



> Numéro 13

> ALLO SUPPORT

Les vues dictionnaires de la SASHELP

Ce sont des vues systèmes permettant de récupérer les informations sur la session SAS en cours.

A partir de ces vues, il est possible, par exemple, de connaître la liste des tables d'une bibliothèque, de récupérer la liste des variables numériques d'une table ou bien de savoir si un index est présent sur une variable.

Les informations contenues dans ces vues sont constamment mises à jour au cours d'une session SAS.

Une identification rapide de ces vues est aisée dans la mesure où elles sont toutes préfixées par la lettre V.

Il existe 2 façons d'utiliser ces vues :

- dans une étape DATA en utilisant le nom de la vue située dans la SASHELP (en gras dans les tableaux ci-dessous),
- dans une PROC SQL en utilisant soit le nom de la vue située dans la SASHELP, soit le nom SQL (en italique dans les tableaux ci-dessous).

Les vues disponibles

> Bibliothèque et fichier externes :

	VSLIB	➔	bibliothèques allouées
	VEXTFL <i>dictionary.extfiles</i>	➔	fichiers externes

> Tables, vues et catalogues :

	VMEMBER <i>dictionary.members</i>	➔	tables, vues et catalogues
	VSTABVW	➔	tables et vues
	VSTABLE	➔	tables
	VTABLE <i>dictionary tables</i>	➔	détail des tables
	VSVIEW	➔	vues
	VVVIEW <i>dictionary .views</i>	➔	détail des vues
	VSACCESS	➔	vues SAS Access
	VSCATLG	➔	catalogues
	VCATALOG <i>dictionary .views</i>	➔	détail des catalogues et de leurs entrées

> Dans ce Numéro

- > Les vues dictionnaires de la SASHELP > 1-2
- > Optimisez vos étapes DATA avec les Array > 3
- > Naviguez depuis vos cartes SAS/GRAPH® > 4-5
- > Comment récupérer des données SAS MVS dans Excel > 6
- > Java Server Pages (JSP) : particularités avec les objets graphiques lors d'un déploiement sous Unix > 7
- > Comment exporter une structure HOLAP en programmation SCL ? > 8-9
- > SAS® Enterprise Miner™ : deux façons d'utiliser le nœud Score pour calculer des prédictions > 10-11

> Ftecnews-l

Ftecnews-l, la lettre électronique de l'équipe du Support Clients SAS, permet de diffuser rapidement toute nouvelle information technique vers les utilisateurs des solutions SAS ou/et aux clients francophones.

Avec ce mode de diffusion privilégié, vous serez informés en avant-première de la disponibilité des nouvelles versions, des correctifs logiciels et des nouvelles documentations techniques (TS).

L'abonnement est simple et gratuit : il vous suffit d'envoyer un message électronique à listserv@vm.sas.com en précisant dans le texte du message l'instruction `subscribe ftecnews-l` suivie de votre prénom et de votre nom.

Ex : `subscribe ftecnews-l Jean Dupond.`


Cette liste de distribution vient en complément des listes déjà existantes sur notre site Internet (TSNEWS-L, SNOTES-L).

Si vous souhaitez recevoir les éditions futures uniquement via ce support, merci d'informer Fabienne Vermast : frafav@fra.sas.com.

The Power to Know™



> Variables et index :

Variable	VCOLUMN → <i>dictionary.columns</i>	détail des variables
	VINDEX → <i>dictionary.indexes</i>	index disponibles
¯ovar	VMACRO → <i>dictionary.macros</i>	macros variables

> Titres, pieds de page et options SAS :

	VTITLE → <i>dictionary.titles</i>	titres et pieds de page
	VOPTIONS → <i>dictionary.options</i>	options SAS

Description d'une vue

Pour obtenir la description d'une vue, il faut utiliser l'instruction DESCRIBE de la PROC SQL.

Le code suivant affiche le nom des variables composant la vue dictionary.indexes :

```
proc sql;
describe table dictionary.indexes;
quit;
```

Journal :

```
NOTE: SQL table DICTIONARY.INDEXES was
created like:
create table DICTIONARY.INDEXES
(
libname char(8) label='Library Name',
memname char(32) label='Member Name',
memtype char(8) label='Member Type',
name char(32) label='Column Name'
```

Utilisation des Macros variables

Afin de stocker les résultats d'une requête dans une Macro variable, il faut utiliser l'instruction INTO: dans la clause SELECT de la requête.

Le code suivant stocke le nom des variables de la table SASHELP.CLASS dans la Macro variable MESVAR en séparant chaque nom par un point virgule :

```
proc sql noprint;
select distinct(name)
into: MESVAR separated by ";"
from dictionary.columns
where libname="SASHELP"
and
memname="CLASS";
quit;
```

Journal :

```
9 %put "&mesvar";
"Age;Height;Name;Sex;Weight"
```

Références :

- Aide en ligne (recherche par index : dictionary tables)
- An Introduction to SASHELP Dictionary Views : <http://www2.sas.com/proceedings/sugi26/p017-26.pdf>
- The SASHELP Library: It Really Does Help You Manage Data: <http://support.sas.com/sassamples/bitsandbytes/sashelp.html>
<http://support.sas.com/sassamples/bitsandbytes/sashelp2.html>

Julien Fages
Ingénieur Consultant





Optimisez vos étapes DATA avec les Array

L'Array, ou tableau, est un choix judicieux pour simplifier les traitements au sein d'une étape DATA.

Présentation

Un Array est l'équivalent d'un tableau virtuel, dont chaque case correspond au contenu d'une variable sur une observation donnée. C'est un ensemble de variables, ordonnées dans une structure logique.

Il permet d'alléger un programme, notamment en résumant en quelques instructions un traitement identique exécuté sur plusieurs variables.

