Object>();


public String status = "";


public static void main(String[] args) {

    final etl_project_process etl_project_processClass = new etl_project_process();

```



```

        int exitCode = etl_project_processClass.runJobInTOS(args);

        System.exit(exitCode);
    }

    public String[][] runJob(String[] args) {

        int exitCode = runJobInTOS(args);
        String[][] bufferValue = new String[][] { { Integer.toString(exitCode) } };

        return bufferValue;
    }

    public boolean hastBufferOutputComponent() {
        boolean hastBufferOutput = false;

        return hastBufferOutput;
    }

    public int runJobInTOS(String[] args) {
        // reset status
        status = "";

        String lastStr = "";
        for (String arg : args) {
            if (arg.equalsIgnoreCase("--context_param")) {
                lastStr = arg;
            } else if (lastStr.equals("")) {
                evalParam(arg);
            } else {

```

```

        evalParam(lastStr + " " + arg);
        lastStr = "";
    }
}

enableLogStash = "true".equalsIgnoreCase(System.getProperty("audit.enabled"));

if (clientHost == null) {
    clientHost = defaultClientHost;
}

if (pid == null || "0".equals(pid)) {
    pid = TalendString.getAsciiRandomString(6);
}

if (rootPid == null) {
    rootPid = pid;
}

if (fatherPid == null) {
    fatherPid = pid;
} else {
    isChildJob = true;
}

boolean inOSGi = routines.system.BundleUtils.inOSGi();

if (inOSGi) {
    java.util.Dictionary<String, Object> jobProperties =
routines.system.BundleUtils.getJobProperties(jobName);

    if (jobProperties != null && jobProperties.get("context") != null) {
        contextStr = (String) jobProperties.get("context");
    }
}

```

```

    }
}

try {
    // call job/subjob with an existing context, like: --context=production. if
    // without this parameter, there will use the default context instead.
    java.io.InputStream inContext =
etl_project_process.class.getClassLoader().getResourceAsStream(
        "local_project/etl_project_process_0_1/contexts/" +
contextStr + ".properties");
    if (inContext == null) {
        inContext = etl_project_process.class.getClassLoader()
            .getResourceAsStream("config/contexts/" +
contextStr + ".properties");
    }
    if (inContext != null) {
        try {
            // defaultProps is in order to keep the original context value
            if (context != null && context.isEmpty()) {
                defaultProps.load(inContext);
                context = new ContextProperties(defaultProps);
            }
        } finally {
            inContext.close();
        }
    } else if (!isDefaultContext) {
        // print info and job continue to run, for case: context_param is not
empty.
        System.err.println("Could not find the context " + contextStr);
    }

    if (!context_param.isEmpty()) {

```

```

        context.putAll(context_param);

        // set types for params from parentJobs
        for (Object key : context_param.keySet()) {
            String context_key = key.toString();

            String context_type =
context_param.getContextType(context_key);

            context.setContextType(context_key, context_type);

        }
    }

    class ContextProcessing {

        private void processContext_0() {

        }

        public void processAllContext() {
            processContext_0();
        }

    }

    new ContextProcessing().processAllContext();
} catch (java.io.IOException ie) {
    System.err.println("Could not load context " + contextStr);
    ie.printStackTrace();
}

// get context value from parent directly
if (parentContextMap != null && !parentContextMap.isEmpty()) {
}

// Resume: init the resumeUtil
resumeEntryMethodName =
ResumeUtil.getResumeEntryMethodName(resuming_checkpoint_path);

```

```

        resumeUtil = new ResumeUtil(resuming_logs_dir_path, isChildJob, rootPid);

        resumeUtil.initCommonInfo(pid, rootPid, fatherPid, projectName, jobName,
contextStr, jobVersion);

        List<String> parametersToEncrypt = new java.util.ArrayList<String>();

        // Resume: jobStart

        resumeUtil.addLog("JOB_STARTED", "JOB:" + jobName, parent_part_launcher,
Thread.currentThread().getId() + "",

                                "", "", "", "", resumeUtil.convertToJsonText(context,
parametersToEncrypt));

        java.util.concurrent.ConcurrentHashMap<Object, Object> concurrentHashMap = new
java.util.concurrent.ConcurrentHashMap<Object, Object>();

        globalMap.put("concurrentHashMap", concurrentHashMap);

        long startUsedMemory = Runtime.getRuntime().totalMemory() -
Runtime.getRuntime().freeMemory();

        long endUsedMemory = 0;

        long end = 0;

        startTime = System.currentTimeMillis();

        this.globalResumeTicket = true;// to run tPreJob

        this.globalResumeTicket = false;// to run others jobs

        try {

            errorCode = null;

            tFileInputDelimited_1Process(globalMap);

            if (!"failure".equals(status)) {

                status = "end";

            }

        } catch (TalendException e_tFileInputDelimited_1) {

```

```

        globalMap.put("tFileInputDelimited_1_SUBPROCESS_STATE", -1);

        e_tFileInputDelimited_1.printStackTrace();

