Support Clients

ALLO SUPPORT

N°12

Dans ce numéro :

Du français dans vos rapports p1

Les formats datetime à la française p2

Documentation en ligne SAS p3

Etendre les fonctionnalités de SAS Enterprise Guide p4

Accès aux données OS390 depuis SAS Enterprise Guide p5

SAS Enterprise Miner : visualiser et imprimer les arbres de décision en un clic p7

Les outils d'analyse du SCL p8

ASP et SAS p10

Utilisation de la mémoire au niveau du système sous Unix p12

www.sas.com/france vous offre la possibilité de contacter le Support clients en permanence.
La messagerie électronique et le formulaire Internet vous permettent de nous faire part de vos questions sur l'utilisation et le fonctionnement de nos logiciels.
http://www.sas.com/offices/europe/france/services/support/index.html informe sur ces différents modes d'accès.

ftecnews-l permet à l'équipe Support Clients de diffuser rapidement toute nouvelle information technique vers les utilisateurs des solutions SAS. Avec ce mode de distribution privilégié, vous serez informés de la disponibilité des nouvelles versions, des correctifs logiciels et des nouvelles documentations techniques (TS).
L'abonnement est simple et gratuit : il 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 (exemple : subscribe ftecnews-l Jean Dupond).
Cette liste de distribution vient en complément des listes déjà existantes sur notre site Internet (TSNEWS-L, SNOTES-L). Elle s'adresse aux clients francophones, utilisateurs des solutions SAS.

Si vous souhaitez recevoir les éditions futures uniquement via ce mode de distribution, merci d'informer Fabienne VERMAST (frafav@fra.sas.com).

Du français dans vos rapports

La version française du Système SAS amène la traduction des résultats produits par les procédures de Base SAS. Depuis la version 8, l'affichage des résultats passe par l'utilisation de modèles (templates). Un modèle est spécifique à chaque procédure et permet de piloter l'affichage des résultats (libellés, justification...). Pour franciser vos rapports, il suffit donc de remplacer les termes anglais par les français dans la définition de ces modèles.
C'est cette technique que nous avons utilisée dans la version 8.2 du Système SAS francisée. Dans l'exemple ci-dessous, les informations sur une table affichée avec la procédure Contents apparaissent en français.

Le Système SAS

La procédure CONTENTS

Nom de la table SAS:	WORK.ANNEE_EN_COURS	Observations:	3318
Type d'entrée:	DATA	Variables:	32
Moteur:	V8	Index:	0
Créé(e):	11:01 Thursday, July 24, 2003	Longueur de l'observation:	960
Dernière modification:	11:01 Thursday, July 24, 2003	Observations supprimées:	0
Protection:		Compressé(e):	NON
Type de table:		Trié(e):	OUI
Libellé:			

-----Informations liées à l'hôte/au moteur-----	
Taille de page pour la table:	16384
Nombre de pages pour la table:	196
Première page de données:	1
Nb max. d'obs. par page:	17
Obs. dans première page de données:	12
Nombre de corrections dans la table:	0
Nom du fichier:	c:\temp_TD512\annee_en_cours.sas7bdat
Version créée:	8.0202M0
Hôte créé:	WIN_PRO

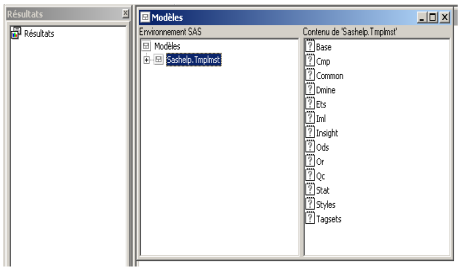
-----Liste alphabétique des variables et attributs-----

#	Variable	Type	Long.	Pos.	Format	Libellé
11	COMPANY1	Alph	50	335		Société

Il faut savoir que les résultats sont découpés en plusieurs objets (un tableau par objet), chacun d'entre eux étant formaté à partir de son modèle propre. Les options *options trace on ;* et *options trace off ;* permettent de trouver le nom du modèle associé à un objet. Ainsi, pour la procédure Contents, ces informations sont affichées dans la log pour le troisième objet :

```
Nom :       VariablesAlpha
Libellé :   Variables
Modèle :    Base.Contents.Variables
Chemin :    Contents.DataSet.VariablesAlpha
```

Pour visualiser la définition du modèle base.contents.variables, il faut se positionner dans la fenêtre Résultats, cliquer droit sur l'élément Résultats et choisir Modèles.



Puis il faut cliquer sur Base (le module auquel la procédure Contents appartient), Contents, et enfin Variables. Le code de la procédure Template correspondant est affiché dans le navigateur de modèles.

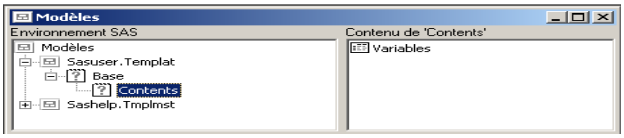
La première instruction de la procédure template est un « define table» du modèle. Supposons que l'on souhaite changer un libellé du tableau, il faut écrire une nouvelle procédure template, où le modèle sera en mode édition (pour permettre sa modification). Voyons directement sur un exemple, où nous allons changer le texte «----- Liste alphabétique des variables et attributs-----» en «Variables par ordre alphabétique» et «Long.» en «Longueur».

```
proc template;
*- remplacement du define en edit -;
  edit base.contents.variables;
  define main;
    *- on conserve uniquement l'instruction text, et
on tape le texte que l'on souhaite voir apparaître
dans la sortie -;
    text "**** Variables par ordre alphabétique
****";
  end;
  define len;
    header= "Longueur";
  end;
  end;
run;
```

On ne conserve que les définitions de colonne qui nous intéressent, avec l'instruction test ou header=. C'est pour éviter d'avoir à rechercher l'instruction adéquate, le nom de la colonne, qu'il est intéressant d'éditer au préalable le code original du modèle.
Après avoir exécuté le code ci-dessus, le résultat de la procédure Contents apparaît sous sa nouvelle forme.

**** Variables par ordre alphabétique ****						
#	Variable	Type	Longueur	Pos.	Format	Libellé
11	COMPANY1	Alph	50	335		Société

Les modèles fournis par SAS sont stockés dans la bibliothèque SASHELP, et sont en accès lecture seule. Quand un modèle est modifié comme ci-dessus et exécuté, il est stocké par défaut dans la bibliothèque SASUSER. Pour le vérifier, il suffit de retourner dans la fenêtre des modèles. En plus de l'item sashelp.tmplmst, il existe désormais sasuser.templat, où est stocké le modèle base.contents.variables modifié.



Comme l'ordre de recherche des modèles est d'abord la bibliothèque sasuser, puis la bibliothèque sashelp, c'est le modèle nouvellement créé qui est utilisé pour produire la sortie.
Cet ordre est défini de cette manière par défaut mais peut être changé avec l'instruction ODS PATH. L'instruction « ODS PATH show ; » permet de vérifier le paramétrage de la session en cours.

```
La liste ODS PATH courante est :

1. SASUSER.TEMPLAT(UPDATE)
2. SASHELP.TMPLMST(READ)
```

Nous venons de voir comment modifier une traduction existante. Le même procédé peut être utilisé pour traduire les résultats des procédures statistiques, par exemple, celles qui n'ont pas été prises en charge par notre service traduction.

Karine Chrillesen
Ingénieur Consultant

Les formats datetime à la française

La procédure FORMAT permet à l'utilisateur de définir ses propres formats et **informats**.
Grâce à la Version 8 du Système SAS®, l'option DATATYPE a été ajoutée à l'instruction PICTURE.
Elle permet de formater des valeurs de type dates, heures et **datetime** en utilisant des «directives».

Prenons le cas où nous souhaitons obtenir l'affichage suivant : 25décembre2003 15:15:00

C'est un affichage où l'heure est séparée de la date par un espace et non par deux points (caractère **:**), standard proposé dans le Système SAS. Les directives de la procédure format permettent de répondre à ce type de question. Il ne reste plus qu'à construire votre format selon l'affichage que vous souhaitez obtenir pour vos valeurs.

Afin d'utiliser les directives, il est nécessaire de positionner l'option DATATYPE à la valeur **date**, **time** ou **datetime**.
Dans notre exemple, il faut l'initialiser à DATETIME.
Le «squelette» de la **proc format** est donc celui-ci :

```
proc format;
      picture nom_du_format
                  low-high=' instructions  '
                  (datatype=datetime);
run;
```

Note : la liste des instructions est disponible sur l'aide en ligne (rubriques Format procedure⇨ Syntax Picture).

Dans le cas présent, la valeur à afficher possède les caractéristiques suivantes :
- le jour sur 2 chiffres : la directive à utiliser est « %d ». Mais avec cette directive, les jours compris entre 1 et 9 seront affichés sur un seul chiffre. Si vous souhaitez avoir un 0 devant les chiffres de 1 à 9, il faut donc utiliser « %0d »,
- le mois en entier : la directive à utiliser est «%B». Pour que le mois soit écrit en français, vous devez modifier la valeur de l'option DFLANG qui est par défaut à anglais,
- l'année sur 4 chiffres : la directive à utiliser est «%Y», mettre un espace entre la date et l'heure,
- les heures, minutes et secondes avec « : » comme séparateur, sur 2 chiffres chacun, les directives respectives sont %H, %M et %S. Mais vous devez ajouter pour chacune d'elles un 0 pour que les valeurs soient sur 2 chiffres.

