

Job code below. Sorry code tags didnt work for this for some reason.

```
// =====  
//  
// Copyright (c) 2005-2007, Talend Inc.  
//  
// This source code has been automatically generated by Talend Open Studio  
// / JobDesigner (CodeGenerator version 2.2.4.r7943).  
// You can find more information about Talend products at www.talend.com.  
// You may distribute this code under the terms of the GNU LGPL license  
// ( http://www.gnu.org/licenses/lgpl.html).  
//  
// =====  
package advice_now.transformlinklibrary;  
import routines.DataOperation;  
import routines.Mathematical;  
import routines.Numeric;  
import routines.Relational;  
import routines.StringHandling;  
import routines.TalendDate;  
import routines.TalendString;  
import routines.system.*;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.Date;  
import java.util.List;  
/**  
 * Job: TransformLinkLibrary Purpose: migrate, transform and input<br>  
 * Description: To migrate, transform and input the link library from the OO4 db  
 * to the oo5 db <br>  
 *  
 * @author harin@othermedia.com  
 * @version 0.1  
 * @status  
 */  
public class TransformLinkLibrary {  
 // create and load default properties  
 private static java.util.Properties defaultProps = new java.util.Properties();  
 // create application properties with default  
 private static java.util.Properties context = new java.util.Properties();  
 private static final String jobName = "TransformLinkLibrary";  
 private static final String projectName = "ADVICE_NOW";  
 public static Integer errorCode = null;  
 private static String currentComponent = "";  
 private static final java.util.Map<String, Long> start_Hash = new java.util.HashMap<String, Long>();  
 private static final java.util.Map<String, Long> end_Hash = new java.util.HashMap<String, Long>();  
 private static final java.util.Map<String, Boolean> ok_Hash = new java.util.HashMap<String, Boolean>();  
 private static final java.util.Map<String, Object> globalMap = new java.util.HashMap<String, Object>();  
 private class TalendException extends Exception {  
 private Exception e = null;  
 private TransformLinkLibrary c = null;  
 private TalendException(TransformLinkLibrary c, Exception e) {  
 this.e = e;
```

```

this.c = c;
}
@Override
public void printStackTrace() {
if (!(e instanceof TalendException || e instanceof TDieException)) {
globalMap.put(currentComponent + "_ERROR_MESSAGE", e
.getMessage());
System.err
.println("Exception in component " + currentComponent);
}
if (!(e instanceof TDieException)) {
e.printStackTrace();
}
if (!(e instanceof TalendException)) {
try {
for (java.lang.reflect.Method m : this.getClass()
.getEnclosingClass().getMethods()) {
if (m.getName().compareTo(currentComponent + "_error") == 0) {
m.invoke(c, (Object[]) null);
break;
}
}
} catch (java.lang.SecurityException e) {
this.e.printStackTrace();
} catch (java.lang.IllegalArgumentException e) {
this.e.printStackTrace();
} catch (java.lang.IllegalAccessException e) {
this.e.printStackTrace();
} catch (java.lang.reflect.InvocationTargetException e) {
this.e.printStackTrace();
}
}
}
}
}
}
public void tPostgresqlInput_1_error() throws TalendException {
end_Hash.put("tPostgresqlInput_1", System.currentTimeMillis());
}
public void tMap_1_error() throws TalendException {
end_Hash.put("tMap_1", System.currentTimeMillis());
}
public void tPostgresqlOutput_1_error() throws TalendException {
end_Hash.put("tPostgresqlOutput_1", System.currentTimeMillis());
}
public void tPostgresqlInput_2_error() throws TalendException {
end_Hash.put("tPostgresqlInput_2", System.currentTimeMillis());
}
public void tAdvancedHash_row2_error() throws TalendException {
end_Hash.put("tAdvancedHash_row2", System.currentTimeMillis());
}
private class adviceNowCmsLinkStruct {
private Integer id;
private String type;
private Boolean deleted;
}

```

```

private String name;
private String keywords;
private String url;
private java.util.Date creation_timestamp;
private Integer creator_user_id;
private java.util.Date modification_timestamp;
private Integer modifier_user_id;
private String object_class;
private Long object_id;
private Long template_mapping_id;
private String caption;
private String target;
}
private class row1Struct {
private int id;
private Integer creator_id;
private java.util.Date creation_date;
private Boolean deleted;
private Integer library_type;
}
private class after_tPostgresqlInput_1Struct {
private static final int DEFAULT_HASHCODE = 1;
private static final int PRIME = 31;
private int hashCode = DEFAULT_HASHCODE;
public boolean hashCodeDirty = true;
private int id;
private Integer creator_id;
private java.util.Date creation_date;
private Boolean deleted;
private Integer library_type;
@Override
public int hashCode() {
if (this.hashCodeDirty) {
final int prime = PRIME;
int result = DEFAULT_HASHCODE;
result = prime * result + (int) this.id;
this.hashCode = result;
this.hashCodeDirty = false;
}
return this.hashCode;
}
@Override
public boolean equals(Object obj) {
if (this == obj)
return true;
if (obj == null)
return false;
if (getClass() != obj.getClass())
return false;
final after_tPostgresqlInput_1Struct other = (after_tPostgresqlInput_1Struct) obj;
if (this.id != other.id)
return false;
return true;
}
}