Ce qu'il faut savoir pour créer un array

- Un Array référence des variables de même type.
Array Identite [3] \$30. Nom Prenom CSP ;
Array Infos [3] 8. Age Taille Poids ;
- Sa structure est simple (exemples ci-dessus) ou multi-dimensionnelle.

Array Score{5,3} score1-score15 ;

Si elle est simple, alors le signe * peut être utilisé pour indiquer le nombre d'éléments de l'Array. Il est alors calculé en fonction du nombre de variables listé. Les syntaxes suivantes sont équivalentes :

Array Notes {3} 8. Note1-Note3 ;

Array Notes {*} 8. Note1-Note3 ;

Il référence des variables :

- existantes :

via une liste de variables (comme ci-dessus).

* **_ALL_** : référence toutes les variables de la table en entrée.

Array AllVar{*} _ALL_ ;

* **_NUMERIC_** : pour les variables numériques.

Array TypeNum{*} _NUMERIC_ ;

* **_CHARACTER_** : pour les variables caractères.

Array TypeChar{*} _CHARACTER_ ;

- nouvelles, via une liste de variables

- temporaires : **_TEMPORARY_** (RETAIN implicite).

Array Calculs {10} 8. **_TEMPORARY_** ;

Exemple

Soit la table Detail.VentesParMois. Elle contient les ventes mensuelles par client en euros, de janvier 2000 à décembre 2002. Voici un extrait :

Client	Ventes 2000_01	Ventes 2000_02	Ventes 2000_03	Ventes 2000_04	Ventes 2000_12
1 M. AUREL	475	315	400	458	500
2 M. BRUNO	504	731	526	689	639
3 M. CELINE	312	462	424	592	674

Le programme ci-dessous crée deux tables, contenant respectivement :

- le calcul des ventes cumulées depuis le début de l'année pour chaque mois (le 'Year-To-Date'),
- les observations sur lesquelles les ventes ont augmenté de 30 % entre 2000 et 2002.

```
data YTD (label="Cumul" drop=_: Ventes:)
  YTD_30(label="Cumul +30%" drop=_: YTD:);
```

```
/* Création de deux Array multi-dimensionnels [Année, Mois],
l'un référençant les variables existantes dans la table, l'autre
créant de nouvelles variables. */
```

```
Array a_VentesMens [3, 12]
  Ventes2000_01-Ventes2000_12
  Ventes2001_01-Ventes2001_12
  Ventes2002_01-Ventes2002_12 ;
Array a_YTD [3, 12]
  YTD2000_01-YTD2000_12
  YTD2001_01-YTD2001_12
  YTD2002_01-YTD2002_12 ;
set detail.VentesParMois ;
```

```
/* Calcul du year-to-date, illustrant la simplification du code :
sans utiliser d'Array, le programme aurait été plus fastidieux */
```

```
Do _annee=1 to 3 ;
  a_YTD[_annee,1]=a_VentesMens[_annee,1];
Do _mois=2 to 12 ;
  a_YTD[_annee,_mois]=
    sum (a_YTD[_annee,_mois-1] ,
        a_VentesMens[_annee,_mois] ) ;
End;
End;
Output YTD;
```

```
/* Recherche les clients dont les ventes ont augmenté de 30%
: les variables désignées par l'Array peuvent être nommées
directement. */
```

```
If YTD2002_12 gt YTD2000_12 * 1.30)
  then output YTD_up30 ;
run ;
```

La table detail.HistoriqueVentesParMois contient les ventes mensuelles par client en francs, de janvier 1996 à décembre 1999. Le programme ci-dessous la convertit en euros. Il illustre l'utilisation du référencement {*} et **_NUMERIC_** : lorsque de nouvelles variables sont ajoutées à la table, le programme n'a pas besoin d'être modifié.

```
data HistoriqueEuros (drop=_:);
  Set detail.HistoriqueVentesParMois ;
  Array A_VarNum (*) _NUMERIC_ ;
  Do _i=1 to dim(A_VarNum) ;
    A_VarNum(_i)=A_VarNum(_i) / 6.55957 ;
  End ;
run ;
```

Pour approfondir le sujet, vous pouvez consulter la SAS Online Documentation, dans les ouvrages " SAS Language reference ". Vous trouverez également d'autres exemples d'utilisation sur le portail du support clients, à la page " SAS sample programs " : http://support.sas.com/techsup/sample/sample_library.html

Géraldine Deschamps
Ingénieur Consultant



Naviguer depuis vos cartes SAS/GRAPH ®

La version 8 introduit une dimension nouvelle dans les sorties graphiques de SAS/GRAPH. Ces sorties graphiques sont devenues dynamiques, la navigation est désormais possible à partir d'une sortie graphique, un clic souris peut vous amener sur un autre graphe, une page HTML avec d'autres informations.

La composante cartographique de SAS/GRAPH au travers de la procédure GMAP n'échappe pas à cette nouveauté. L'exemple présenté dans cet article permet d'associer une URL à une région administrative française aboutissant sur le site du Conseil Régional de cette même région. Deux voies sont alors possibles : le clic activant l'URL se fait uniquement à partir du centre de la région administrative, ou bien à partir de n'importe quel point dans la région administrative.

A chacune de ces voies correspond une méthode.

1ERE METHODE : table d'annotation- clic point central

Il va donc être ici question de calculer le point central d'une région d'une carte, d'afficher la valeur d'une variable (le nom de chaque région administrative dans notre cas) et d'associer un lien, ceci à l'aide d'une table d'annotation. Cette méthode se décompose en 4 étapes.

Etape 1 : récupération des données géographiques.

Pour rappel : la bibliothèque MAPS regroupe l'ensemble des tables cartographiques du Système SAS.