La **proc format** ainsi obtenue est la suivante :

```
proc format;
 picture test low-high='%0d%B%Y %0H:%0M:%0S'
    (datatype = datetime);
run;
```

Le code final en vue de l'objectif recherché prend donc la forme suivante :

```
option dflang=french;

data _null_;
aujourdhui ="20dec2003:15:15:00"dt ;
put aujourdhui datetime.;
put aujourdhui eurdfdt. ;
put aujourdhui test23. ;
run;
```

La log indique alors :

20DEC03:15:15:00
20dec03:15:15:00
20décembre2003 15:15:00

Damien Michot
Ingénieur Consultant

Documentation en ligne : particularités avec IE6 sous Windows XP

Les dernières versions d'Internet Explorer 6 sur les environnements Windows XP ne contiennent plus de Machine Virtuelle Java (JVM) (source : http://www.microsoft.com/mscorp/java/).
Dans ce cas, la documentation en ligne 'Sas OnlineDoc' avec les fonctions de recherche et d'index ne peut plus être utilisée, car elle s'appuie sur une applet qui s'exécute nécessairement avec la machine virtuelle Java. Néanmoins une solution a été trouvée pour utiliser malgré tout la documentation en ligne dans cette configuration technique.
Nous vous présentons, au travers de cet article, la démarche à suivre.

Remarque :
La documentation en ligne au format HTML sans les fonctions de recherches n'est pas impactée par ce problème (aide sans java).

1. Java Plug-In 1.1.3

Pour avoir une machine virtuelle java correcte sur le poste de travail, vous devez vous procurer un Plug-In Java. Dans le cas présent, vous devez installer le Plug-In java 1.1.3 qui intègre une machine virtuelle Java de version 1.1.8 (version compatible avec l'applet de l'aide en ligne et équivalente à la JVM des navigateurs IE).

Ou le trouver ?
Vous pouvez le récupérer sur le site internet de SUN à cette adresse :
http://java.sun.com/products/archive/plugin/1.1.3_007/index.html
Vous devez choisir la version 'internationalized' pour l'environnement Windows.

Où s'installe t'il ?
Il doit s'installer dans l'arborescence suivante :
C:\Program Files\Java Plug-in 1.1

Une fois cet outil installé, vous devez suivre les actions suivantes :
a) Copier le fichier sasixt.jar (répertoire sasdoc\sashtml) de l'aide en ligne SAS dans le répertoire C:\Program Files\Java Plug-in 1.1\lib.

b) Lancer le programme se trouvant dans « Démarrer => Programmes => java Plug-in Control Panel » et valider les champs suivants :
Onglet « Basic » :
Cache Jar in memory : Coché
Enable Java Plug-In : Coché
Network access : Restricted
Onglet « Advanced » :
Java Runtime Environment : Use Java Plug-In Default
Enable Just in time Compiler : Coché
Onglet « Proxies » :
Use Browser Settings : Coché

Remarque :
Lors des premiers tests, vous pouvez aussi cocher 'Show Java Console' (onglet Basic) qui permettra de visualiser le déroulement du chargement de l'applet.

2. Le code HTML

Pour utiliser le Plug-In Java installé, le code HTML de la page applet.htm (répertoire sasdoc\sashtml) de la documentation en ligne SAS doit être réécrite.
Il est judicieux de sauvegarder la page applet.htm sous un autre nom avant de la modifier. Ce nouveau code permet d'utiliser le Plug-In java automatiquement.

Nouveau code HTML de applet.htm :

```
<html>
<!-- Applet HTML for left frame of OnlineDoc application -->
<head>
<title>SAS OnlineDoApplet Frame</title>
</head>c
<BODY bgcolor="white">
<SCRIPT LANGUAGE="JavaScript"><!--
  document.write('<OBJECT ');
  document.write(' CLASSID="clsid:8AD9C840-044E-11D1-B3E9-
00805F499D93"');
  document.write(' CODEBASE="http://java.sun.com/products/
plugin/1.1.3/jinstall-113-win32.cab#Version=1,1,3,0"');
  document.write(' WIDTH = 98%');
  document.write(' HEIGHT = 100%');
  document.writeln('><NOEMBED><XMP>');
//--></SCRIPT>
<APPLET CODE = "com/sas/ixt/cd/CDApplet.class"
          WIDTH =98% HEIGHT =100% vspace=0 hspace=0>
          </XMP>
<PARAM NAME=CODE       VALUE = "com/sas/ixt/cd/CDApplet.
class">
<PARAM NAME="CODEBASE" VALUE=".">
<PARAM NAME="type"     VALUE = "application/x-java-applet;
version=1.1.3">
<param name=HomeURL value="contents.txt">
<param name=TabBackground value=white>
<param name=Background value=#EEE8AA>
<param name=HelpButton1Label value="Quick Tour">
<param name=HelpButton1URL value="odquick/splash2.htm">
</APPLET>
</NOEMBED></EMBED></OBJECT>
</BODY>
</HTML>
```

La ligne en gras du code HTML représente la ligne à modifier selon si le chargement de l'aide en ligne est effectué à partir d'un serveur Web ou de votre poste en local.

2.1 Chargement à partir d'un serveur Web :
Cette ligne est à supprimer dans ce cas. Tous les composants de l'aide en ligne doivent être trouvés correctement.

2.2 Chargement depuis un poste local ou par l'intermédiaire d'un chemin Windows :
Vous devez positionner la ligne suivante : <PARAM NAME="CODEBASE" VALUE=".">
Cela permettra à l'applet de trouver toutes ces ressources correctement.

Point important:
Si le chemin Windows vers la documentation en ligne SAS comporte des espaces, vous devez faire un «mapping» Windows vers ce répertoire.

Prenons un exemple classique. Le répertoire d'installation par défaut de l'aide en ligne est :
C:\Program Files\SAS Institute\SASOnlineDocV8\sasdoc\sashtml.
Ce chemin contient des espaces et empêchera un chargement correct de la documentation en ligne.
Pour contourner ce problème, nous vous proposons les étapes suivantes :
- Dans l'explorateur Windows, sélectionner le répertoire sashtml qui se trouve dans C:\Program Files\SAS Institute\SASOnlineDocV8\sasdoc et vous affichez ces propriétés.
- Dans l'onglet «Partage», cocher «Partager ce dossier» (par défaut, le partage prend le nom du répertoire) et appliquer. Il ne faut pas que ce partage contienne un espace.
- Maintenant, charger la page onldoc.htm à partir de «\\sashtml\onldoc.htm » au lieu de «C:\Program Files\SAS Institute\SASOnlineDocV8\sasdoc\sashtml\onldoc.htm».

Au terme de ces différentes étapes, la documentation en ligne disposera de l'intégralité de ses fonctionnalités avec Internet Explorer 6 sous Windows XP.

Pascal Nicolas
Ingénieur Consultant

Étendre les fonctionnalités de SAS® Enterprise Guide® avec ses propres tâches.

1. Pourquoi créer une nouvelle tâche ?

Une des innovations de SAS® Enterprise Guide® version 2 est la possibilité de créer ses propres tâches.

Les tâches personnalisées ouvrent de nouvelles possibilités :
- livrer aux utilisateurs des outils adaptés à leur métier,
- normaliser des programmes SAS (pour le même traitement le même code sera exécuté).

Leur utilisation est transparente pour les utilisateurs. Elles s'intègrent parfaitement à l'interface de SAS Enterprise Guide et s'utilisent en tout point comme celles déjà existantes.

Ces extensions peuvent être créées à partir d'un environnement de développement supportant la technologie COM (Component Object Model) de Microsoft comme Visual Basic, Visual C++ ainsi que les langages de programmation .NET.

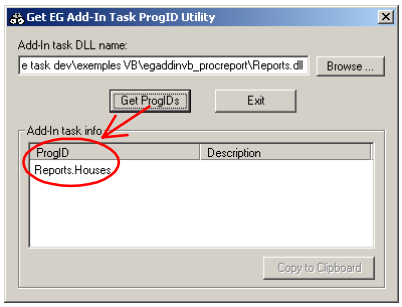
2. Comment ajouter une tâche personnalisée dans SAS Enterprise Guide ?

Une tâche SAS Enterprise Guide se présente sous la forme d'un fichier .DLL (Dynamic Link Library). Quelques exemples de tâches sont fournis sur le site Internet SAS avec leur code source (Cf 4.).

