

The generated code :

```
// =====  
//  
// Copyright (c) 2005-2007, Talend Inc.  
//  
// This source code has been automatically generated by Talend Open Studio  
// / JobDesigner (CodeGenerator version 2.3.0.r8623).  
// 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 gespat.migresite;  
import routines.DataOperation;  
import routines.Mathematical;  
import routines.Numeric;  
import routines.Relational;  
import routines.StringHandling;  
import routines.TalendDataGenerator;  
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;  
import java.math.BigDecimal;  
/**  
 * Job: MigreSite Purpose: <br>  
 * Description: <br>  
 *  
 * @author jean-michel.souchard@interieur.gouv.fr  
 * @version 0.1  
 * @status  
 */  
public class MigreSite {  
 // create and load default properties  
 private static java.util.Properties defaultProps = new java.util.Properties();  
 // create application properties with default  
 private static class ContextProperties extends java.util.Properties {  
 public ContextProperties(java.util.Properties properties) {  
 super(properties);  
 }  
 public ContextProperties() {  
 super();  
 }  
 }  
 private static ContextProperties context = new ContextProperties();  
 private static final String jobName = "MigreSite";  
 private static final String projectName = "GESPAT";  
 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>();
public static final java.util.List<String[]> globalBuffer = new java.util.ArrayList<String[]>();
private class TalendException extends Exception {
private Exception e = null;
private MigreSite c = null;
private TalendException(MigreSite 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, new Object[] { e });
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 tMSSqlInput_2_error(Exception exception) throws TalendException {
end_Hash.put("tMSSqlInput_2", System.currentTimeMillis());
tMSSqlInput_2_onSubJobError(exception);
}
public void tFilterRow_1_error(Exception exception) throws TalendException {
end_Hash.put("tFilterRow_1", System.currentTimeMillis());
tMSSqlInput_2_onSubJobError(exception);
}
public void tJavaRow_1_error(Exception exception) throws TalendException {
end_Hash.put("tJavaRow_1", System.currentTimeMillis());
}

```

```

tMSSqlInput_2_onSubJobError(exception);
}
public void tMap_1_error(Exception exception) throws TalendException {
end_Hash.put("tMap_1", System.currentTimeMillis());
tMSSqlInput_2_onSubJobError(exception);
}
public void tPostgresqlOutput_1_error(Exception exception)
throws TalendException {
end_Hash.put("tPostgresqlOutput_1", System.currentTimeMillis());
tMSSqlInput_2_onSubJobError(exception);
}
public void tMSSqlInput_2_onSubJobError(Exception exception)
throws TalendException {
}
private class row4Struct {
private static final int DEFAULT_HASHCODE = 1;
private static final int PRIME = 31;
private int hashCode = DEFAULT_HASHCODE;
public boolean hashCodeDirty = true;
Integer site_id;
Integer doma_id;
Integer util_id;
Integer com_id;
Integer tprop_id;
Integer tsite_id;
Integer icpe_id;
Integer erp_id;
Integer circo_id;
Integer sterr_id;
String site_lib;
String site_adr1;
String site_adr2;
String site_cp;
String site_obser;
Integer site_dom;
Integer site_surf;
Integer site_nbpa;
Integer site_pkex;
Integer site_pkco;
Integer site_bati;
Integer site_perim_mh;
java.util.Date date_modifs;
@Override
public int hashCode() {
if (this.hashCodeDirty) {
final int prime = PRIME;
int result = DEFAULT_HASHCODE;
result = prime
* result
+ ((this.site_id == null) ? 0 : this.site_id.hashCode());
this.hashCode = result;
this.hashCodeDirty = false;
}
}

```

```
return this.hashCode();
}
@Override
public boolean equals(Object obj) {
if (this == obj)
return true;
if (obj == null)
return false;
if (getClass() != obj.getClass())
return false;
final row4Struct other = (row4Struct) obj;
if (this.site_id == null) {
if (other.site_id != null)
return false;
} else if (!this.site_id.equals(other.site_id))
return false;
return true;
}
}
private class row3Struct {
Integer id_site;
Byte tpge_o_n;
String tgpe_site_numimmat;
String tgpe_site_numattri;
String tgpe_site_numtitre;
String tgpe_site_numprop;
String tgpe_site_numgeo;
String tgpe_site_cle;
String nom_site;
Integer destination_site;
Integer domaine_site;
Byte at_pn_site;
Integer site_circo;
Integer site_com;
String site_adr1;
String site_adr2;
Integer site_poste;
String site_cp;
String site_burdis;
String site_oberv;