```

```

}
}
public void tPostgresqlInput_1Process() throws TalendException {
try {
tPostgresqlInput_2Process();
row1Struct row1 = new row1Struct();
adviceNowCmsLinkStruct adviceNowCmsLink = new adviceNowCmsLinkStruct();
/**
* start
*/
ok_Hash.put("tPostgresqlOutput_1", false);
start_Hash.put("tPostgresqlOutput_1", System.currentTimeMillis());
currentComponent = "tPostgresqlOutput_1";
String dbschema_tPostgresqlOutput_1 = null;
dbschema_tPostgresqlOutput_1 = "";
String tableName_tPostgresqlOutput_1 = null;
if (dbschema_tPostgresqlOutput_1 == null
|| dbschema_tPostgresqlOutput_1.trim().length() == 0) {
tableName_tPostgresqlOutput_1 = "\"" + "cms_link_test" + "\"";
} else {
tableName_tPostgresqlOutput_1 = "\""
+ dbschema_tPostgresqlOutput_1 + "\".\"\"
+ "cms_link_test" + "\"";
}
int nb_line_tPostgresqlOutput_1 = 0;
int nb_line_update_tPostgresqlOutput_1 = 0;
int nb_line_inserted_tPostgresqlOutput_1 = 0;
int nb_line_deleted_tPostgresqlOutput_1 = 0;
int deletedCount_tPostgresqlOutput_1 = 0;
int updatedCount_tPostgresqlOutput_1 = 0;
int insertedCount_tPostgresqlOutput_1 = 0;
boolean whetherReject_tPostgresqlOutput_1 = false;
java.sql.Connection conn_tPostgresqlOutput_1 = null;
java.lang.Class.forName("org.postgresql.Driver");
String url_tPostgresqlOutput_1 = "jdbc:postgresql://" + "maia"
+ ":" + "5432" + "/" + "advicenow-rebuild";
String dbUser_tPostgresqlOutput_1 = "advicenow";
String dbPwd_tPostgresqlOutput_1 = "chaepu";
conn_tPostgresqlOutput_1 = java.sql.DriverManager.getConnection(
url_tPostgresqlOutput_1, dbUser_tPostgresqlOutput_1,
dbPwd_tPostgresqlOutput_1);
java.sql.PreparedStatement pstmt_tPostgresqlOutput_1 = null;
pstmt_tPostgresqlOutput_1 = conn_tPostgresqlOutput_1
.prepareStatement("INSERT INTO "
+ tableName_tPostgresqlOutput_1
+ "
(\"id\", \"type\", \"deleted\", \"name\", \"keywords\", \"url\", \"creation_timestamp\", \"creator_user_id\", \"modification_timestamp\", \"modifier_user_id\", \"object_class\", \"object_id\", \"template_mapping_id\", \"
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
int commitEvery_tPostgresqlOutput_1 = 100;
int commitCounter_tPostgresqlOutput_1 = 0;
conn_tPostgresqlOutput_1.setAutoCommit(false);
/**
* stop