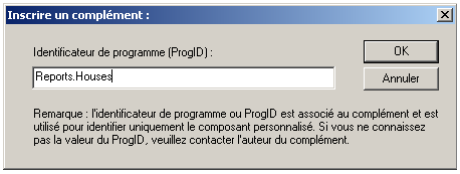
Pour implémenter une tâche, il suffit que le fichier DLL soit accessible et qu'il soit enregistré sur le poste de chaque utilisateur indépendamment du mode de déploiement de SAS Enterprise Guide.. Ensuite, il faut intégrer cette tâche à l'interface de SAS Enterprise Guide en procédant de la manière suivante :

a. Fermer SAS Enterprise Guide.
b. Copier le fichier DLL sur le poste de l'utilisateur.
c. Enregistrer le fichier avec la commande ci-dessous. Une fenêtre s'ouvre pour confirmer que tout s'est bien déroulé : regsvr32 c:\addins\reports.dll

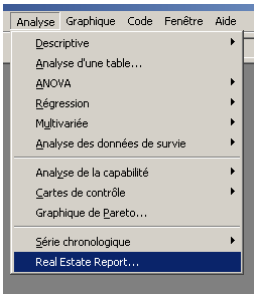
d. Utiliser l'utilitaire *getprogid.exe* téléchargeable à l'adresse ci-dessous pour obtenir l'identifiant de la tâche :
http://www.sas.com/technologies/bi/query_reporting/guide/getprogid.zip



e. Ouvrir SAS Enterprise Guide et aller dans le menu *Outils* ➔ *Personnaliser* puis dans l'onglet *Compléments* et enfin cliquer sur le bouton Ajouter.
Saisir l'identifiant de la tâche puis valider.



f. Fermer puis ouvrir SAS Enterprise Guide, la tâche a été ajoutée à l'interface.



3. Précautions à prendre

Le fichier *tabctl32.ocx* doit être enregistré (Cf. SN-009447) sur le poste de l'utilisateur.
Ce fichier peut être facilement téléchargé sur Internet ou être copié depuis un poste sur lequel le fichier est présent.

Afin d'enregistrer ce fichier il faut utiliser la commande Windows suivante :

regsvr32 tabctl32.ocx

4. Références :

La documentation sur la réalisation des tâches, des exemples prêts à l'emploi avec leur code source ainsi que des utilitaires sont disponibles à l'adresse suivante:
http://www.sas.com/technologies/bi/query_reporting/guide/customtasks.html

La note SN-009447 :
http://support.sas.com/techsup/unotes/SN/009/009447.html

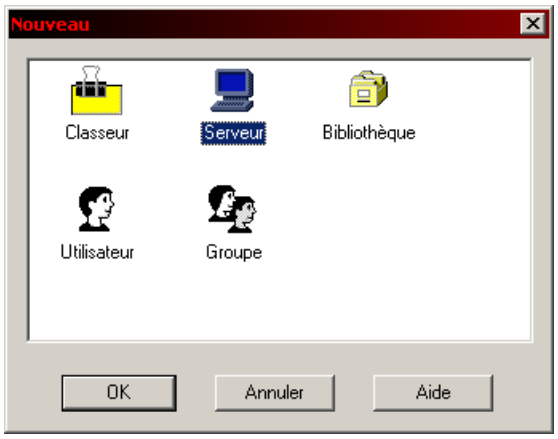
Julien Fages
Ingénieur Consultant

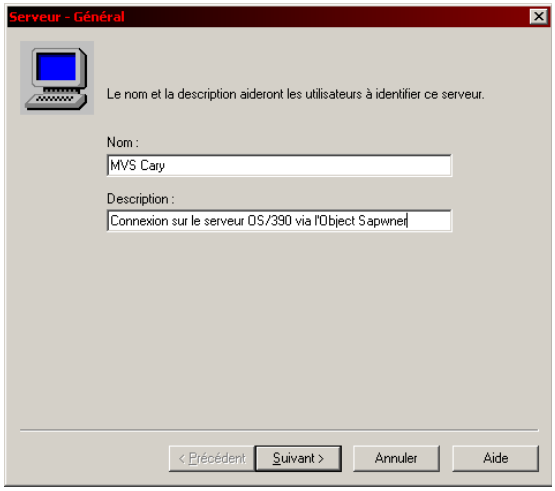
Accès aux données OS390 depuis SAS® Enterprise Guide®

Dans Allô Support n°5, nous avions vu comment paramétrer le module SAS/IT® et l'Object Spawner sur OS/390, afin de permettre à un client SAS Enterprise Guide de se connecter à un serveur SAS sur mainframe. Nous allons voir aujourd'hui les spécificités d'un tel serveur dans l'interface SAS Enterprise Guide.

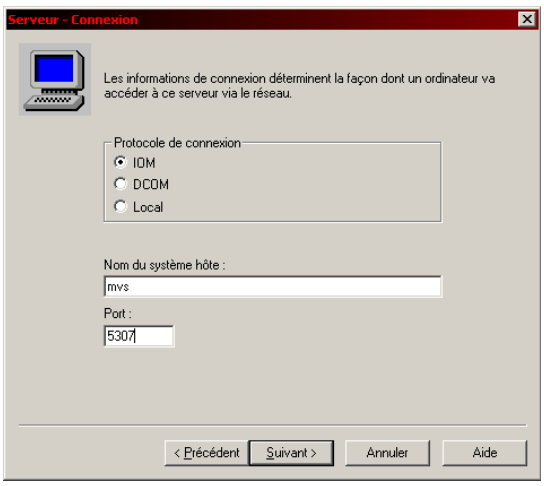
Définition d'un serveur SAS sur OS/390

Dans l'Administrateur de SAS Enterprise Guide :
Nous allons définir un nouveau serveur, avec le menu Fichier > Nouveau…

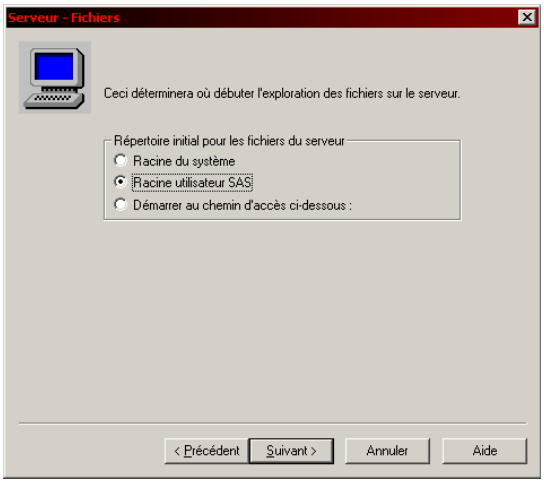




On indique ensuite le nom TCP du serveur OS/390, et le port utilisateur sur lequel le Spawner est en écoute.

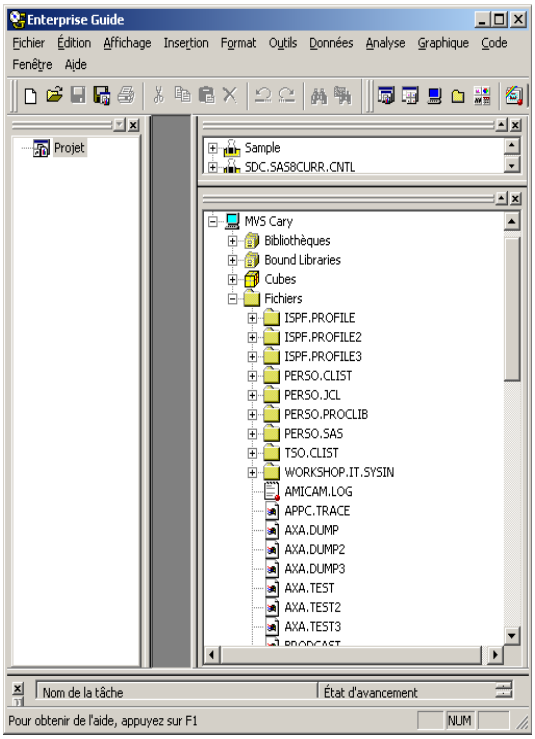


ATTENTION : Il faut impérativement que le répertoire initial pour l'exploration des fichiers soit la racine utilisateur :



Visualisation d'un fichier depuis SAS Enterprise Guide

Dans SAS Enterprise Guide, on peut accéder directement aux fichiers (séquentiels et partitionnés) préfixés par le user TSO de l'utilisateur.
Menu Affichage > Liste des serveurs :

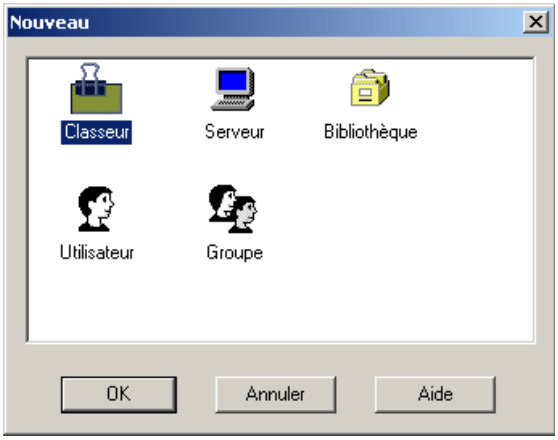


Les fichiers partitionnés (PDS) sont représentés par un répertoire, les fichiers séquentiels par un fichier simple.

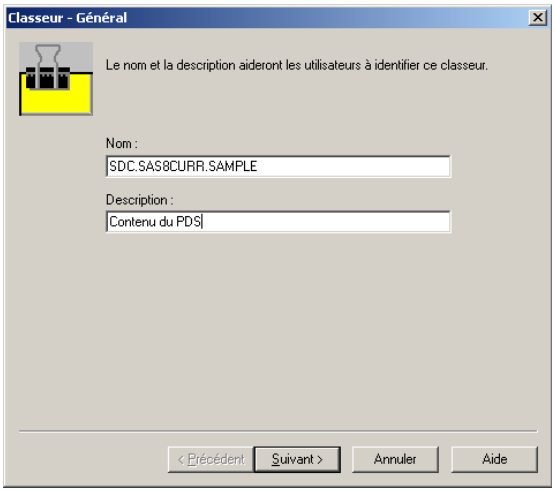
Pour accéder à des fichiers partitionnés autres que ceux préfixés par le user TSO, on peut créer des Classeurs (ou Binders).

