PILOTER VOS JOBS SAS® DATA MANAGEMENT DEPUIS SAS BASE

Le contexte économique actuel associé à l'explosion des volumes de données donne aujourd'hui aux entreprises un défi important à relever. Dans ce contexte Big Data, **SAS Data Management** aide les entreprises à améliorer et piloter les données sur lesquelles elles s'appuient pour prendre leurs décisions stratégiques.

Caractéristiques :
Catégories : Data Management
OS : Windows
Version : 9.4
Vérifié en Mars 2014

La technologie **SAS** de gestion de la qualité des données vous permet ainsi de créer des « jobs » qui normalisent, valident et intègrent les données à chaque étape de leur cycle de vie.

Une fois ces jobs créés, il est intéressant de pouvoir les utiliser dans vos programmes **SAS.**

Cet article présente les fonctions **SAS** à utiliser ainsi que les clés pour vous aider à piloter vos jobs SAS Data management.

Table des matières

Nouveauté SAS 9.4

SAS a maintenant entièrement intégré la suite DataFlux. Cela permettra de mettre en œuvre une stratégie de gestion de l'information mieux intégrée, allant au-delà de la gestion des données et de la gouvernance, en s'appuyant sur l'analytique pour une meilleure prise de décision.

Certains produits DataFlux sont en train de changer de noms et d'autres le seront prochainement. Pour plus d'information, n'hésitez pas à consulter les pages :

http://support.sas.com/software/products/dataflux/

http://support.sas.com/software/products/entdis/index.html

Les modifications des offres pourraient affecter vos renouvellements de licences.

Pré-requis

Pour pouvoir suivre le scenario de cet article, il est nécessaire de posséder SAS® Data Management Advanced.

SAS Data Management Advanced propose une palette d'outils complète pour la qualité et la gouvernance des données, conçue pour répondre aux besoins d'intégration de données d'entreprise. Les produits compris dans cette solution sont :

Base SAS®

SAS® Metadata Server, qui fournit un référentiel ouvert et centralisé pour le stockage et la gestion de métadonnées d'entreprise

DataFlux® Data Management Server

DataFlux® Data Management Studio

SAS® Data Integration Server, qui assure l'extraction, le nettoyage, la transformation, la mise en conformité, l'agrégation, le chargement et la gestion des données. Permet également la réutilisation des processus développés par d'autres équipes.

Pour le scenario de cet article, nous avons également utilisé :

SAS/ACCESS® to Mysql

Scénario

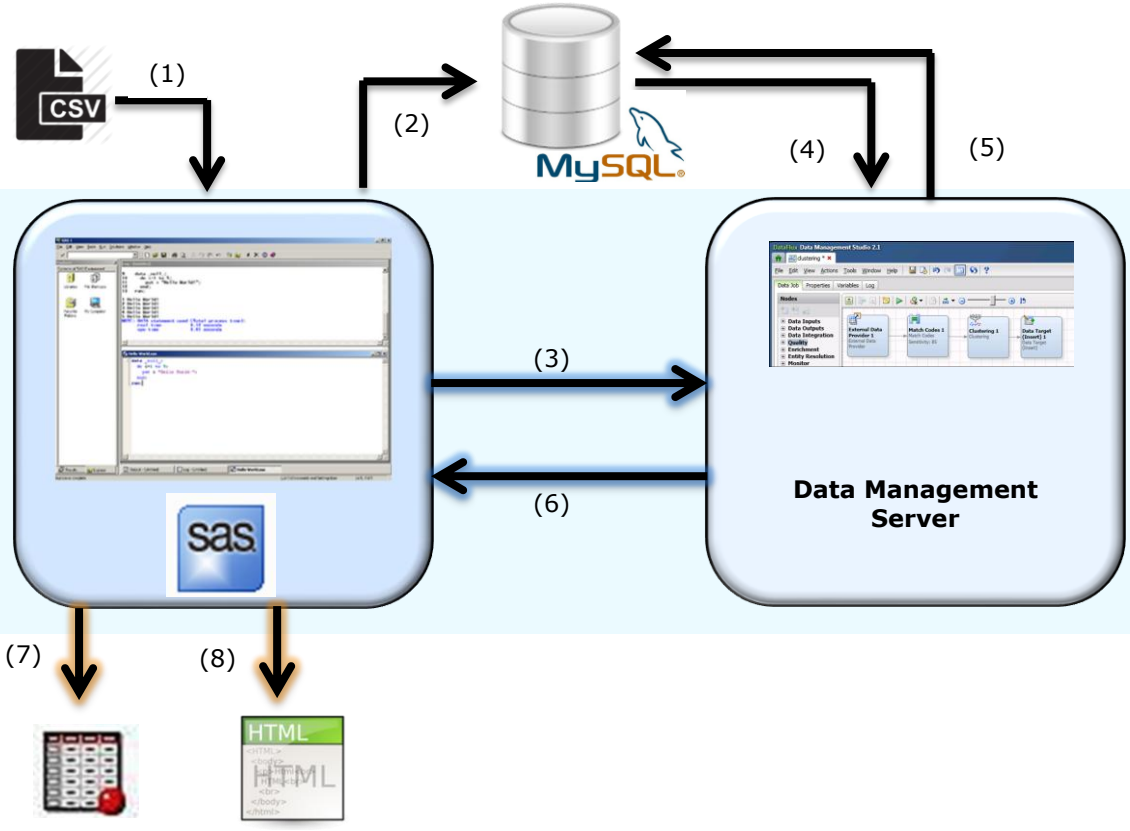
Dans cet article, nous allons utiliser un exemple de projet afin de faciliter la compréhension de l'interaction entre **SAS®** et **les jobs SAS Data Management**. L'objectif est de rapprocher deux tables d'une base de données afin de voir si les données d'une des tables correspondent aux données de la seconde. Ainsi, imaginons qu'une table contienne les clients d'une entreprise, et la seconde table une liste de clients à surveiller (par exemple, des clients inscrits sur une liste noire). L'opération consiste donc à rapprocher les deux tables afin de vérifier si un client de l'entreprise n'est pas inscrit dans cette liste noire.