```

```

*/
/**
* start
*/
ok_Hash.put("tMap_1", false);
start_Hash.put("tMap_1", System.currentTimeMillis());
currentComponent = "tMap_1";
// #####
// # Lookup's keys initialization
org.talend.designer.components.commons.AdvancedLookup<row2Struct> tHash_Lookup_row2 = (org.talend.designer.components.commons.AdvancedLookup<row2Struct>) globalMap
.get("tHash_Lookup_row2");
row2Struct row2HashKey = new row2Struct();
row2Struct row2Default = new row2Struct();
// #####
// #####
// # Vars initialization
// #####
// #####
// # Outputs initialization
adviceNowCmsLinkStruct adviceNowCmsLink_tmp = new adviceNowCmsLinkStruct();
// #####
/**
* stop
*/
/**
* start
*/
ok_Hash.put("tPostgresqlInput_1", false);
start_Hash.put("tPostgresqlInput_1", System.currentTimeMillis());
currentComponent = "tPostgresqlInput_1";
int nb_line_tPostgresqlInput_1 = 0;
java.sql.Connection conn_tPostgresqlInput_1 = null;
java.lang.Class.forName("org.postgresql.Driver");
String url_tPostgresqlInput_1 = "jdbc:postgresql://" + "electra"
+ "." + "5432" + "/" + "advicenow";
String dbUser_tPostgresqlInput_1 = "advicenow";
String dbPwd_tPostgresqlInput_1 = "liepai";
conn_tPostgresqlInput_1 = java.sql.DriverManager.getConnection(
url_tPostgresqlInput_1, dbUser_tPostgresqlInput_1,
dbPwd_tPostgresqlInput_1);
java.sql.Statement stmt_tPostgresqlInput_1 = conn_tPostgresqlInput_1
.createStatement();
java.sql.ResultSet rs_tPostgresqlInput_1 = stmt_tPostgresqlInput_1
.executeQuery("SELECT cms_library.\"id\", cms_library.\"creator_id\", cms_library.\"creation_date\", cms_library.\"deleted\", cms_library.\"library_type\" FROM cms_library");
java.sql.ResultSetMetaData rsmd_tPostgresqlInput_1 = rs_tPostgresqlInput_1
.getMetaData();
int colQtyInRs_tPostgresqlInput_1 = rsmd_tPostgresqlInput_1
.getColumnCount();
globalMap
.put(
"tPostgresqlInput_1_QUERY",
"SELECT cms_library.\"id\", cms_library.\"creator_id\", cms_library.\"creation_date\", cms_library.\"deleted\", cms_library.\"library_type\" FROM cms_library");
while (rs_tPostgresqlInput_1.next()) {

```

```
nb_line_tPostgresqlInput_1++;
if (colQtyInRs_tPostgresqlInput_1 < 1) {
row1.id = 0;
} else {
if (rs_tPostgresqlInput_1.getObject(1) != null) {
row1.id = rs_tPostgresqlInput_1.getInt(1);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_1 < 2) {
row1.creator_id = null;
} else {
if (rs_tPostgresqlInput_1.getObject(2) != null) {
row1.creator_id = rs_tPostgresqlInput_1.getInt(2);
} else {
row1.creator_id = null;
}
}
if (colQtyInRs_tPostgresqlInput_1 < 3) {
row1.creation_date = null;
} else {
if (rs_tPostgresqlInput_1.getTimestamp(3) != null) {
row1.creation_date = new java.util.Date(
rs_tPostgresqlInput_1.getTimestamp(3).getTime());
} else {
row1.creation_date = null;
}
}
if (colQtyInRs_tPostgresqlInput_1 < 4) {
row1.deleted = null;
} else {
if (rs_tPostgresqlInput_1.getObject(4) != null) {
row1.deleted = rs_tPostgresqlInput_1.getBoolean(4);
} else {
row1.deleted = null;
}
}
if (colQtyInRs_tPostgresqlInput_1 < 5) {
row1.library_type = null;
} else {
if (rs_tPostgresqlInput_1.getObject(5) != null) {
row1.library_type = rs_tPostgresqlInput_1.getInt(5);
} else {
row1.library_type = null;
}
}
/**
* stop
*/
/**
* start
```

```

*/
currentComponent = "tPostgresqlInput_1";
/**
* stop
*/
/**
* start
*/
currentComponent = "tMap_1";
boolean rejectedInnerJoin_tMap_1 = false;
// #####
// # Input tables (lookups)
row2HashKey.library_id = row1.id;
boolean forceLooprow2 = false;
int sizeResultsFromLookup_row2 = -1;
java.util.List<row2Struct> listFromLookup_row2 = null;
row2Struct row2ObjectFromLookup = null;
if (!rejectedInnerJoin_tMap_1) { // G 20
row2HashKey.hashCodeDirty = true;
tHash_Lookup_row2.get(row2HashKey);
if (tHash_Lookup_row2.hasResult()) { // G 90
if (tHash_Lookup_row2.resultIsObject()) { // G 49
row2ObjectFromLookup = (row2Struct) tHash_Lookup_row2
.getResultObject();
sizeResultsFromLookup_row2 = row2ObjectFromLookup != null ? 1
: -1;
} else { // G 49
listFromLookup_row2 = (java.util.List<row2Struct>) tHash_Lookup_row2
.getResultList();
sizeResultsFromLookup_row2 = listFromLookup_row2 != null ? listFromLookup_row2
.size()
: -1;
}
} // G 90
else { // G 91
rejectedInnerJoin_tMap_1 = true;
forceLooprow2 = true;
} // G 91
} // G 20
else { // G 21
forceLooprow2 = true;
} // G 21
for (int irow2 = 0; sizeResultsFromLookup_row2 != -1
&& irow2 < sizeResultsFromLookup_row2 || forceLooprow2; irow2++) {
row2Struct fromLookup_row2 = null;
row2Struct row2 = row2Default;
if (!forceLooprow2) { // G 46
if (row2ObjectFromLookup != null) {
fromLookup_row2 = row2ObjectFromLookup;
row2 = fromLookup_row2;
}
else if (listFromLookup_row2 != null) {
fromLookup_row2 = listFromLookup_row2.get(irow2);
}
}
}

