Moreover here is my Job in Java Code

```
// ============================================================================
//
// Copyright (c) 2006-2015, Talend Inc.
//
// This source code has been automatically generated by_Talend Open Studio for Data Integration
// / Licensed under the Apache License, Version 2.0 (the "License");
// you may not use this file except in compliance with the License.
// You may obtain a copy of the License at
// http://www.apache.org/licenses/LICENSE-2.0
//
// Unless required by applicable law or agreed to in writing, software
// distributed under the License is distributed on an "AS IS" BASIS,
// WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
// See the License for the specific language governing permissions and
// limitations under the License.
package sf_csv.sf_prod_data_migration_q_ktxt_acc_u_0_1;
import routines.Mathematical;
import routines.DataOperation;
import routines.Relational;
import routines.TalendDate;
import routines.TalendDataGenerator;
import routines.Numeric;
import routines.TalendString;
import routines.StringHandling;
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: SF_Prod_Data_Migration_q_ktxt_acc_u Purpose: <br>
 * Description:  <br>
 * @author user@talend.com
 * @version 6.1.0.20151029_1337
 * @status
 */
public class SF_Prod_Data_Migration_q_ktxt_acc_u implements TalendJob {
 public final Object obj = new Object();
 // for transmiting 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";
// create and load default properties
private java.util.Properties defaultProps = new java.util.Properties();
// create application properties with default
public class ContextProperties extends java.util.Properties {
 private static final long serialVersionUID = 1L;
 public ContextProperties(java.util.Properties properties) {
  super(properties);
 }
 public ContextProperties() {
  super();
 }
 public void synchronizeContext() {
  if (new1 != null) {
   this.setProperty("new1", new1.toString());
  }
 }
 public String new1;
 public String getNew1() {
  return this.new1;
 }
}
private ContextProperties context = new ContextProperties();
public ContextProperties getContext() {
 return this.context;
}
private final String jobVersion = "0.1";
private final String jobName = "SF_Prod_Data_Migration_q_ktxt_acc_u";
private final String projectName = "SF_CSV";
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";
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);
```

```
}
LogCatcherUtils tLogCatcher_1 = new LogCatcherUtils();
StatCatcherUtils tStatCatcher_1 = new StatCatcherUtils(
  "_FBfMIMmHEeW456YArOnjeA", "0.1");
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);
 }
 if (!(e instanceof TDieException)) {
  if (e instanceof TalendException) {
   e.printStackTrace();
  } else {
   e.printStackTrace();
   e.printStackTrace(errorMessagePS);
   SF_Prod_Data_Migration_q_ktxt_acc_u.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(SF_Prod_Data_Migration_q_ktxt_acc_u.this,
       new Object[] { e, currentComponent,
         globalMap });
     break;
    }
   }
   if (!(e instanceof TDieException)) {
    tLogCatcher_1.addMessage("Java Exception",
      currentComponent, 6, e.getClass().getName()
        + ":" + e.getMessage(), 1);
    tLogCatcher_1Process(globalMap);
   }
  } catch (TalendException e) {
   // do nothing
  } catch (Exception e) {
   this.e.printStackTrace();
  }
 }
}
public void tDie_1_error(Exception exception, String errorComponent,
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tDie_1_onSubJobError(exception, errorComponent, globalMap);
```

```
}
public void tDie_4_error(Exception exception, String errorComponent,
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tDie_4_onSubJobError(exception, errorComponent, globalMap);
}
public void tLogCatcher_1_error(Exception exception, String errorComponent,
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tLogCatcher_1_onSubJobError(exception, errorComponent, globalMap);
}
public void tSendMail_1_error(Exception exception, String errorComponent,
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tLogCatcher_1_onSubJobError(exception, errorComponent, globalMap);
}
public void tOracleInput_2_error(Exception exception,
  String errorComponent, final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 try {
  errorCode = null;
  tDie_4Process(globalMap);
  if (!"failure".equals(status)) {
   status = "end";
  }
 } catch (Exception e) {
  e.printStackTrace();
 }
 tOracleInput_2_onSubJobError(exception, errorComponent, globalMap);
}
public void tMap_2_error(Exception exception, String errorComponent,
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tOracleInput_2_onSubJobError(exception, errorComponent, globalMap);
}
public void tConvertType_1_error(Exception exception,
  String errorComponent, final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tOracleInput_2_onSubJobError(exception, errorComponent, globalMap);
}
public void tSalesforceOutput_1_error(Exception exception,
  String errorComponent, final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
```

```
 tOracleInput_2_onSubJobError(exception, errorComponent, globalMap);
}
public void tOracleInput_3_error(Exception exception,
  String errorComponent, final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 try {
  errorCode = null;
  tDie_1Process(globalMap);
  if (!"failure".equals(status)) {
   status = "end";
  }
 } catch (Exception e) {
  e.printStackTrace();
 }
 tOracleInput_3_onSubJobError(exception, errorComponent, globalMap);
}
public void tStatCatcher_1_error(Exception exception,
  String errorComponent, final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tStatCatcher_1_onSubJobError(exception, errorComponent, globalMap);
}
public void tSendMail_2_error(Exception exception, String errorComponent,
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tStatCatcher_1_onSubJobError(exception, errorComponent, globalMap);
}
public void tAdvancedHash_row3_error(Exception exception,
  String errorComponent, final java.util.Map<String, Object> globalMap)
  throws TalendException {
 end_Hash.put(errorComponent, System.currentTimeMillis());
 status = "failure";
 tOracleInput_3_onSubJobError(exception, errorComponent, globalMap);
}
public void tDie_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 tDie_4_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 tLogCatcher_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 tOracleInput_2_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 tOracleInput_3_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 tStatCatcher_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 tDie_1Process(final java.util.Map<String, Object> globalMap)
  throws TalendException {
 globalMap.put("tDie_1_SUBPROCESS_STATE", 0);
 final boolean execStat = this.execStat;
 String iterateId = "";
 String currentComponent = "";
 java.util.Map<String, Object> resourceMap = new java.util.HashMap<String, Object>();
 try {
  String currentMethodName = new java.lang.Exception()
    .getStackTrace().getMethodName();
  boolean resumeIt = currentMethodName.equals(resumeEntryMethodName);
  if (resumeEntryMethodName == null || resumeIt || globalResumeTicket) {// start
                    // the
                    // resume
   globalResumeTicket = true;
   /**
    *  start
    */
   ok_Hash.put("tDie_1", false);
   start_Hash.put("tDie_1", System.currentTimeMillis());
   currentComponent = "tDie_1";
   int tos_count_tDie_1 = 0;
   /**
    *  stop
    */
   /**
```

```
 *  start
 */
currentComponent = "tDie_1";
tLogCatcher_1
  .addMessage(
    "tDie",
    "tDie_1",
    5,
    "Some Database Error occured plz. Check Database Settings",
    4);
tLogCatcher_1Process(globalMap);
globalMap.put("tDie_1_DIE_PRIORITY", 5);
System.err
  .println("Some Database Error occured plz. Check Database Settings");
globalMap
  .put("tDie_1_DIE_MESSAGE",
    "Some Database Error occured plz. Check Database Settings");
globalMap
  .put("tDie_1_DIE_MESSAGES",
    "Some Database Error occured plz. Check Database Settings");
currentComponent = "tDie_1";
status = "failure";
errorCode = new Integer(4);
globalMap.put("tDie_1_DIE_CODE", errorCode);
if (true) {
 throw new TDieException();
}
tos_count_tDie_1++;
  /**
   *  stop
   */
  /**
   *  start
   */
  currentComponent = "tDie_1";
  ok_Hash.put("tDie_1", true);
  end_Hash.put("tDie_1", System.currentTimeMillis());
  /**
   *  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 {
  /**
   *  start
   */
  currentComponent = "tDie_1";
  /**
   *  stop
   */
```

```
  } catch (java.lang.Exception e) {
   // ignore
  } catch (java.lang.Error error) {
   // ignore
  }
  resourceMap = null;
 }
 globalMap.put("tDie_1_SUBPROCESS_STATE", 1);
}
public void tDie_4Process(final java.util.Map<String, Object> globalMap)
  throws TalendException {
 globalMap.put("tDie_4_SUBPROCESS_STATE", 0);
 final boolean execStat = this.execStat;
 String iterateId = "";
 String currentComponent = "";
 java.util.Map<String, Object> resourceMap = new java.util.HashMap<String, Object>();
 try {
  String currentMethodName = new java.lang.Exception()
    .getStackTrace().getMethodName();
  boolean resumeIt = currentMethodName.equals(resumeEntryMethodName);
  if (resumeEntryMethodName == null || resumeIt || globalResumeTicket) {// start
                    // the
                    // resume
   globalResumeTicket = true;
   /**
    *  start
    */
   ok_Hash.put("tDie_4", false);
start_Hash.put("tDie_4", System.currentTimeMillis());
currentComponent = "tDie_4";
int tos_count_tDie_4 = 0;
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tDie_4";
tLogCatcher_1
  .addMessage(
    "tDie",
    "tDie_4",
    5,
    "Some Database Error occured plz. Check Database Settings",
    4);
tLogCatcher_1Process(globalMap);
globalMap.put("tDie_4_DIE_PRIORITY", 5);
System.err
  .println("Some Database Error occured plz. Check Database Settings");
globalMap
  .put("tDie_4_DIE_MESSAGE",
    "Some Database Error occured plz. Check Database Settings");
globalMap
  .put("tDie_4_DIE_MESSAGES",
    "Some Database Error occured plz. Check Database Settings");
currentComponent = "tDie_4";
```

```
  status = "failure";
  errorCode = new Integer(4);
  globalMap.put("tDie_4_DIE_CODE", errorCode);
  if (true) {
   throw new TDieException();
  }
  tos_count_tDie_4++;
  /**
   *  stop
   */
  /**
   *  start
   */
  currentComponent = "tDie_4";
  ok_Hash.put("tDie_4", true);
  end_Hash.put("tDie_4", System.currentTimeMillis());
  /**
   *  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 {
  /**
   *  start
   */
  currentComponent = "tDie_4";
  /**
   *  stop
   */
  } catch (java.lang.Exception e) {
   // ignore
  } catch (java.lang.Error error) {
   // ignore
  }
  resourceMap = null;
 }
 globalMap.put("tDie_4_SUBPROCESS_STATE", 1);
}
public static class row5Struct implements
  routines.system.IPersistableRow<row5Struct> {
 final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 public java.util.Date moment;
 public java.util.Date getMoment() {
  return this.moment;
 }
 public String pid;
 public String getPid() {
  return this.pid;
 }
```

```
public String root_pid;
public String getRoot_pid() {
 return this.root_pid;
}
public String father_pid;
public String getFather_pid() {
 return this.father_pid;
}
public String project;
public String getProject() {
 return this.project;
}
public String job;
public String getJob() {
 return this.job;
}
public String context;
public String getContext() {
 return this.context;
}
public Integer priority;
public Integer getPriority() {
 return this.priority;
}
public String type;
public String getType() {
 return this.type;
}
public String origin;
public String getOrigin() {
 return this.origin;
}
public String message;
public String getMessage() {
 return this.message;
}
public Integer code;
public Integer getCode() {
 return this.code;
}
private java.util.Date readDate(ObjectInputStream dis)
  throws IOException {
 java.util.Date dateReturn = null;
 int length = 0;
 length = dis.readByte();
 if (length == -1) {
  dateReturn = null;
 } else {
  dateReturn = new Date(dis.readLong());
 }
 return dateReturn;
}
private void writeDate(java.util.Date date1, ObjectOutputStream dos)
  throws IOException {
 if (date1 == null) {
  dos.writeByte(-1);
```

```
 } else {
  dos.writeByte(0);
  dos.writeLong(date1.getTime());
 }
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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 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 void writeInteger(Integer intNum, ObjectOutputStream dos)
  throws IOException {
 if (intNum == null) {
  dos.writeByte(-1);
 } else {
  dos.writeByte(0);
```

```
  dos.writeInt(intNum);
 }
}
public void readData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   this.moment = readDate(dis);
   this.pid = readString(dis);
   this.root_pid = readString(dis);
   this.father_pid = readString(dis);
   this.project = readString(dis);
   this.job = readString(dis);
   this.context = readString(dis);
   this.priority = readInteger(dis);
   this.type = readString(dis);