Integer statut_terrain;
Integer type_prop;
Byte Column22;
Integer surface_totale;
Integer nombre_pers;
Integer places_surf;
Integer places_couv;
Byte nature;
Integer statut_occup;
Integer Column29;
Integer Column30;
Integer Column31;
Object Column32;
```

```
java.util.Date date_estim;
Object valeur_locat;
java.util.Date date_valeur_locat;
Integer type_prop_bati;
Integer Column37;
Byte maintenance;
Byte ouvrage;
Object montant_loyer;
java.util.Date date_valeur_loyer;
Integer IGH;
Integer ERP;
Integer ICPE;
Integer Column45;
Integer site_mh;
Integer action_site;
java.util.Date Column48;
Integer id_user;
java.util.Date date_user;
Integer id_group;
java.util.Date date_echeance_bail;
String bn_supprimeSite;
}
private class row2Struct {
Integer id_site;
Byte tpge_o_n;
String tgpe_site_numimmat;
String tgpe_site_numattri;
String tgpe_site_numtitre;
String tgpe_site_numprop;
String tgpe_site_numgeo;
String tgpe_site_cle;
String nom_site;
Integer destination_site;
Integer domaine_site;
Byte at_pn_site;
Integer site_circo;
Integer site_com;
String site_adr1;
String site_adr2;
Integer site_poste;
String site_cp;
String site_burdis;
String site_oberv;
Integer statut_terrain;
Integer type_prop;
Byte Column22;
Integer surface_totale;
Integer nombre_pers;
Integer places_surf;
Integer places_couv;
Byte nature;
Integer statut_occup;
Integer Column29;
```

```
Integer Column30;
Integer Column31;
Object Column32;
java.util.Date date_estim;
Object valeur_locat;
java.util.Date date_valeur_locat;
Integer type_prop_bati;
Integer Column37;
Byte maintenance;
Byte ouvrage;
Object montant_loyer;
java.util.Date date_valeur_loyer;
Integer IGH;
Integer ERP;
Integer ICPE;
Integer Column45;
Integer site_mh;
Integer action_site;
java.util.Date Column48;
Integer id_user;
java.util.Date date_user;
Integer id_group;
java.util.Date date_echeance_bail;
String bn_supprimeSite;
}
private class row1Struct {
Integer id_site;
Byte tpge_o_n;
String tgpe_site_numimmat;
String tgpe_site_numattri;
String tgpe_site_numtitre;
String tgpe_site_numprop;
String tgpe_site_numgeo;
String tgpe_site_cle;
String nom_site;
Integer destination_site;
Integer domaine_site;
Byte at_pn_site;
Integer site_circo;
Integer site_com;
String site_adr1;
String site_adr2;
Integer site_poste;
String site_cp;
String site_burdis;
String site_oberv;
Integer statut_terrain;
Integer type_prop;
Byte Column22;
Integer surface_totale;
Integer nombre_pers;
Integer places_surf;
Integer places_couv;
```

```

Byte nature;
Integer statut_occup;
Integer Column29;
Integer Column30;
Integer Column31;
Object Column32;
java.util.Date date_estim;
Object valeur_locat;
java.util.Date date_valeur_locat;
Integer type_prop_bati;
Integer Column37;
Byte maintenance;
Byte ouvrage;
Object montant_loyer;
java.util.Date date_valeur_loyer;
Integer IGH;
Integer ERP;
Integer ICPE;
Integer Column45;
Integer site_mh;
Integer action_site;
java.util.Date Column48;
Integer id_user;
java.util.Date date_user;
Integer id_group;
java.util.Date date_echeance_bail;
String bn_supprimeSite;
}
public void tMSSqlInput_2Process() throws TalendException {
try {
row1Struct row1 = new row1Struct();
row2Struct row2 = new row2Struct();
row3Struct row3 = new row3Struct();
row4Struct row4 = new row4Struct();
/**
* 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 = "public";
String tableName_tPostgresqlOutput_1 = null;
if (dbschema_tPostgresqlOutput_1 == null
|| dbschema_tPostgresqlOutput_1.trim().length() == 0) {
tableName_tPostgresqlOutput_1 = "site_temp";
} else {
tableName_tPostgresqlOutput_1 = dbschema_tPostgresqlOutput_1
+ "\".\"\" + "site_temp";
}
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://"
+ "10.237.4.117" + ":" + "5432" + "/" + "DPAT";
String dbUser_tPostgresqlOutput_1 = "gespat";
String dbPwd_tPostgresqlOutput_1 = "gespat";
conn_tPostgresqlOutput_1 = java.sql.DriverManager.getConnection(
url_tPostgresqlOutput_1, dbUser_tPostgresqlOutput_1,
dbPwd_tPostgresqlOutput_1);
java.sql.PreparedStatement pstmt_tPostgresqlOutput_1 = conn_tPostgresqlOutput_1
.prepareStatement("INSERT INTO \"
+ tableName_tPostgresqlOutput_1
+ "\"
(\"site_id\", \"doma_id\", \"util_id\", \"com_id\", \"tprop_id\", \"tsite_id\", \"icpe_id\", \"erp_id\", \"circo_id\", \"sterr_id\", \"site_lib\", \"site_adr1\", \"site_adr2\", \"site_cp\", \"site_obser\", \"site_dom\", \"site_surf\", \"sit
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
int commitEvery_tPostgresqlOutput_1 = 10000;