```

```

row2 = fromLookup_row2;
}
if (!rejectedInnerJoin_tMap_1 &&
!(
row1.library_type == 3
)
) { // G 16
fromLookup_row2 = null;
rejectedInnerJoin_tMap_1 = true;
} // G 16
} // G 46
forceLooprow2 = false;
// #####
{ // start of Var scope
// #####
// # Vars tables
// #####
// #####
// # Output tables
adviceNowCmsLink = null;
if (!rejectedInnerJoin_tMap_1) {
// # Output table : 'adviceNowCmsLink'
adviceNowCmsLink_tmp.id = row2.id;
adviceNowCmsLink_tmp.type = "E";
adviceNowCmsLink_tmp.deleted = null;
adviceNowCmsLink_tmp.name = row2.name;
adviceNowCmsLink_tmp.keywords = row2.keywords;
adviceNowCmsLink_tmp.url = row2.url;
adviceNowCmsLink_tmp.creation_timestamp = null;
adviceNowCmsLink_tmp.creator_user_id = 1;
adviceNowCmsLink_tmp.modification_timestamp = null;
adviceNowCmsLink_tmp.modifier_user_id = 1;
adviceNowCmsLink_tmp.object_class = null;
adviceNowCmsLink_tmp.object_id = null;
adviceNowCmsLink_tmp.template_mapping_id = null;
adviceNowCmsLink_tmp.caption = null;
adviceNowCmsLink_tmp.target = null;
adviceNowCmsLink = adviceNowCmsLink_tmp;
} // closing inner join bracket (2)
// #####
} // end of Var scope
rejectedInnerJoin_tMap_1 = false;
/**
* stop
*/
// Start of branch "adviceNowCmsLink"
if (adviceNowCmsLink != null) {
/**
* start
*/
currentComponent = "tPostgresqlOutput_1";
whetherReject_tPostgresqlOutput_1 = false;
if (adviceNowCmsLink.id == null) {

```

```
pstmt_tPostgresqlOutput_1.setNull(1,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(1,
adviceNowCmsLink.id);
}
if (adviceNowCmsLink.type == null) {
pstmt_tPostgresqlOutput_1.setNull(2,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(2,
adviceNowCmsLink.type);
}
if (adviceNowCmsLink.deleted == null) {
pstmt_tPostgresqlOutput_1.setNull(3,
java.sql.Types.BOOLEAN);
} else {
pstmt_tPostgresqlOutput_1.setBoolean(3,
adviceNowCmsLink.deleted);
}
if (adviceNowCmsLink.name == null) {
pstmt_tPostgresqlOutput_1.setNull(4,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(4,
adviceNowCmsLink.name);
}
if (adviceNowCmsLink.keywords == null) {
pstmt_tPostgresqlOutput_1.setNull(5,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(5,
adviceNowCmsLink.keywords);
}
if (adviceNowCmsLink.url == null) {
pstmt_tPostgresqlOutput_1.setNull(6,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(6,
adviceNowCmsLink.url);
}
if (adviceNowCmsLink.creation_timestamp != null) {
pstmt_tPostgresqlOutput_1.setTimestamp(7,
new java.sql.Timestamp(
adviceNowCmsLink.creation_timestamp
.getTime()));
} else {
pstmt_tPostgresqlOutput_1.setNull(7,
java.sql.Types.DATE);
}
if (adviceNowCmsLink.creator_user_id == null) {
pstmt_tPostgresqlOutput_1.setNull(8,
java.sql.Types.INTEGER);
}
```

```
} else {
pstmt_tPostgresqlOutput_1.setInt(8,
adviceNowCmsLink.creator_user_id);
}
if (adviceNowCmsLink.modification_timestamp != null) {
pstmt_tPostgresqlOutput_1
.setTimestamp(
9,
new java.sql.Timestamp(
adviceNowCmsLink.modification_timestamp
.getTime()));
} else {
pstmt_tPostgresqlOutput_1.setNull(9,
java.sql.Types.DATE);
}
if (adviceNowCmsLink.modifier_user_id == null) {
pstmt_tPostgresqlOutput_1.setNull(10,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(10,
adviceNowCmsLink.modifier_user_id);
}
if (adviceNowCmsLink.object_class == null) {
pstmt_tPostgresqlOutput_1.setNull(11,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(11,
adviceNowCmsLink.object_class);
}
if (adviceNowCmsLink.object_id == null) {
pstmt_tPostgresqlOutput_1.setNull(12,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setLong(12,
adviceNowCmsLink.object_id);
}
if (adviceNowCmsLink.template_mapping_id == null) {
pstmt_tPostgresqlOutput_1.setNull(13,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setLong(13,
adviceNowCmsLink.template_mapping_id);
}
if (adviceNowCmsLink.caption == null) {
pstmt_tPostgresqlOutput_1.setNull(14,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(14,
adviceNowCmsLink.caption);
}
if (adviceNowCmsLink.target == null) {
pstmt_tPostgresqlOutput_1.setNull(15,
java.sql.Types.VARCHAR);
}
```

```

} else {
pstmt_tPostgresqlOutput_1.setString(15,
adviceNowCmsLink.target);
}
try {
insertedCount_tPostgresqlOutput_1 = insertedCount_tPostgresqlOutput_1
+ pstmt_tPostgresqlOutput_1.executeUpdate();
nb_line_tPostgresqlOutput_1++;
} catch (Exception e) {
whetherReject_tPostgresqlOutput_1 = true;
throw (e);
}
commitCounter_tPostgresqlOutput_1++;
if (commitEvery_tPostgresqlOutput_1 <= commitCounter_tPostgresqlOutput_1) {
conn_tPostgresqlOutput_1.commit();
commitCounter_tPostgresqlOutput_1 = 0;
}
/**
* stop
*/
} // End of branch "adviceNowCmsLink"
} // close loop of lookup 'row2'
/**
* start
*/
currentComponent = "tPostgresqlInput_1";
}
stmt_tPostgresqlInput_1.close();
conn_tPostgresqlInput_1.close();
globalMap.put("tPostgresqlInput_1_NB_LINE",
nb_line_tPostgresqlInput_1);
ok_Hash.put("tPostgresqlInput_1", true);
end_Hash.put("tPostgresqlInput_1", System.currentTimeMillis());
/**
* stop
*/
/**
* start
*/
currentComponent = "tMap_1";
// #####
// # Lookup hashes releasing
globalMap.remove("tHash_row2");
// #####
ok_Hash.put("tMap_1", true);
end_Hash.put("tMap_1", System.currentTimeMillis());
/**
* stop
*/
/**
* start
*/
currentComponent = "tPostgresqlOutput_1";