La table REG_FR va être créée à partir des tables MAPS.FRANCE et MAPS.FRANCE2. Elle regroupe les coordonnées de la France triées en fonction de la variable REGION.

```
data reg_fr;
merge maps.france maps.france2;
by id;
run;
```

```
proc sort data=reg_fr;
by region;
run;
```

```
proc gremove data=reg_fr out=reg_fr;
id id;
by region;
run;
```

La procédure GREMOVE permet dans notre cas de supprimer les coordonnées concernant les départements et de ne conserver que celles des régions administratives.

Etape 2 : utilisation des macros programmes

CENTROID et MAPLABEL disponibles à l'adresse suivante : <http://ftp.sas.com/techsup/download/graph/maplabelmacros.txt>

- CENTROID permet de calculer le centre de chaque région,
- MAPLABEL permet de créer la table d'annotation utilisée avec la procédure GMAP (vous pouvez modifier ici la police utilisée pour les labels ainsi que sa taille et sa couleur), notons au passage que CENTROID est appelé dans ce macro programme.

La seule modification à apporter est d'ajouter la variable HTML qui correspondra au lien affecté à chaque région :

```
[.] Bloc if-Then- Else du macro programme MAPLABEL
else
text=left(input(&textvar,BEST.));
html='href="http://www.cr-||trim(left(region )) || '.fr"';
[.]
```

Vous devez compiler ces macros (soumettre le code).

Etape 3 : exécution de la macro MAPLABEL pour obtenir la table d'annotation CENTERS.

```
%maplabel(reg_fr, reg_fr, centers, region, region);
```

- le 1er attribut correspond à la table contenant les coordonnées cartographiques,
- le 2ème attribut correspond à la table possédant la variable utilisée pour les labels,
- le 3ème attribut correspond à la table d'annotation qui sera créée,
- le 4ème attribut correspond à la variable utilisée pour les labels,
- le 5ème attribut correspond à la variable triée de la table.



La table d'annotation ainsi obtenue (pour les 5 premières observations) est la suivante :

	position	xsys	ysys	when	hsys	text	function	style	color	size	x
1	5	2	2	a	3	ALSACE	label	swis	black	1.5	540.83327233
2	5	2	2	a	3	AQUITAINE	label	swis	black	1.5	214.24999237
3	5	2	2	a	3	AUVERGNE	label	swis	black	1.5	365.5416476
4	5	2	2	a	3	BASSE-NORMANDIE	label	swis	black	1.5	200.12496568
5	5	2	2	a	3	BOURGOGNE	label	swis	black	1.5	405.66666667

	y	Name of Regions	html
1	483.55627441	ALSACE	href="http://www.cr-ALSACE.fr"
2	212.75042725	AQUITAINE	href="http://www.cr-AQUITAINE.fr"
3	310.35614014	AUVERGNE	href="http://www.cr-AUVERGNE.fr"
4	516.90686035	BASSE-NORMANDIE	href="http://www.cr-BASSE-NORMANDIE.fr"
5	412.85327148	BOURGOGNE	href="http://www.cr-BOURGOGNE.fr"

Elle contient :

- le nom (TEXT) de chaque région administrative,
- la police (STYLE) utilisée pour afficher le nom des régions administratives,
- la taille (SIZE) de la police utilisée pour afficher le nom des régions administratives,
- les coordonnées (X,Y) du point central de chaque région administrative,
- le lien (HTML) associé à chaque région administrative.

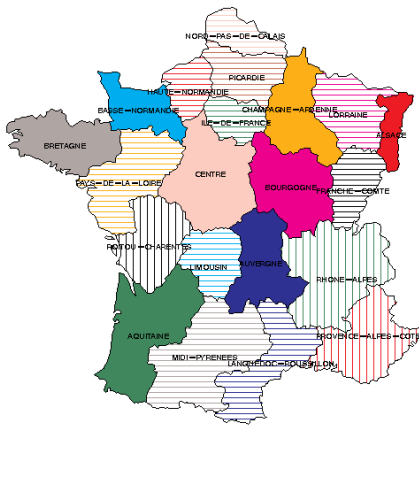
Etape 4 : création de la carte

Pour créer votre carte, vous devez exécuter la procédure GMAP correspondante :

```
ods html file="graph.html" path="c:\temp";
proc gmap data=reg_fr map=reg_fr;
  id region ;
choro region / discrete nolegend      anno=centers;
run;
quit;
ods html close;
```

La table CENTERS est utilisée en tant que table d'annotation (instruction ANNO).

La carte ainsi obtenue (avec une taille de police de 1,5) est la suivante :



Vous pouvez alors cliquer sur le libellé d'une région pour être orienté vers le site du Conseil Régional correspondant. Le clic doit se faire sur le centre de la région administrative, le curseur de la souris se transforme en  une fois que ce dernier est localisé.

Pour information, les liens générés sont de la forme <http://www.cr-REGION.fr>

Cependant, notons que cette règle n'est pas valable pour l'ensemble des conseils régionaux, le code de cet article n'est fourni qu'à titre d'exemple pour illustrer une fonctionnalité.

2ÈME MÉTHODE : option HTML - clic au hasard

L'association réalisée précédemment peut se faire à l'aide de l'option HTML de l'instruction CHORO de la procédure GMAP, comme montré dans l'exemple suivant (la table REG_FR précédemment créée reste valide pour cet exemple).

```
ods html file="graph.html" path="c:\temp";

data lien;
  length htmlvar $80;
  set reg_fr;
  htmlvar="href="http://www.cr- ||trim(left(region))||.fr";
run;

data anno;
  set CENTERS(drop=HTML);
run;

proc gmap map=reg_fr data=lien all;
  id region;
  choro region / discrete
                annotate=anno
                html=htmlvar
                outline=black
                nolegend;

run;
quit;

ods html close;
```

La carte obtenue est identique en apparence à la précédente mais cette fois ci, le lien n'est pas activé à partir d'un point précis d'une région administrative mais à partir de n'importe quel point appartenant à la région.