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
// #####
// #####
// # Vars initialization
// #####
// #####
// # Outputs initialization
row4Struct row4_tmp = new row4Struct();
// #####
/**
 * stop
 */
/**
 * start
 */
ok_Hash.put("tJavaRow_1", false);
start_Hash.put("tJavaRow_1", System.currentTimeMillis());
currentComponent = "tJavaRow_1";
int nb_line_tJavaRow_1 = 0;
/**

```

```

* stop
*/
/**
* start
*/
ok_Hash.put("tFilterRow_1", false);
start_Hash.put("tFilterRow_1", System.currentTimeMillis());
currentComponent = "tFilterRow_1";
int nb_line_tFilterRow_1 = 0;
int nb_line_ok_tFilterRow_1 = 0;
/**
* stop
*/
/**
* start
*/
ok_Hash.put("tMSSqlInput_2", false);
start_Hash.put("tMSSqlInput_2", System.currentTimeMillis());
currentComponent = "tMSSqlInput_2";
int nb_line_tMSSqlInput_2 = 0;
java.lang.Class.forName("net.sourceforge.jtds.jdbc.Driver");
String url_tMSSqlInput_2 = "jdbc:jtds:sqlserver://"
+ "10.237.4.113" + ":" + "1433" + "/" + "IMMOBILIER"
+ ";appName=" + projectName + ";" + "";
String dbUser_tMSSqlInput_2 = "sa";
String dbPwd_tMSSqlInput_2 = "";
java.sql.Connection conn_tMSSqlInput_2 = java.sql.DriverManager
.getConnection(url_tMSSqlInput_2, dbUser_tMSSqlInput_2,
dbPwd_tMSSqlInput_2);
java.sql.Statement stmt_tMSSqlInput_2 = conn_tMSSqlInput_2
.createStatement();
java.sql.ResultSet rs_tMSSqlInput_2 = stmt_tMSSqlInput_2
.executeQuery("SELECT Site.id_site, Site.tpge_o_n, Site.tpge_site_numimmat, Site.tpge_site_numattri, Site.tpge_site_numtitre, Site.tpge_site_numprop, Site.tpge_site_numgeo,
Site.tpge_site_cle, Site.nom_site, Site.destination_site, Site.domaine_site, Site.at_pn_site, Site.site_circo, Site.site_com, Site.site_adr1, Site.site_adr2, Site.site_poste, Site.site_cp,
Site.site_burdis, Site.site_oberv, Site.statut_terrain, Site.type_prop, Site.domianialité, Site.surface_totale, Site.nombre_pers, Site.places_surf, Site.places_couv, Site.nature, Site.statut_occup,
Site.mode_entrée, Site.année_entrée, Site.année_construc, Site.montant_estimé, Site.date_estim, Site.valeur_locat, Site.date_valeur_locat, Site.type_prop_bati, Site.année_entrée_bati,
Site.maintenance, Site.ouvrage, Site.montant_loyer, Site.date_valeur_loyer, Site.IGH, Site.ERP, Site.ICPE, Site.périm_mh, Site.site_mh, Site.action_site, Site.date_appréciation, Site.id_user,
Site.date_user, Site.id_group, Site.date_echeance_bail, Site.bn_supprimeSite FROM Site WHERE Site.bn_supprimeSite=0 AND Site.id_site IS NOT null AND Site.id_site != -1");
java.sql.ResultSetMetaData rsmd_tMSSqlInput_2 = rs_tMSSqlInput_2
.getMetaData();
int colQtyInRs_tMSSqlInput_2 = rsmd_tMSSqlInput_2.getColumnCount();
globalMap
.put(
"tMSSqlInput_2_QUERY",
"SELECT Site.id_site, Site.tpge_o_n, Site.tpge_site_numimmat, Site.tpge_site_numattri, Site.tpge_site_numtitre, Site.tpge_site_numprop, Site.tpge_site_numgeo, Site.tpge_site_cle,
Site.nom_site, Site.destination_site, Site.domaine_site, Site.at_pn_site, Site.site_circo, Site.site_com, Site.site_adr1, Site.site_adr2, Site.site_poste, Site.site_cp, Site.site_burdis,
Site.site_oberv, Site.statut_terrain, Site.type_prop, Site.domianialité, Site.surface_totale, Site.nombre_pers, Site.places_surf, Site.places_couv, Site.nature, Site.statut_occup,
Site.mode_entrée, Site.année_entrée, Site.année_construc, Site.montant_estimé, Site.date_estim, Site.valeur_locat, Site.date_valeur_locat, Site.type_prop_bati, Site.année_entrée_bati,
Site.maintenance, Site.ouvrage, Site.montant_loyer, Site.date_valeur_loyer, Site.IGH, Site.ERP, Site.ICPE, Site.périm_mh, Site.site_mh, Site.action_site, Site.date_appréciation, Site.id_user,
Site.date_user, Site.id_group, Site.date_echeance_bail, Site.bn_supprimeSite FROM Site WHERE Site.bn_supprimeSite=0 AND Site.id_site IS NOT null AND Site.id_site != -1");
while (rs_tMSSqlInput_2.next()) {
nb_line_tMSSqlInput_2++;
if (colQtyInRs_tMSSqlInput_2 < 1) {

```

```
row1.id_site = null;
} else {
if (rs_tMSSqlInput_2.getObject(1) != null) {
row1.id_site = rs_tMSSqlInput_2.getInt(1);
} else {
row1.id_site = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 2) {
row1.tpge_o_n = null;
} else {
if (rs_tMSSqlInput_2.getObject(2) != null) {
row1.tpge_o_n = rs_tMSSqlInput_2.getBytes(2);
} else {
row1.tpge_o_n = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 3) {
row1.tgpe_site_numimmat = null;
} else {
if (rs_tMSSqlInput_2.getObject(3) != null) {
row1.tgpe_site_numimmat = rs_tMSSqlInput_2.getString(3);
} else {