   this.origin = readString(dis);
   this.message = readString(dis);
   this.code = readInteger(dis);
  } catch (IOException e) {
   throw new RuntimeException(e);
  }
 }
}
public void writeData(ObjectOutputStream dos) {
 try {
  // java.util.Date
  writeDate(this.moment, dos);
  // String
  writeString(this.pid, dos);
  // String
  writeString(this.root_pid, dos);
  // String
  writeString(this.father_pid, dos);
  // String
  writeString(this.project, dos);
  // String
  writeString(this.job, dos);
  // String
  writeString(this.context, dos);
  // Integer
  writeInteger(this.priority, dos);
  // String
  writeString(this.type, dos);
  // String
  writeString(this.origin, dos);
  // String
  writeString(this.message, dos);
  // Integer
  writeInteger(this.code, dos);
 } catch (IOException e) {
  throw new RuntimeException(e);
 }
}
public String toString() {
 StringBuilder sb = new StringBuilder();
```

```
 sb.append(super.toString());
 sb.append("");
 return sb.toString();
}
/**
 * Compare keys
 */
public int compareTo(row5Struct 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 tLogCatcher_1Process(
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 globalMap.put("tLogCatcher_1_SUBPROCESS_STATE", 0);
 final boolean execStat = this.execStat;
 String iterateId = "";
 String currentComponent = "";
 java.util.Map<String, Object> resourceMap = new java.util.HashMap<String, Object>();
 try {
  String currentMethodName = new java.lang.Exception()
    .getStackTrace().getMethodName();
  boolean resumeIt = currentMethodName.equals(resumeEntryMethodName);
  if (resumeEntryMethodName == null || resumeIt || globalResumeTicket) {// start
                    // the
                    // resume
   globalResumeTicket = true;
   row5Struct row5 = new row5Struct();
   /**
    *  start
    */
   ok_Hash.put("tSendMail_1", false);
   start_Hash.put("tSendMail_1", System.currentTimeMillis());
   currentComponent = "tSendMail_1";
   int tos_count_tSendMail_1 = 0;
   /**
```

```
 *  stop
 */
/**
 *  start
 */
ok_Hash.put("tLogCatcher_1", false);
start_Hash.put("tLogCatcher_1", System.currentTimeMillis());
currentComponent = "tLogCatcher_1";
int tos_count_tLogCatcher_1 = 0;
for (LogCatcherUtils.LogCatcherMessage lcm : tLogCatcher_1
  .getMessages()) {
 row5.type = lcm.getType();
 row5.origin = (lcm.getOrigin() == null
   || lcm.getOrigin().length() < 1 ? null : lcm
   .getOrigin());
 row5.priority = lcm.getPriority();
 row5.message = lcm.getMessage();
 row5.code = lcm.getCode();
 row5.moment = java.util.Calendar.getInstance().getTime();
 row5.pid = pid;
 row5.root_pid = rootPid;
 row5.father_pid = fatherPid;
 row5.project = projectName;
 row5.job = jobName;
 row5.context = contextStr;
 /**
  *  stop
  */
 /**
  *  start
  */
currentComponent = "tLogCatcher_1";
tos_count_tLogCatcher_1++;
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tSendMail_1";
String smtpHost_tSendMail_1 = "srv72.sbah.local";
String smtpPort_tSendMail_1 = "25";
String from_tSendMail_1 = ("Talend-error@localhost");
String to_tSendMail_1 = ("edv@sba.de").replace(";", ",");
String cc_tSendMail_1 = (("") == null || "".equals("")) ? null
  : ("").replace(";", ",");
String bcc_tSendMail_1 = (("") == null || "".equals("")) ? null
  : ("").replace(";", ",");
String subject_tSendMail_1 = ("Error  from Talend "
  + row5.job + "");
java.util.List<java.util.Map<String, String>> headers_tSendMail_1 = new java.util.ArrayList<java.util.Map<String, String>>();
java.util.List<String> attachments_tSendMail_1 = new java.util.ArrayList<String>();
java.util.List<String> contentTransferEncoding_tSendMail_1 = new java.util.ArrayList<String>();
String message_tSendMail_1 = (("Hello,\n\nThere is a problem.\n\nJob "
  + row5.job
  + "\n\n"
```

```
    + row5.moment
    + "\n"
    + row5.project
    + " ("
    + row5.context
    + ") / "
    + row5.origin
    + "\n"
    + row5.type
    + " : "
    + row5.message + " (" + row5.code + ")\n\nGood Luck.") == null || ""
    .equals("Hello,\n\nThere is a problem.\n\nJob "
      + row5.job + "\n\n" + row5.moment + "\n"
      + row5.project + " (" + row5.context
      + ") / " + row5.origin + "\n" + row5.type
      + " : " + row5.message + " (" + row5.code
      + ")\n\nGood Luck.")) ? "\"\""
    : ("Hello,\n\nThere is a problem.\n\nJob "
      + row5.job + "\n\n" + row5.moment + "\n"
      + row5.project + " (" + row5.context
      + ") / " + row5.origin + "\n" + row5.type
      + " : " + row5.message + " (" + row5.code + ")\n\nGood Luck.");
java.util.Properties props_tSendMail_1 = System
  .getProperties();
props_tSendMail_1.put("mail.smtp.host",
  smtpHost_tSendMail_1);
props_tSendMail_1.put("mail.smtp.port",
  smtpPort_tSendMail_1);
props_tSendMail_1.put("mail.mime.encodefilename", "true");
try {
 props_tSendMail_1.put("mail.smtp.auth", "false");
 javax.mail.Session session_tSendMail_1 = javax.mail.Session
   .getInstance(props_tSendMail_1, null);
 javax.mail.Message msg_tSendMail_1 = new javax.mail.internet.MimeMessage(
   session_tSendMail_1);
 msg_tSendMail_1
   .setFrom(new javax.mail.internet.InternetAddress(
     from_tSendMail_1, null));
 msg_tSendMail_1.setRecipients(
   javax.mail.Message.RecipientType.TO,
   javax.mail.internet.InternetAddress.parse(
     to_tSendMail_1, false));
 if (cc_tSendMail_1 != null)
  msg_tSendMail_1.setRecipients(
    javax.mail.Message.RecipientType.CC,
    javax.mail.internet.InternetAddress.parse(
      cc_tSendMail_1, false));
 if (bcc_tSendMail_1 != null)
  msg_tSendMail_1.setRecipients(
    javax.mail.Message.RecipientType.BCC,
    javax.mail.internet.InternetAddress.parse(
      bcc_tSendMail_1, false));
 msg_tSendMail_1.setSubject(subject_tSendMail_1);
 for (int i_tSendMail_1 = 0; i_tSendMail_1 < headers_tSendMail_1
   .size(); i_tSendMail_1++) {
  java.util.Map<String, String> header_tSendMail_1 = headers_tSendMail_1
```

```
   .get(i_tSendMail_1);
 msg_tSendMail_1.setHeader(
   header_tSendMail_1.get("KEY"),
   header_tSendMail_1.get("VALUE"));
}
msg_tSendMail_1.setSentDate(new Date());
msg_tSendMail_1.setHeader("X-Priority", "5"); // High->1
            // Normal->3
            // Low->5
javax.mail.Multipart mp_tSendMail_1 = new javax.mail.internet.MimeMultipart();
javax.mail.internet.MimeBodyPart mbpText_tSendMail_1 = new javax.mail.internet.MimeBodyPart();
mbpText_tSendMail_1.setText(message_tSendMail_1,
  "ISO-8859-15", "plain");
mp_tSendMail_1.addBodyPart(mbpText_tSendMail_1);
javax.mail.internet.MimeBodyPart mbpFile_tSendMail_1 = null;
for (int i_tSendMail_1 = 0; i_tSendMail_1 < attachments_tSendMail_1
  .size(); i_tSendMail_1++) {
 String filename_tSendMail_1 = attachments_tSendMail_1
   .get(i_tSendMail_1);
 javax.activation.FileDataSource fds_tSendMail_1 = null;
 java.io.File file_tSendMail_1 = new java.io.File(
   filename_tSendMail_1);
 if (!file_tSendMail_1.exists()) {
  continue;
 }
 if (file_tSendMail_1.isDirectory()) {
  java.io.File[] subFiles_tSendMail_1 = file_tSendMail_1
    .listFiles();
  for (java.io.File subFile_tSendMail_1 : subFiles_tSendMail_1) {
   if (subFile_tSendMail_1.isFile()) {
    fds_tSendMail_1 = new javax.activation.FileDataSource(
      subFile_tSendMail_1
        .getAbsolutePath());
    mbpFile_tSendMail_1 = new javax.mail.internet.MimeBodyPart();
    mbpFile_tSendMail_1
      .setDataHandler(new javax.activation.DataHandler(
        fds_tSendMail_1));
    mbpFile_tSendMail_1
      .setFileName(javax.mail.internet.MimeUtility
        .encodeText(fds_tSendMail_1
          .getName()));
    if (contentTransferEncoding_tSendMail_1
      .get(i_tSendMail_1)
      .equalsIgnoreCase("base64")) {
     mbpFile_tSendMail_1
       .setHeader(
         "Content-Transfer-Encoding",
         "base64");
    }
    mp_tSendMail_1
      .addBodyPart(mbpFile_tSendMail_1);
   }
  }
 } else {
  mbpFile_tSendMail_1 = new javax.mail.internet.MimeBodyPart();
  fds_tSendMail_1 = new javax.activation.FileDataSource(
```

```
    filename_tSendMail_1);
  mbpFile_tSendMail_1
    .setDataHandler(new javax.activation.DataHandler(
      fds_tSendMail_1));
  mbpFile_tSendMail_1
    .setFileName(javax.mail.internet.MimeUtility
      .encodeText(fds_tSendMail_1
        .getName()));
  if (contentTransferEncoding_tSendMail_1.get(
    i_tSendMail_1).equalsIgnoreCase(
    "base64")) {
   mbpFile_tSendMail_1.setHeader(
     "Content-Transfer-Encoding",
     "base64");
  }
  mp_tSendMail_1.addBodyPart(mbpFile_tSendMail_1);
 }
}
// -- set the content --
msg_tSendMail_1.setContent(mp_tSendMail_1);
// add handlers for main MIME types
javax.activation.MailcapCommandMap mc_tSendMail_1 = (javax.activation.MailcapCommandMap) javax.activation.CommandMap
  .getDefaultCommandMap();
mc_tSendMail_1
  .addMailcap("text/html;; x-java-content-handler=com.sun.mail.handlers.text_html");
mc_tSendMail_1
  .addMailcap("text/xml;; x-java-content-handler=com.sun.mail.handlers.text_xml");
mc_tSendMail_1
  .addMailcap("text/plain;; x-java-content-handler=com.sun.mail.handlers.text_plain");
 mc_tSendMail_1
  .addMailcap("multipart/*;; x-java-content-handler=com.sun.mail.handlers.multipart_mixed");
 mc_tSendMail_1
  .addMailcap("message/rfc822;; x-java-content-handler=com.sun.mail.handlers.message_rfc822");
 javax.activation.CommandMap
  .setDefaultCommandMap(mc_tSendMail_1);
 // -- Send the message --
 javax.mail.Transport.send(msg_tSendMail_1);
} catch (java.lang.Exception e) {
 System.err.println(e.toString());
} finally {
 props_tSendMail_1.remove("mail.smtp.host");
 props_tSendMail_1.remove("mail.smtp.port");
 props_tSendMail_1.remove("mail.mime.encodefilename");
 props_tSendMail_1.remove("mail.smtp.auth");
}
tos_count_tSendMail_1++;
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tLogCatcher_1";
}
ok_Hash.put("tLogCatcher_1", true);
end_Hash.put("tLogCatcher_1", System.currentTimeMillis());
```

```
  /**
   *  stop
   */
  /**
   *  start
   */
  currentComponent = "tSendMail_1";
  ok_Hash.put("tSendMail_1", true);
  end_Hash.put("tSendMail_1", System.currentTimeMillis());
  /**
   *  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 {
  /**
   *  start
   */
  currentComponent = "tLogCatcher_1";
  /**
   *  stop
   */
    /**
     *  start
     */
    currentComponent = "tSendMail_1";
    /**
     *  stop
     */
  } catch (java.lang.Exception e) {
   // ignore
  } catch (java.lang.Error error) {
   // ignore
  }
  resourceMap = null;
 }
 globalMap.put("tLogCatcher_1_SUBPROCESS_STATE", 1);
}
public static class row1Struct implements
  routines.system.IPersistableRow<row1Struct> {
 final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 public String ABC_Klassifizierung__c;
 public String getABC_Klassifizierung__c() {
  return this.ABC_Klassifizierung__c;
 }
 public Double Aktueller_Umsatz__c;
 public Double getAktueller_Umsatz__c() {
  return this.Aktueller_Umsatz__c;
 }
```

```
public java.util.Date Bearbeitungsdatum__c;
public java.util.Date getBearbeitungsdatum__c() {
 return this.Bearbeitungsdatum__c;
}
public String Bemerkung_Infor__c;
public String getBemerkung_Infor__c() {
 return this.Bemerkung_Infor__c;
}
public String BillingCity;
public String getBillingCity() {
 return this.BillingCity;
}
public String BillingCountry;
public String getBillingCountry() {
 return this.BillingCountry;
}
public String BillingPostalCode;
public String getBillingPostalCode() {
 return this.BillingPostalCode;
}
public String BillingState;
public String getBillingState() {
 return this.BillingState;
}
public String BillingStreet;
public String getBillingStreet() {
 return this.BillingStreet;
}
public java.util.Date Erstelldatum__c;
public java.util.Date getErstelldatum__c() {
 return this.Erstelldatum__c;
}
public String KTXT__c;
public String getKTXT__c() {
 return this.KTXT__c;
}
public String Lieferbedingungen__c;
public String getLieferbedingungen__c() {
 return this.Lieferbedingungen__c;
}
public String Name;
public String getName() {
 return this.Name;
}
public String Spedition_Versand__c;
public String getSpedition_Versand__c() {
 return this.Spedition_Versand__c;
}
public String UST_ID_NR__c;
public String getUST_ID_NR__c() {
 return this.UST_ID_NR__c;
}
public Double Umsatz_VorVorjahr__c;
public Double getUmsatz_VorVorjahr__c() {
 return this.Umsatz_VorVorjahr__c;
}
```

```
public Double Umsatz_Vorjahr__c;
public Double getUmsatz_Vorjahr__c() {
 return this.Umsatz_Vorjahr__c;
}
public String VertreterNummer__c;
public String getVertreterNummer__c() {
 return this.VertreterNummer__c;
}
public String Verwendung__c;
public String getVerwendung__c() {
 return this.Verwendung__c;
}
public String Zahlungsbedingungen__c;
public String getZahlungsbedingungen__c() {
 return this.Zahlungsbedingungen__c;
}
public String f42_INFOR_FirmenNr__c;
public String getF42_INFOR_FirmenNr__c() {
 return this.f42_INFOR_FirmenNr__c;
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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 java.util.Date readDate(ObjectInputStream dis)
  throws IOException {
```

```
 java.util.Date dateReturn = null;
 int length = 0;
 length = dis.readByte();
 if (length == -1) {
  dateReturn = null;
 } else {