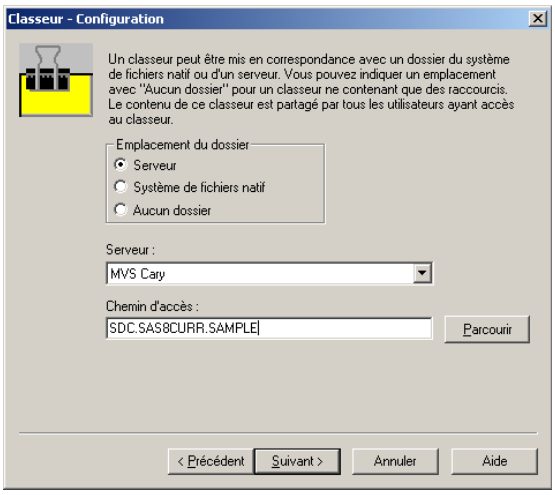
Dans l'Administrateur :

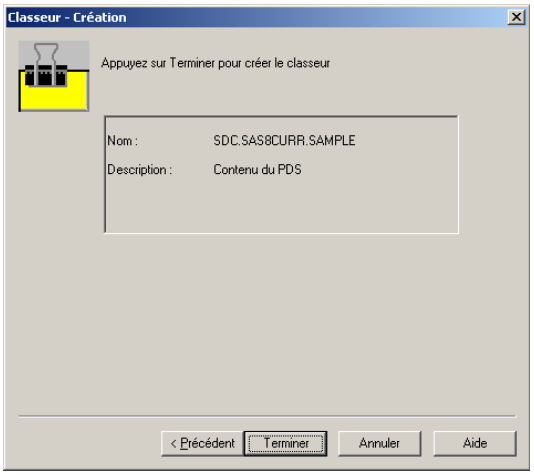
On crée ici un nouveau classeur :



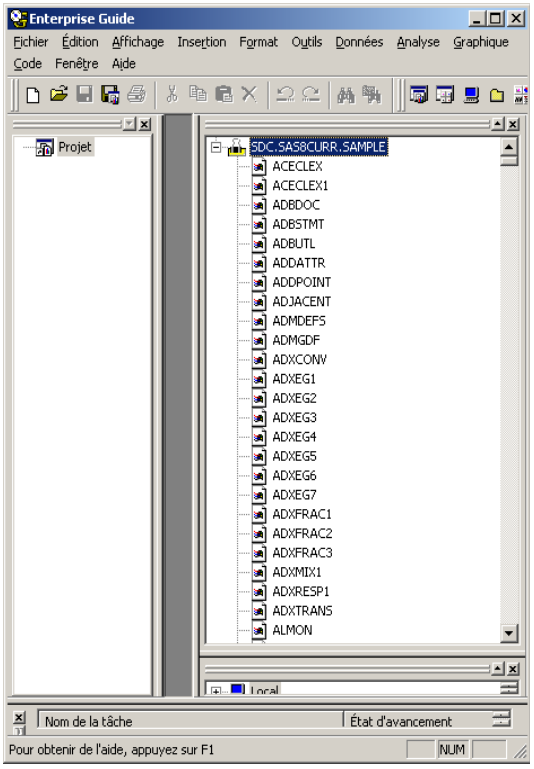
On choisit ensuite le serveur OS/390, et le fichier PDS à lire :







Dans SAS Enterprise Guide : le nouveau classeur apparaît, et permet de lister les membres du PDS accédé (menu Affichage > Liste des classeurs)



Le «glisser/déplacer» ne fonctionne pas, car les noms de fichiers sur OS/390 n'ont pas d'extension qui permettrait de les identifier comme du code SAS, des données…
Attention au cas particulier exposé dans la note suivante : http://support.sas.com/techsup/unotes/SN/008/008971.html

---

SN-008971
Unable to import a .txt file from MVS using Enterprise Guide

You may be unable to import .txt files on MVS when using Enterprise Guide. The problem will most likely occur if the DCB RECFM is not VB. To attempt to circumvent the problem, give the text file a setting of RECFM=VB and then import the file into Enterprise Guide.

This issue is scheduled to be fixed for Enterprise Guide 3.0.

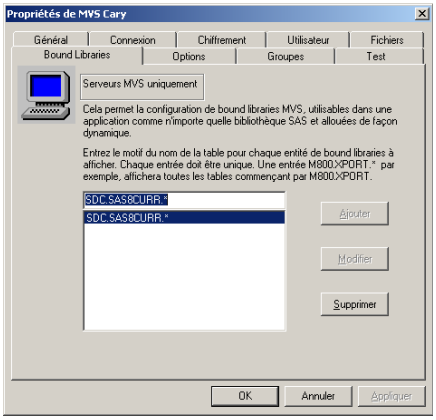
Un correctif est disponible, sur demande auprès du Support Clients.

« Bound Libraries »

Cette fonctionnalité, depuis la version 2 de SAS Enterprise Guide, est accessible dès la création du serveur, ou via ses propriétés. Dans la liste des serveurs, la rubrique « Bibliothèques » ne permet de visualiser que les bibliothèques qui ont fait l'objet d'une allocation explicite (soit par défaut, soit grâce à une instruction libname).
La rubrique « Bound Libraries » permet, avec un serveur SAS sur MVS, d'accéder directement à toutes les tables SAS existantes sous un préfixe donné.

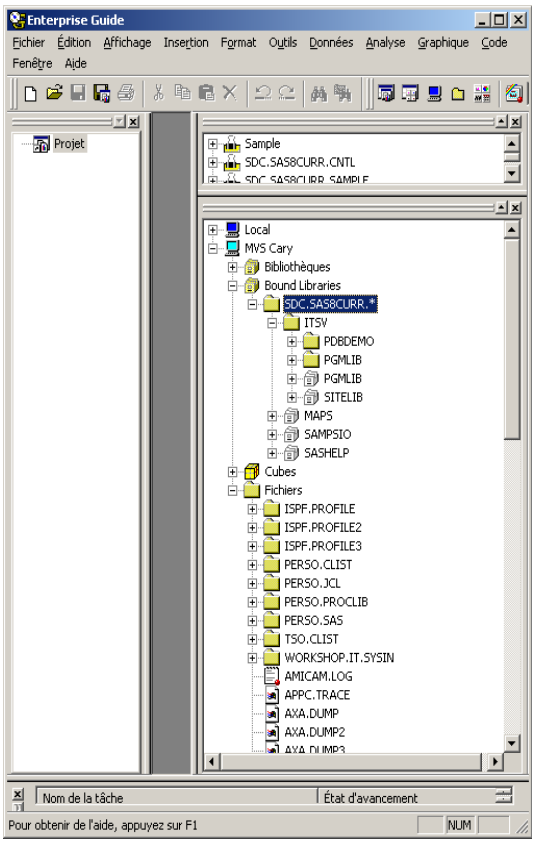
Dans l'Administrateur :

Cette définition peut être faite à la création du serveur, ou en modifiant ces propriétés :



Tous les fichiers commençant par SDC.SAS8CURR.*, et qui sont des bibliothèques SAS, seront présentés sous cette entrée.

Dans SAS Enterprise Guide : les bibliothèques sont accessibles dans l'interface de SAS Enterprise Guide comme suit :



L'arborescence est créée par l'environnement Windows.
Les fichiers physiques accédés sont :
SDC.SAS8CURR.ITSV.PDBDEMO.*
SDC.SAS8CURR.ITSV.PGMLIB.*
SDC.SAS8CURR.ITSV.PGMLIB
SDC.SAS8CURR.ITSV.SITELIB
SDC.SAS8CURR.MAPS
SDC.SAS8CURR.SAMPIO
SDC.SAS8CURR.SASHELP

Conclusion:

Avec SAS Enterprise Guide et un serveur SAS sur OS/390, il est possible de visualiser les fichiers partitionnés soit depuis l'explorateur de fichiers de la liste des serveurs, soit en créant un classeur.
Pour accéder aux bibliothèques SAS sans faire d'allocation explicite, les «Bound Libraries» peuvent être utilisées.

Ludovic Breysse
Ingénieur Consultant

SAS® Enterprise Miner™ - Visualiser et imprimer les arbres de décision en un clic...

Il est parfois difficile de visualiser ou d'imprimer des arbres de décision volumineux obtenus avec SAS Enterprise Miner.
A partir de la version 4.1, SAS Enterprise Miner met à votre disposition, à titre expérimental, un composant Microsoft '**Tree Results Viewer'**, améliorant nettement la qualité des graphiques et leurs impressions.

Accès à l'application 'Tree Results Viewer'

Cette application est automatiquement installée avec Enterprise Miner (à partir de la version SAS 8.2).