Techniquement, nous avons un Datajob développé sous **DataFlux Data Management Studio.** Ce Datajob est exécuté et déployé sous DataFlux Data Management Server. **Ce Datajob est, en quelque sorte, l'intelligence de notre projet**. Celui-ci lit les données d'une table Mysql et crée des matchcodes pour chaque nom et prénom de la table Mysql. Un matchcode est un code alphanumérique identifiant le nom et l'adresse complète d'un client. Notons également que la taille de ces matchcodes n'est pas forcément identique selon les locales, les data types, les définitions… Il permet de comparer, trier, regrouper ou dédoublonner des fichiers. Le choix du matchcode est déterminant pour ne pas omettre de rapprochement. Enfin, les matchcodes sont enregistrés dans une seconde table Mysql.

Le but est donc de créer un programme **SAS®** qui va :

1/ Enrichir la table Mysql utilisée en entrée du Datajob **DataFlux®**
2/ Exécuter le DataJob sur **SAS Management Server**
3/ Afficher les résultats sous forme de fichier html

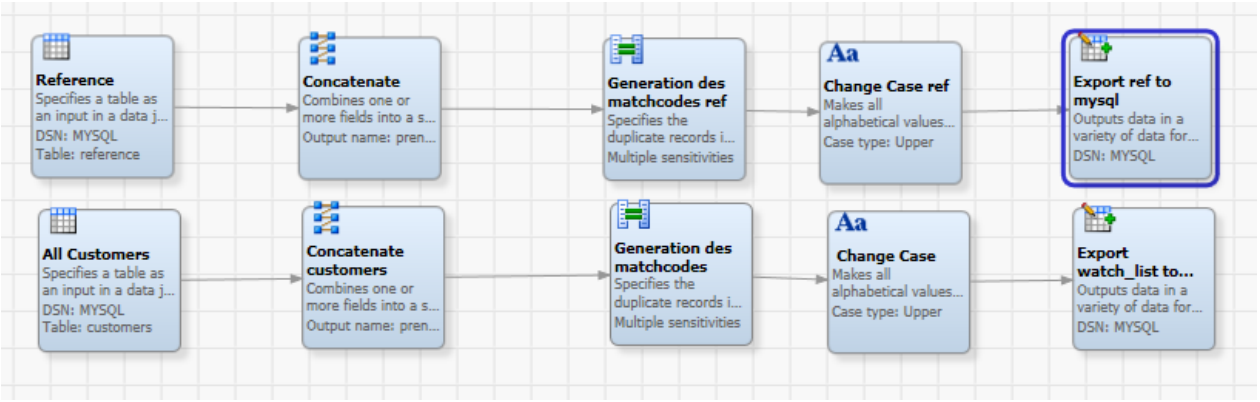
Le schéma ci-dessous présente les différentes étapes de notre scenario :