  dateReturn = new Date(dis.readLong());
 }
 return dateReturn;
}
private void writeDate(java.util.Date date1, ObjectOutputStream dos)
  throws IOException {
 if (date1 == null) {
  dos.writeByte(-1);
 } else {
  dos.writeByte(0);
  dos.writeLong(date1.getTime());
 }
}
public void readData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   this.ABC_Klassifizierung__c = readString(dis);
   length = dis.readByte();
   if (length == -1) {
    this.Aktueller_Umsatz__c = null;
   } else {
    this.Aktueller_Umsatz__c = dis.readDouble();
   }
   this.Bearbeitungsdatum__c = readDate(dis);
   this.Bemerkung_Infor__c = readString(dis);
   this.BillingCity = readString(dis);
   this.BillingCountry = readString(dis);
   this.BillingPostalCode = readString(dis);
   this.BillingState = readString(dis);
   this.BillingStreet = readString(dis);
   this.Erstelldatum__c = readDate(dis);
   this.KTXT__c = readString(dis);
   this.Lieferbedingungen__c = readString(dis);
   this.Name = readString(dis);
   this.Spedition_Versand__c = readString(dis);
   this.UST_ID_NR__c = readString(dis);
   length = dis.readByte();
   if (length == -1) {
    this.Umsatz_VorVorjahr__c = null;
   } else {
    this.Umsatz_VorVorjahr__c = dis.readDouble();
   }
   length = dis.readByte();
   if (length == -1) {
    this.Umsatz_Vorjahr__c = null;
   } else {
    this.Umsatz_Vorjahr__c = dis.readDouble();
   }
   this.VertreterNummer__c = readString(dis);
```

```
   this.Verwendung__c = readString(dis);
   this.Zahlungsbedingungen__c = readString(dis);
   this.f42_INFOR_FirmenNr__c = readString(dis);
  } catch (IOException e) {
   throw new RuntimeException(e);
  }
 }
}
public void writeData(ObjectOutputStream dos) {
 try {
  // String
  writeString(this.ABC_Klassifizierung__c, dos);
  // Double
  if (this.Aktueller_Umsatz__c == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeDouble(this.Aktueller_Umsatz__c);
  }
  // java.util.Date
  writeDate(this.Bearbeitungsdatum__c, dos);
  // String
  writeString(this.Bemerkung_Infor__c, dos);
  // String
  writeString(this.BillingCity, dos);
  // String
  writeString(this.BillingCountry, dos);
  // String
  writeString(this.BillingPostalCode, dos);
  // String
  writeString(this.BillingState, dos);
  // String
  writeString(this.BillingStreet, dos);
  // java.util.Date
  writeDate(this.Erstelldatum__c, dos);
  // String
  writeString(this.KTXT__c, dos);
  // String
  writeString(this.Lieferbedingungen__c, dos);
  // String
  writeString(this.Name, dos);
  // String
  writeString(this.Spedition_Versand__c, dos);
  // String
  writeString(this.UST_ID_NR__c, dos);
  // Double
  if (this.Umsatz_VorVorjahr__c == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeDouble(this.Umsatz_VorVorjahr__c);
  }
  // Double
  if (this.Umsatz_Vorjahr__c == null) {
   dos.writeByte(-1);
  } else {
```

```
    dos.writeByte(0);
    dos.writeDouble(this.Umsatz_Vorjahr__c);
   }
   // String
   writeString(this.VertreterNummer__c, dos);
   // String
   writeString(this.Verwendung__c, dos);
   // String
   writeString(this.Zahlungsbedingungen__c, dos);
   // String
   writeString(this.f42_INFOR_FirmenNr__c, dos);
  } catch (IOException e) {
   throw new RuntimeException(e);
  }
 }
 public String toString() {
  StringBuilder sb = new StringBuilder();
  sb.append(super.toString());
  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 static class out_sfStruct implements
  routines.system.IPersistableRow<out_sfStruct> {
 final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 public Float Aktueller_Umsatz__c;
 public Float getAktueller_Umsatz__c() {
  return this.Aktueller_Umsatz__c;
 }
```

```
public java.util.Date Bearbeitungsdatum__c;
public java.util.Date getBearbeitungsdatum__c() {
 return this.Bearbeitungsdatum__c;
}
public String BillingCity;
public String getBillingCity() {
 return this.BillingCity;
}
public String BillingCountry;
public String getBillingCountry() {
 return this.BillingCountry;
}
public String BillingPostalCode;
public String getBillingPostalCode() {
 return this.BillingPostalCode;
}
public String BillingState;
public String getBillingState() {
 return this.BillingState;
}
public String ABC_Klassifizierung__c;
public String getABC_Klassifizierung__c() {
 return this.ABC_Klassifizierung__c;
}
public String BillingStreet;
public String getBillingStreet() {
 return this.BillingStreet;
}
public java.util.Date Erstelldatum__c;
public java.util.Date getErstelldatum__c() {
 return this.Erstelldatum__c;
}
public String Name;
public String getName() {
 return this.Name;
}
public String UST_ID_NR__c;
public String getUST_ID_NR__c() {
 return this.UST_ID_NR__c;
}
public Float Umsatz_VorVorjahr__c;
public Float getUmsatz_VorVorjahr__c() {
 return this.Umsatz_VorVorjahr__c;
}
public Float Umsatz_Vorjahr__c;
public Float getUmsatz_Vorjahr__c() {
 return this.Umsatz_Vorjahr__c;
}
public String VertreterNummer__c;
public String getVertreterNummer__c() {
 return this.VertreterNummer__c;
}
public BigDecimal Verwendung__c;
public BigDecimal getVerwendung__c() {
 return this.Verwendung__c;
}
```

```
public String f42_INFOR_FirmenNr__c;
public String getF42_INFOR_FirmenNr__c() {
 return this.f42_INFOR_FirmenNr__c;
}
public String Bemerkung_Infor__c;
public String getBemerkung_Infor__c() {
 return this.Bemerkung_Infor__c;
}
public String Lieferbedingungen__c;
public String getLieferbedingungen__c() {
 return this.Lieferbedingungen__c;
}
public String Spedition_Versand__c;
public String getSpedition_Versand__c() {
 return this.Spedition_Versand__c;
}
public String Zahlungsbedingungen__c;
public String getZahlungsbedingungen__c() {
 return this.Zahlungsbedingungen__c;
}
public String KTXT__c;
public String getKTXT__c() {
 return this.KTXT__c;
}
private java.util.Date readDate(ObjectInputStream dis)
  throws IOException {
 java.util.Date dateReturn = null;
 int length = 0;
 length = dis.readByte();
 if (length == -1) {
  dateReturn = null;
 } else {
  dateReturn = new Date(dis.readLong());
 }
 return dateReturn;
}
private void writeDate(java.util.Date date1, ObjectOutputStream dos)
  throws IOException {
 if (date1 == null) {
  dos.writeByte(-1);
 } else {
  dos.writeByte(0);
  dos.writeLong(date1.getTime());
 }
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
```

```
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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);
 }
}
public void readData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   length = dis.readByte();
if (length == -1) {
 this.Aktueller_Umsatz__c = null;
} else {
 this.Aktueller_Umsatz__c = dis.readFloat();
}
this.Bearbeitungsdatum__c = readDate(dis);
this.BillingCity = readString(dis);
this.BillingCountry = readString(dis);
this.BillingPostalCode = readString(dis);
this.BillingState = readString(dis);
this.ABC_Klassifizierung__c = readString(dis);
this.BillingStreet = readString(dis);
this.Erstelldatum__c = readDate(dis);
this.Name = readString(dis);
this.UST_ID_NR__c = readString(dis);
length = dis.readByte();
if (length == -1) {
 this.Umsatz_VorVorjahr__c = null;
} else {
 this.Umsatz_VorVorjahr__c = dis.readFloat();
}
length = dis.readByte();
if (length == -1) {
 this.Umsatz_Vorjahr__c = null;
} else {
 this.Umsatz_Vorjahr__c = dis.readFloat();
}
this.VertreterNummer__c = readString(dis);
```

```
    this.Verwendung__c = (BigDecimal) dis.readObject();
    this.f42_INFOR_FirmenNr__c = readString(dis);
    this.Bemerkung_Infor__c = readString(dis);
    this.Lieferbedingungen__c = readString(dis);
    this.Spedition_Versand__c = readString(dis);
    this.Zahlungsbedingungen__c = readString(dis);
    this.KTXT__c = readString(dis);
   } catch (IOException e) {
    throw new RuntimeException(e);
   } catch (ClassNotFoundException eCNFE) {
    throw new RuntimeException(eCNFE);
   }
  }
}
public void writeData(ObjectOutputStream dos) {
 try {
  // Float
  if (this.Aktueller_Umsatz__c == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeFloat(this.Aktueller_Umsatz__c);
  }
  // java.util.Date
  writeDate(this.Bearbeitungsdatum__c, dos);
  // String
  writeString(this.BillingCity, dos);
  // String
  writeString(this.BillingCountry, dos);
  // String
  writeString(this.BillingPostalCode, dos);
  // String
  writeString(this.BillingState, dos);
  // String
  writeString(this.ABC_Klassifizierung__c, dos);
  // String
  writeString(this.BillingStreet, dos);
  // java.util.Date
  writeDate(this.Erstelldatum__c, dos);
  // String
  writeString(this.Name, dos);
  // String
  writeString(this.UST_ID_NR__c, dos);
  // Float
  if (this.Umsatz_VorVorjahr__c == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeFloat(this.Umsatz_VorVorjahr__c);
  }
  // Float
  if (this.Umsatz_Vorjahr__c == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeFloat(this.Umsatz_Vorjahr__c);
```

```
    }
    // String
    writeString(this.VertreterNummer__c, dos);
    // BigDecimal
    dos.writeObject(this.Verwendung__c);
    // String
    writeString(this.f42_INFOR_FirmenNr__c, dos);
    // String
    writeString(this.Bemerkung_Infor__c, dos);
    // String
    writeString(this.Lieferbedingungen__c, dos);
    // String
    writeString(this.Spedition_Versand__c, dos);
    // String
    writeString(this.Zahlungsbedingungen__c, dos);
    // String
    writeString(this.KTXT__c, dos);
   } catch (IOException e) {
    throw new RuntimeException(e);
   }
  }
  public String toString() {
   StringBuilder sb = new StringBuilder();
   sb.append(super.toString());
   sb.append("");
   return sb.toString();
  }
  /**
   * Compare keys
   */
   public int compareTo(out_sfStruct 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 row2Struct implements
    routines.system.IPersistableRow<row2Struct> {
```

```
final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
public String FIRMANR;
public String getFIRMANR() {
 return this.FIRMANR;
}
public String FIRMANR1;
public String getFIRMANR1() {
 return this.FIRMANR1;
}
public String STRASSE;
public String getSTRASSE() {
 return this.STRASSE;
}
public String ORT;
public String getORT() {
 return this.ORT;
}
public String LAND;
public String getLAND() {
 return this.LAND;
}
public String STAAT;
public String getSTAAT() {
 return this.STAAT;
}
public String PLZORT;
public String getPLZORT() {
 return this.PLZORT;
}
public String POSTFACH;
public String getPOSTFACH() {
 return this.POSTFACH;
}
public String VERTRETER1;
public String getVERTRETER1() {
 return this.VERTRETER1;
}
public String LANDKNG;
public String getLANDKNG() {
 return this.LANDKNG;
}
public String ANSCHRIFTNR;
public String getANSCHRIFTNR() {
 return this.ANSCHRIFTNR;
}
public String ABCKLAS;
public String getABCKLAS() {
 return this.ABCKLAS;
}
public String USTIDNR;
public String getUSTIDNR() {
 return this.USTIDNR;
}
public String KTXT;
public String getKTXT() {
```

```
 return this.KTXT;
}
public String BEMERKUNG;
public String getBEMERKUNG() {
 return this.BEMERKUNG;
}
public BigDecimal VERWENDUNG1;
public BigDecimal getVERWENDUNG1() {
 return this.VERWENDUNG1;
}
public java.util.Date CREATEDATE;
public java.util.Date getCREATEDATE() {
 return this.CREATEDATE;
}
public java.util.Date MODIFYDATE;
public java.util.Date getMODIFYDATE() {
 return this.MODIFYDATE;
}
public String ABCKLAS1;
public String getABCKLAS1() {
 return this.ABCKLAS1;
}
public String NAME;
public String getNAME() {
 return this.NAME;
}
public String TEXT0;
public String getTEXT0() {
 return this.TEXT0;
}
public String TEXT1;
public String getTEXT1() {
 return this.TEXT1;
}
public String TEXT2;
public String getTEXT2() {
 return this.TEXT2;
}
public String TEXT3;
public String getTEXT3() {
 return this.TEXT3;
}
public String TEXT4;
public String getTEXT4() {
 return this.TEXT4;
}
public BigDecimal ZBED;
public BigDecimal getZBED() {
 return this.ZBED;
}
public String LIEFERBEDINGUNGEN;
public String getLIEFERBEDINGUNGEN() {
 return this.LIEFERBEDINGUNGEN;
}
public String Spedition_Versand;
public String getSpedition_Versand() {
```

```
 return this.Spedition_Versand;
}
public String ZAHLUNGSBEDINGUNGEN;
public String getZAHLUNGSBEDINGUNGEN() {
 return this.ZAHLUNGSBEDINGUNGEN;
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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 java.util.Date readDate(ObjectInputStream dis)
  throws IOException {
 java.util.Date dateReturn = null;
 int length = 0;
 length = dis.readByte();
 if (length == -1) {
  dateReturn = null;
 } else {
  dateReturn = new Date(dis.readLong());
 }
 return dateReturn;
}
private void writeDate(java.util.Date date1, ObjectOutputStream dos)
  throws IOException {
 if (date1 == null) {
  dos.writeByte(-1);
```

```
 } else {
  dos.writeByte(0);
  dos.writeLong(date1.getTime());
 }
}
public void readData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   this.FIRMANR = readString(dis);
   this.FIRMANR1 = readString(dis);
   this.STRASSE = readString(dis);
   this.ORT = readString(dis);
   this.LAND = readString(dis);
   this.STAAT = readString(dis);
   this.PLZORT = readString(dis);
   this.POSTFACH = readString(dis);
   this.VERTRETER1 = readString(dis);