Cependant, pour l'activer, la commande suivante doit être exécutée avant de lancer SAS Enterprise Miner :

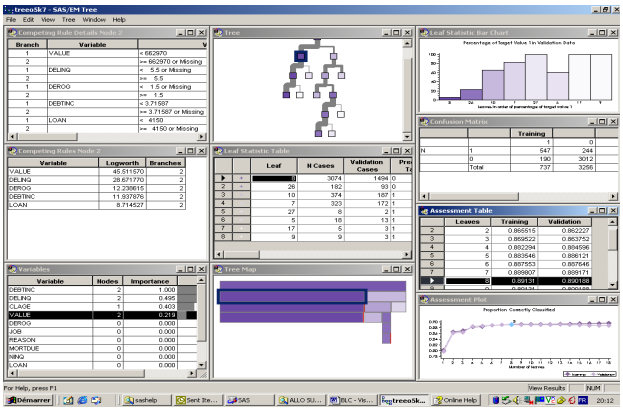
```
%let emv4tree=1 ;
```

Pour ne pas avoir à effectuer cette manipulation à chaque fois, cette commande peut être saisie dans le fichier autoexec.sas.

L'application Tree Results Viewer est alors accessible, par un clic droit sur le nœud Tree concerné et en choisissant le menu '**New view** '.

<u>Présentation de l'application Tree Results Viewer :</u>

Lors de l'ouverture de l'application 'Tree Results Viewer', un certain nombre de fenêtres (ou vues), contenant des tables ou des graphiques, apparaissent à l'écran :



Les vues suivantes s'obtiennent en sélectionnant tous les items du menu **View** :
- Assessment Plot
- Assessment Table
- Confusion Matrix
- Tree
- Tree Map
- Leaf Statistics Bar Chart
- Leaf Statistics Table
- Variables
- Competing Rules
- Competing Rules Details
- Surrogate Rules
- Surrogate Rules Details

Toutes ces vues sont des fenêtres Windows redimensionnables qui peuvent s'agencer automatiquement grâce au menu **Windows** dans la barre de menus de l'application.

Elles présentent, d'autre part, la particularité de se synchroniser, en fonction de l'élément sélectionné.

Exemple : Un clic sur une feuille d'arbre dans la fenêtre **Tree view** mettra en surbrillance la feuille dans 4 autres fenêtres: Tree, Tree Map, Leaf Statistics Bar Chart, et Leaf Statistics Table.

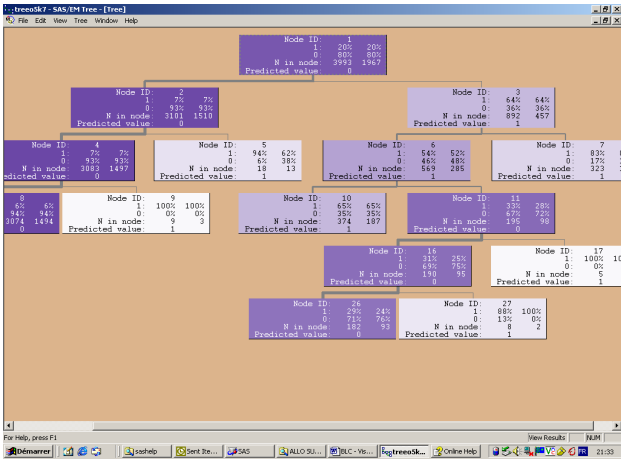
Enfin, il est possible de les imprimer indépendamment les unes des autres, par le menu **File > Print**.

Personnalisation des graphiques :

Les menus **Tree > View Properties** et **Tree > Map Properties** vous permettent de modifier la couleur et l'aspect des divers éléments de l'arbre, ainsi que la taille ou le style de la police utilisée.

Certaines modifications, comme la couleur de l'arbre ou le choix d'une police, se répercutent instantanément sur les autres graphiques.
Dans tous les cas, les paramètres spécifiques à chaque vue, sont accessibles via un clic droit sur la vue concernée.

Ainsi, un clic droit sur la fenêtre **Tree View** nous donne accès au menu **Node Statistics**, qui permet de spécifier les informations statistiques relatives à chaque feuille de l'arbre. En cochant **Display Text**, par un nouveau clic droit, on peut obtenir un arbre comme suit:



Vous trouverez tous les détails concernant les menus, dans l'aide en ligne : **Help > Help Topics**

Blandine Colas
Ingénieur Consultant

Les outils d'analyses du SCL

Pour aider au développement SAS/AF® et SAS/EIS®, et à la maintenance des applications, SAS fournit une panoplie d'outils dont un utilitaire d'analyse du SCL, le **sclprof**.
Il se compose de 4 éléments essentiels :
- ***Coverage Analyzer*** : il permet de suivre l'exécution d'une application et procure des rapports listant les instructions SCL non exécutées ;
- ***List Diagnostic Utility*** : il trace l'exécution d'une application et génère des rapports sur l'utilisation des listes SCL, précisant les listes qui ne sont pas supprimées ;
- ***Performance Analyzer*** : il suit le déroulement d'une application et fournit des rapports qui permettent d'identifier les goulots d'étranglement dans le code SCL
- ***Static Analyzer*** : il utilise les données collectées avec la procédure BUILD et met à disposition des informations relatives à la structure de l'application, notamment un graphique des flux, des rapports sur les fonctions, les méthodes, les variables, les entrées/sorties ainsi que les messages d'erreur et d'avertissement.

L'objectif de cet article est de vous montrer brièvement l'utilisation de chacun de ces outils.

► Coverage Analyzer
Cet outil de diagnostic se décompose en 2 parties : une phase collecte des données puis une phase présentation des données.

Collecte des données
Deux types de collecte sont possibles selon le besoin. Les informations peuvent être récupérées sur l'exécution de l'ensemble d'une application ou seulement sur une partie.

Pour suivre l'intégralité de l'exécution d'une application, l'instruction «**sclprof=coverage**» doit être placée après la commande de lancement de l'application.

Par exemple :
af c=libname.moncatal.monappli.frame sclprof = coverage

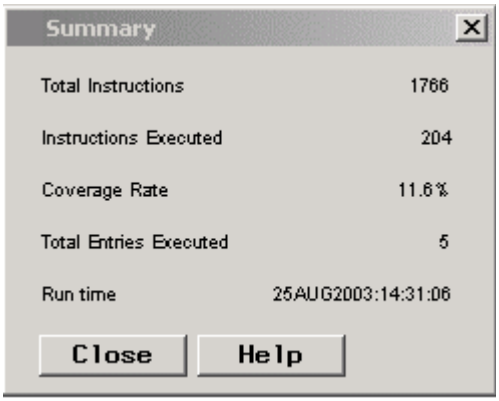
Pour suivre une partie de l'exécution d'une application, la commande « **sclprof coverage on**» doit être entrée en ligne de commande lorsque la phase de collecte doit démarrer. Pour l'arrêter, il suffit de saisir la commande « **sclprof coverage off** ».

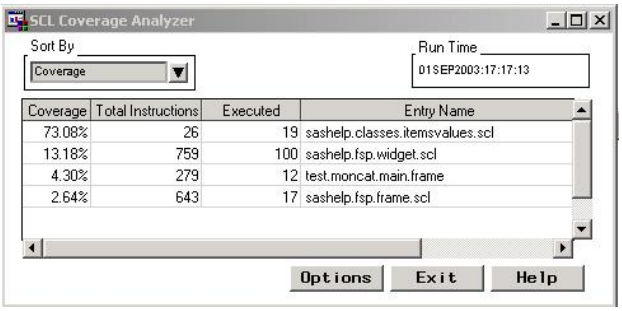
Restitution des informations

Cette phase démarre automatiquement à la fin de celle de collecte. Elle peut apparaître soit à la fin de l'exécution d'une application soit lorsque la commande «**sclprof coverage off**» est exécutée.

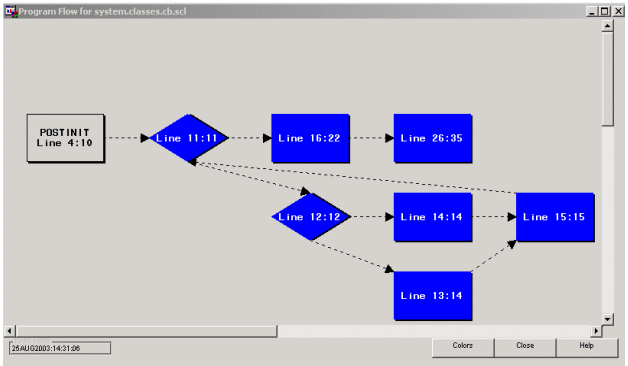
Les principales fonctionnalités sont regroupées au sein de deux fenêtres principales :

- Fenêtre **Summary** :





- Fenêtre SCL Coverage Analyzer
Cet écran permet d'accéder à d'autres fonctionnalités (notamment un diagramme des flux) en double cliquant sur une des lignes.



►List diagnostic utility

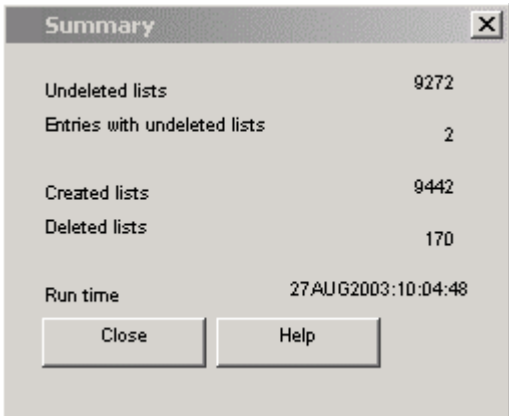
Deux phases sont nécessaires pour obtenir les informations finales : une phase collecte des données puis une phase de présentation.