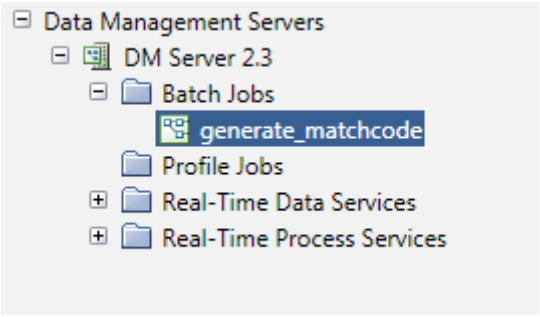
1. SAS Base : Lecture du fichier csv,
2. SAS/ACCESS to Mysql : Import des données dans Mysql,
3. Déclenchement du job,
4. DATAFLUX MANAGEMENT SERVER : Exécution du DataJob (création des matchcodes),
5. DATAFLUX MANAGEMENT SERVER : Export des résultats dans la base Mysql (via ODBC),
6. Envoi du code retour du Datajob à SAS,
7. SAS Base : Création d'une table,
8. SAS Base : Création d'une page html affichant les résultats.

Présentation du job Dataflux

Nous partons du principe que la conception du job **DataFlux®** ait déjà été faite et que celui-ci fonctionne. Ce job utilisé pour notre scenario a pour objectif la création d'un code matchcode pour un couple nom/prénom.



Le job est déployé sur le serveur (c'est-à-dire qu'il peut être exécuté par celui-ci) :



Le programme SAS en détail

Etape 1 et 2 : Préparation des données

Examinons maintenant le code SAS en détail :

La première étape consiste à créer une « connexion » à la base Mysql. Pour cela, nous allons créer un libname en utilisant le module SAS/ACCESS to Mysql.

```
libname myData mysql user=root database=sas server=localhost port=3306;
```

Une fois le libname créé, nous pouvons purger la table customers. Cette table va contenir le contenu du fichier csv.

```
proc delete data = myData.customers;
run;
```

Nous pouvons importer le fichier csv dans la table customers de notre base de données Mysql.

```
filename import "customers.csv" ;

data myData.customers;
      infile import delimiter=";";
      input    nom $ prenom $;
run;
```

Nous créons ensuite une référence qui servira de comparaison avec la table customers :

```
data myData.reference;
      length nom prenom $50.;
      input nom $ prenom $;
      datalines;
      Housset Nicolas
;
run;
```

Etape 3 à 6 : Interaction SAS et Dataflux

Nous arrivons maintenant dans le cœur du programme.
Nous allons en effet utiliser la fonction *dmsrvbatchjob* afin d'exécuter le job via **DataFlux® Data Management Server.**

```
data _null_;
LENGTH jobid $52;

      jobid = dmsrvbatchjob ('generate_matchcode.djf', 'localhost', 21036);

      jobrc = dmsrvjobstatus (jobid, 'localhost', 21036);

      copyrc = dmsrvCopyLog (jobid, 'localhost', 21036,
'C:\temp\Dataflux\logs\my.log');

      put "jobid = " jobid;
      put "jobrc = " jobrc;

run;
```

Dans notre programme, nous utilisons trois fonctions :

Dmsrvbatchjob	Cette fonction va déclencher l'exécution du job passé en paramètre sur le serveur. Elle retourne le jobid créé par Dataflux Data Management Server.
Dmsrvjobstatus	Cette fonction permet de récupérer le code retour du jobid passé en paramètre.
DmsrvCopyLog	Cette fonction permet de rapatrier la log d'exécution du job.

Ces trois fonctions prennent le nom du serveur et le port du serveur en paramètres. Dans notre exemple, il s'agit de *localhost* et de *21036*, *21036* étant le port par défaut de Data Management Server.

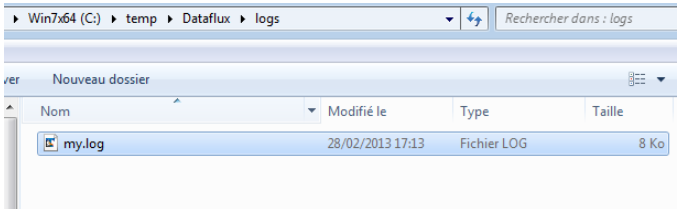
Vérification de l'exécution du job

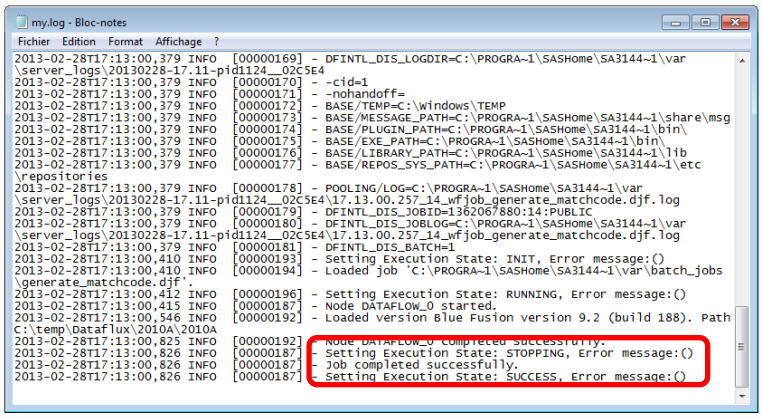
Il est possible de vérifier l'exécution du job de plusieurs manières :