```

```

pstmt_tPostgresqlOutput_1.close();
conn_tPostgresqlOutput_1.commit();
conn_tPostgresqlOutput_1.close();
nb_line_deleted_tPostgresqlOutput_1 = nb_line_deleted_tPostgresqlOutput_1
+ deletedCount_tPostgresqlOutput_1;
nb_line_update_tPostgresqlOutput_1 = nb_line_update_tPostgresqlOutput_1
+ updatedCount_tPostgresqlOutput_1;
nb_line_inserted_tPostgresqlOutput_1 = nb_line_inserted_tPostgresqlOutput_1
+ insertedCount_tPostgresqlOutput_1;
globalMap.put("tPostgresqlOutput_1_NB_LINE",
nb_line_tPostgresqlOutput_1);
globalMap.put("tPostgresqlOutput_1_NB_LINE_UPDATED",
nb_line_update_tPostgresqlOutput_1);
globalMap.put("tPostgresqlOutput_1_NB_LINE_INSERTED",
nb_line_inserted_tPostgresqlOutput_1);
globalMap.put("tPostgresqlOutput_1_NB_LINE_DELETED",
nb_line_deleted_tPostgresqlOutput_1);
ok_Hash.put("tPostgresqlOutput_1", true);
end_Hash.put("tPostgresqlOutput_1", System.currentTimeMillis());
/**
 * stop
 */
} catch (Exception e) {
throw new TalendException(this, e);
}
}
private class row2Struct {
private static final int DEFAULT_HASHCODE = 1;
private static final int PRIME = 31;
private int hashCode = DEFAULT_HASHCODE;
public boolean hashCodeDirty = true;
private int id;
private int library_id;
private short library_type_del;
private boolean current;
private String name;
private String description;
private String author;
private String publisher;
private String keywords;
private String copyright;
private int language_id;
private String text;
private int category_id;
private String client_ref;
private int modifier_id;
private java.util.Date modified_date;
private String file_name;
private String original_file_name;
private int file_type;
private int file_size;
private int width;
private int height;