   this.LANDKNG = readString(dis);
   this.ANSCHRIFTNR = readString(dis);
   this.ABCKLAS = readString(dis);
   this.USTIDNR = readString(dis);
   this.KTXT = readString(dis);
   this.BEMERKUNG = readString(dis);
   this.VERWENDUNG1 = (BigDecimal) dis.readObject();
   this.CREATEDATE = readDate(dis);
   this.MODIFYDATE = readDate(dis);
   this.ABCKLAS1 = readString(dis);
   this.NAME = readString(dis);
   this.TEXT0 = readString(dis);
   this.TEXT1 = readString(dis);
   this.TEXT2 = readString(dis);
   this.TEXT3 = readString(dis);
   this.TEXT4 = readString(dis);
   this.ZBED = (BigDecimal) dis.readObject();
   this.LIEFERBEDINGUNGEN = readString(dis);
   this.Spedition_Versand = readString(dis);
   this.ZAHLUNGSBEDINGUNGEN = readString(dis);
  } catch (IOException e) {
   throw new RuntimeException(e);
  } catch (ClassNotFoundException eCNFE) {
   throw new RuntimeException(eCNFE);
  }
 }
}
public void writeData(ObjectOutputStream dos) {
 try {
  // String
  writeString(this.FIRMANR, dos);
  // String
  writeString(this.FIRMANR1, dos);
  // String
  writeString(this.STRASSE, dos);
  // String
  writeString(this.ORT, dos);
  // String
```

```
writeString(this.LAND, dos);
// String
writeString(this.STAAT, dos);
// String
writeString(this.PLZORT, dos);
// String
writeString(this.POSTFACH, dos);
// String
writeString(this.VERTRETER1, dos);
// String
writeString(this.LANDKNG, dos);
// String
writeString(this.ANSCHRIFTNR, dos);
// String
writeString(this.ABCKLAS, dos);
// String
writeString(this.USTIDNR, dos);
// String
writeString(this.KTXT, dos);
// String
writeString(this.BEMERKUNG, dos);
// BigDecimal
dos.writeObject(this.VERWENDUNG1);
// java.util.Date
writeDate(this.CREATEDATE, dos);
// java.util.Date
writeDate(this.MODIFYDATE, dos);
// String
 writeString(this.ABCKLAS1, dos);
 // String
 writeString(this.NAME, dos);
 // String
 writeString(this.TEXT0, dos);
 // String
 writeString(this.TEXT1, dos);
 // String
 writeString(this.TEXT2, dos);
 // String
 writeString(this.TEXT3, dos);
 // String
 writeString(this.TEXT4, dos);
 // BigDecimal
 dos.writeObject(this.ZBED);
 // String
 writeString(this.LIEFERBEDINGUNGEN, dos);
 // String
 writeString(this.Spedition_Versand, dos);
 // String
 writeString(this.ZAHLUNGSBEDINGUNGEN, dos);
 } catch (IOException e) {
  throw new RuntimeException(e);
 }
}
public String toString() {
 StringBuilder sb = new StringBuilder();
 sb.append(super.toString());
```

```
 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 after_tOracleInput_2Struct implements
  routines.system.IPersistableRow<after_tOracleInput_2Struct> {
 final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 public String FIRMANR;
 public String getFIRMANR() {
  return this.FIRMANR;
 }
 public String FIRMANR1;
 public String getFIRMANR1() {
  return this.FIRMANR1;
 }
 public String STRASSE;
 public String getSTRASSE() {
  return this.STRASSE;
 }
 public String ORT;
 public String getORT() {
  return this.ORT;
 }
 public String LAND;
 public String getLAND() {
  return this.LAND;
 }
 public String STAAT;
 public String getSTAAT() {
```

```
 return this.STAAT;
}
public String PLZORT;
public String getPLZORT() {
 return this.PLZORT;
}
public String POSTFACH;
public String getPOSTFACH() {
 return this.POSTFACH;
}
public String VERTRETER1;
public String getVERTRETER1() {
 return this.VERTRETER1;
}
public String LANDKNG;
public String getLANDKNG() {
 return this.LANDKNG;
}
public String ANSCHRIFTNR;
public String getANSCHRIFTNR() {
 return this.ANSCHRIFTNR;
}
public String ABCKLAS;
public String getABCKLAS() {
 return this.ABCKLAS;
}
public String USTIDNR;
public String getUSTIDNR() {
 return this.USTIDNR;
}
public String KTXT;
public String getKTXT() {
 return this.KTXT;
}
public String BEMERKUNG;
public String getBEMERKUNG() {
 return this.BEMERKUNG;
}
public BigDecimal VERWENDUNG1;
public BigDecimal getVERWENDUNG1() {
 return this.VERWENDUNG1;
}
public java.util.Date CREATEDATE;
public java.util.Date getCREATEDATE() {
 return this.CREATEDATE;
}
public java.util.Date MODIFYDATE;
public java.util.Date getMODIFYDATE() {
 return this.MODIFYDATE;
}
public String ABCKLAS1;
public String getABCKLAS1() {
 return this.ABCKLAS1;
}
public String NAME;
public String getNAME() {
```

```
 return this.NAME;
}
public String TEXT0;
public String getTEXT0() {
 return this.TEXT0;
}
public String TEXT1;
public String getTEXT1() {
 return this.TEXT1;
}
public String TEXT2;
public String getTEXT2() {
 return this.TEXT2;
}
public String TEXT3;
public String getTEXT3() {
 return this.TEXT3;
}
public String TEXT4;
public String getTEXT4() {
 return this.TEXT4;
}
public BigDecimal ZBED;
public BigDecimal getZBED() {
 return this.ZBED;
}
public String LIEFERBEDINGUNGEN;
public String getLIEFERBEDINGUNGEN() {
 return this.LIEFERBEDINGUNGEN;
}
public String Spedition_Versand;
public String getSpedition_Versand() {
 return this.Spedition_Versand;
}
public String ZAHLUNGSBEDINGUNGEN;
public String getZAHLUNGSBEDINGUNGEN() {
 return this.ZAHLUNGSBEDINGUNGEN;
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
```

```
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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 java.util.Date readDate(ObjectInputStream dis)
  throws IOException {
 java.util.Date dateReturn = null;
 int length = 0;
 length = dis.readByte();
 if (length == -1) {
  dateReturn = null;
 } else {
  dateReturn = new Date(dis.readLong());
 }
 return dateReturn;
}
private void writeDate(java.util.Date date1, ObjectOutputStream dos)
  throws IOException {
 if (date1 == null) {
  dos.writeByte(-1);
 } else {
  dos.writeByte(0);
  dos.writeLong(date1.getTime());
 }
}
public void readData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   this.FIRMANR = readString(dis);
   this.FIRMANR1 = readString(dis);
   this.STRASSE = readString(dis);
   this.ORT = readString(dis);
   this.LAND = readString(dis);
   this.STAAT = readString(dis);
   this.PLZORT = readString(dis);
   this.POSTFACH = readString(dis);
   this.VERTRETER1 = readString(dis);
   this.LANDKNG = readString(dis);
   this.ANSCHRIFTNR = readString(dis);
   this.ABCKLAS = readString(dis);
   this.USTIDNR = readString(dis);
   this.KTXT = readString(dis);
   this.BEMERKUNG = readString(dis);
```

```
    this.VERWENDUNG1 = (BigDecimal) dis.readObject();
    this.CREATEDATE = readDate(dis);
    this.MODIFYDATE = readDate(dis);
    this.ABCKLAS1 = readString(dis);
    this.NAME = readString(dis);
    this.TEXT0 = readString(dis);
    this.TEXT1 = readString(dis);
    this.TEXT2 = readString(dis);
    this.TEXT3 = readString(dis);
    this.TEXT4 = readString(dis);
    this.ZBED = (BigDecimal) dis.readObject();
    this.LIEFERBEDINGUNGEN = readString(dis);
    this.Spedition_Versand = readString(dis);
    this.ZAHLUNGSBEDINGUNGEN = readString(dis);
   } catch (IOException e) {
    throw new RuntimeException(e);
   } catch (ClassNotFoundException eCNFE) {
    throw new RuntimeException(eCNFE);
   }
  }
}
public void writeData(ObjectOutputStream dos) {
 try {
  // String
  writeString(this.FIRMANR, dos);
  // String
  writeString(this.FIRMANR1, dos);
  // String
  writeString(this.STRASSE, dos);
  // String
  writeString(this.ORT, dos);
  // String
  writeString(this.LAND, dos);
  // String
  writeString(this.STAAT, dos);
  // String
  writeString(this.PLZORT, dos);
  // String
  writeString(this.POSTFACH, dos);
  // String
  writeString(this.VERTRETER1, dos);
  // String
  writeString(this.LANDKNG, dos);
  // String
  writeString(this.ANSCHRIFTNR, dos);
  // String
  writeString(this.ABCKLAS, dos);
  // String
  writeString(this.USTIDNR, dos);
  // String
  writeString(this.KTXT, dos);
  // String
  writeString(this.BEMERKUNG, dos);
  // BigDecimal
  dos.writeObject(this.VERWENDUNG1);
  // java.util.Date
```

```
 writeDate(this.CREATEDATE, dos);
 // java.util.Date
 writeDate(this.MODIFYDATE, dos);
 // String
 writeString(this.ABCKLAS1, dos);
 // String
 writeString(this.NAME, dos);
 // String
 writeString(this.TEXT0, dos);
 // String
 writeString(this.TEXT1, dos);
 // String
 writeString(this.TEXT2, dos);
 // String
 writeString(this.TEXT3, dos);
 // String
 writeString(this.TEXT4, dos);
 // BigDecimal
 dos.writeObject(this.ZBED);
 // String
 writeString(this.LIEFERBEDINGUNGEN, dos);
 // String
 writeString(this.Spedition_Versand, dos);
 // String
 writeString(this.ZAHLUNGSBEDINGUNGEN, dos);
} catch (IOException e) {
 throw new RuntimeException(e);
}
}
public String toString() {
 StringBuilder sb = new StringBuilder();
 sb.append(super.toString());
 sb.append("");
 return sb.toString();
}
/**
 * Compare keys
 */
public int compareTo(after_tOracleInput_2Struct 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 tOracleInput_2Process(
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 globalMap.put("tOracleInput_2_SUBPROCESS_STATE", 0);
 final boolean execStat = this.execStat;
 String iterateId = "";
 String currentComponent = "";
 java.util.Map<String, Object> resourceMap = new java.util.HashMap<String, Object>();
 try {
  String currentMethodName = new java.lang.Exception()
    .getStackTrace().getMethodName();
  boolean resumeIt = currentMethodName.equals(resumeEntryMethodName);
  if (resumeEntryMethodName == null || resumeIt || globalResumeTicket) {// start
                    // the
                    // resume
   globalResumeTicket = true;
   tOracleInput_3Process(globalMap);
   row2Struct row2 = new row2Struct();
   out_sfStruct out_sf = new out_sfStruct();
   row1Struct row1 = new row1Struct();
   /**
    *  start
    */
ok_Hash.put("tSalesforceOutput_1", false);
start_Hash.put("tSalesforceOutput_1",
  System.currentTimeMillis());
currentComponent = "tSalesforceOutput_1";
int tos_count_tSalesforceOutput_1 = 0;
int nb_line_tSalesforceOutput_1 = 0;
int nb_success_tSalesforceOutput_1 = 0;
int nb_reject_tSalesforceOutput_1 = 0;
System.setProperty("org.apache.commons.logging.Log",
  "org.apache.commons.logging.impl.NoOpLog");
org.talend.salesforce.SforceConnection sforceConn_tSalesforceOutput_1 = null;
final String decryptedPassword_tSalesforceOutput_1 = routines.system.PasswordEncryptUtil
  .decryptPassword("e4583c29fe79de56740d90ac22eb1f1926c80f6f4eeb7649d4c053b42a70f2f1f39ec871341f632507e2f2d0547b9f97");
sforceConn_tSalesforceOutput_1 = new org.talend.salesforce.SforceBasicConnection.Builder(
  "https://login.salesforce.com/services/Soap/u/34.0",
  "sf.admin@sba.de",
  decryptedPassword_tSalesforceOutput_1)
  .setTimeout(60000).needCompression(false)
  .useHttpChunked(true).build();
org.talend.salesforce.SforceManagement sforceManagement_tSalesforceOutput_1 = new org.talend.salesforce.SforceManagementImpl(
  sforceConn_tSalesforceOutput_1, 200, true, "");
org.apache.axiom.om.OMFactory fac_tSalesforceOutput_1 = org.apache.axiom.om.OMAbstractFactory
  .getOMFactory();
/**
 *  stop
 */
/**
```

```
 *  start
 */
ok_Hash.put("tConvertType_1", false);
start_Hash.put("tConvertType_1", System.currentTimeMillis());
currentComponent = "tConvertType_1";
int tos_count_tConvertType_1 = 0;
int nb_line_tConvertType_1 = 0;
/**
 *  stop
 */
/**
 *  start
 */
ok_Hash.put("tMap_2", false);
start_Hash.put("tMap_2", System.currentTimeMillis());
currentComponent = "tMap_2";
int tos_count_tMap_2 = 0;
// ###############################
// # Lookup's keys initialization
org.talend.designer.components.lookup.memory.AdvancedMemoryLookup<row3Struct> tHash_Lookup_row3 = (org.talend.designer.components.lookup.memor

  .get("tHash_Lookup_row3"));
row3Struct row3HashKey = new row3Struct();
row3Struct row3Default = new row3Struct();
// ###############################
// ###############################
// # Vars initialization
class Var__tMap_2__Struct {
}
Var__tMap_2__Struct Var__tMap_2 = new Var__tMap_2__Struct();
// ###############################
// ###############################
// # Outputs initialization
out_sfStruct out_sf_tmp = new out_sfStruct();
// ###############################
/**
 *  stop
 */
/**
 *  start
 */
ok_Hash.put("tOracleInput_2", false);
start_Hash.put("tOracleInput_2", System.currentTimeMillis());
currentComponent = "tOracleInput_2";
int tos_count_tOracleInput_2 = 0;
int nb_line_tOracleInput_2 = 0;
java.sql.Connection conn_tOracleInput_2 = null;
String driverClass_tOracleInput_2 = "oracle.jdbc.driver.OracleDriver";
java.lang.Class.forName(driverClass_tOracleInput_2);
String url_tOracleInput_2 = null;
url_tOracleInput_2 = "jdbc:oracle:thin:@srv43:1591:PROD";
String dbUser_tOracleInput_2 = "SBA_READ";
final String decryptedPassword_tOracleInput_2 = routines.system.PasswordEncryptUtil
  .decryptPassword("b2e1ec96dcc3d7cff4f7aba1746784ea");
String dbPwd_tOracleInput_2 = decryptedPassword_tOracleInput_2;
conn_tOracleInput_2 = java.sql.DriverManager.getConnection(
```

```
  url_tOracleInput_2, dbUser_tOracleInput_2,
  dbPwd_tOracleInput_2);
java.sql.Statement stmtGetTZ_tOracleInput_2 = conn_tOracleInput_2
  .createStatement();
java.sql.ResultSet rsGetTZ_tOracleInput_2 = stmtGetTZ_tOracleInput_2
  .executeQuery("select sessiontimezone from dual");
String sessionTimezone_tOracleInput_2 = java.util.TimeZone
  .getDefault().getID();