row1.tgpe_site_numimmat = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 4) {
row1.tgpe_site_numattri = null;
} else {
if (rs_tMSSqlInput_2.getObject(4) != null) {
row1.tgpe_site_numattri = rs_tMSSqlInput_2.getString(4);
} else {
row1.tgpe_site_numattri = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 5) {
row1.tgpe_site_numtitre = null;
} else {
if (rs_tMSSqlInput_2.getObject(5) != null) {
row1.tgpe_site_numtitre = rs_tMSSqlInput_2.getString(5);
} else {
row1.tgpe_site_numtitre = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 6) {
row1.tgpe_site_numprop = null;
} else {
if (rs_tMSSqlInput_2.getObject(6) != null) {
row1.tgpe_site_numprop = rs_tMSSqlInput_2.getString(6);
} else {
row1.tgpe_site_numprop = null;
}
}
}
```

```
if (colQtyInRs_tMSSqllInput_2 < 7) {
row1.tgpe_site_numgeo = null;
} else {
if (rs_tMSSqllInput_2.getObject(7) != null) {
row1.tgpe_site_numgeo = rs_tMSSqllInput_2.getString(7);
} else {
row1.tgpe_site_numgeo = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 8) {
row1.tgpe_site_cle = null;
} else {
if (rs_tMSSqllInput_2.getObject(8) != null) {
row1.tgpe_site_cle = rs_tMSSqllInput_2.getString(8);
} else {
row1.tgpe_site_cle = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 9) {
row1.nom_site = null;
} else {
if (rs_tMSSqllInput_2.getObject(9) != null) {
row1.nom_site = rs_tMSSqllInput_2.getString(9);
} else {
row1.nom_site = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 10) {
row1.destination_site = null;
} else {
if (rs_tMSSqllInput_2.getObject(10) != null) {
row1.destination_site = rs_tMSSqllInput_2.getInt(10);
} else {
row1.destination_site = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 11) {
row1.domaine_site = null;
} else {
if (rs_tMSSqllInput_2.getObject(11) != null) {
row1.domaine_site = rs_tMSSqllInput_2.getInt(11);
} else {
row1.domaine_site = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 12) {
row1.at_pn_site = null;
} else {
if (rs_tMSSqllInput_2.getObject(12) != null) {
row1.at_pn_site = rs_tMSSqllInput_2.getBytes(12);
} else {
row1.at_pn_site = null;
}
}
```

```
}
if (colQtyInRs_tMSSqllInput_2 < 13) {
row1.site_circo = null;
} else {
if (rs_tMSSqllInput_2.getObject(13) != null) {
row1.site_circo = rs_tMSSqllInput_2.getInt(13);
} else {
row1.site_circo = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 14) {
row1.site_com = null;
} else {
if (rs_tMSSqllInput_2.getObject(14) != null) {
row1.site_com = rs_tMSSqllInput_2.getInt(14);
} else {
row1.site_com = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 15) {
row1.site_adr1 = null;
} else {
if (rs_tMSSqllInput_2.getObject(15) != null) {
row1.site_adr1 = rs_tMSSqllInput_2.getString(15);
} else {
row1.site_adr1 = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 16) {
row1.site_adr2 = null;
} else {
if (rs_tMSSqllInput_2.getObject(16) != null) {
row1.site_adr2 = rs_tMSSqllInput_2.getString(16);
} else {
row1.site_adr2 = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 17) {
row1.site_poste = null;
} else {
if (rs_tMSSqllInput_2.getObject(17) != null) {
row1.site_poste = rs_tMSSqllInput_2.getInt(17);
} else {
row1.site_poste = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 18) {
row1.site_cp = null;
} else {
if (rs_tMSSqllInput_2.getObject(18) != null) {
row1.site_cp = rs_tMSSqllInput_2.getString(18);
} else {
row1.site_cp = null;
}
```

```
}  
}  
if (colQtyInRs_tMSSqllInput_2 < 19) {  
row1.site_burdis = null;  
} else {  
if (rs_tMSSqllInput_2.getObject(19) != null) {  
row1.site_burdis = rs_tMSSqllInput_2.getString(19);  
} else {  
row1.site_burdis = null;  
}  
}  
if (colQtyInRs_tMSSqllInput_2 < 20) {  
row1.site_oberv = null;  
} else {  
if (rs_tMSSqllInput_2.getObject(20) != null) {  
row1.site_oberv = rs_tMSSqllInput_2.getString(20);  
} else {  
row1.site_oberv = null;  
}  
}  
if (colQtyInRs_tMSSqllInput_2 < 21) {  
row1.statut_terrain = null;  
} else {  
if (rs_tMSSqllInput_2.getObject(21) != null) {  
row1.statut_terrain = rs_tMSSqllInput_2.getInt(21);  
} else {  
row1.statut_terrain = null;  
}  
}  
if (colQtyInRs_tMSSqllInput_2 < 22) {  
row1.type_prop = null;  
} else {  
if (rs_tMSSqllInput_2.getObject(22) != null) {  
row1.type_prop = rs_tMSSqllInput_2.getInt(22);  
} else {  
row1.type_prop = null;  
}  
}  
if (colQtyInRs_tMSSqllInput_2 < 23) {  
row1.Column22 = null;  
} else {  
if (rs_tMSSqllInput_2.getObject(23) != null) {  
row1.Column22 = rs_tMSSqllInput_2.getBytes(23);  
} else {  
row1.Column22 = null;  