```

```

private String url;
private boolean active_url;
@Override
public int hashCode() {
if (this.hashCodeDirty) {
final int prime = PRIME;
int result = DEFAULT_HASHCODE;
result = prime * result + (int) this.library_id;
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 row2Struct other = (row2Struct) obj;
if (this.library_id != other.library_id)
return false;
return true;
}
}
public void tPostgresqlInput_2Process() throws TalendException {
try {
row2Struct row2 = new row2Struct();
/**
* start
*/
ok_Hash.put("tAdvancedHash_row2", false);
start_Hash.put("tAdvancedHash_row2", System.currentTimeMillis());
currentComponent = "tAdvancedHash_row2";
org.talend.designer.components.commons.AdvancedLookup.MATCHING_MODE matchingModeEnum_row2 =
org.talend.designer.components.commons.AdvancedLookup.MATCHING_MODE.ALL_MATCHES;
org.talend.designer.components.commons.AdvancedLookup<row2Struct> tHash_Lookup_row2 = org.talend.designer.components.commons.AdvancedLookup
.<row2Struct> getLookup(matchingModeEnum_row2);
globalMap.put("tHash_Lookup_row2", tHash_Lookup_row2);
/**
* stop
*/
/**
* start
*/
ok_Hash.put("tPostgresqlInput_2", false);
start_Hash.put("tPostgresqlInput_2", System.currentTimeMillis());
currentComponent = "tPostgresqlInput_2";
int nb_line_tPostgresqlInput_2 = 0;
java.sql.Connection conn_tPostgresqlInput_2 = null;
java.lang.Class.forName("org.postgresql.Driver");

```

```

String url_tPostgresqlInput_2 = "jdbc 🍌ostgresql://" + "electra"
+ ":" + "5432" + "/" + "advicenow";
String dbUser_tPostgresqlInput_2 = "advicenow";
String dbPwd_tPostgresqlInput_2 = "liepai";
conn_tPostgresqlInput_2 = java.sql.DriverManager.getConnection(
url_tPostgresqlInput_2, dbUser_tPostgresqlInput_2,
dbPwd_tPostgresqlInput_2);
java.sql.Statement stmt_tPostgresqlInput_2 = conn_tPostgresqlInput_2
.createStatement();
java.sql.ResultSet rs_tPostgresqlInput_2 = stmt_tPostgresqlInput_2
.executeQuery("SELECT \"cms_library_data\".\"id\", \"cms_library_data\".\"library_id\", \"cms_library_data\".\"library_type_del\", \"cms_library_data\".\"current\", \"cms_library_data\".\"name\",
\"cms_library_data\".\"description\", \"cms_library_data\".\"author\", \"cms_library_data\".\"publisher\", \"cms_library_data\".\"keywords\", \"cms_library_data\".\"copyright\",
\"cms_library_data\".\"language_id\", \"cms_library_data\".\"text\", \"cms_library_data\".\"category_id\", \"cms_library_data\".\"client_ref\", \"cms_library_data\".\"modifier_id\",
\"cms_library_data\".\"modified_date\", \"cms_library_data\".\"file_name\", \"cms_library_data\".\"original_file_name\", \"cms_library_data\".\"file_type\", \"cms_library_data\".\"file_size\",
\"cms_library_data\".\"width\", \"cms_library_data\".\"height\", \"cms_library_data\".\"url\", \"cms_library_data\".\"active_url\" FROM \"cms_library_data\"");
java.sql.ResultSetMetaData rsmd_tPostgresqlInput_2 = rs_tPostgresqlInput_2
.getMetaData();
int colQtyInRs_tPostgresqlInput_2 = rsmd_tPostgresqlInput_2
.getColumnCount();
globalMap
.put(
"tPostgresqlInput_2_QUERY",
"SELECT \"cms_library_data\".\"id\", \"cms_library_data\".\"library_id\", \"cms_library_data\".\"library_type_del\", \"cms_library_data\".\"current\", \"cms_library_data\".\"name\",
\"cms_library_data\".\"description\", \"cms_library_data\".\"author\", \"cms_library_data\".\"publisher\", \"cms_library_data\".\"keywords\", \"cms_library_data\".\"copyright\",
\"cms_library_data\".\"language_id\", \"cms_library_data\".\"text\", \"cms_library_data\".\"category_id\", \"cms_library_data\".\"client_ref\", \"cms_library_data\".\"modifier_id\",
\"cms_library_data\".\"modified_date\", \"cms_library_data\".\"file_name\", \"cms_library_data\".\"original_file_name\", \"cms_library_data\".\"file_type\", \"cms_library_data\".\"file_size\",
\"cms_library_data\".\"width\", \"cms_library_data\".\"height\", \"cms_library_data\".\"url\", \"cms_library_data\".\"active_url\" FROM \"cms_library_data\"");
while (rs_tPostgresqlInput_2.next()) {
nb_line_tPostgresqlInput_2++;
if (colQtyInRs_tPostgresqlInput_2 < 1) {
row2.id = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(1) != null) {
row2.id = rs_tPostgresqlInput_2.getInt(1);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 2) {
row2.library_id = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(2) != null) {
row2.library_id = rs_tPostgresqlInput_2.getInt(2);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 3) {
row2.library_type_del = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(3) != null) {