- Dans le journal SAS, en utilisant la macro variable jobrc :

```
jobid = 1362059267:1150:PUBLIC
jobrc = 0
NOTE: L'étape DATA used (Total process time):
      real time           1.06 secondes
      cpu time            0.06 secondes
```

- En examinant le fichier de log rapatrié via la fonction dmsrvCopyLog :





- Directement dans Dataflux Data Management Studio :

  Data management Servers > Batch Jobs > Run history :

Etape 7 et 8 : Génération du rapport html

Afin de compléter notre scenario, nous créons deux tables de rapprochement basées sur les résultats de l'exécution du job par Dataflux.

```
proc sql;
create table Rappro50 as
      select      a.prenom_nom  as nomA,
                  a.prenom_nom_MatchCode_50 as prenom_nom_MatchCodeA,
                  b.prenom_nom as nomB,
                  b.prenom_nom_MatchCode_50 as prenom_nom_MatchCodeB
      from myData.reference_match_code a
      inner join myData.watch_list b
            on (b.prenom_nom_MatchCode_50=a.prenom_nom_MatchCode_50)
;quit;

proc sql;
create table Rappro as
      select      a.prenom_nom  as nomA,
                  a.prenom_nom_MatchCode as prenom_nom_MatchCodeA,
                  b.prenom_nom as nomB,
                  b.prenom_nom_MatchCode as prenom_nom_MatchCodeB
      from myData.reference_match_code a
      inner join myData.watch_list b
            on (b.prenom_nom_MatchCode=a.prenom_nom_MatchCode)
;quit;
```

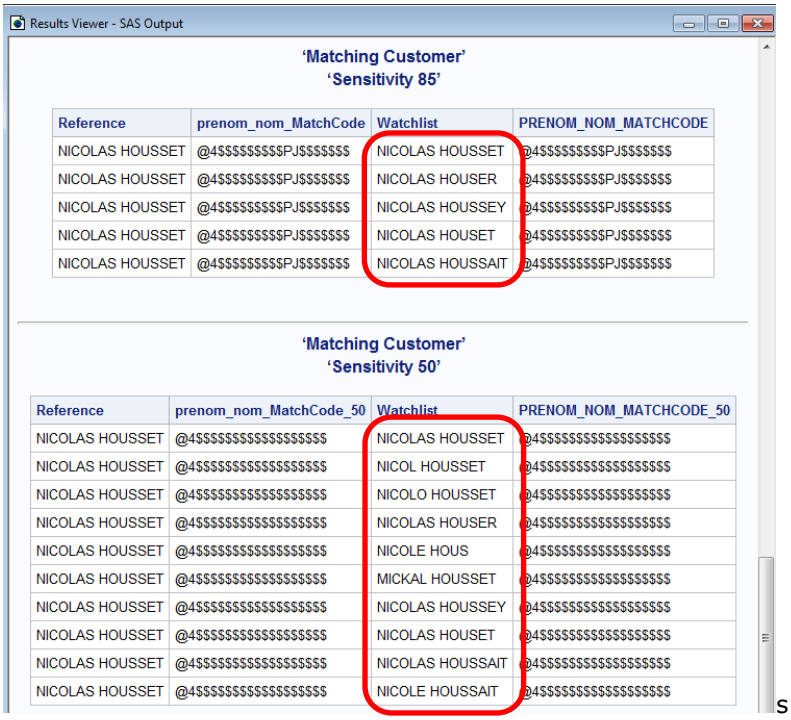
Nous pouvons maintenant créer deux fichiers HTML :

```
title1 'Matching Customer';
title2 'Sensitivity 85';
proc print data=Rappro label noobs;
label nomA ="Reference" nomB ="Watchlist";
run;

title1 'Matching Customer';
title2 'Sensitivity 50';
proc print data=Rappro50 label noobs;
label nomA ="Reference" nomB ="Watchlist";
run;
```

Nous obtenons ainsi deux listes. Dans la première, avec une sensibilité de 85, nous constatons que « NICOLAS HOUSSET » ressemble à « NICOLAS HOUSER », « NICOLAS HOUSSEY », « NICOLAS HOUSET» et «NICOLAS HOUSSAIT»

Dans la deuxième liste, avec une sensibilité de 50, la liste est plus importante.