while (rsGetTZ_tOracleInput_2.next()) {
 sessionTimezone_tOracleInput_2 = rsGetTZ_tOracleInput_2
   .getString(1);
}
((oracle.jdbc.OracleConnection) conn_tOracleInput_2)
  .setSessionTimeZone(sessionTimezone_tOracleInput_2);
java.sql.Statement stmt_tOracleInput_2 = conn_tOracleInput_2
  .createStatement();
String dbquery_tOracleInput_2 = "SELECT \n  DISTINCT\n (RELFIRMA.FIRMANR),\n RELFIRMA.FIRMANR,\n  RELANSCH.STRASSE,\n  RELANSCH.ORT,\n  RELANS

globalMap.put("tOracleInput_2_QUERY", dbquery_tOracleInput_2);
java.sql.ResultSet rs_tOracleInput_2 = null;
try {
 rs_tOracleInput_2 = stmt_tOracleInput_2
   .executeQuery(dbquery_tOracleInput_2);
 java.sql.ResultSetMetaData rsmd_tOracleInput_2 = rs_tOracleInput_2
   .getMetaData();
 int colQtyInRs_tOracleInput_2 = rsmd_tOracleInput_2
   .getColumnCount();
 String tmpContent_tOracleInput_2 = null;
while (rs_tOracleInput_2.next()) {
 nb_line_tOracleInput_2++;
 if (colQtyInRs_tOracleInput_2 < 1) {
  row2.FIRMANR = null;
 } else {
  tmpContent_tOracleInput_2 = rs_tOracleInput_2
    .getString(1);
  if (tmpContent_tOracleInput_2 != null) {
   row2.FIRMANR = tmpContent_tOracleInput_2.trim();
  } else {
   row2.FIRMANR = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 2) {
  row2.FIRMANR1 = null;
 } else {
  tmpContent_tOracleInput_2 = rs_tOracleInput_2
    .getString(2);
  if (tmpContent_tOracleInput_2 != null) {
   row2.FIRMANR1 = tmpContent_tOracleInput_2
     .trim();
  } else {
   row2.FIRMANR1 = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 3) {
  row2.STRASSE = null;
 } else {
```

```
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(3);
 if (tmpContent_tOracleInput_2 != null) {
  row2.STRASSE = tmpContent_tOracleInput_2.trim();
 } else {
  row2.STRASSE = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 4) {
 row2.ORT = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(4);
 if (tmpContent_tOracleInput_2 != null) {
  row2.ORT = tmpContent_tOracleInput_2.trim();
 } else {
  row2.ORT = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 5) {
 row2.LAND = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(5);
 if (tmpContent_tOracleInput_2 != null) {
  row2.LAND = tmpContent_tOracleInput_2.trim();
 } else {
  row2.LAND = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 6) {
 row2.STAAT = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(6);
 if (tmpContent_tOracleInput_2 != null) {
  row2.STAAT = tmpContent_tOracleInput_2.trim();
 } else {
  row2.STAAT = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 7) {
 row2.PLZORT = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(7);
 if (tmpContent_tOracleInput_2 != null) {
  row2.PLZORT = tmpContent_tOracleInput_2.trim();
 } else {
  row2.PLZORT = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 8) {
 row2.POSTFACH = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
```

```
    .getString(8);
  if (tmpContent_tOracleInput_2 != null) {
   row2.POSTFACH = tmpContent_tOracleInput_2
     .trim();
  } else {
   row2.POSTFACH = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 9) {
  row2.VERTRETER1 = null;
 } else {
  tmpContent_tOracleInput_2 = rs_tOracleInput_2
    .getString(9);
  if (tmpContent_tOracleInput_2 != null) {
   row2.VERTRETER1 = tmpContent_tOracleInput_2
     .trim();
  } else {
   row2.VERTRETER1 = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 10) {
  row2.LANDKNG = null;
 } else {
  tmpContent_tOracleInput_2 = rs_tOracleInput_2
    .getString(10);
  if (tmpContent_tOracleInput_2 != null) {
   row2.LANDKNG = tmpContent_tOracleInput_2.trim();
  } else {
   row2.LANDKNG = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 11) {
  row2.ANSCHRIFTNR = null;
 } else {
  tmpContent_tOracleInput_2 = rs_tOracleInput_2
    .getString(11);
  if (tmpContent_tOracleInput_2 != null) {
   row2.ANSCHRIFTNR = tmpContent_tOracleInput_2
     .trim();
  } else {
   row2.ANSCHRIFTNR = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 12) {
  row2.ABCKLAS = null;
 } else {
  tmpContent_tOracleInput_2 = rs_tOracleInput_2
    .getString(12);
  if (tmpContent_tOracleInput_2 != null) {
   row2.ABCKLAS = tmpContent_tOracleInput_2.trim();
  } else {
   row2.ABCKLAS = null;
  }
 }
 if (colQtyInRs_tOracleInput_2 < 13) {
  row2.USTIDNR = null;
```

```
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(13);
 if (tmpContent_tOracleInput_2 != null) {
  row2.USTIDNR = tmpContent_tOracleInput_2.trim();
 } else {
  row2.USTIDNR = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 14) {
 row2.KTXT = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(14);
 if (tmpContent_tOracleInput_2 != null) {
  row2.KTXT = tmpContent_tOracleInput_2.trim();
 } else {
  row2.KTXT = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 15) {
 row2.BEMERKUNG = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(15);
 if (tmpContent_tOracleInput_2 != null) {
  row2.BEMERKUNG = tmpContent_tOracleInput_2
    .trim();
 } else {
  row2.BEMERKUNG = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 16) {
 row2.VERWENDUNG1 = null;
} else {
 if (rs_tOracleInput_2.getObject(16) != null) {
  row2.VERWENDUNG1 = rs_tOracleInput_2
    .getBigDecimal(16);
 } else {
  row2.VERWENDUNG1 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 17) {
 row2.CREATEDATE = null;
} else {
 if (rs_tOracleInput_2.getTimestamp(17) != null) {
  row2.CREATEDATE = new java.util.Date(
    rs_tOracleInput_2.getTimestamp(17)
      .getTime());
 } else {
  row2.CREATEDATE = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 18) {
 row2.MODIFYDATE = null;
} else {
```

```
 if (rs_tOracleInput_2.getTimestamp(18) != null) {
  row2.MODIFYDATE = new java.util.Date(
    rs_tOracleInput_2.getTimestamp(18)
      .getTime());
 } else {
  row2.MODIFYDATE = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 19) {
 row2.ABCKLAS1 = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(19);
 if (tmpContent_tOracleInput_2 != null) {
  row2.ABCKLAS1 = tmpContent_tOracleInput_2
    .trim();
 } else {
  row2.ABCKLAS1 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 20) {
 row2.NAME = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(20);
 if (tmpContent_tOracleInput_2 != null) {
  row2.NAME = tmpContent_tOracleInput_2.trim();
 } else {
  row2.NAME = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 21) {
 row2.TEXT0 = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(21);
 if (tmpContent_tOracleInput_2 != null) {
  row2.TEXT0 = tmpContent_tOracleInput_2.trim();
 } else {
  row2.TEXT0 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 22) {
 row2.TEXT1 = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(22);
 if (tmpContent_tOracleInput_2 != null) {
  row2.TEXT1 = tmpContent_tOracleInput_2.trim();
 } else {
  row2.TEXT1 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 23) {
 row2.TEXT2 = null;
} else {
```

```
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(23);
 if (tmpContent_tOracleInput_2 != null) {
  row2.TEXT2 = tmpContent_tOracleInput_2.trim();
 } else {
  row2.TEXT2 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 24) {
 row2.TEXT3 = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(24);
 if (tmpContent_tOracleInput_2 != null) {
  row2.TEXT3 = tmpContent_tOracleInput_2.trim();
 } else {
  row2.TEXT3 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 25) {
 row2.TEXT4 = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(25);
 if (tmpContent_tOracleInput_2 != null) {
  row2.TEXT4 = tmpContent_tOracleInput_2.trim();
 } else {
  row2.TEXT4 = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 26) {
 row2.ZBED = null;
} else {
 if (rs_tOracleInput_2.getObject(26) != null) {
  row2.ZBED = rs_tOracleInput_2.getBigDecimal(26);
 } else {
  row2.ZBED = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 27) {
 row2.LIEFERBEDINGUNGEN = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(27);
 if (tmpContent_tOracleInput_2 != null) {
  row2.LIEFERBEDINGUNGEN = tmpContent_tOracleInput_2
    .trim();
 } else {
  row2.LIEFERBEDINGUNGEN = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 28) {
 row2.Spedition_Versand = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(28);
```

```
 if (tmpContent_tOracleInput_2 != null) {
  row2.Spedition_Versand = tmpContent_tOracleInput_2
    .trim();
 } else {
  row2.Spedition_Versand = null;
 }
}
if (colQtyInRs_tOracleInput_2 < 29) {
 row2.ZAHLUNGSBEDINGUNGEN = null;
} else {
 tmpContent_tOracleInput_2 = rs_tOracleInput_2
   .getString(29);
 if (tmpContent_tOracleInput_2 != null) {
  row2.ZAHLUNGSBEDINGUNGEN = tmpContent_tOracleInput_2
    .trim();
 } else {
  row2.ZAHLUNGSBEDINGUNGEN = null;
 }
}
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tOracleInput_2";
tos_count_tOracleInput_2++;
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tMap_2";
boolean hasCasePrimitiveKeyWithNull_tMap_2 = false;
// ##############################
// # Input tables (lookups)
boolean rejectedInnerJoin_tMap_2 = false;
boolean mainRowRejected_tMap_2 = false;
// ////////////////////////////////////////////
// Starting Lookup Table "row3"
// ////////////////////////////////////////////
boolean forceLooprow3 = false;
row3Struct row3ObjectFromLookup = null;
if (!rejectedInnerJoin_tMap_2) { // G_TM_M_020
 hasCasePrimitiveKeyWithNull_tMap_2 = false;
 row3HashKey.DATENFELDER_1 = row2.FIRMANR;
 row3HashKey.hashCodeDirty = true;
 tHash_Lookup_row3.lookup(row3HashKey);
} // G_TM_M_020
if (tHash_Lookup_row3 != null
  && tHash_Lookup_row3.getCount(row3HashKey) > 1) { // G
                // 071
 // System.out.println("WARNING: UNIQUE MATCH is configured for the lookup 'row3' and it contains more one result from keys :  row3.DATENFEl

 // + row3HashKey.DATENFELDER_1 + "'");
```

```
} // G 071
row3Struct row3 = null;
row3Struct fromLookup_row3 = null;
row3 = row3Default;
if (tHash_Lookup_row3 != null
  && tHash_Lookup_row3.hasNext()) { // G 099
 fromLookup_row3 = tHash_Lookup_row3.next();
} // G 099
if (fromLookup_row3 != null) {
 row3 = fromLookup_row3;
}
// ################################
{ // start of Var scope
 // ################################
 // # Vars tables
 Var__tMap_2__Struct Var = Var__tMap_2;// ################################
 // ################################
 // # Output tables
 out_sf = null;
 // # Output table : 'out_sf'
 out_sf_tmp.Aktueller_Umsatz__c = row3.AKTJAHR;
 out_sf_tmp.Bearbeitungsdatum__c = row2.MODIFYDATE;
 out_sf_tmp.BillingCity = row2.ORT;
 out_sf_tmp.BillingCountry = row2.LAND;
 out_sf_tmp.BillingPostalCode = row2.PLZORT;
 out_sf_tmp.BillingState = row2.STAAT;
 out_sf_tmp.ABC_Klassifizierung__c = row2.ABCKLAS;
 out_sf_tmp.BillingStreet = row2.STRASSE;
 out_sf_tmp.Erstelldatum__c = row2.CREATEDATE;
 out_sf_tmp.Name = row2.NAME;
 out_sf_tmp.UST_ID_NR__c = row2.USTIDNR;
 out_sf_tmp.Umsatz_VorVorjahr__c = row3.VORVORJAHR;
 out_sf_tmp.Umsatz_Vorjahr__c = row3.VORJAHR;
 out_sf_tmp.VertreterNummer__c = row2.VERTRETER1;
 out_sf_tmp.Verwendung__c = row2.VERWENDUNG1;
 out_sf_tmp.f42_INFOR_FirmenNr__c = row2.FIRMANR1;
 out_sf_tmp.Bemerkung_Infor__c = row2.BEMERKUNG;
 out_sf_tmp.Lieferbedingungen__c = row2.LIEFERBEDINGUNGEN;
 out_sf_tmp.Spedition_Versand__c = row2.Spedition_Versand;
 out_sf_tmp.Zahlungsbedingungen__c = row2.ZAHLUNGSBEDINGUNGEN;
 out_sf_tmp.KTXT__c = row2.KTXT;
 out_sf = out_sf_tmp;
 // ################################
} // end of Var scope
rejectedInnerJoin_tMap_2 = false;
tos_count_tMap_2++;
/**
 *  stop
 */
// Start of branch "out_sf"
if (out_sf != null) {
 /**
  *  start
  */
 currentComponent = "tConvertType_1";
 row1 = new row1Struct();
```

```
boolean bHasError_tConvertType_1 = false;
try {
 row1.ABC_Klassifizierung__c = TypeConvert
   .String2String(out_sf.ABC_Klassifizierung__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Aktueller_Umsatz__c = TypeConvert
   .Float2Double(out_sf.Aktueller_Umsatz__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Bearbeitungsdatum__c = TypeConvert
   .Date2Date(out_sf.Bearbeitungsdatum__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Bemerkung_Infor__c = TypeConvert
   .String2String(out_sf.Bemerkung_Infor__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.BillingCity = TypeConvert
   .String2String(out_sf.BillingCity);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.BillingCountry = TypeConvert
   .String2String(out_sf.BillingCountry);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.BillingPostalCode = TypeConvert
   .String2String(out_sf.BillingPostalCode);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.BillingState = TypeConvert
   .String2String(out_sf.BillingState);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
```

```
}
try {
 row1.BillingStreet = TypeConvert
   .String2String(out_sf.BillingStreet);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Erstelldatum__c = TypeConvert
   .Date2Date(out_sf.Erstelldatum__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.KTXT__c = TypeConvert
   .String2String(out_sf.KTXT__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Lieferbedingungen__c = TypeConvert
   .String2String(out_sf.Lieferbedingungen__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Name = TypeConvert