}  
}  
if (colQtyInRs_tMSSqllInput_2 < 24) {  
row1.surface_totale = null;  
} else {  
if (rs_tMSSqllInput_2.getObject(24) != null) {  
row1.surface_totale = rs_tMSSqllInput_2.getInt(24);  
} else {
```

```
row1.surface_totale = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 25) {
row1.nombre_pers = null;
} else {
if (rs_tMSSqllInput_2.getObject(25) != null) {
row1.nombre_pers = rs_tMSSqllInput_2.getInt(25);
} else {
row1.nombre_pers = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 26) {
row1.places_surf = null;
} else {
if (rs_tMSSqllInput_2.getObject(26) != null) {
row1.places_surf = rs_tMSSqllInput_2.getInt(26);
} else {
row1.places_surf = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 27) {
row1.places_couv = null;
} else {
if (rs_tMSSqllInput_2.getObject(27) != null) {
row1.places_couv = rs_tMSSqllInput_2.getInt(27);
} else {
row1.places_couv = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 28) {
row1.nature = null;
} else {
if (rs_tMSSqllInput_2.getObject(28) != null) {
row1.nature = rs_tMSSqllInput_2.getBytes(28);
} else {
row1.nature = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 29) {
row1.statut_occup = null;
} else {
if (rs_tMSSqllInput_2.getObject(29) != null) {
row1.statut_occup = rs_tMSSqllInput_2.getInt(29);
} else {
row1.statut_occup = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 30) {
row1.Column29 = null;
} else {
if (rs_tMSSqllInput_2.getObject(30) != null) {
row1.Column29 = rs_tMSSqllInput_2.getInt(30);
}
```

```
} else {
row1.Column29 = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 31) {
row1.Column30 = null;
} else {
if (rs_tMSSqlInput_2.getObject(31) != null) {
row1.Column30 = rs_tMSSqlInput_2.getInt(31);
} else {
row1.Column30 = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 32) {
row1.Column31 = null;
} else {
if (rs_tMSSqlInput_2.getObject(32) != null) {
row1.Column31 = rs_tMSSqlInput_2.getInt(32);
} else {
row1.Column31 = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 33) {
row1.Column32 = null;
} else {
if (rs_tMSSqlInput_2.getObject(33) != null) {
row1.Column32 = rs_tMSSqlInput_2.getObject(33);
} else {
row1.Column32 = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 34) {
row1.date_estim = null;
} else {
if (rs_tMSSqlInput_2.getTimestamp(34) != null) {
row1.date_estim = new java.util.Date(rs_tMSSqlInput_2
.getTimestamp(34).getTime());
} else {
row1.date_estim = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 35) {
row1.valeur_locat = null;
} else {
if (rs_tMSSqlInput_2.getObject(35) != null) {
row1.valeur_locat = rs_tMSSqlInput_2.getObject(35);
} else {
row1.valeur_locat = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 36) {
row1.date_valeur_locat = null;
} else {
```

```
if (rs_tMSSqlInput_2.getTimestamp(36) != null) {
row1.date_valeur_locat = new java.util.Date(
rs_tMSSqlInput_2.getTimestamp(36).getTime());
} else {
row1.date_valeur_locat = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 37) {
row1.type_prop_bati = null;
} else {
if (rs_tMSSqlInput_2.getObject(37) != null) {
row1.type_prop_bati = rs_tMSSqlInput_2.getInt(37);
} else {
row1.type_prop_bati = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 38) {
row1.Column37 = null;
} else {
if (rs_tMSSqlInput_2.getObject(38) != null) {
row1.Column37 = rs_tMSSqlInput_2.getInt(38);
} else {
row1.Column37 = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 39) {
row1.maintenance = null;
} else {
if (rs_tMSSqlInput_2.getObject(39) != null) {
row1.maintenance = rs_tMSSqlInput_2.getBytes(39);
} else {
row1.maintenance = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 40) {
row1.ouvrage = null;
} else {
if (rs_tMSSqlInput_2.getObject(40) != null) {
row1.ouvrage = rs_tMSSqlInput_2.getBytes(40);
} else {
row1.ouvrage = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 41) {
row1.montant_loyer = null;
} else {
if (rs_tMSSqlInput_2.getObject(41) != null) {
row1.montant_loyer = rs_tMSSqlInput_2.getObject(41);
} else {
row1.montant_loyer = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 42) {
```

```
row1.date_valeur_loyer = null;
} else {
if (rs_tMSSqlInput_2.getTimestamp(42) != null) {
row1.date_valeur_loyer = new java.util.Date(
rs_tMSSqlInput_2.getTimestamp(42).getTime());
} else {
row1.date_valeur_loyer = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 43) {
row1.IGH = null;
} else {
if (rs_tMSSqlInput_2.getObject(43) != null) {
row1.IGH = rs_tMSSqlInput_2.getInt(43);