Results Viewer - SAS Output

'Matching Customer'
'Sensitivity 85'

Reference	prenom_nom_MatchCode	Watchlist	PRENOM_NOM_MATCHCODE
NICOLAS HOUSSET	@4$$$$$$$$$PJ$$$$$$$	NICOLAS HOUSSET	@4$$$$$$$$$PJ$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$PJ$$$$$$$	NICOLAS HOUSER	@4$$$$$$$$$PJ$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$PJ$$$$$$$	NICOLAS HOUSSEY	@4$$$$$$$$$PJ$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$PJ$$$$$$$	NICOLAS HOUSET	@4$$$$$$$$$PJ$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$PJ$$$$$$$	NICOLAS HOUSSAIT	@4$$$$$$$$$PJ$$$$$$$

'Matching Customer'
'Sensitivity 50'

Reference	prenom_nom_MatchCode_50	Watchlist	PRENOM_NOM_MATCHCODE_50
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOL HOUSSET	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLO HOUSSET	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLAS HOUSER	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLE HOUS	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	MICKAL HOUSSET	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLAS HOUSSEY	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLAS HOUSET	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLAS HOUSSAIT	@4$$$$$$$$$$$$$$$$$$
NICOLAS HOUSSET	@4$$$$$$$$$$$$$$$$$$	NICOLE HOUSSAIT	@4$$$$$$$$$$$$$$$$$$

s

En cas de problème

Si vous rencontrez des problèmes lors de l'utilisation des fonctions SAS présentées dans cet article, vous pouvez nous écrire à support@sas.com, en attachant à votre message un document explicitant votre problème, ce dernier accompagné, si possible, des jobs Dataflux, d'un jeu de données permettant de reproduire le comportement et les fichiers de logs retournés par la fonction dmsrvCopyLog.

```
ERROR: Unrecognized ACJ error 28.
```

Cette erreur signifie que le serveur Data Management Server auquel vous tentez de vous connecter n'est pas démarré.

Cette erreur peut également survenir si les informations de connexion, passées en argument de la

fonction *dmsrvbatchjob* ne sont pas correctes.

ERROR: Architect job generate_matchcode.djf failed (Job Not Found,).

Soit le job n'est pas déployé sur le serveur.
Soit le nom du job n'est pas correct dans la fonction *dmsrvbatchjob*.

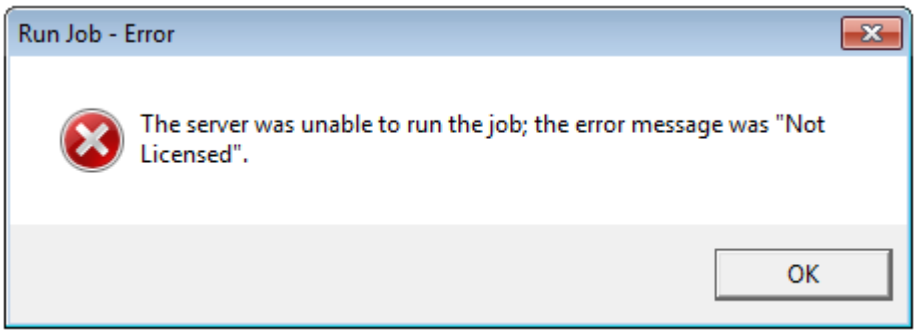
Si la syntaxe est correcte :

- Vérifiez, via Data Management Studio, si le job est correctement déployé sur le serveur,
- Vérifiez si le fichier du job est bien présent physiquement sur le serveur.

ERROR: Architect job generate_matchcode.djf failed (Not Licensed,)

Ce message indique que la licence sur le serveur n'est pas valide.

Si vous essayez de lancer votre job directement sur le serveur depuis **Data Management Studio**, vous obtiendrez le message suivant :



Le message suivant : `*** SERVER IS NOT LICENSED AND WILL NOT EXECUTE JOBS OR SERVICES ***` devrait également être présent dans la log du serveur.

Conclusion

Comme nous avons pu le constater, la mise en place d'un pilotage des Datajobs développé sous Dataflux Data Management Studio par SAS ® est simple à mettre en œuvre.

A travers cet exemple simple, nous avons pu utiliser la puissance de **SAS** sans avoir à développer de nouveau programme spécifique. Cela promet des perspectives intéressantes.

Nicolas Housset
Consultant Support Clients SAS France