   .String2String(out_sf.Name);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Spedition_Versand__c = TypeConvert
   .String2String(out_sf.Spedition_Versand__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.UST_ID_NR__c = TypeConvert
   .String2String(out_sf.UST_ID_NR__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Umsatz_VorVorjahr__c = TypeConvert
   .Float2Double(out_sf.Umsatz_VorVorjahr__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
```

```
}
try {
 row1.Umsatz_Vorjahr__c = TypeConvert
   .Float2Double(out_sf.Umsatz_Vorjahr__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.VertreterNummer__c = TypeConvert
   .String2String(out_sf.VertreterNummer__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Verwendung__c = TypeConvert
   .BigDecimal2String(out_sf.Verwendung__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.Zahlungsbedingungen__c = TypeConvert
   .String2String(out_sf.Zahlungsbedingungen__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
try {
 row1.f42_INFOR_FirmenNr__c = TypeConvert
   .String2String(out_sf.f42_INFOR_FirmenNr__c);
} catch (java.lang.Exception e) {
 bHasError_tConvertType_1 = true;
 throw e;
}
if (bHasError_tConvertType_1) {
 row1 = null;
}
nb_line_tConvertType_1++;
tos_count_tConvertType_1++;
/**
 *  stop
 */
// Start of branch "row1"
if (row1 != null) {
 /**
  *  start
  */
 currentComponent = "tSalesforceOutput_1";
 // ////////////////////
 Object[] resultMessageObj_tSalesforceOutput_1 = null;
 java.util.List<org.apache.axiom.om.OMElement> list_tSalesforceOutput_1 = new java.util.ArrayList<org.apache.axiom.om.OMElement>();
 // the null value will be updated or inserted,so
 // we can't treat it as key for updating(only Id
 // can be the key) or upserting(Id,external id
```

```
// or indexed id)
java.util.List<String> nullList_tSalesforceOutput_1 = new java.util.ArrayList<String>();
if (row1.ABC_Klassifizierung__c != null
  && !"".equals(String
    .valueOf(row1.ABC_Klassifizierung__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "ABC_Klassifizierung__c",
       String.valueOf(row1.ABC_Klassifizierung__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("ABC_Klassifizierung__c");
}
if (row1.Aktueller_Umsatz__c != null
  && !"".equals(String
    .valueOf(row1.Aktueller_Umsatz__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Aktueller_Umsatz__c",
       String.valueOf(row1.Aktueller_Umsatz__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Aktueller_Umsatz__c");
}
if (row1.Bearbeitungsdatum__c != null) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Bearbeitungsdatum__c",
       FormatterUtils
         .format_Date(
           row1.Bearbeitungsdatum__c,
           "yyyy-MM-dd")));
}
else {
 nullList_tSalesforceOutput_1
   .add("Bearbeitungsdatum__c");
}
if (row1.Bemerkung_Infor__c != null
  && !"".equals(String
    .valueOf(row1.Bemerkung_Infor__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Bemerkung_Infor__c",
       String.valueOf(row1.Bemerkung_Infor__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Bemerkung_Infor__c");
}
if (row1.BillingCity != null
```

```
  && !"".equals(String
    .valueOf(row1.BillingCity))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "BillingCity",
       String.valueOf(row1.BillingCity)));
}
else {
 nullList_tSalesforceOutput_1
   .add("BillingCity");
}
if (row1.BillingCountry != null
  && !"".equals(String
    .valueOf(row1.BillingCountry))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "BillingCountry",
       String.valueOf(row1.BillingCountry)));
}
else {
 nullList_tSalesforceOutput_1
   .add("BillingCountry");
}
if (row1.BillingPostalCode != null
  && !"".equals(String
    .valueOf(row1.BillingPostalCode))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "BillingPostalCode",
       String.valueOf(row1.BillingPostalCode)));
}
else {
 nullList_tSalesforceOutput_1
   .add("BillingPostalCode");
}
if (row1.BillingState != null
  && !"".equals(String
    .valueOf(row1.BillingState))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "BillingState",
       String.valueOf(row1.BillingState)));
}
else {
 nullList_tSalesforceOutput_1
   .add("BillingState");
}
if (row1.BillingStreet != null
  && !"".equals(String
    .valueOf(row1.BillingStreet))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
```

```
      .newOMElement(
        "BillingStreet",
        String.valueOf(row1.BillingStreet)));
}
else {
 nullList_tSalesforceOutput_1
   .add("BillingStreet");
}
if (row1.Erstelldatum__c != null) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Erstelldatum__c",
       FormatterUtils
         .format_Date(
           row1.Erstelldatum__c,
           "yyyy-MM-dd")));
}
else {
 nullList_tSalesforceOutput_1
   .add("Erstelldatum__c");
}
if (row1.KTXT__c != null
  && !"".equals(String
    .valueOf(row1.KTXT__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "KTXT__c",
       String.valueOf(row1.KTXT__c)));
}
else {
 nullList_tSalesforceOutput_1.add("KTXT__c");
}
if (row1.Lieferbedingungen__c != null
  && !"".equals(String
    .valueOf(row1.Lieferbedingungen__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Lieferbedingungen__c",
       String.valueOf(row1.Lieferbedingungen__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Lieferbedingungen__c");
}
if (row1.Name != null
  && !"".equals(String.valueOf(row1.Name))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Name",
       String.valueOf(row1.Name)));
}
else {
```

```
 nullList_tSalesforceOutput_1.add("Name");
}
if (row1.Spedition_Versand__c != null
  && !"".equals(String
    .valueOf(row1.Spedition_Versand__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Spedition_Versand__c",
       String.valueOf(row1.Spedition_Versand__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Spedition_Versand__c");
}
if (row1.UST_ID_NR__c != null
  && !"".equals(String
    .valueOf(row1.UST_ID_NR__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "UST_ID_NR__c",
       String.valueOf(row1.UST_ID_NR__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("UST_ID_NR__c");
}
if (row1.Umsatz_VorVorjahr__c != null
  && !"".equals(String
    .valueOf(row1.Umsatz_VorVorjahr__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Umsatz_VorVorjahr__c",
       String.valueOf(row1.Umsatz_VorVorjahr__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Umsatz_VorVorjahr__c");
}
if (row1.Umsatz_Vorjahr__c != null
  && !"".equals(String
    .valueOf(row1.Umsatz_Vorjahr__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Umsatz_Vorjahr__c",
       String.valueOf(row1.Umsatz_Vorjahr__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Umsatz_Vorjahr__c");
}
if (row1.VertreterNummer__c != null
  && !"".equals(String
```

```
    .valueOf(row1.VertreterNummer__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "VertreterNummer__c",
       String.valueOf(row1.VertreterNummer__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("VertreterNummer__c");
}
if (row1.Verwendung__c != null
  && !"".equals(String
    .valueOf(row1.Verwendung__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Verwendung__c",
       String.valueOf(row1.Verwendung__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Verwendung__c");
}
if (row1.Zahlungsbedingungen__c != null
  && !"".equals(String
    .valueOf(row1.Zahlungsbedingungen__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "Zahlungsbedingungen__c",
       String.valueOf(row1.Zahlungsbedingungen__c)));
}
else {
 nullList_tSalesforceOutput_1
   .add("Zahlungsbedingungen__c");
}
if (row1.f42_INFOR_FirmenNr__c != null
  && !"".equals(String
    .valueOf(row1.f42_INFOR_FirmenNr__c))) {
 list_tSalesforceOutput_1
   .add(org.talend.salesforce.SforceManagementUtil
     .newOMElement(
       "f42_INFOR_FirmenNr__c",
       String.valueOf(row1.f42_INFOR_FirmenNr__c)));
}
resultMessageObj_tSalesforceOutput_1 = sforceManagement_tSalesforceOutput_1
  .upsert("Account",
    "f42_INFOR_FirmenNr__c",
    (org.apache.axiom.om.OMElement[]) list_tSalesforceOutput_1
      .toArray(new org.apache.axiom.om.OMElement),
    (String[]) nullList_tSalesforceOutput_1
      .toArray(new String));
nb_line_tSalesforceOutput_1++;
nb_success_tSalesforceOutput_1++;
// ////////////////////
```

```
    tos_count_tSalesforceOutput_1++;
    /**
     *  stop
     */
   } // End of branch "row1"
  } // End of branch "out_sf"
  /**
   *  start
   */
  currentComponent = "tOracleInput_2";
 }
} finally {
 stmt_tOracleInput_2.close();
 if (conn_tOracleInput_2 != null
   && !conn_tOracleInput_2.isClosed()) {
  conn_tOracleInput_2.close();
 }
}
globalMap.put("tOracleInput_2_NB_LINE", nb_line_tOracleInput_2);
ok_Hash.put("tOracleInput_2", true);
end_Hash.put("tOracleInput_2", System.currentTimeMillis());
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tMap_2";
// ###############################
// # Lookup hashes releasing
if (tHash_Lookup_row3 != null) {
 tHash_Lookup_row3.endGet();
}
globalMap.remove("tHash_Lookup_row3");
// ###############################
ok_Hash.put("tMap_2", true);
end_Hash.put("tMap_2", System.currentTimeMillis());
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tConvertType_1";
globalMap.put("tConvertType_1_NB_LINE", nb_line_tConvertType_1);
ok_Hash.put("tConvertType_1", true);
end_Hash.put("tConvertType_1", System.currentTimeMillis());
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tSalesforceOutput_1";
sforceManagement_tSalesforceOutput_1.logout();
globalMap.put("tSalesforceOutput_1_NB_LINE",
```

```
    nb_line_tSalesforceOutput_1);
   globalMap.put("tSalesforceOutput_1_NB_SUCCESS",
    nb_success_tSalesforceOutput_1);
   globalMap.put("tSalesforceOutput_1_NB_REJECT",
    nb_reject_tSalesforceOutput_1);
   ok_Hash.put("tSalesforceOutput_1", true);
   end_Hash.put("tSalesforceOutput_1", System.currentTimeMillis());
   /**
    *  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_2"
  globalMap.remove("tHash_Lookup_row3");
  try {
   /**
    *  start
    */
   currentComponent = "tOracleInput_2";
   /**
    *  stop
    */
   /**
    *  start
    */
   currentComponent = "tMap_2";
   /**
    *  stop
    */
   /**
    *  start
    */
   currentComponent = "tConvertType_1";
   /**
    *  stop
    */
   /**
    *  start
    */
   currentComponent = "tSalesforceOutput_1";
   /**
    *  stop
    */
  } catch (java.lang.Exception e) {
   // ignore
  } catch (java.lang.Error error) {
   // ignore
  }
  resourceMap = null;
 }
```

```
 globalMap.put("tOracleInput_2_SUBPROCESS_STATE", 1);
}
public static class row3Struct implements
  routines.system.IPersistableComparableLookupRow<row3Struct> {
 final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
 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 String DATENFELDER_1;
 public String getDATENFELDER_1() {
  return this.DATENFELDER_1;
 }
 public Float VORVORJAHR;
 public Float getVORVORJAHR() {
  return this.VORVORJAHR;
 }
 public Float AKTJAHR;
 public Float getAKTJAHR() {
  return this.AKTJAHR;
 }
 public Float VORJAHR;
 public Float getVORJAHR() {
  return this.VORJAHR;
 }
 public String SATZART;
 public String getSATZART() {
  return this.SATZART;
 }
 @Override
 public int hashCode() {
  if (this.hashCodeDirty) {
   final int prime = PRIME;
   int result = DEFAULT_HASHCODE;
   result = prime
     * result
     + ((this.DATENFELDER_1 == null) ? 0
       : this.DATENFELDER_1.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 row3Struct other = (row3Struct) obj;
  if (this.DATENFELDER_1 == null) {
   if (other.DATENFELDER_1 != null)
```

```
   return false;
 } else if (!this.DATENFELDER_1.equals(other.DATENFELDER_1))
  return false;
 return true;
}
public void copyDataTo(row3Struct other) {
 other.DATENFELDER_1 = this.DATENFELDER_1;
 other.VORVORJAHR = this.VORVORJAHR;
 other.AKTJAHR = this.AKTJAHR;
 other.VORJAHR = this.VORJAHR;
 other.SATZART = this.SATZART;
}
public void copyKeysDataTo(row3Struct other) {
 other.DATENFELDER_1 = this.DATENFELDER_1;
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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 String readString(DataInputStream dis, ObjectInputStream ois)
  throws IOException {
 String strReturn = null;
 int length = 0;
 length = dis.readInt();
 if (length == -1) {
  strReturn = null;
```

```
 } else {
  byte[] byteArray = new byte;
  dis.read(byteArray);
  strReturn = new String(byteArray, utf8Charset);
 }
 return strReturn;
}
private void writeString(String str, DataOutputStream dos,
  ObjectOutputStream oos) throws IOException {
 if (str == null) {
  dos.writeInt(-1);
 } else {
  byte[] byteArray = str.getBytes(utf8Charset);
  dos.writeInt(byteArray.length);
  dos.write(byteArray);
 }
}
public void readKeysData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   this.DATENFELDER_1 = readString(dis);
  } catch (IOException e) {
   throw new RuntimeException(e);
  }
 }
}
public void writeKeysData(ObjectOutputStream dos) {
 try {
  // String
  writeString(this.DATENFELDER_1, dos);
 } catch (IOException e) {
  throw new RuntimeException(e);
 }
}
/**
 * Fill Values data by reading ObjectInputStream.