} else {
row1.IGH = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 44) {
row1.ERP = null;
} else {
if (rs_tMSSqlInput_2.getObject(44) != null) {
row1.ERP = rs_tMSSqlInput_2.getInt(44);
} else {
row1.ERP = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 45) {
row1.ICPE = null;
} else {
if (rs_tMSSqlInput_2.getObject(45) != null) {
row1.ICPE = rs_tMSSqlInput_2.getInt(45);
} else {
row1.ICPE = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 46) {
row1.Column45 = null;
} else {
if (rs_tMSSqlInput_2.getObject(46) != null) {
row1.Column45 = rs_tMSSqlInput_2.getInt(46);
} else {
row1.Column45 = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 47) {
row1.site_mh = null;
} else {
if (rs_tMSSqlInput_2.getObject(47) != null) {
row1.site_mh = rs_tMSSqlInput_2.getInt(47);
} else {
row1.site_mh = null;
}
}
```

```
}
if (colQtyInRs_tMSSqllInput_2 < 48) {
row1.action_site = null;
} else {
if (rs_tMSSqllInput_2.getObject(48) != null) {
row1.action_site = rs_tMSSqllInput_2.getInt(48);
} else {
row1.action_site = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 49) {
row1.Column48 = null;
} else {
if (rs_tMSSqllInput_2.getTimestamp(49) != null) {
row1.Column48 = new java.util.Date(rs_tMSSqllInput_2
.getTimestamp(49).getTime());
} else {
row1.Column48 = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 50) {
row1.id_user = null;
} else {
if (rs_tMSSqllInput_2.getObject(50) != null) {
row1.id_user = rs_tMSSqllInput_2.getInt(50);
} else {
row1.id_user = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 51) {
row1.date_user = null;
} else {
if (rs_tMSSqllInput_2.getTimestamp(51) != null) {
row1.date_user = new java.util.Date(rs_tMSSqllInput_2
.getTimestamp(51).getTime());
} else {
row1.date_user = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 52) {
row1.id_group = null;
} else {
if (rs_tMSSqllInput_2.getObject(52) != null) {
row1.id_group = rs_tMSSqllInput_2.getInt(52);
} else {
row1.id_group = null;
}
}
if (colQtyInRs_tMSSqllInput_2 < 53) {
row1.date_echeance_bail = null;
} else {
if (rs_tMSSqllInput_2.getTimestamp(53) != null) {
row1.date_echeance_bail = new java.util.Date(
```

```

rs_tMSSqlInput_2.getTimestamp(53).getTime());
} else {
row1.date_echeance_bail = null;
}
}
if (colQtyInRs_tMSSqlInput_2 < 54) {
row1.bn_supprimeSite = null;
} else {
if (rs_tMSSqlInput_2.getObject(54) != null) {
row1.bn_supprimeSite = rs_tMSSqlInput_2.getString(54);
} else {
row1.bn_supprimeSite = null;
}
}
/**
 * stop
 */
/**
 * start
 */
currentComponent = "tMSSqlInput_2";
/**
 * stop
 */
/**
 * start
 */
currentComponent = "tFilterRow_1";
row2 = null;
if (
row1.id_site != -1
)
{
if (row2 == null) {
row2 = new row2Struct();
}
row2.id_site = row1.id_site;
row2.tpge_o_n = row1.tpge_o_n;
row2.tgpe_site_numimmat = row1.tgpe_site_numimmat;
row2.tgpe_site_numattri = row1.tgpe_site_numattri;
row2.tgpe_site_numtitre = row1.tgpe_site_numtitre;
row2.tgpe_site_numprop = row1.tgpe_site_numprop;
row2.tgpe_site_numgeo = row1.tgpe_site_numgeo;
row2.tgpe_site_cle = row1.tgpe_site_cle;
row2.nom_site = row1.nom_site;
row2.destination_site = row1.destination_site;
row2.domaine_site = row1.domaine_site;
row2.at_pn_site = row1.at_pn_site;
row2.site_circo = row1.site_circo;
row2.site_com = row1.site_com;
row2.site_adr1 = row1.site_adr1;
row2.site_adr2 = row1.site_adr2;
row2.site_poste = row1.site_poste;

```

```

row2.site_cp = row1.site_cp;
row2.site_burdis = row1.site_burdis;
row2.site_oberv = row1.site_oberv;
row2.statut_terrain = row1.statut_terrain;
row2.type_prop = row1.type_prop;
row2.Column22 = row1.Column22;
row2.surface_totale = row1.surface_totale;
row2.nombre_pers = row1.nombre_pers;
row2.places_surf = row1.places_surf;
row2.places_couv = row1.places_couv;
row2.nature = row1.nature;
row2.statut_occup = row1.statut_occup;
row2.Column29 = row1.Column29;
row2.Column30 = row1.Column30;
row2.Column31 = row1.Column31;
row2.Column32 = row1.Column32;
row2.date_estim = row1.date_estim;
row2.valeur_locat = row1.valeur_locat;
row2.date_valeur_locat = row1.date_valeur_locat;
row2.type_prop_bati = row1.type_prop_bati;
row2.Column37 = row1.Column37;
row2.maintenance = row1.maintenance;
row2.ouvrage = row1.ouvrage;
row2.montant_loyer = row1.montant_loyer;
row2.date_valeur_loyer = row1.date_valeur_loyer;
row2.IGH = row1.IGH;
row2.ERP = row1.ERP;
row2.ICPE = row1.ICPE;
row2.Column45 = row1.Column45;
row2.site_mh = row1.site_mh;
row2.action_site = row1.action_site;