Collecte des données

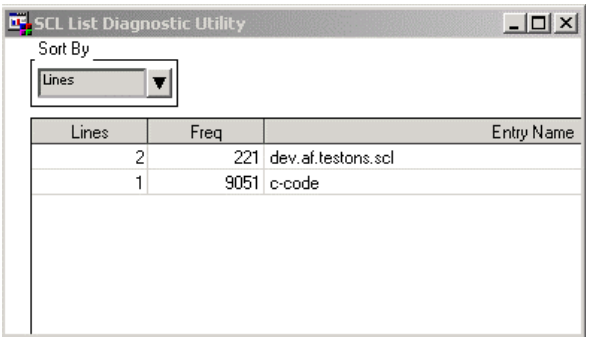
Pour débuter cette phase, vous devez saisir la commande « **sclprof list on** » depuis la ligne de commande. La commande « **sclprof list off** » arrêtera la collecte des informations.

Restitution des informations

Cette phase démarre automatiquement à la fin de la phase de collecte des informations.
Tout comme l'utilitaire précédent, deux fenêtres principales fournissent des informations :



- Fenêtre **Summary**
- Fenêtre SCL List Diagnostic Utility



►Performance Analyzer

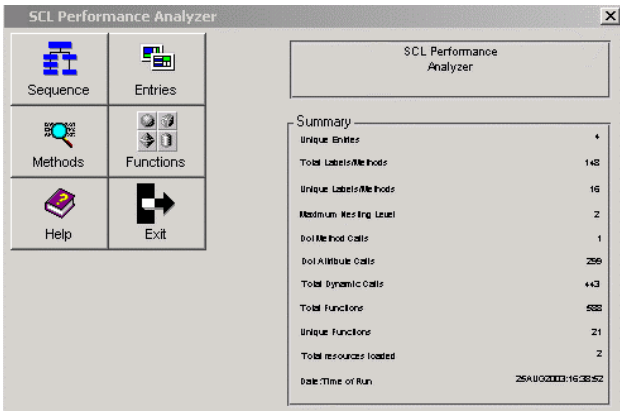
Cet outil nécessitent deux parties : la collecte puis la restitution des données.

Collecte des données
Vous avez la possibilité d'obtenir des informations sur l'ensemble d'une application ou sur seulement une partie d'une application.

Pour avoir les informations sur une application complète, vous devez ajouter la commande «**sclprof = timer**» au niveau de la commande de lancement de votre application.
Par exemple :
af c=libname.moncatal.monappli.frame sclprof = timer

Pour lancer l'exécution de l'outil «Performance Analyser» seulement sur une partie d'une application, vous devez saisir la commande suivante en ligne de commande : «**sclprof timer on**».
La commande «**sclprof timer off**» arrêtera la collecte de données.

Restitution des informations
Cette phase de restitution commence dès que la phase de collecte des données est terminée (soit par la fermeture de l'application soit par l'utilisation de la commande «**sclprof timer off**»).
Une fenêtre principale s'ouvre donnant des statistiques générales et permettant d'accéder à 4 niveaux d'informations :



Fenêtre 'SCL Calling Sequence' (bouton «Sequence») : cette fenêtre contient un graphique hiérarchique qui permet de visualiser les successions d'étiquettes et de méthodes exécutées durant l'analyse.
- Fenêtre 'Entry Statistics' (bouton « Entries ») : informations sur les entrées
- Fenêtre 'Method/Label Statistic' (bouton « Methods ») : information sur les méthodes
- fenêtre 'Functions Statistics' (bouton « Functions ») : information sur les fonctions

►Static Analyzer

Ce dernier outil d'aide au développement ou à la maintenance d'applications se décompose en une phase de la collecte de l'information puis en une phase de restitution des données.

Collecte des données
La collecte d'informations par le « Static Analyser » s'effectue en utilisant l'instruction **Crossref** dans une procédure **Proc Build**.

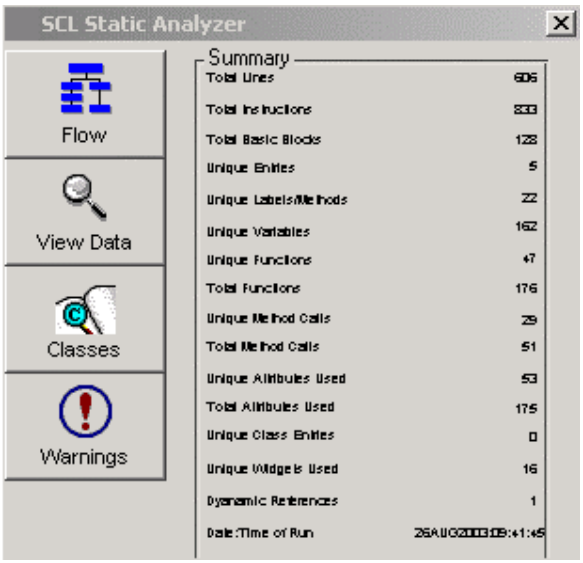
Par exemple:

```
proc build batch;
  crossref project=monlib.moncat
  outlib=malib;
run;
```

Dix tables SAS seront alors créées dans la bibliothèque **malib** contenant chacune des informations relatives à l'application.

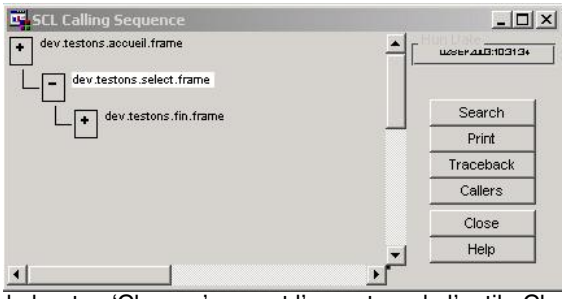
Restitution des données

Une fois les données collectées, il suffit de saisir la commande suivante en ligne de commande : «**sclprof static malib**» et la fenêtre suivante est générée.

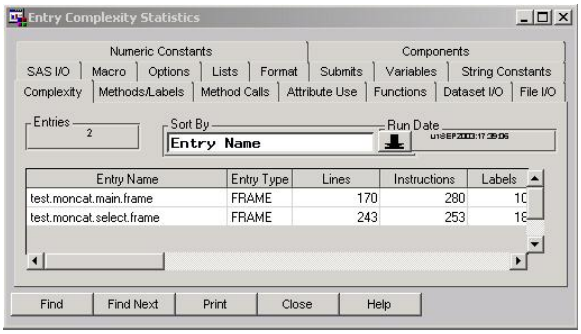


Cette fenêtre permet d'accéder à quatre domaines d'analyse :

- le bouton 'Flow' permet d'accéder à un graphique hiérarchique des entrées de catalogue appelées lors de la phase d'analyse. Un double-clic sur une entrée de catalogue permet de visualiser le code SCL.
- le bouton 'View Data' ouvre une fenêtre contenant quinze onglets qui fournissent des informations sur les fonctions, les variables, les méthodes, les entrées/sorties….



- le bouton 'Classes' permet l'ouverture de l'outil «Class browser» qui produit un affichage hiérarchique des classes.
- le bouton 'Warning' propose un listing des avertissements et erreurs générés en précisant l'entrée de catalogue en cause.



Tous ces outils doivent vous permettre d'optimiser votre code SCL. Pour toutes informations complémentaires sur les outils d'analyse, je vous invite à consulter l'aide en ligne du Système SAS : dans l'onglet « Sommaire » choisissez « Help on SAS Software Products » > « SAS/AF Software » > « SAS/AF development Tools ».

Cécile Guillouzo
Ingénieur Consultant

ASP et SAS

Depuis quelques années SAS propose des solutions pour diffuser de l'information sur le web, notamment avec les technologies CGI avec SAS/Intrnet et JAVA avec AppDev Studio.

Mais SAS n'est pas uniquement ouvert sur ces deux technologies, notamment depuis la version 8 et la venue de IOM (Integrated Object Model). En effet, grâce à cette technologie, la manipulation de table, l'exécution de code SAS et la restitution peuvent être réalisées par un programmeur connaissant les technologies Microsoft.

Une question revient de façon récurrente : comment peut–on se servir de la puissance de SAS et récupérer des résultats depuis une ASP (Active Server Page) ?

Cet article présente deux exemples de restitution d'une table SAS via une ASP Microsoft afin d'illustrer cette technologie :

- Utilisation d'une procédure SAS stockée(SAS Stored Process)
- Utilisation de la couche Microsoft ADO et OLEDB.

ASP est un standard Microsoft et ne peut être utilisé qu'avec des serveurs Web de Microsoft.

L'environnement de développement et d'exécution des exemples est le suivant:

- Windows 2000
- IIS 5
- 8.2

Qu'est-que l'IOM ?

IOM permet à SAS de s'interfacer avec les technologies COM, DCOM et CORBA.

Il peut également être exploité via différents langages de programmation (Visual Basic, Java, C++) par les développeurs pour utiliser facilement le système SAS sans avoir à connaître le langage de programmation SAS.