Le curseur a une forme constante () quel que soit l'endroit où il est positionné sur la carte.

Notons également qu'un clic souris entraîne la sélection complète du contour de la région administrative. Au travers de cet exemple simple, nous avons illustré comment introduire une dynamique dans la consultation d'une carte SAS/GRAPH, soit à partir d'un point précis d'une carte ou bien à partir d'une région.

En complément d'information, nous vous conseillons de lire le document intitulé "SAS Mapping : technologies, techniques, tips and tricks" de Darrell Massengill.

<http://support.sas.com/md/datavisualization/papers/SASMapping.pdf>

Damien Michot
Ingénieur consultant



Comment récupérer des données SAS MVS dans Excel ?

Cet article explique comment récupérer des données SAS provenant de MVS sous Excel.

Pour récupérer ces données dans des fichiers au format Excel, il faut dans un premier temps générer, sous MVS, un fichier HTML à l'aide de l'ODS HTML.

```
ods html body="mon.fichier.excel" rs=none;
Title;
proc print data=sashelp.class;
run;
ods html close;
```

Il faut ensuite transférer ce fichier sous Windows via FTP afin de pouvoir l'ouvrir dans Excel et l'enregistrer au format Excel.

Le transfert peut également se faire de manière plus directe, en utilisant l'envoi de fichier au travers des logiciels SAS. Pour cela, il suffit d'utiliser le filename email, et de spécifier dans les options que le fichier joint sera de type xls :

```
filename outbox email
TO='user@fra.sas.com'
TYPE='text/html'
SUBJECT='Fichier excel'
ATTACH=("mon.fichier.excel"
CONTENT_TYPE='text/html'
EXTENSION='xls');
```

```
data _null_;
file outbox;
put "Envoi de fichiers excel ";
run;
```

Voici la description des options :

- TO : le destinataire du mail
- TYPE : spécifie le type du contenu du corps du mail
- SUBJECT : le sujet du mail
- ATTACH : le fichier joint
- CONTENT_TYPE : spécifie le type du contenu de la pièce jointe
- EXTENSION : spécifie l'extension du fichier joint

Pour plus de précisions sur les options de l'envoi de mail, vous pouvez vous référer à la note technique TS-605 consultable sur notre site Internet à l'adresse suivante :

<http://ftp.sas.com/techsup/download/technote/ts605.html>

Le fichier Excel obtenu soit par le transfert via FTP soit directement en pièce jointe d'un mail sera de la forme suivante :

	A	B	C	D	E	F	G
1							
2	Obs	Name	Sex	Age	Height	Weight	
3	1	Alfred	M	14	69.0	112.5	
4	2	Alice	F	13	56.5	84.0	
5	3	Barbara	F	13	65.3	98.0	
6	4	Carol	F	14	62.8	102.5	
7	5	Henry	M	14	63.5	102.5	

Il est possible d'utiliser un autre style (ex : police, couleur, ...) lors de la création du fichier HTML ou d'en définir un nouveau à l'aide de la procédure TEMPLATE. Des informations sur cette procédure sont disponibles à l'adresse suivante :

<http://support.sas.com/rnd/base/topics/templateFAQ/Template.html>

Le nouveau style sera appliqué au fichier Excel final.

Par exemple, si le style " RTF " est spécifié au niveau de l'instruction ODS HTML, le fichier suivant sera obtenu :

	A	B	C	D	E	F	G	H
1								
2	Obs	Name	Sex	Age	Height	Weight		
3	1	Alfred	M	14	69.0	112.5		
4	2	Alice	F	13	56.5	84.0		
5	3	Barbara	F	13	65.3	98.0		
6	4	Carol	F	14	62.8	102.5		
7	5	Henry	M	14	63.5	102.5		
8	6	James	M	12	57.3	83.0		
9	7	Jane	F	12	59.8	84.5		

À l'adresse suivante, vous trouverez le document " Generating Custom Excel Spreadsheets using ODS " vous permettant de personnaliser vos fichiers Excel :

<http://www2.sas.com/proceedings/sugi28/012-28.pdf>

L'utilisation du filename email pour récupérer des données SAS directement dans Excel permet d'effectuer tous les traitements à partir de SAS. Ce procédé a l'avantage d'éviter le transfert du fichier via ftp et permet l'envoi du fichier directement au destinataire final sans avoir besoin de passer par l'interface Windows.

Clarisse Audoucet
Ingénieur Consultant



Java Server Pages (JSP) : particularités avec les objets graphiques lors d'un déploiement sous Unix.

D'année en année vous êtes de plus en plus nombreux à utiliser le web pour mettre à disposition vos rapports. Aujourd'hui, pour la réalisation de ces rapports, le langage JAVA est une technologie des plus utilisées, notamment les JSP/Servlet (exécution réalisée sur le serveur).

En effet, cette technologie, couplée à la puissance de SAS vous permet de générer et de mettre à disposition vos rapports à n'importe quel moment, depuis un poste client "léger" (poste disposant seulement d'un navigateur web).

Dans cette optique, AppDev Studio™ est un atelier de développement dédié à la conception d'applications décisionnelles et analytiques. Il offre notamment tous les composants nécessaires pour concevoir une application JSP/Servlets.

La richesse des objets graphiques proposés par AppDev Studio™ constitue un atout. Ces objets sont primordiaux à l'interprétation synthétique et simplifiée de vos résultats.

Cet article se propose de vous commenter une des difficultés que vous êtes susceptible de rencontrer lors d'un déploiement sous un environnement Unix (votre Serveur Web Java est installé sur une machine Unix).