```

```
row2.library_type_del = rs_tPostgresqlInput_2
.getShort(3);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 4) {
row2.current = false;
} else {
if (rs_tPostgresqlInput_2.getObject(4) != null) {
row2.current = rs_tPostgresqlInput_2.getBoolean(4);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 5) {
row2.name = null;
} else {
if (rs_tPostgresqlInput_2.getObject(5) != null) {
row2.name = rs_tPostgresqlInput_2.getString(5);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 6) {
row2.description = null;
} else {
if (rs_tPostgresqlInput_2.getObject(6) != null) {
row2.description = rs_tPostgresqlInput_2.getString(6);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 7) {
row2.author = null;
} else {
if (rs_tPostgresqlInput_2.getObject(7) != null) {
row2.author = rs_tPostgresqlInput_2.getString(7);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 8) {
row2.publisher = null;
} else {
if (rs_tPostgresqlInput_2.getObject(8) != null) {
row2.publisher = rs_tPostgresqlInput_2.getString(8);
} else {
```

```
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 9) {
row2.keywords = null;
} else {
if (rs_tPostgresqlInput_2.getObject(9) != null) {
row2.keywords = rs_tPostgresqlInput_2.getString(9);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 10) {
row2.copyright = null;
} else {
if (rs_tPostgresqlInput_2.getObject(10) != null) {
row2.copyright = rs_tPostgresqlInput_2.getString(10);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 11) {
row2.language_id = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(11) != null) {
row2.language_id = rs_tPostgresqlInput_2.getInt(11);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 12) {
row2.text = null;
} else {
if (rs_tPostgresqlInput_2.getObject(12) != null) {
row2.text = rs_tPostgresqlInput_2.getString(12);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 13) {
row2.category_id = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(13) != null) {
row2.category_id = rs_tPostgresqlInput_2.getInt(13);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
```

```
}
if (colQtyInRs_tPostgresqlInput_2 < 14) {
row2.client_ref = null;
} else {
if (rs_tPostgresqlInput_2.getObject(14) != null) {
row2.client_ref = rs_tPostgresqlInput_2.getString(14);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 15) {
row2.modifier_id = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(15) != null) {
row2.modifier_id = rs_tPostgresqlInput_2.getInt(15);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 16) {
row2.modified_date = null;
} else {
if (rs_tPostgresqlInput_2.getTimestamp(16) != null) {
row2.modified_date = new java.util.Date(
rs_tPostgresqlInput_2.getTimestamp(16)
.getTime());
} else {
row2.modified_date = null;
}
}
if (colQtyInRs_tPostgresqlInput_2 < 17) {
row2.file_name = null;
} else {
if (rs_tPostgresqlInput_2.getObject(17) != null) {
row2.file_name = rs_tPostgresqlInput_2.getString(17);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 18) {
row2.original_file_name = null;
} else {
if (rs_tPostgresqlInput_2.getObject(18) != null) {
row2.original_file_name = rs_tPostgresqlInput_2
.getString(18);
} else {
row2.original_file_name = null;
}
}
if (colQtyInRs_tPostgresqlInput_2 < 19) {
```

```
row2.file_type = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(19) != null) {
row2.file_type = rs_tPostgresqlInput_2.getInt(19);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 20) {
row2.file_size = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(20) != null) {
row2.file_size = rs_tPostgresqlInput_2.getInt(20);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 21) {
row2.width = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(21) != null) {
row2.width = rs_tPostgresqlInput_2.getInt(21);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 22) {
row2.height = 0;
} else {
if (rs_tPostgresqlInput_2.getObject(22) != null) {
row2.height = rs_tPostgresqlInput_2.getInt(22);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 23) {
row2.url = null;
} else {
if (rs_tPostgresqlInput_2.getObject(23) != null) {
row2.url = rs_tPostgresqlInput_2.getString(23);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
if (colQtyInRs_tPostgresqlInput_2 < 24) {
row2.active_url = false;
} else {
if (rs_tPostgresqlInput_2.getObject(24) != null) {
```

```

row2.active_url = rs_tPostgresqlInput_2.getBoolean(24);
} else {
throw new RuntimeException(
"Null value in non-Nullable column");
}
}
/**
* stop
*/
/**
* start
*/
currentComponent = "tPostgresqlInput_2";
/**
* stop
*/
/**
* start
*/
currentComponent = "tAdvancedHash_row2";
row2Struct row2_HashRow = new row2Struct();
row2_HashRow.id = row2.id;
row2_HashRow.library_id = row2.library_id;
row2_HashRow.library_type_del = row2.library_type_del;
row2_HashRow.current = row2.current;
row2_HashRow.name = row2.name;
row2_HashRow.description = row2.description;
row2_HashRow.author = row2.author;
row2_HashRow.publisher = row2.publisher;
row2_HashRow.keywords = row2.keywords;
row2_HashRow.copyright = row2.copyright;
row2_HashRow.language_id = row2.language_id;
row2_HashRow.text = row2.text;
row2_HashRow.category_id = row2.category_id;
row2_HashRow.client_ref = row2.client_ref;
row2_HashRow.modifier_id = row2.modifier_id;
row2_HashRow.modified_date = row2.modified_date;
row2_HashRow.file_name = row2.file_name;
row2_HashRow.original_file_name = row2.original_file_name;
row2_HashRow.file_type = row2.file_type;
row2_HashRow.file_size = row2.file_size;
row2_HashRow.width = row2.width;
row2_HashRow.height = row2.height;
row2_HashRow.url = row2.url;
row2_HashRow.active_url = row2.active_url;
tHash_Lookup_row2.put(row2_HashRow);
/**
* stop
*/
/**
* start
*/
currentComponent = "tPostgresqlInput_2";