row2.Column48 = row1.Column48;
row2.id_user = row1.id_user;
row2.date_user = row1.date_user;
row2.id_group = row1.id_group;
row2.date_echeance_bail = row1.date_echeance_bail;
row2.bn_supprimeSite = row1.bn_supprimeSite;
nb_line_ok_tFilterRow_1++;
} else {
}
nb_line_tFilterRow_1++;
/**
 * stop
 */
// Start of branch "row2"
if (row2 != null) {
/**
 * start
 */
currentComponent = "tJavaRow_1";
if (row2.Column45 == 1) {
row3.Column45 = 1;
} else if (row2.Column45 == 2) {

```

```

row3.Column45 = 0;
} else if (row2.Column45 == 3) {
row3.Column45 = 0;
} else {
row3.Column45 = null;
}
if (row2.site_circo == 0) {
row3.site_circo = null;
}
nb_line_tJavaRow_1++;
/**
 * stop
 */
/**
 * start
 */
currentComponent = "tMap_1";
boolean rejectedInnerJoin_tMap_1 = false;
// #####
// # Input tables (lookups)
// #####
{ // start of Var scope
// #####
// # Vars tables
// #####
// #####
// # Output tables
row4 = null;
// # Output table : 'row4'
// # Filter conditions
if (
row3.id_site != null
) {
row4_tmp.site_id = row3.id_site;
row4_tmp.doma_id = row3.domaine_site;
row4_tmp.util_id = null;
row4_tmp.com_id = null;
row4_tmp.tprop_id = null;
row4_tmp.tsite_id = null;
row4_tmp.icpe_id = null;
row4_tmp.erp_id = null;
row4_tmp.circo_id = null;
row4_tmp.sterr_id = null;
row4_tmp.site_lib = null;
row4_tmp.site_adr1 = null;
row4_tmp.site_adr2 = null;
row4_tmp.site_cp = null;
row4_tmp.site_obser = null;
row4_tmp.site_dom = null;
row4_tmp.site_surf = null;
row4_tmp.site_nbpa = null;
row4_tmp.site_pkex = null;
row4_tmp.site_pkco = null;

```

```
row4_tmp.site_bati = null;
row4_tmp.site_perim_mh = null;
row4_tmp.date_modifs = null;
row4 = row4_tmp;
} // closing filter/reject
// #####
} // end of Var scope
rejectedInnerJoin_tMap_1 = false;
/**
 * stop
 */
// Start of branch "row4"
if (row4 != null) {
/**
 * start
 */
currentComponent = "tPostgresqlOutput_1";
whetherReject_tPostgresqlOutput_1 = false;
if (row4.site_id == null) {
pstmt_tPostgresqlOutput_1.setNull(1,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(1, row4.site_id);
}
if (row4.doma_id == null) {
pstmt_tPostgresqlOutput_1.setNull(2,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(2, row4.doma_id);
}
if (row4.util_id == null) {
pstmt_tPostgresqlOutput_1.setNull(3,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(3, row4.util_id);
}
if (row4.com_id == null) {
pstmt_tPostgresqlOutput_1.setNull(4,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(4, row4.com_id);
}
if (row4.tprop_id == null) {
pstmt_tPostgresqlOutput_1.setNull(5,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(5, row4.tprop_id);
}
if (row4.tsite_id == null) {
pstmt_tPostgresqlOutput_1.setNull(6,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(6, row4.tsite_id);
}
```

```
}
if (row4.icpe_id == null) {
pstmt_tPostgresqlOutput_1.setNull(7,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(7, row4.icpe_id);
}
if (row4.erp_id == null) {
pstmt_tPostgresqlOutput_1.setNull(8,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(8, row4.erp_id);
}
if (row4.circo_id == null) {
pstmt_tPostgresqlOutput_1.setNull(9,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(9, row4.circo_id);
}
if (row4.sterr_id == null) {
pstmt_tPostgresqlOutput_1.setNull(10,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(10, row4.sterr_id);
}
if (row4.site_lib == null) {
pstmt_tPostgresqlOutput_1.setNull(11,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(11,
row4.site_lib);
}
if (row4.site_adr1 == null) {
pstmt_tPostgresqlOutput_1.setNull(12,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(12,
row4.site_adr1);
}
if (row4.site_adr2 == null) {
pstmt_tPostgresqlOutput_1.setNull(13,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(13,
row4.site_adr2);
}
if (row4.site_cp == null) {
pstmt_tPostgresqlOutput_1.setNull(14,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(14,
row4.site_cp);
}
}
```

```
if (row4.site_obser == null) {
pstmt_tPostgresqlOutput_1.setNull(15,
java.sql.Types.VARCHAR);
} else {
pstmt_tPostgresqlOutput_1.setString(15,
row4.site_obser);
}
if (row4.site_dom == null) {
pstmt_tPostgresqlOutput_1.setNull(16,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(16, row4.site_dom);
}
if (row4.site_surf == null) {
pstmt_tPostgresqlOutput_1.setNull(17,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1
.setInt(17, row4.site_surf);
}
if (row4.site_nbpa == null) {
pstmt_tPostgresqlOutput_1.setNull(18,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1