Dans un environnement UNIX, hébergeant un serveur Web JAVA qui utilise JDK 1.3.x, un des messages suivants peut apparaître lors de l'affichage de graphiques :

```
java.lang.InternalError:
Can't connect to X11 window server using ':0.0'
as the value of the DISPLAY variable.
at sun.awt.X11GraphicsEnvironment.initDisplay (Native
Method)
at sun.awt.X11GraphicsEnvironment.<clinit>(X11
GraphicsEnvironment.java:63)
```

ou

```
Internal error:
exception thrown from the servlet service function
(uri=/xxx/xxx2.jsp):
java.lang.NoClassDefFoundError: java/awt/SystemColor, stack:
java.lang.NoClassDefFoundError: java/awt/SystemColor
at com.sas.visuals.BaseBorder.<init>(BaseBorder.java:209)
```

Ce message résulte d'un dysfonctionnement connu de Sun, il est référencé dans la base " Bug Parade " à l'adresse suivante :

<http://developer.java.sun.com/developer/bugParade/bugs/4281163.html>

Ce comportement intervient dès lors que vous ne lancez pas votre Serveur Web Java depuis une fenêtre X ou que cette fenêtre ne reste pas active continuellement.

Les différentes possibilités pour pallier à l'apparition de ce message sont décrites ci dessous :

1. JDK 1.4

Ce dysfonctionnement est résolu avec un JDK 1.4, ce qui signifie une montée en version du JDK. Dans les améliorations du JDK 1.4 sur le site de Sun, on peut trouver que ce comportement n'intervient plus :

<http://java.sun.com/j2se/1.4/docs/guide/awt/AWTChanges.html#headless>

Une fois la montée de version de JDK réalisée, il faut mettre en place les options " headless " :

```
-Djava.awt.headless=true
-Djava.awt.headlesslib=true
```

Les spécifications de ces options changeront selon le Java Web Serveur utilisé.

Par exemple pour Tomcat 3.x, vous pouvez spécifier ces options dans le fichier tomcat.sh au niveau du paramètre TOMCAT_OPTS.

Evidemment une montée de JDK n'est pas toujours envisageable.

Il existe alors d'autres manières de contourner cette difficulté sans devoir changer de version de JDK.

NB : Le JDK 1.4 n'est pas supporté en mode développement avec AppDev Studio™ 2.0.x., mais il peut être utilisé lors de vos déploiements..

2. Gestion avec une Fenêtre X

Cette méthode fonctionnera toujours mais elle n'est pas forcément la plus pratique : il faut démarrer sur le moniteur rattaché au serveur une fenêtre X dans laquelle on va démarrer le Serveur Web Java.

Pour faire fonctionner une telle architecture, les pré-requis sont les suivants :

- un moniteur/carte vidéo doit être physiquement rattaché au serveur concerné,
- une fenêtre X doit être démarrée en permanence sur le serveur et affichée sur le moniteur du serveur.

3. Emulateur graphique

Une autre possibilité est de pointer vers un PC qui possède une session d'émulateur graphique en cours (par exemple : Exceed), et d'ajouter au lancement du Serveur Web Java la commande suivante :

```
export DISPLAY=machineipaddress:0.0
```

DISPLAY est une variable d'environnement permettant de désigner l'écran graphique utilisé par défaut par les processus clients du serveur X.

4. Classes Java PJA

Les classes PJA (Pure Java AWT) n'utilisent aucune ressource graphique du système sur lequel fonctionne le Serveur Web Java, à l'inverse des classes AWT.

Les classes PJA ne nécessitent pas de serveur X.

Vous pouvez récupérer ces classes sur Internet à l'adresse suivante : <http://www.eteks.com/pja/>

La documentation sur cette couche et l'explication de sa mise en place y sont détaillées.

5. Xvfb (Serveur X virtuel)

Vous pouvez aussi mettre en place un serveur X virtuel sur la machine Unix pour donner accès aux ressources X de la machine Unix par le Serveur Web Java, sans devoir avoir de fenêtre X ouverte.

Il faut donc installer et lancer un serveur X virtuel sur la machine Unix. Ce serveur virtuel s'appelle "Xvfb".

Vous trouverez des informations sur cette solution à l'adresse suivante :

<http://www.xfree86.org>

L'apparition du message évoqué au début de cet article est certes gênante, néanmoins il existe différentes façons simples de le contourner.

Un résumé des différentes actions possibles est disponible au travers de la note SAS à l'adresse suivante :

<http://support.sas.com/techsup/unotes/SN/007/007313.html>

Fabrice POLLET
Ingénieur Consultant



Comment exporter une structure HOLAP en programmation SCL ?

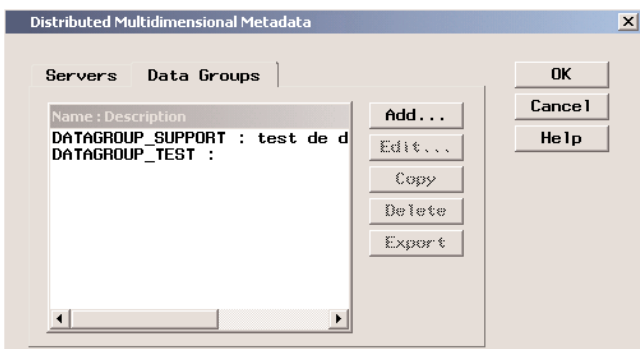
L'utilisation de la classe `HOLAP Metadata` (`sashelp.holapmd.Md.class`) permet en version 8 de mettre à jour les informations relatives aux structures HOLAP à exporter, les "data groups", visibles dans l'interface `Distributed Multidimensional Metadata` (accessible par la commande `MDMDDB`).