 */
public void readValuesData(DataInputStream dis, ObjectInputStream ois) {
 try {
  int length = 0;
  length = dis.readByte();
  if (length == -1) {
   this.VORVORJAHR = null;
  } else {
   this.VORVORJAHR = dis.readFloat();
  }
  length = dis.readByte();
  if (length == -1) {
   this.AKTJAHR = null;
  } else {
   this.AKTJAHR = dis.readFloat();
  }
  length = dis.readByte();
  if (length == -1) {
   this.VORJAHR = null;
```

```
  } else {
   this.VORJAHR = dis.readFloat();
  }
  this.SATZART = readString(dis, ois);
 } catch (IOException e) {
  throw new RuntimeException(e);
 }
}
/**
 * Return a byte array which represents Values data.
 */
public void writeValuesData(DataOutputStream dos, ObjectOutputStream oos) {
 try {
  if (this.VORVORJAHR == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeFloat(this.VORVORJAHR);
  }
  if (this.AKTJAHR == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeFloat(this.AKTJAHR);
  }
  if (this.VORJAHR == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeFloat(this.VORJAHR);
  }
  writeString(this.SATZART, dos, oos);
 } catch (IOException e) {
  throw new RuntimeException(e);
 }
}
public String toString() {
 StringBuilder sb = new StringBuilder();
 sb.append(super.toString());
 sb.append("");
 return sb.toString();
}
/**
 * Compare keys
 */
public int compareTo(row3Struct other) {
 int returnValue = -1;
 returnValue = checkNullsAndCompare(this.DATENFELDER_1,
   other.DATENFELDER_1);
 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 tOracleInput_3Process(
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 globalMap.put("tOracleInput_3_SUBPROCESS_STATE", 0);
 final boolean execStat = this.execStat;
 String iterateId = "";
 String currentComponent = "";
 java.util.Map<String, Object> resourceMap = new java.util.HashMap<String, Object>();
 try {
  String currentMethodName = new java.lang.Exception()
    .getStackTrace().getMethodName();
boolean resumeIt = currentMethodName.equals(resumeEntryMethodName);
if (resumeEntryMethodName == null || resumeIt || globalResumeTicket) {// start
                  // the
                  // resume
 globalResumeTicket = true;
 row3Struct row3 = new row3Struct();
 /**
  *  start
  */
 ok_Hash.put("tAdvancedHash_row3", false);
 start_Hash
   .put("tAdvancedHash_row3", System.currentTimeMillis());
 currentComponent = "tAdvancedHash_row3";
 int tos_count_tAdvancedHash_row3 = 0;
 // connection name:row3
 // source node:tOracleInput_3 - inputs:(after_tOracleInput_2)
 // outputs:(row3,OnComponentError,row3) | target
 // node:tAdvancedHash_row3 - inputs:(row3) outputs:()
 // linked node: tMap_2 - inputs:(row2,row3) outputs:(out_sf)
 org.talend.designer.components.lookup.common.ICommonLookup.MATCHING_MODE matchingModeEnum_row3 = org.talend.designer.components.lookup.common.

 org.talend.designer.components.lookup.memory.AdvancedMemoryLookup<row3Struct> tHash_Lookup_row3 = org.talend.designer.components.lookup.memory

   .<row3Struct> getLookup(matchingModeEnum_row3);
 globalMap.put("tHash_Lookup_row3", tHash_Lookup_row3);
 /**
  *  stop
```

```
 */
/**
 *  start
 */
ok_Hash.put("tOracleInput_3", false);
start_Hash.put("tOracleInput_3", System.currentTimeMillis());
currentComponent = "tOracleInput_3";
int tos_count_tOracleInput_3 = 0;
int nb_line_tOracleInput_3 = 0;
java.sql.Connection conn_tOracleInput_3 = null;
String driverClass_tOracleInput_3 = "oracle.jdbc.driver.OracleDriver";
java.lang.Class.forName(driverClass_tOracleInput_3);
String url_tOracleInput_3 = null;
url_tOracleInput_3 = "jdbc:oracle:thin:@srv43:1591:PROD";
String dbUser_tOracleInput_3 = "SBA_READ";
final String decryptedPassword_tOracleInput_3 = routines.system.PasswordEncryptUtil
  .decryptPassword("b2e1ec96dcc3d7cff4f7aba1746784ea");
String dbPwd_tOracleInput_3 = decryptedPassword_tOracleInput_3;
conn_tOracleInput_3 = java.sql.DriverManager.getConnection(
  url_tOracleInput_3, dbUser_tOracleInput_3,
  dbPwd_tOracleInput_3);
java.sql.Statement stmtGetTZ_tOracleInput_3 = conn_tOracleInput_3
  .createStatement();
java.sql.ResultSet rsGetTZ_tOracleInput_3 = stmtGetTZ_tOracleInput_3
  .executeQuery("select sessiontimezone from dual");
String sessionTimezone_tOracleInput_3 = java.util.TimeZone
  .getDefault().getID();
while (rsGetTZ_tOracleInput_3.next()) {
 sessionTimezone_tOracleInput_3 = rsGetTZ_tOracleInput_3
   .getString(1);
}
((oracle.jdbc.OracleConnection) conn_tOracleInput_3)
  .setSessionTimeZone(sessionTimezone_tOracleInput_3);
java.sql.Statement stmt_tOracleInput_3 = conn_tOracleInput_3
  .createStatement();
String dbquery_tOracleInput_3 = "SELECT DISTINCT \n(RELFIRMA.FIRMANR),\nDATENFELDER_1,\nRELES.VORVORJAHR,\n  RELES.AktJahr,\n  RELES.VorJahr,\

globalMap.put("tOracleInput_3_QUERY", dbquery_tOracleInput_3);
java.sql.ResultSet rs_tOracleInput_3 = null;
try {
 rs_tOracleInput_3 = stmt_tOracleInput_3
   .executeQuery(dbquery_tOracleInput_3);
 java.sql.ResultSetMetaData rsmd_tOracleInput_3 = rs_tOracleInput_3
   .getMetaData();
 int colQtyInRs_tOracleInput_3 = rsmd_tOracleInput_3
   .getColumnCount();
 String tmpContent_tOracleInput_3 = null;
 while (rs_tOracleInput_3.next()) {
  nb_line_tOracleInput_3++;
  if (colQtyInRs_tOracleInput_3 < 1) {
   row3.DATENFELDER_1 = null;
  } else {
   tmpContent_tOracleInput_3 = rs_tOracleInput_3
     .getString(1);
   if (tmpContent_tOracleInput_3 != null) {
    row3.DATENFELDER_1 = tmpContent_tOracleInput_3
```

```
    .trim();
 } else {
  row3.DATENFELDER_1 = null;
 }
}
if (colQtyInRs_tOracleInput_3 < 2) {
 row3.VORVORJAHR = null;
} else {
 if (rs_tOracleInput_3.getObject(2) != null) {
  row3.VORVORJAHR = rs_tOracleInput_3.getFloat(2);
 } else {
  row3.VORVORJAHR = null;
 }
}
if (colQtyInRs_tOracleInput_3 < 3) {
 row3.AKTJAHR = null;
} else {
 if (rs_tOracleInput_3.getObject(3) != null) {
  row3.AKTJAHR = rs_tOracleInput_3.getFloat(3);
 } else {
  row3.AKTJAHR = null;
 }
}
if (colQtyInRs_tOracleInput_3 < 4) {
 row3.VORJAHR = null;
} else {
 if (rs_tOracleInput_3.getObject(4) != null) {
  row3.VORJAHR = rs_tOracleInput_3.getFloat(4);
 } else {
  row3.VORJAHR = null;
 }
}
if (colQtyInRs_tOracleInput_3 < 5) {
 row3.SATZART = null;
} else {
 tmpContent_tOracleInput_3 = rs_tOracleInput_3
   .getString(5);
 if (tmpContent_tOracleInput_3 != null) {
  row3.SATZART = tmpContent_tOracleInput_3.trim();
 } else {
  row3.SATZART = null;
 }
}
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tOracleInput_3";
tos_count_tOracleInput_3++;
/**
 *  stop
 */
/**
 *  start
```

```
   */
  currentComponent = "tAdvancedHash_row3";
  row3Struct row3_HashRow = new row3Struct();
  row3_HashRow.DATENFELDER_1 = row3.DATENFELDER_1;
  row3_HashRow.VORVORJAHR = row3.VORVORJAHR;
  row3_HashRow.AKTJAHR = row3.AKTJAHR;
  row3_HashRow.VORJAHR = row3.VORJAHR;
  row3_HashRow.SATZART = row3.SATZART;
  tHash_Lookup_row3.put(row3_HashRow);
  tos_count_tAdvancedHash_row3++;
  /**
   *  stop
   */
  /**
   *  start
   */
  currentComponent = "tOracleInput_3";
 }
} finally {
 stmt_tOracleInput_3.close();
 if (conn_tOracleInput_3 != null
   && !conn_tOracleInput_3.isClosed()) {
  conn_tOracleInput_3.close();
 }
}
globalMap.put("tOracleInput_3_NB_LINE", nb_line_tOracleInput_3);
ok_Hash.put("tOracleInput_3", true);
end_Hash.put("tOracleInput_3", System.currentTimeMillis());
  /**
   *  stop
   */
  /**
   *  start
   */
  currentComponent = "tAdvancedHash_row3";
  tHash_Lookup_row3.endPut();
  ok_Hash.put("tAdvancedHash_row3", true);
  end_Hash.put("tAdvancedHash_row3", System.currentTimeMillis());
  /**
   *  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 {
  /**
   *  start
   */
  currentComponent = "tOracleInput_3";
  /**
   *  stop
```

```
    */
    /**
     *  start
     */
    currentComponent = "tAdvancedHash_row3";
    /**
     *  stop
     */
   } catch (java.lang.Exception e) {
    // ignore
   } catch (java.lang.Error error) {
    // ignore
   }
   resourceMap = null;
  }
  globalMap.put("tOracleInput_3_SUBPROCESS_STATE", 1);
 }
 public static class row4Struct implements
   routines.system.IPersistableRow<row4Struct> {
  final static byte[] commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
  static byte[] commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
  public java.util.Date moment;
  public java.util.Date getMoment() {
   return this.moment;
  }
  public String pid;
  public String getPid() {
   return this.pid;
  }
  public String father_pid;
  public String getFather_pid() {
   return this.father_pid;
  }
  public String root_pid;
  public String getRoot_pid() {
   return this.root_pid;
  }
  public Long system_pid;
  public Long getSystem_pid() {
   return this.system_pid;
  }
  public String project;
  public String getProject() {
   return this.project;
  }
  public String job;
  public String getJob() {
   return this.job;
  }
  public String job_repository_id;
  public String getJob_repository_id() {
   return this.job_repository_id;
  }
  public String job_version;
  public String getJob_version() {
   return this.job_version;
```

```
}
public String context;
public String getContext() {
 return this.context;
}
public String origin;
public String getOrigin() {
 return this.origin;
}
public String message_type;
public String getMessage_type() {
 return this.message_type;
}
public String message;
public String getMessage() {
 return this.message;
}
public Long duration;
public Long getDuration() {
 return this.duration;
}
private java.util.Date readDate(ObjectInputStream dis)
  throws IOException {
 java.util.Date dateReturn = null;
 int length = 0;
 length = dis.readByte();
 if (length == -1) {
  dateReturn = null;
 } else {
  dateReturn = new Date(dis.readLong());
 }
 return dateReturn;
}
private void writeDate(java.util.Date date1, ObjectOutputStream dos)
  throws IOException {
 if (date1 == null) {
  dos.writeByte(-1);
 } else {
  dos.writeByte(0);
  dos.writeLong(date1.getTime());
 }
}
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_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length) {
   if (length < 1024
     && commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u.length == 0) {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   } else {
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u = new byte;
   }
```

```
  }
  dis.readFully(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    0, length);
  strReturn = new String(
    commonByteArray_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u,
    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);
 }
}
public void readData(ObjectInputStream dis) {
 synchronized (commonByteArrayLock_SF_CSV_SF_Prod_Data_Migration_q_ktxt_acc_u) {
  try {
   int length = 0;
   this.moment = readDate(dis);
   this.pid = readString(dis);
   this.father_pid = readString(dis);
   this.root_pid = readString(dis);
   length = dis.readByte();
   if (length == -1) {
    this.system_pid = null;
   } else {
    this.system_pid = dis.readLong();
   }
   this.project = readString(dis);
   this.job = readString(dis);
   this.job_repository_id = readString(dis);
   this.job_version = readString(dis);
   this.context = readString(dis);
   this.origin = readString(dis);
   this.message_type = readString(dis);
   this.message = readString(dis);
   length = dis.readByte();
   if (length == -1) {
    this.duration = null;
   } else {
    this.duration = dis.readLong();
   }
  } catch (IOException e) {
   throw new RuntimeException(e);
  }
 }
}
public void writeData(ObjectOutputStream dos) {
 try {
  // java.util.Date
```

```
writeDate(this.moment, dos);
// String
writeString(this.pid, dos);
// String
writeString(this.father_pid, dos);
// String
writeString(this.root_pid, dos);
// Long
if (this.system_pid == null) {
 dos.writeByte(-1);
} else {
 dos.writeByte(0);
 dos.writeLong(this.system_pid);
}
// String
writeString(this.project, dos);
// String
writeString(this.job, dos);
// String
writeString(this.job_repository_id, dos);
// String
writeString(this.job_version, dos);
// String
writeString(this.context, dos);
// String
writeString(this.origin, dos);
// String
writeString(this.message_type, dos);
  // String
  writeString(this.message, dos);
  // Long
  if (this.duration == null) {
   dos.writeByte(-1);
  } else {
   dos.writeByte(0);
   dos.writeLong(this.duration);
  }
 } catch (IOException e) {
  throw new RuntimeException(e);
 }
}
public String toString() {
 StringBuilder sb = new StringBuilder();
 sb.append(super.toString());
 sb.append("");
 return sb.toString();
}
/**
 * Compare keys
 */
public int compareTo(row4Struct 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 tStatCatcher_1Process(
