



> Numéro 22

> ALLO SUPPORT



> Dans ce numéro

Cliquez sur un article du sommaire pour vous y rendre directement... et revenez au sommaire à tout moment !

> A la découverte du symbole « : »	> 1
> Les procédures IMPORT et EXPORT	> 3
> Interaction entre SAS et le système	> 7
> Migration de SAS® Enterprise Miner™ 5.1 vers 5.2	> 8
> La procédure METAOPERATE	> 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'en informer Fabienne Vermast : frafav@fra.sas.com.

A la découverte du symbole « : »

Dans la catégorie "Trucs et Astuces", les deux-points peuvent être utilisés dans différents contextes en Base SAS® :

- pour spécifier un format dans l'instruction PUT,
- pour spécifier un informat dans l'instruction INPUT,
- pour désigner des variables ou des valeurs en donnant juste un préfixe,
- pour comparer des valeurs à des préfixes.

Cet article propose quelques illustrations. Les exemples sont construits sur la table WORK.EXEMPLE, contenant une variable NOM, une variable formatée DATE, et une variable par année de 1995 à 2000.

```
Data Exemple ;
  length Nom $10. ;
  attrib date format=ddmmyy10. informat=ddmmyy6. length=5;
  input Nom $ date an95 an96 an97 an98 an99 an00 ;
  cards ;
  Arnold 010190 15 9 16 10 12 8
  Barnabé 010290 36 38 40 37 38 39
  Cerise 010892 5 10 15 18 14 16
  Didier 010992 5 9 10 8 7 3 5
  Eglantine 011292 45 85 69 21 26 56
  Fanny 010794 25 26 27 28 29 25
  ;
run ;
```

1. Instruction PUT

L'instruction PUT permet d'écrire une ligne de sortie, celle-ci s'inscrivant :

- dans un fichier si une instruction FILE définit un fichier en sortie,
- sinon dans la LOG.

Cette ligne contient du texte et/ou des variables formatées ou non formatées.

Lorsqu'une variable a déjà un format défini au niveau de l'étape DATA, il n'est pas nécessaire de le préciser dans l'instruction PUT. De plus, avec cette syntaxe, dite "listée", un blanc est inséré entre chaque champ.

Exemple :

```
put nom date an95 an96 an97 ;
```

Extrait du résultat :

```
Arnold 01/01/1990 15 9 16
Barnabé 01/02/1990 36 38 40
Cerise 01/08/1992 5 10 15
```

Seule la variable DATE a un format. Dans le résultat, son format a bien été appliqué, et un espace sépare chaque champ.

Il est possible d'ajouter un format pour chaque variable au sein de l'instruction PUT. Il est valable pour cette instruction, il ne devient pas le format attribué de la variable. Pour les variables ayant déjà un format défini au niveau de l'étape data, celui indiqué dans l'instruction le remplace uniquement pour cette écriture.



Il y a 2 notations possibles :

- sans les deux points : la sortie est dite “formatée”. Dans ce cas, la longueur indiquée sera totalement inscrite quelle que soit la longueur réelle de la valeur, et le champ suivant sera inscrit directement à la suite.

Exemple :

```
put nom $10. date date7. an95 z2. an96 z2. an97 z2.;
```

Extrait du résultat :

```
Arnold 01JAN90150916
Barnabé 01FEB90363840
Cerise 01AUG92051015
```

Ici, le nom est systématiquement inscrit sur 10 caractères, puis la date est notée à la 11ème position, avec un autre format que le sien. Celle-ci ayant un format sur 7 positions, la variable suivante s'inscrit directement à la 18ème position.

- avec les deux points : la sortie est dite “listée modifiée”. Dans ce cas, chaque valeur est écrite, et un espace sépare chaque champ.

Exemple :

```
put nom : $10. date : date7. an95 : z2. an96 : z2. an97 : z2.;
```

Extrait du résultat :

```
Arnold 01JAN90 15 09 16
Barnabé 01FEB90 36 38 40
Cerise 01AUG92 05 10 15
```

Chaque valeur est écrite avec le format donné, et séparé d'un blanc.

Si vous souhaitez des précisions sur cette syntaxe, ou connaître les autres options de l'instruction PUT, tapez “Statements : PUT Statement, List” (avec les guillemets) dans la zone de recherche de l'aide en ligne.