Cet article présente quelques méthodes de la classe `HOLAP Metadata`, notamment les méthodes permettant d'exporter un data group afin de créer un proxy.

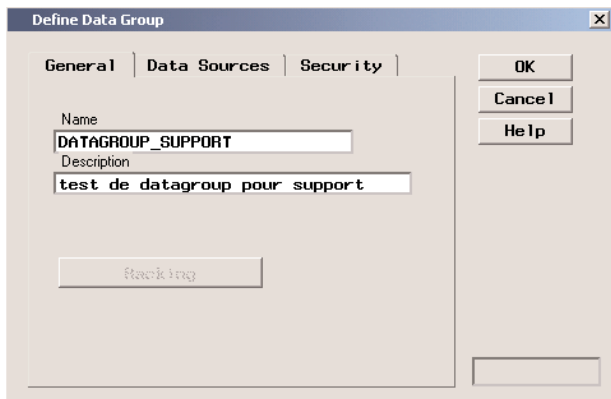
Vocabulaire utilisé

Afin de mieux comprendre les termes utilisés dans cet article, les quelques définitions qui suivent sont illustrées de copies d'écrans issues de l'interface `Distributed Multidimensional Metadata`.

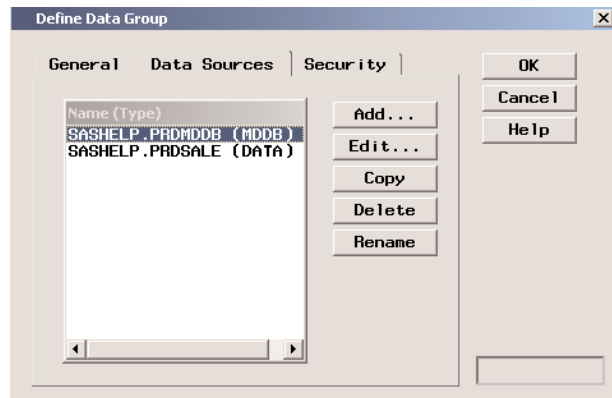
Data group : structure contenant les informations relatives à la structure HOLAP à exporter, comme le nom des sources de données ("data sources"), les éventuels mots de passe etc.



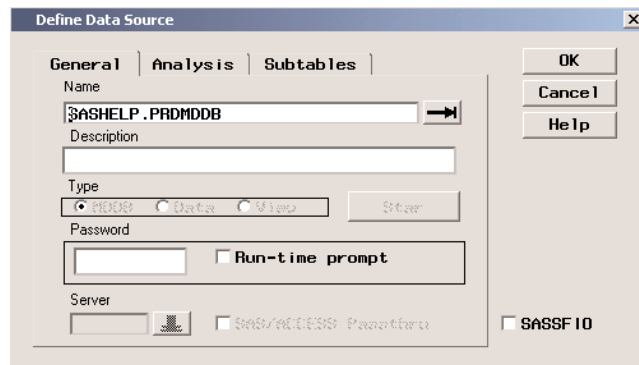
Détails d'un data group.



Data source : source de données référencée dans un data group où sont stockées les informations, comme le nom de la base de données, les variables d'analyse et leurs statistiques associées, les éventuelles hiérarchies etc.



Détails d'un data source



Repository : référentiel où sont enregistrées les données (cubes, tables ou vues).

Proxy : structure multi-dimensionnelle résultant de l'export du data group.

Initialisation de l'environnement OLAP

L'import et l'instanciation de la classe `RESENV` (`sashelp.mb.resenv.class`) sont nécessaires pour récupérer les informations sur le repository sur lequel vous souhaitez travailler :

```
Import sashelp.md.resenv ;
Dcl resenv resenvid = _new_ resenv();
```

Vérification de l'existence du repository hébergeant le data group

L'utilisation de la classe `METABASE` (`sashelp.mb.Metabase.class`) est alors nécessaire. L'identifiant de l'instance de cette classe peut être récupéré dans la liste d'environnement locale et la méthode `_repositoryExist` permet de vérifier l'existence du repository :

```
Dcl Object mbid ;
Dcl Char(1) exist ;
Mbid = getnitemn(envlist('L'),'METABASE');
Mbid._repositoryExist(repname, exist);
```



La variable SCL "exist" contiendra soit "Y" si le repository existe, soit "N" s'il n'existe pas.

La variable SCL "rename" doit être initialisée au préalable avec le nom du repository concerné.

La classe RESENV n'étant plus utilisée par la suite, la méthode _term() permet de supprimer l'objet resenvid :

```
Resenvid._term();
```

Utilisation de la classe HOLAP Metadata

Cette classe offre de nombreuses possibilités comme récupérer les informations relatives aux data groups créés dans un repository, mais elle permet également la mise à jour de ces informations.

Dans l'exemple de code qui suit, un data group préalablement créé va être mis à jour avec l'ajout de data sources, et exporté pour créer le proxy.

Import et instanciation de la classe HOLAP Metadata

```
Import sashelp.holapmd.md ;
Dcl md mid = _new_ md();
```

Chargement des informations relatives au repository

```
mid._setup(repository);
```

Ajout des data sources

Dans cet exemple, deux cubes sont ajoutés dans le data group. Il est également possible d'y ajouter des tables et/ou des vues.

Deux listes SCL (ds_I et dss_I) doivent être alimentées pour retrouver exactement la même structure que les informations présentes dans le data group.