SAS IOM peut donc être utilisé à travers différents environnements de programmation :

- Suite Microsoft de Visual Basic (incluant Visual Basic for Applications et VBScript), Visual C++ et Visual J++;
- Environnements JAVA incluant IBM's Visual Age, Symantec's Visual Café, Borland's JBuilder et bien évidemment webAF, l'atelier de développement JAVA fourni avec la suite AppDev Studio;
- Des environnements de programmation propriétaire comme Borland's Delphi et Sybase's PowerBuilder

Pour de plus amples informations, vous pouvez vous rendre à l'adresse suivante :
http://support.sas.com/rnd/itech/doc/dist-obj/iom.html

Qu'est-ce qu'une ASP ?

Une page ASP (Active Server Page) est une page HTML créée dynamiquement par le serveur. Elle inclut des scripts qui seront exécutés lors de son appel. Cette technologie permet d'adapter l'information présentée sur la page en fonction de la requête faite par l'utilisateur.

Le script de la page ASP exécute la requête pour accéder à des sources de données stockées sur un serveur. L'ensemble de ces données est traité puis diffusé vers les postes clients.

Remarque :
Le module Integration Technologies doit être installé et sous licence pour l'exécution du premier exemple (SAS Stored Process).

1. Exemple utilisant un SAS Stored Process

Qu'est-ce qu'un SAS Stored Process (procédure SAS stockée)**?**

Une procédure SAS stockée est un programme SAS qui est exécuté par SAS via une requête. Il peut recevoir des paramètres de la requête qu'il pourra utiliser lors de son exécution dans SAS.
L'avantage de l'utilisation d'une procédure SAS stockée réside dans le fait que la modification du code SAS utilisé n'impacte pas le code de l'application Web en elle même. Seule la procédure SAS stockée est modifiée.

Le code ASP suivant (écrit en VBScript) montre l'appel à un SAS Stored Process. Le traitement exécute un simple code SAS ODS qui affiche une table et un graphique. Le résultat est renvoyé par le biais du flux de connexion vers le navigateur Web.

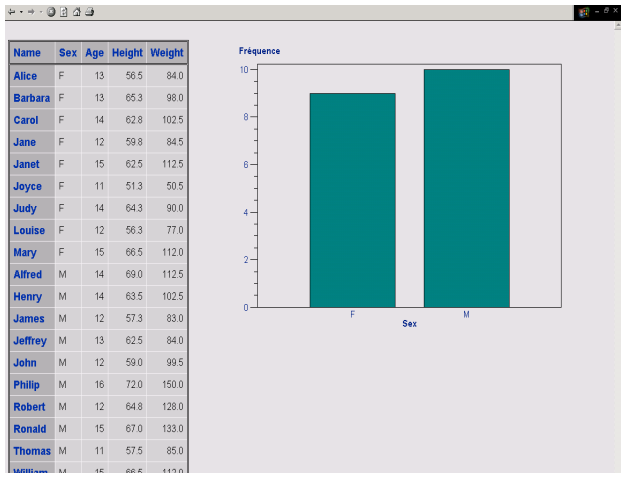
Code ASP :

```
<%

' Déclaration des variables
  Dim wsm, ws, wsStoredProcessService, wsFref, wsBinaryStream,
wsStreamHelper, strParams

  Dim VisibilityProcess
  VisibilityProcess = 1

' Déclaration d'une connexion vers une session SAS
  Set wsm = Server.CreateObject("SASWorkspaceManager.
WorkspaceManager")
  Set ws  = wsm.Workspaces.CreateWorkspaceByServer("server",
VisibilityProcess, nothing, "", "", "")

' Utilisation de l'objet LanguageService associé avec la
session SAS pour permettre  l'utilisation de l'objet Sto-
redProcessService et renseigner l'emplacement de la
procédure stockée
  set wsStoredProcessService = ws.LanguageService.
StoredProcessService
  wsStoredProcessService.Repository = "file:c:\temp\"

' Exécution d'une procédure stockée appelée simplehtml
sans paramètres
   wsStoredProcessService.Execute "simplehtml", ""

' Ecriture des résultats
   set wsFref = ws.FileService.UseFileref("b")
  set wsBinaryStream = wsFref.OpenBinaryStream(0)
  set wsStreamHelper = CreateObject("SASScripto.StreamHelper")

 response.binarywrite wsStreamHelper.ReadBinaryArray
(wsBinaryStream, 0)

' Fermeture des objets
  wsBinaryStream.Close
  wsm.Workspaces.RemoveWorkspace ws
  ws.Close

%>
```

Code SAS (Stored Process):

```
*ProcessBody;
proc template;
  define style styles.test;
    parent=styles.default;
      style body from body /
          pagebreakhtml=_undef_;
      style table from table /
          tagattr='align=left'
          just=left;
     end;
run;
filename b temp ;
ods html body=b (url="body.html" title="Utilisation de la Technolo-
gie ASP avec SAS") style=styles.test ;

proc print data=sashelp.class  ;
            id name ;
            title ;
run ;

ods html style=styles.default;
goption dev=activex ;
proc gchart data=sashelp.class ;
            vbar sex ;
run ;quit ;

ods html close ;
```

Résultat :



2. Exemple avec Microsoft ADO et standard OLE DB

Qu'est-ce ADO ?

Les composants ActiveX Data Objects (ADO) permettent d'établir une interaction entre une application Windows et des sources de données en s'appuyant sur la couche OLEDB. ADO est installé avec les composants MDAC, distribué indépendamment du serveur Web de Microsoft (IIS).
Vous pouvez trouver les composants MDAC en téléchargement sur le site de Microsoft :
http://msdn.microsoft.com/library/default.asp?url=/downloads/list/dataaccess.asp

Qu'est-ce que OLEDB ?

OLEDB est une technologie qui a pour but de résoudre certaines contraintes d'ODBC. Cette technologie autorise l'accès à tout type de données (sous réserve qu'un fournisseur OLEDB existe pour ces données), et prend en compte les contraintes du Web. Cette technologie a pour but à terme de remplacer la technologie ODBC. ADO est le modèle objet permettant de simplifier l'accès à cette technologie.

L'architecture IOM supporte Microsoft ADO et les standards OLE DB.
Notez, comme nous l'avons évoqué dans Allo Support n°10 (JDBC, JSP et SAS), que l'architecture IOM supporte également JDBC.
Le code suivant illustre l'utilisation du fournisseur OLEDB « SAS Local Data Provider » qui accède directement au fichier représentant une table SAS, récupère son contenu et le retourne dans un simple tableau HTML.
«SAS Local Data Provider» est disponible avec le module Base du Système SAS en environnement Windows. Ce «provider» permet d'accéder aux tables SAS en lecture seule.
Le code sera familier aux programmeurs connaissant ADO. Vous trouverez plus d'informations sur cette technologie à cette adresse:
http://support.sas.com/rnd/eai/oledb/index.htm

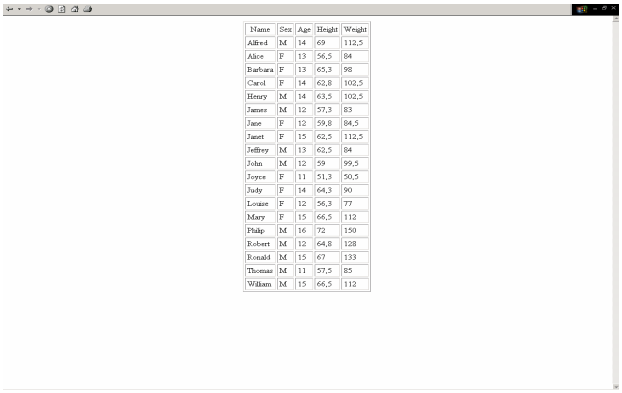
Voici les étapes que nous allons suivre dans cet exemple :
* Création d'une connexion ADO vers la base de données
* Ouverture de la connexion à la base de données
* Création d'un ADO Recordset (vecteur d'enregistrements) utilisé pour accéder aux enregistrements de la table

* Ouverture du Recordset
* Extraction des données dont nous avons besoin depuis le Recordset
* Fermeture du Recordset
* Fermeture de la connexion

Code ASP :

```
<%

Const adLockReadOnly = 1
'Type de lock sur les enregistrements dans le Recordset
Const adCmdTableDirect = 512
'Méthode d'extraction des données par le RecordSet

'Création d'une connexion ADO vers la base de données spécification du « provider » utilisé (Voir http://support.sas.com/rnd/eai/oledb/gs_best_prov.htm)
'Spécification du chemin où les données sont stockées
DIM obConnection
Set obConnection = Server.CreateObject("ADODB.Connection")
obConnection.Provider = "sas.LocalProvider.1"
obConnection.Properties("Data Source") = "C:\Program Files\SAS Institute\SAS\V8\core\sashelp"

'Ouverture de  la connexion de la base de données
obConnection.Open

' Création d'un ADO Recordset utilisé pour accéder aux enregistrements de la table
Set obRecordset = Server.CreateObject("ADODB.Recordset")

'Ouverture du Recordset
obRecordset.Open "CLASS", obConnection,  , adLockReadOnly, adCmdTableDirect

'Construction du tableau HTML en bouclant sur les enregistrements du Recordset
<table border=1 cellpadding=3 cellspacing=3 align=center>
<tr>
<% For Each column In obRecordset.Fields %>
<td align=center><%= column.Name %></td>
<% Next %>
</tr>
<% Do While Not obRecordset.EOF %>
<tr>
<% For Each column In obRecordset.Fields %>
<td><%= column.Value %></td>
<% Next %>
</tr>
<%
obRecordset.MoveNext
Loop
%>
</table>

'Fermeture de la connexion
obRecordset.Close
set obRecordSet = Nothing

'Fermeture de la connexion
obConnection.Close
%>
```