  final java.util.Map<String, Object> globalMap)
  throws TalendException {
 globalMap.put("tStatCatcher_1_SUBPROCESS_STATE", 0);
 final boolean execStat = this.execStat;
 String iterateId = "";
 String currentComponent = "";
 java.util.Map<String, Object> resourceMap = new java.util.HashMap<String, Object>();
 try {
  String currentMethodName = new java.lang.Exception()
    .getStackTrace().getMethodName();
boolean resumeIt = currentMethodName.equals(resumeEntryMethodName);
if (resumeEntryMethodName == null || resumeIt || globalResumeTicket) {// start
                  // the
                  // resume
 globalResumeTicket = true;
 row4Struct row4 = new row4Struct();
 /**
  *  start
  */
 ok_Hash.put("tSendMail_2", false);
 start_Hash.put("tSendMail_2", System.currentTimeMillis());
 currentComponent = "tSendMail_2";
 int tos_count_tSendMail_2 = 0;
 /**
  *  stop
  */
 /**
  *  start
  */
 ok_Hash.put("tStatCatcher_1", false);
 start_Hash.put("tStatCatcher_1", System.currentTimeMillis());
 currentComponent = "tStatCatcher_1";
 int tos_count_tStatCatcher_1 = 0;
 for (StatCatcherUtils.StatCatcherMessage scm : tStatCatcher_1
   .getMessages()) {
  row4.pid = pid;
  row4.root_pid = rootPid;
```

```
row4.father_pid = fatherPid;
row4.project = projectName;
row4.job = jobName;
row4.context = contextStr;
row4.origin = (scm.getOrigin() == null
  || scm.getOrigin().length() < 1 ? null : scm
  .getOrigin());
row4.message = scm.getMessage();
row4.duration = scm.getDuration();
row4.moment = scm.getMoment();
row4.message_type = scm.getMessageType();
row4.job_version = scm.getJobVersion();
row4.job_repository_id = scm.getJobId();
row4.system_pid = scm.getSystemPid();
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tStatCatcher_1";
tos_count_tStatCatcher_1++;
/**
 *  stop
 */
/**
 *  start
 */
currentComponent = "tSendMail_2";
String smtpHost_tSendMail_2 = "srv72.sbah.local";
String smtpPort_tSendMail_2 = "25";
String from_tSendMail_2 = ("Talend-sucess@localhost");
String to_tSendMail_2 = ("edv@sba.de").replace(";", ",");
String cc_tSendMail_2 = (("") == null || "".equals("")) ? null
  : ("").replace(";", ",");
String bcc_tSendMail_2 = (("") == null || "".equals("")) ? null
  : ("").replace(";", ",");
String subject_tSendMail_2 = ("Talend " + row4.job + "");
java.util.List<java.util.Map<String, String>> headers_tSendMail_2 = new java.util.ArrayList<java.util.Map<String, String>>();
java.util.List<String> attachments_tSendMail_2 = new java.util.ArrayList<String>();
java.util.List<String> contentTransferEncoding_tSendMail_2 = new java.util.ArrayList<String>();
String message_tSendMail_2 = (("Hello,\n\nJob was started/ended  Successfully \n\nJob "
  + row4.job
  + "\n\n"
  + row4.moment
  + "\n"
  + row4.project
  + " ("
  + row4.context
  + ") / "
  + row4.origin
  + "\n"
  + row4.message_type
  + " : "
  + row4.message + " (" + row4.origin + ")\n\nGood Luck.") == null || ""
  .equals("Hello,\n\nJob was started/ended  Successfully \n\nJob "
```

```
    + row4.job
    + "\n\n"
    + row4.moment
    + "\n"
    + row4.project
    + " ("
    + row4.context
    + ") / "
    + row4.origin
    + "\n"
    + row4.message_type
    + " : "
    + row4.message
    + " (" + row4.origin + ")\n\nGood Luck.")) ? "\"\""
  : ("Hello,\n\nJob was started/ended  Successfully \n\nJob "
    + row4.job
    + "\n\n"
    + row4.moment
    + "\n"
    + row4.project
    + " ("
    + row4.context
    + ") / "
    + row4.origin
    + "\n"
    + row4.message_type
    + " : "
    + row4.message
    + " (" + row4.origin + ")\n\nGood Luck.");
java.util.Properties props_tSendMail_2 = System
  .getProperties();
props_tSendMail_2.put("mail.smtp.host",
  smtpHost_tSendMail_2);
props_tSendMail_2.put("mail.smtp.port",
  smtpPort_tSendMail_2);
props_tSendMail_2.put("mail.mime.encodefilename", "true");
try {
 props_tSendMail_2.put("mail.smtp.auth", "false");
 javax.mail.Session session_tSendMail_2 = javax.mail.Session
   .getInstance(props_tSendMail_2, null);
 javax.mail.Message msg_tSendMail_2 = new javax.mail.internet.MimeMessage(
   session_tSendMail_2);
 msg_tSendMail_2
   .setFrom(new javax.mail.internet.InternetAddress(
     from_tSendMail_2, null));
 msg_tSendMail_2.setRecipients(
   javax.mail.Message.RecipientType.TO,
   javax.mail.internet.InternetAddress.parse(
     to_tSendMail_2, false));
 if (cc_tSendMail_2 != null)
  msg_tSendMail_2.setRecipients(
    javax.mail.Message.RecipientType.CC,
    javax.mail.internet.InternetAddress.parse(
      cc_tSendMail_2, false));
 if (bcc_tSendMail_2 != null)
  msg_tSendMail_2.setRecipients(
```

```
   javax.mail.Message.RecipientType.BCC,
   javax.mail.internet.InternetAddress.parse(
     bcc_tSendMail_2, false));
msg_tSendMail_2.setSubject(subject_tSendMail_2);
for (int i_tSendMail_2 = 0; i_tSendMail_2 < headers_tSendMail_2
  .size(); i_tSendMail_2++) {
 java.util.Map<String, String> header_tSendMail_2 = headers_tSendMail_2
   .get(i_tSendMail_2);
 msg_tSendMail_2.setHeader(
   header_tSendMail_2.get("KEY"),
   header_tSendMail_2.get("VALUE"));
}
msg_tSendMail_2.setSentDate(new Date());
msg_tSendMail_2.setHeader("X-Priority", "5"); // High->1
            // Normal->3
            // Low->5
javax.mail.Multipart mp_tSendMail_2 = new javax.mail.internet.MimeMultipart();
javax.mail.internet.MimeBodyPart mbpText_tSendMail_2 = new javax.mail.internet.MimeBodyPart();
mbpText_tSendMail_2.setText(message_tSendMail_2,
  "ISO-8859-15", "plain");
mp_tSendMail_2.addBodyPart(mbpText_tSendMail_2);
javax.mail.internet.MimeBodyPart mbpFile_tSendMail_2 = null;
for (int i_tSendMail_2 = 0; i_tSendMail_2 < attachments_tSendMail_2
  .size(); i_tSendMail_2++) {
 String filename_tSendMail_2 = attachments_tSendMail_2
   .get(i_tSendMail_2);
 javax.activation.FileDataSource fds_tSendMail_2 = null;
 java.io.File file_tSendMail_2 = new java.io.File(
  filename_tSendMail_2);
if (!file_tSendMail_2.exists()) {
 continue;
}
if (file_tSendMail_2.isDirectory()) {
 java.io.File[] subFiles_tSendMail_2 = file_tSendMail_2
   .listFiles();
 for (java.io.File subFile_tSendMail_2 : subFiles_tSendMail_2) {
  if (subFile_tSendMail_2.isFile()) {
   fds_tSendMail_2 = new javax.activation.FileDataSource(
     subFile_tSendMail_2
       .getAbsolutePath());
   mbpFile_tSendMail_2 = new javax.mail.internet.MimeBodyPart();
   mbpFile_tSendMail_2
     .setDataHandler(new javax.activation.DataHandler(
       fds_tSendMail_2));
   mbpFile_tSendMail_2
     .setFileName(javax.mail.internet.MimeUtility
       .encodeText(fds_tSendMail_2
         .getName()));
   if (contentTransferEncoding_tSendMail_2
     .get(i_tSendMail_2)
     .equalsIgnoreCase("base64")) {
    mbpFile_tSendMail_2
      .setHeader(
        "Content-Transfer-Encoding",
        "base64");
   }
```

```
    mp_tSendMail_2
      .addBodyPart(mbpFile_tSendMail_2);
   }
  }
 } else {
  mbpFile_tSendMail_2 = new javax.mail.internet.MimeBodyPart();
  fds_tSendMail_2 = new javax.activation.FileDataSource(
    filename_tSendMail_2);
  mbpFile_tSendMail_2
    .setDataHandler(new javax.activation.DataHandler(
      fds_tSendMail_2));
  mbpFile_tSendMail_2
    .setFileName(javax.mail.internet.MimeUtility
      .encodeText(fds_tSendMail_2
        .getName()));
  if (contentTransferEncoding_tSendMail_2.get(
    i_tSendMail_2).equalsIgnoreCase(
    "base64")) {
   mbpFile_tSendMail_2.setHeader(
     "Content-Transfer-Encoding",
     "base64");
  }
  mp_tSendMail_2.addBodyPart(mbpFile_tSendMail_2);
 }
}
// -- set the content --
msg_tSendMail_2.setContent(mp_tSendMail_2);
// add handlers for main MIME types
 javax.activation.MailcapCommandMap mc_tSendMail_2 = (javax.activation.MailcapCommandMap) javax.activation.CommandMap
   .getDefaultCommandMap();
 mc_tSendMail_2
   .addMailcap("text/html;; x-java-content-handler=com.sun.mail.handlers.text_html");
 mc_tSendMail_2
   .addMailcap("text/xml;; x-java-content-handler=com.sun.mail.handlers.text_xml");
 mc_tSendMail_2
   .addMailcap("text/plain;; x-java-content-handler=com.sun.mail.handlers.text_plain");
 mc_tSendMail_2
   .addMailcap("multipart/*;; x-java-content-handler=com.sun.mail.handlers.multipart_mixed");
 mc_tSendMail_2
   .addMailcap("message/rfc822;; x-java-content-handler=com.sun.mail.handlers.message_rfc822");
 javax.activation.CommandMap
   .setDefaultCommandMap(mc_tSendMail_2);
 // -- Send the message --
 javax.mail.Transport.send(msg_tSendMail_2);
} catch (java.lang.Exception e) {
 System.err.println(e.toString());
} finally {
 props_tSendMail_2.remove("mail.smtp.host");
 props_tSendMail_2.remove("mail.smtp.port");
 props_tSendMail_2.remove("mail.mime.encodefilename");
 props_tSendMail_2.remove("mail.smtp.auth");
}
tos_count_tSendMail_2++;
/**
 *  stop
 */
```

```
    /**
     *  start
     */
    currentComponent = "tStatCatcher_1";
   }
   ok_Hash.put("tStatCatcher_1", true);
   end_Hash.put("tStatCatcher_1", System.currentTimeMillis());
   /**
    *  stop
    */
   /**
    *  start
    */
   currentComponent = "tSendMail_2";
   ok_Hash.put("tSendMail_2", true);
   end_Hash.put("tSendMail_2", System.currentTimeMillis());
   /**
    *  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 {
    /**
     *  start
     */
    currentComponent = "tStatCatcher_1";
    /**
     *  stop
     */
    /**
     *  start
     */
    currentComponent = "tSendMail_2";
    /**
     *  stop
     */
   } catch (java.lang.Exception e) {
    // ignore
   } catch (java.lang.Error error) {
    // ignore
   }
   resourceMap = null;
  }
  globalMap.put("tStatCatcher_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 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;
 };
};
private java.util.Properties context_param = new java.util.Properties();
public java.util.Map<String, Object> parentContextMap = new java.util.HashMap<String, Object>();
public String status = "";
public static void main(String[] args) {
 final SF_Prod_Data_Migration_q_ktxt_acc_u SF_Prod_Data_Migration_q_ktxt_acc_uClass = new SF_Prod_Data_Migration_q_ktxt_acc_u();
 int exitCode = SF_Prod_Data_Migration_q_ktxt_acc_uClass
   .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 = "";
  }
 }
```

```
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;
}
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 = SF_Prod_Data_Migration_q_ktxt_acc_u.class
   .getClassLoader().getResourceAsStream(
     "sf_csv/sf_prod_data_migration_q_ktxt_acc_u_0_1/contexts/"
       + contextStr + ".properties");
 if (isDefaultContext && inContext == null) {
 } else {
  if (inContext != null) {
   // defaultProps is in order to keep the original context
   // value
   defaultProps.load(inContext);
   inContext.close();
   context = new ContextProperties(defaultProps);
  } else {
   // 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);
 }
 context.new1 = (String) context.getProperty("new1");
} 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()) {
 if (parentContextMap.containsKey("new1")) {
  context.new1 = (String) parentContextMap.get("new1");
 }
}
// 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();
tStatCatcher_1.addMessage("begin");
this.globalResumeTicket = true;// to run tPreJob
try {
 tStatCatcher_1Process(globalMap);
} catch (java.lang.Exception e) {
 e.printStackTrace();
}
this.globalResumeTicket = false;// to run others jobs
try {
 errorCode = null;
 tOracleInput_2Process(globalMap);
 if (!"failure".equals(status)) {
  status = "end";
 }
} catch (TalendException e_tOracleInput_2) {
 globalMap.put("tOracleInput_2_SUBPROCESS_STATE", -1);
 e_tOracleInput_2.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 : SF_Prod_Data_Migration_q_ktxt_acc_u");
}
tStatCatcher_1.addMessage(status == "" ? "end" : status,
  (end - startTime));
try {
 tStatCatcher_1Process(globalMap);
} catch (java.lang.Exception e) {
 e.printStackTrace();
}
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_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);
```

```
 }
}
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;
  // judege if the left string includes escape chars
  for (String[] strArray : escapeChars) {
   index = keyValue.indexOf(strArray, currIndex);
   if (index >= 0) {
    result.append(keyValue.substring(currIndex,
      index + strArray.length()).replace(strArray,
      strArray));
    currIndex = index + strArray.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;
}
/************************************************************************************************
 * 171631 characters generated by Talend Open Studio for Data Integration on the
 * February 25, 2016 5:16:30 PM CET
 ************************************************************************************************/
```