```

```

}
stmt_tPostgresqlInput_2.close();
conn_tPostgresqlInput_2.close();
globalMap.put("tPostgresqlInput_2_NB_LINE",
nb_line_tPostgresqlInput_2);
ok_Hash.put("tPostgresqlInput_2", true);
end_Hash.put("tPostgresqlInput_2", System.currentTimeMillis());
/**
 * stop
 */
/**
 * start
 */
currentComponent = "tAdvancedHash_row2";
ok_Hash.put("tAdvancedHash_row2", true);
end_Hash.put("tAdvancedHash_row2", System.currentTimeMillis());
/**
 * stop
 */
} catch (Exception e) {
throw new TalendException(this, e);
}
}
public static boolean watch = false;
public static int portStats = 3334;
public static int portTraces = 4334;
public static String clientHost;
public static String defaultClientHost = "localhost";
public static String contextStr = "Default";
public static String pid = "0";
public static String rootPid = null;
public static String fatherPid = null;
private static java.util.Properties context_param = new java.util.Properties();
public static String status = "";
public static void main(String[] args) {
int exitCode = runJob(args);
System.exit(exitCode);
}
public static synchronized int runJob(String[] args) {
init();
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;

```

```

}
pid = TalendString.getAsciiRandomString(6);
if (rootPid == null) {
rootPid = pid;
}
if (fatherPid == null) {
fatherPid = pid;
}
try {
java.io.InputStream inContext = TranformLinkLibrary.class
.getClassLoader()
.getResourceAsStream(
"advice_now/tranformlinklibrary/contexts/Default.properties");
if (inContext != null) {
defaultProps.load(inContext);
inContext.close();
context = new java.util.Properties(defaultProps);
}
if (contextStr.compareTo("Default") != 0) {
inContext = TranformLinkLibrary.class.getClassLoader()
.getResourceAsStream(
"advice_now/tranformlinklibrary/contexts/"
+ contextStr + ".properties");
if (inContext != null) {
context.load(inContext);
inContext.close();
}
}
if (!context_param.isEmpty()) {
context.putAll(context_param);
}
} catch (java.io.IOException ie) {
System.err.println("Could not load context " + contextStr);
ie.printStackTrace();
}
long end = 0;
long start = System.currentTimeMillis();
TranformLinkLibrary TranformLinkLibraryClass = new TranformLinkLibrary();
try {
TranformLinkLibraryClass.tPostgresqlInput_1Process();
} catch (TalendException e) {
status = "failure";
e.printStackTrace();
} finally {
end = System.currentTimeMillis();
if (watch) {
System.out.println((end - start) + " milliseconds");
}
}
if (errorCode == null) {
return status != null && status.equals("failure") ? 1 : 0;
} else {
return errorCode.intValue();
}

```

```

}
}
public static void evalParam(String arg) {
if (arg.startsWith("--watch")) {
watch = true;
} else if (arg.startsWith("--stat_port=")) {
portStats = Integer.parseInt(arg.substring(12));
} 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);
} else if (arg.startsWith("--father_pid=")) {
fatherPid = arg.substring(13);
} else if (arg.startsWith("--root_pid=")) {
rootPid = arg.substring(11);
} else if (arg.startsWith("--context_param")) {
String keyValue = arg.substring(16);
int index = -1;
if (keyValue != null && (index = keyValue.indexOf('=') > -1) {
context_param.put(keyValue.substring(0, index), keyValue
.substring(index + 1));
}
}
}
private static void init() {
defaultProps.clear();
context.clear();
errorCode = null;
currentComponent = "";
start_Hash.clear();
end_Hash.clear();
ok_Hash.clear();
globalMap.clear();
watch = false;
portStats = 3334;
portTraces = 4334;
clientHost = null;
defaultClientHost = "localhost";
contextStr = "Default";
pid = "0";
rootPid = null;
fatherPid = null;
context_param.clear();
status = "";
}
}
/*****
* 44514 characters generated by Talend OpenStudio on the 11 January 2008
* 11:40:18 GMT
*****/

```