Résultat :



Name	Sex	Age	Height	Weight
Alfred	M	14	69	112,5
Alice	F	13	56,5	84
Barbara	F	13	65,3	98
Carol	F	14	62,8	102,5
Henry	M	14	63,5	102,5
James	M	12	57,3	83
Jane	F	12	59,8	84,5
Janet	F	15	62,5	112,5
Jeffrey	M	13	62,5	84
John	M	12	59	99,5
Joyce	F	11	51,3	50,5
Judy	F	14	64,3	90
Louise	F	12	56,3	77
Mary	F	15	66,5	112
Philip	M	16	72	150
Robert	M	12	64,8	128
Ronald	M	15	67	133
Thomas	M	11	57,5	85
William	M	15	66,5	112

Les techniques évoquées présentent donc un grand avantage, puisqu'un programmeur ne connaissant pas forcément le fonctionnement du Système SAS peut tout à fait exploiter la puissance du Système SAS et obtenir des restitutions Web facilement.
Les serveurs Web Microsoft étant très répandus, les ASP peuvent être très intéressantes à utiliser.
Les entreprises se retrouvent souvent confrontées à des choix de technologies concernant des applications Web (JSP/Servlet ou ASP). Le Système SAS s'interfa-ce avec ces différentes technologies et laisse donc le choix de la technologie à utiliser.

Informations complémentaires :

Quelques exemples :
http://support.sas.com/rnd/eai/samples/index.html

Documentation SAS Integration Technologies 8.2 :
http://support.sas.com/rnd/itech/library/library82.html

Référence ADO/OLEDB :
http://support.sas.com/rnd/eai/oledb/index.htm

Fabrice Pollet
Ingénieur Consultant

Utilisation de la mémoire au niveau du système SAS sous Unix

Lors de l'installation, le système SAS est paramétré par rapport à un fonctionnement standard. Mais nous sommes amenés à travailler sur des volumes de plus en plus importants de données. Dans ce cas, l'utilisation de certaines procédures très consommatrices en terme de ressources mémoire peut produire des erreurs du type :

ERROR: Out of Memory.

Ceci nous oblige à paramétrer SAS d'une façon appropriée au travail effectué.

Les principaux paramètres liés à l'utilisation de la mémoire sur SAS sont les suivants :

Memsize
Fullstimer
Sortsize
Ubufsize
Bufno

1. Memsize représente la mémoire allouée au processus SAS au moment du démarrage de la session. Par défaut ce paramètre est positionné à 64Mo et peut prendre jusqu'à la valeur de 2Go (limite système).
Il est possible aussi de spécifier une valeur nulle et dans ce cas SAS va allouer au fur et à mesure des traitements la mémoire qui lui est nécessaire. Ce dernier cas est recommandé uniquement pour les besoins d'un test, pour un utilisateur donné. L'emploi d'une telle option en production et pour l'ensemble des utilisateurs peut en effet provoquer une consommation trop importante de mémoire. Donc, parfois un message «out of memory» est préférable à une défaillance générale du serveur !

Le paramètre **memsize** doit être en relation avec le paramètre système (mémoire par processus) que l'on peut visualiser avec la commande Unix : ulimit –d

Autrement dit, même si le paramètre **memsize** a une valeur supérieure au paramètre système, le processus SAS ne pourra jamais dépasser en taille mémoire la valeur donnée par le système.

Exemple:

```
# ulimit -a
time(seconds)        unlimited
file(blocks)         unlimited
data(kbytes)         65536
stack(kbytes)        8192
memory(kbytes)       unlimited
coredump(blocks)     4194303
nofiles(descriptors) 60
```

Le paramètre qui nous intéresse est **data** (kbytes). Par défaut sur une machine Unix, sa valeur est 64Mo (d'où le défaut de **memsize)**.

Il est difficile parfois de choisir la valeur optimum pour **memsize**. Dans ce cas l'option «fullstimer» peut aider à ajuster cette valeur.

2. Fullstimer permet d'afficher dans la log après chaque procédure la mémoire consommée par le traitement. En cas d'interruption incorrecte du traitement elle donne la taille mémoire maximum qui a pu être allouée.

Exemple :

```
options fullstimer ;
3   proc mddb data=sashelp.prdsale out = work.cube ;
4          class product month region quarter division year coun-
try ;
6          hierarchy year quarter month product ;
7          hierarchy year quarter month ;
8          hierarchy year quarter product ;
9          hierarchy year quarter ;
10         hierarchy year product ;
11         hierarchy country region division ;
12         hierarchy country region  ;
13         hierarchy country ;
14         var actual predict / sum ;
15   run ;
NOTE:  SAS/MDDB(R) Server Software has been initialized.
NOTE: N-way complete  cells=1440.
NOTE: "HIER1" computed from "NWAY"  cells=120.
NOTE: "HIER2" computed from "HIER1"  cells=24.
NOTE: "HIER3" computed from "HIER1"  cells=40.
NOTE: "HIER4" computed from "HIER2"  cells=8.
NOTE: "HIER5" computed from "HIER3"  cells=10.
NOTE: "HIER6" computed from "NWAY"  cells=12.
NOTE: "HIER7" computed from "HIER6"  cells=6.
NOTE: "HIER8" computed from "HIER7"  cells=3.
NOTE: PROCEDURE MDDB used:
     real time          0.55 seconds
     user cpu time      0.05 seconds
     system cpu time    0.01 seconds
     Memory                    1289k
     Page Faults               61
     Page Reclaims               0
     Page Swaps                  0
     Voluntary Context Switches        118
     Involuntary Context Switches      3
     Block Input Operations          46
     Block Output Operations          19
```

Dans ce cas nous observons que la création du cube (Proc MDDB) a utilisé 1,289 Mo de mémoire sans faire appel à de la mémoire paginée (swap).

3. Sortsize représente la mémoire nécessaire à la procédure Sort. Cette valeur est par défaut positionné à 48Mo et doit être inférieure d'au moins 8Mo à la valeur de **memsize.**
Si la valeur de **sortsize** est insuffisante, le système SAS va créer des fichiers temporaires sur le disque dur. Du fait d'un nombre de lectures/écritures sur le disque plus important, le temps de traitement augmente également. A l'inverse une valeur trop grande peut amener une utilisation excessive de la mémoire swap (qui consiste aussi à lire/écrire sur le disque dur).

4. Ubufsize représente la taille du tampon utilisé par la Proc Sort pour écrire les fichiers temporaires sur le disque dur (dans la Work). **Ubufsize** peut prendre les valeurs suivantes : 8K (valeur par défaut), 16K, 24K et 32K. La taille des fichiers temporaires peut atteindre trois fois la taille de la table à trier. Si les opérations de lectures/écritures sur votre disque dur sont lentes nous vous recommandons la valeur 32K.

5. Bufno représente le nombre de tampons utilisés par SAS pour les opérations de lecture /écriture sur le disque dur.
Par défaut la valeur de **bufno** est égale à 1 et il est recommandé de ne pas dépasser une valeur égale à 10. Le positionnement d'une valeur trop élevée peut provoquer un dépassement de la mémoire réelle et diminuer les performances des traitements du fait de la pagination
S'il est intéressant de choisir une valeur optimum pour **bufno**, il est conseillé de modifier celle de **bufsize** sur un traitement ponctuel (option **Dataset**). Le paramètre **bufsize** représente la taille d'un tampon alloué pour les opérations de lecture/écriture et sa valeur est calculée dynamiquement par SAS en fonction de la longueur de l'observation de la table.
Si ce paramètre est déclaré dans le fichier **sasv8.cfg**, sa valeur est utilisée pour toutes les lectures/écritures de table, et vous risquez de diminuer les performances des traitements dans la plupart des cas.

6. Conclusion
Dans le cas où de façon ponctuelle vous souhaitez modifier certains paramètres mémoire, il est utile de créer un fichier de configuration **sasv8.cfg** autre que celui qui se trouve dans la SASROOT et de lancer le traitement en question avec l'option **–config /chemin/sasv8.cfg** .
Pour les procédures les plus consommatrices en terme de mémoire une formule vous permet de calculer la taille mémoire requise pour un traitement donné. Vous trouverez cette formule en consultant la «Doc OnLine» dans la partie «Computational Resources» de chaque procédure.

Catalin ALDEA
Ingénieur Consultant

Directeurs de la publication :
Philippe Hoffmann
Pierre Lorenzetti

Comité de rédaction :
Blandine Colas
Catalin Aldéa
Cécile Guillouzo
Damien Michot
Fabrice Pollet
Julien Fages
Karine Chrillesen
Pascal Nicolas
Ludovic Breysse

Comité de relecture :
Fabienne Bernard
Mouloud Dey
Philippe Hoffmann

Mise en page :
Service reproduction
SAS France

SAS France
Domaine de Grégy - BP 5
77166 Grégy-sur-Yerres
Tél. : 01 60 62 11 11
Fax : 01 60 62 11 99

SAS Europe, Middle East & Africa
P.O. Box 10 53 40
Neuenheimer Landstr. 28-30
D-69043 Heidelberg, Germany
Tel: +49 6221 4160, Fax: +49 6221 474850

SAS est une marque déposée de SAS Institute Inc., Cary NC, USA.

www.sas.com/france