    }

    this.globalResumeTicket = true;// to run tPostJob

    end = System.currentTimeMillis();

    if (watch) {
        System.out.println((end - startTime) + " milliseconds");
    }

    endUsedMemory = Runtime.getRuntime().totalMemory() -
Runtime.getRuntime().freeMemory();
    if (false) {
        System.out.println(
            (endUsedMemory - startUsedMemory) + " bytes memory
increase when running : etl_project_process");
    }

    int returnCode = 0;

    if (errorCode == null) {
        returnCode = status != null && status.equals("failure") ? 1 : 0;
    } else {
        returnCode = errorCode.intValue();
    }

    resumeUtil.addLog("JOB_ENDED", "JOB:" + jobName, parent_part_launcher,
Thread.currentThread().getId() + "", "",
        "" + returnCode, "", "", "");

```

```

        return returnCode;

    }

    // only for OSGi env
    public void destroy() {

    }

    private java.util.Map<String, Object> getSharedConnections4REST() {
        java.util.Map<String, Object> connections = new java.util.HashMap<String, Object>();

        return connections;
    }

    private void evalParam(String arg) {
        if (arg.startsWith("--resuming_logs_dir_path")) {
            resuming_logs_dir_path = arg.substring(25);
        } else if (arg.startsWith("--resuming_checkpoint_path")) {
            resuming_checkpoint_path = arg.substring(27);
        } else if (arg.startsWith("--parent_part_launcher")) {
            parent_part_launcher = arg.substring(23);
        } else if (arg.startsWith("--watch")) {
            watch = true;
        } else if (arg.startsWith("--stat_port=")) {
            String portStatsStr = arg.substring(12);
            if (portStatsStr != null && !portStatsStr.equals("null")) {
                portStats = Integer.parseInt(portStatsStr);
            }
        } else if (arg.startsWith("--trace_port=")) {

```

```

        portTraces = Integer.parseInt(arg.substring(13));
    } else if (arg.startsWith("--client_host=")) {
        clientHost = arg.substring(14);
    } else if (arg.startsWith("--context=")) {
        contextStr = arg.substring(10);
        isDefaultContext = false;
    } else if (arg.startsWith("--father_pid=")) {
        fatherPid = arg.substring(13);
    } else if (arg.startsWith("--root_pid=")) {
        rootPid = arg.substring(11);
    } else if (arg.startsWith("--father_node=")) {
        fatherNode = arg.substring(14);
    } else if (arg.startsWith("--pid=")) {
        pid = arg.substring(6);
    } else if (arg.startsWith("--context_type=")) {
        String keyValue = arg.substring(15);
        int index = -1;
        if (keyValue != null && (index = keyValue.indexOf('=')) > -1) {
            if (fatherPid == null) {
                context_param.setContextType(keyValue.substring(0, index),

replaceEscapeChars(keyValue.substring(index + 1)));
            } else { // the subjob won't escape the especial chars
                context_param.setContextType(keyValue.substring(0, index),
keyValue.substring(index + 1));
            }
        }

    }

    } else if (arg.startsWith("--context_param=")) {
        String keyValue = arg.substring(16);
        int index = -1;

```



```

        if (keyValue != null && (index = keyValue.indexOf('=') > -1) {
            if (fatherPid == null) {
                context_param.put(keyValue.substring(0, index),
replaceEscapeChars(keyValue.substring(index + 1)));
            } else { // the subjob won't escape the especial chars
                context_param.put(keyValue.substring(0, index),
keyValue.substring(index + 1));
            }
        }
    } else if (arg.startsWith("--log4jLevel=")) {
        log4jLevel = arg.substring(13);
    } else if (arg.startsWith("--audit.enabled") && arg.contains("=")) { // for trunjob call
        final int equal = arg.indexOf('=');
        final String key = arg.substring("--".length(), equal);
        System.setProperty(key, arg.substring(equal + 1));
    }
}

```

```

    private static final String
NULL_VALUE_EXPRESSION_IN_COMMAND_STRING_FOR_CHILD_JOB_ONLY = "<TALEND_NULL>";

```

```

    private final String[][] escapeChars = { { "\\\"", "\\\" }, { "\\n", "\\n" }, { "\\\"", "\\\" }, { "\\r", "\\r" },
        { "\\f", "\\f" }, { "\\b", "\\b" }, { "\\t", "\\t" } };

```

```

    private String replaceEscapeChars(String keyValue) {

```

```

        if (keyValue == null || ("").equals(keyValue.trim())) {
            return keyValue;
        }

```

```

        StringBuilder result = new StringBuilder();
        int currIndex = 0;

```

```

while (currIndex < keyValue.length()) {
    int index = -1;
    // judge if the left string includes escape chars
    for (String[] strArray : escapeChars) {
        index = keyValue.indexOf(strArray[0], currIndex);
        if (index >= 0) {

            result.append(keyValue.substring(currIndex, index +
strArray[0].length()).replace(strArray[0],
                                strArray[1]));
            currIndex = index + strArray[0].length();
            break;
        }
    }
    // if the left string doesn't include escape chars, append the left into the
    // result
    if (index < 0) {
        result.append(keyValue.substring(currIndex));
        currIndex = currIndex + keyValue.length();
    }
}

return result.toString();
}

public Integer getErrorCode() {
    return errorCode;
}

public String getStatus() {
    return status;
}

```

}

ResumeUtil resumeUtil = null;

}

/\*\*\*\*\*  
\*\*\*\*\*

\* 210863 characters generated by Talend Open Studio for Data Integration on the

\* 21 janvier 2024 à 6:12:11 PM CET

\*\*\*\*\*  
\*\*\*\*\*/