La liste ds_I est créée à vide pour que la liste dss_I ait la structure suivante :

```
( SASHELP.PRDMDDDB=(
SASHELP.PRDMDDDB3=(
)
```

Le code se présente comme cela :

```
dss_I = makelist();
mddb=makearray(2);
mddb[1]='SASHELP.PRDMDDDB';
mddb[2]='SASHELP.PRDMDDDB3';
i=1;
do while (i<=2);
ds_I = makelist();
dss_I = setniteml(dss_I,ds_I,mddb[i],1,1,I);
mid._createDataSourceDefault(mddb[i],"ds_I");
i=i+1;
end;
dg_I = makelist();
dg_I = setniteml(dg_I,dss_I,'DATASOURCES');
dg_I = setnitemc(dg_I,"DESC");
dg_I = setnitemc(dg_I,"PWR");
dg_I = setnitemc(dg_I,"PWW");
```

Mise à jour des informations du data group

```
mid._setHolapGroupInfo(datagroupname,dg_I);
```

La variable SCL " datagroupname " doit être initialisée au préalable avec le nom du data group concerné.

La liste SCL " dg_I " est créée à l'étape précédente.

Enregistrement du Data Group

```
mid._saveMetadata();
```

Export du data group et création du proxy

```
rc = mid._export('EIS',rename,proxynome,
datagroupname,dg_I,'REPLACE', template,1);
```

Notons que la méthode _export, comme le bouton "Export" dans l'interface Distributed Multidimensional Metadata, permet également l'enregistrement du proxy dans le repository.

Les différentes étapes présentées dans ce document peuvent être incluses dans un même fichier SCL, l'ensemble de ce code ayant pour but final de mettre à jour les informations d'un data group et d'exporter ce dernier afin de créer un proxy.

Pour plus de détails sur les classes utilisées dans cet article, et plus particulièrement sur les paramètres des méthodes, vous pouvez vous référer à l'aide en ligne.

Sylvain BERTIN
Ingénieur Consultant



SAS® Enterprise Miner™ : deux façons d'utiliser le nœud Score pour calculer des prédictions

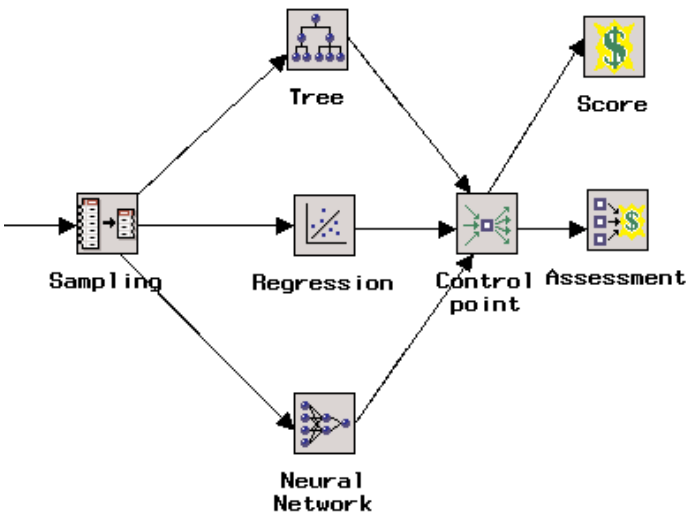
Le nœud Score se positionne à la suite d'un processus de modélisation.

Il est utilisé pour générer du code SAS, afin d'appliquer des modèles à de nouvelles populations (scoring).

Cela peut se faire de plusieurs façons : soit en récupérant le code et en l'exécutant à l'aide du module SAS Base, soit directement dans l'environnement de SAS Enterprise Miner. Ces deux approches sont présentées ci-après.

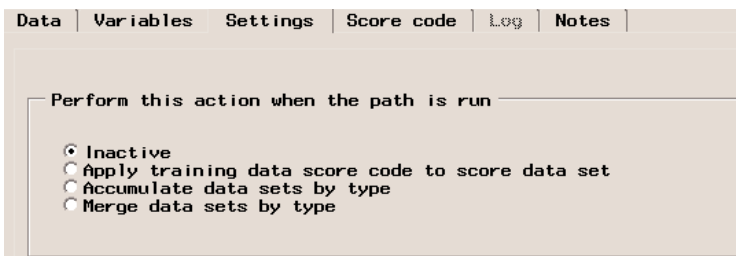
Scoring à partir du code généré par SAS Enterprise Miner

Le nœud score est placé derrière un ou plusieurs nœuds de modélisation, comme dans l'exemple suivant :



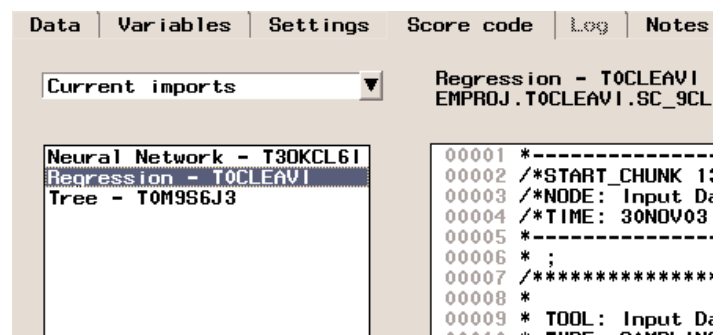
Configuration du nœud Score

Pour obtenir le code nécessaire au scoring, il faut cocher l'option 'Inactive' pour indiquer que le scoring aura lieu au sein d'une session SAS :

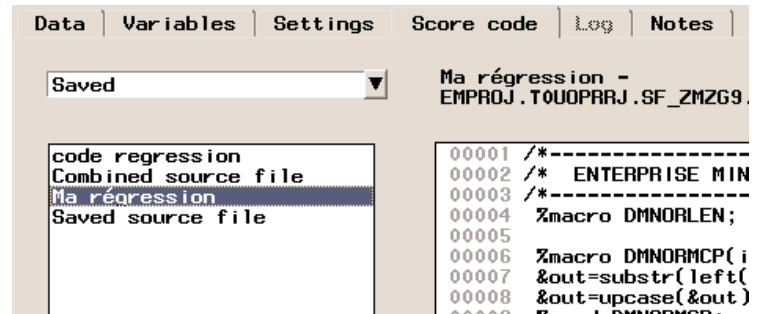


L'onglet 'Score code' permet de visualiser le code relatif à chacun des modèles connectés au nœud Score (directement ou par l'intermédiaire d'un nœud Control Point).