.setInt(18, row4.site_nbpa);
}
if (row4.site_pkex == null) {
pstmt_tPostgresqlOutput_1.setNull(19,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1
.setInt(19, row4.site_pkex);
}
if (row4.site_pkco == null) {
pstmt_tPostgresqlOutput_1.setNull(20,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1
.setInt(20, row4.site_pkco);
}
if (row4.site_bati == null) {
pstmt_tPostgresqlOutput_1.setNull(21,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1
.setInt(21, row4.site_bati);
}
if (row4.site_perim_mh == null) {
pstmt_tPostgresqlOutput_1.setNull(22,
java.sql.Types.INTEGER);
} else {
pstmt_tPostgresqlOutput_1.setInt(22,
```

```

row4.site_perim_mh);
}
if (row4.date_modifs != null) {
pstmt_tPostgresqlOutput_1.setTimestamp(23,
new java.sql.Timestamp(row4.date_modifs
.getTime()));
} else {
pstmt_tPostgresqlOutput_1.setNull(23,
java.sql.Types.DATE);
}
try {
insertedCount_tPostgresqlOutput_1 = insertedCount_tPostgresqlOutput_1
+ pstmt_tPostgresqlOutput_1.executeUpdate();
nb_line_tPostgresqlOutput_1++;
} catch (Exception e) {
whetherReject_tPostgresqlOutput_1 = true;
System.err.print(e.getMessage());
}
commitCounter_tPostgresqlOutput_1++;
if (commitEvery_tPostgresqlOutput_1 <= commitCounter_tPostgresqlOutput_1) {
conn_tPostgresqlOutput_1.commit();
commitCounter_tPostgresqlOutput_1 = 0;
}
/**
* stop
*/
} // End of branch "row4"
} // End of branch "row2"
/**
* start
*/
currentComponent = "tMSSqlInput_2";
}
stmt_tMSSqlInput_2.close();
conn_tMSSqlInput_2.close();
globalMap.put("tMSSqlInput_2_NB_LINE", nb_line_tMSSqlInput_2);
ok_Hash.put("tMSSqlInput_2", true);
end_Hash.put("tMSSqlInput_2", System.currentTimeMillis());
/**
* stop
*/
/**
* start
*/
currentComponent = "tFilterRow_1";
globalMap.put("tFilterRow_1_NB_LINE", nb_line_tFilterRow_1);
globalMap.put("tFilterRow_1_NB_LINE_OK", nb_line_ok_tFilterRow_1);
ok_Hash.put("tFilterRow_1", true);
end_Hash.put("tFilterRow_1", System.currentTimeMillis());
/**
* stop
*/
/**

```

```

* start
*/
currentComponent = "tJavaRow_1";
globalMap.put("tJavaRow_1_NB_LINE", nb_line_tJavaRow_1);
ok_Hash.put("tJavaRow_1", true);
end_Hash.put("tJavaRow_1", System.currentTimeMillis());
/**
* stop
*/
/**
* start
*/
currentComponent = "tMap_1";
// #####
// # Lookup hashes releasing
ok_Hash.put("tMap_1", true);
end_Hash.put("tMap_1", System.currentTimeMillis());
/**
* stop
*/
/**
* start
*/
currentComponent = "tPostgresqlOutput_1";
if (pstmt_tPostgresqlOutput_1 != null) {
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);
}
}
}
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 = runJobInTOS(args);
globalBuffer.clear();
System.exit(exitCode);
}
public static String[][] runJob(String[] args) {
int exitCode = runJobInTOS(args);
String[][] bufferValue = new String[][] { { Integer.toString(exitCode) } };
globalBuffer.clear();
return bufferValue;
}
public static synchronized int runJobInTOS(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 = MigreSite.class.getClassLoader()
.getResourceAsStream(
"gespat/migresite/contexts/Default.properties");
if (inContext != null) {
defaultProps.load(inContext);
inContext.close();
context = new ContextProperties(defaultProps);
}
}

```

```

if (contextStr.compareTo("Default") != 0) {
inContext = MigreSite.class.getClassLoader()
.getResourceAsStream(
"gespat/migresite/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 startUsedMemory = Runtime.getRuntime().totalMemory()
- Runtime.getRuntime().freeMemory();
long endUsedMemory = 0;
long end = 0;
long start = System.currentTimeMillis();
final MigreSite MigreSiteClass = new MigreSite();
try {
errorCode = null;
MigreSiteClass.tMSSqlInput_2Process();
status = "end";
} catch (TalendException e_tMSSqlInput_2) {
status = "failure";
e_tMSSqlInput_2.printStackTrace();
} finally {
}
end = System.currentTimeMillis();
if (watch) {
System.out.println((end - start) + " milliseconds");
}
endUsedMemory = Runtime.getRuntime().totalMemory()
- Runtime.getRuntime().freeMemory();
if (false) {
System.out.println((endUsedMemory - startUsedMemory)
+ " bytes memory increase when running : MigreSite");
}
reset();
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() {
errorCode = null;
status = "";
}
private static void reset() {
defaultProps.clear();
context.clear();
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();
System.gc();
}
}
}
/*****
* 54189 characters generated by Talend OpenStudio on the 13 février 2008
* 11:11:10 CET
*****/

```