Ces codes ne sont rien d'autre que des étapes data du module SAS Base .



Après avoir sélectionné un modèle par double-clic dans la fenêtre de gauche, le code apparaît dans la fenêtre de droite. Un clic droit propose l'accès au menu 'Save', qui permet de personnaliser le nom de ce code, et de le rendre accessible dans la liste déroulante 'Saved' :



Un nouveau clic droit sur le code précédemment sauvegardé donne accès au menu 'Export', qui génère ce code dans un fichier SAS.

Utilisation du code dans le L4G de SAS :

Le fichier sas généré à l'étape précédente est directement utilisable avec le module Base SAS.

Toutefois, les lignes suivantes doivent être ajoutées, afin de préciser le nom de la table à laquelle appliquer le modèle, ainsi que le nom de la table dans laquelle seront stockées les prédictions :

```
%let _SCORE = Table_d_entree;
%let _PREDICT = Table_de_sortie;
```

Les résultats contenus dans la table de sortie peuvent ensuite être manipulés et affichés, comme on le souhaite grâce aux procédures SAS.

L'avantage de passer par un code SAS est de rendre le projet de Data Mining entièrement indépendant de SAS Enterprise Miner, et de pouvoir ainsi l'exécuter sur tout poste où SAS est installé.

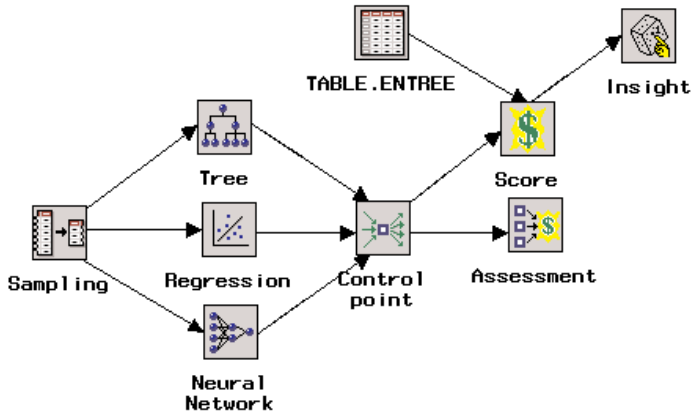
A noter qu'il existe également un nœud ScoreC qui, comme son nom le suggère, génère du code C au lieu du code SAS.

Néanmoins, pour ceux qui ne souhaitent pas manipuler de code SAS, la solution qui suit est préférable.



Scoring au sein de SAS Enterprise Miner

Pour appliquer un modèle à de nouvelles données sans sortir de SAS Enterprise Miner, il faut ajouter au diagramme précédent un nœud Input Data Source en entrée du nœud score.

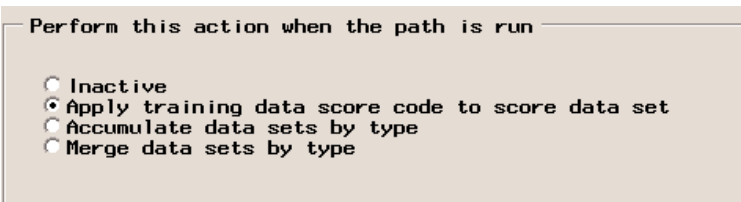


Configuration du nœud Input Data Source :

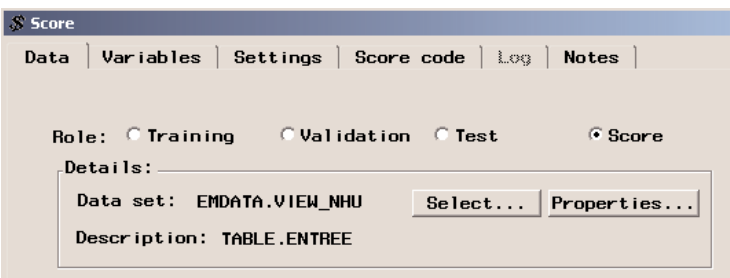
Le nœud Input Data Source doit être configuré de manière à ce que la source de données soit la table contenant les nouvelles données, et le 'rôle' doit être 'SCORE' et non 'RAW'.

Configuration du nœud Score :

Afin que les informations concernant la prédiction soient générées, il faut cocher l'option 'Apply training data score code to score data set', comme ci-dessous



L'onglet Data permet de préciser la table sur laquelle appliquer le modèle :



La visualisation des prédictions calculées peut se faire, par exemple, grâce à un nœud Insight (l'option 'Entire Data Set' doit être cochée dans l'onglet Data).

Les méthodes présentées ci-dessus fournissent toutes les deux les prédictions sur des données. Choisissez celle qui vous convient le mieux.

Vous trouverez plus de précisions sur le nœud Score dans l'aide en ligne de SAS Enterprise Miner, accessible par le menu Help > Em Reference, ainsi que dans le menu Help > Getting Started with Enterprise Miner Software.

Blandine Colas
Ingénieur Consultant

Directeurs de la publication :
Philippe Hoffmann
Pierre Lorenzetti

Comité de rédaction :
Julien Fages
Géraldine Deschamps
Damien Michot
Clarisse Audoucet
Fabrice Pollet
Sylvain Bertin
Blandine Colas

Comité de relecture :
Fabienne Bernard
Mouloud Dey
Philippe Hoffmann

Mise en page :
Agence FC²