2. Instruction INPUT

L'instruction INPUT permet de lire des données, issues d'un fichier (instruction INFILE) ou listées dans l'étape DATA (instruction CARDS).

Le signe « : » permet de spécifier un informat directement au niveau de l'instruction INPUT.

L'exemple ci-dessous présente deux syntaxes équivalentes. L'instruction ATTRIB définit l'informat de la variable DATE :

```
attrib date format=ddmmyy10. informat=ddmmyy6.
length=5;
input Nom $ date an95 an96 an97 an98 an99 an00 ;
```

Ce qui équivaut à :

```
attrib date format=ddmmyy10. length=5;
input Nom $ date : ddmmyy6. an95 an96 an97 an98
an99 an00 ;
```

3. Désigner un groupe de variables

Les deux-points permettent de désigner un groupe de variables, en ne nommant qu'un préfixe commun.

Exemple 1 : retirer / garder un ensemble de variables.

Lors de la lecture des données en entrée, ce programme garde les variables correspondant aux années 90. Il retire de la table en sortie toutes les variables de type “compteur” (_i et _j dans notre cas).

```
data work.ex1 (drop = _: );
set work.Exemple (keep=nom an9:);
do _i=1 to 10 ;
do _j=20 to 100 by 2 ;
/* ... plus d'instructions .... */
end ;
end ;
run ;
```

Exemple 2 : faire une somme.

Grâce au deux-points, il n'est pas nécessaire de nommer les variables une à une pour faire la somme de plusieurs variables.

```
data work.ex2 ;
set work.Exemple ;
Total=sum(of an: ) ;
run ;
```

4. Désigner un groupe de tables

La procédure DATASETS accepte également ce signe. Ainsi, le programme ci-dessous supprime de la bibliothèque WORK toutes les tables dont le nom commence par EX.

```
proc datasets lib=work ;
delete ex: / memtype=data ;
quit ;
```

5. Comparer des valeurs caractères

Les chaînes de caractères peuvent être comparées à un préfixe grâce au signe « : » qui peut être ajouté à n'importe quel opérateur de comparaison, tel que = ou EQ par exemple.

Ci-dessous, les noms sont catégorisés par ordre alphabétique via des fourchettes de valeurs. De plus, les noms commençant par S, T ou Z ne sont pas conservés.

```
data work.ex3 ;
set work.Exemple ;
select ;
when ('A' =<: nom <: 'F') ordre=1;
when ('F' =<: nom <: 'K') ordre=2;
when ('K' =<: nom <: 'O') ordre=3;
otherwise ordre=3 ;
end ;

If nom not in: ('S', 'T', 'Z');
run ;
```

L'utilisation des deux-points est généralisée dans le L4G SAS, aussi d'autres applications sont possibles. Si vous souhaitez en savoir plus, vous pouvez taper “colon” dans le champ de recherche de l'aide en ligne.

Géraldine Deschamps

[RETOUR SOMMAIRE](#)



Les procédures IMPORT et EXPORT

Avec le Service Pack 3 de la version SAS® 9.1.3 quelques nouvelles options ont été implémentées au niveau des procédures IMPORT et EXPORT.

L'objet de cet article est de présenter ces nouveautés mais aussi le fonctionnement de ces procédures, plus particulièrement lors de l'interaction avec des fichiers Excel. En effet les questions qui nous sont posées au Support Clients, portent très souvent sur ce type de fichiers (en tous cas plus fréquemment que sur les autres types de fichiers accessibles via ces deux procédures).

Après un rappel de l'objet de ces deux procédures, nous allons voir leurs deux modes d'utilisation (via du code ou via un assistant). Ensuite, nous verrons dans le détail plusieurs options utiles lors de l'importation de fichiers, ainsi que lors de l'exportation. Un dernier point sera fait sur une nouveauté intéressante en environnement Unix.

1. Introduction sur les procédures import et export

Les procédures import et export présentent la particularité d'appartenir au module Base SAS®, mais, pour bénéficier de certaines de ses fonctionnalités, une licence du module SAS/Access® to PC Files Format est nécessaire.

Ainsi, avec uniquement une licence pour le module Base SAS®, il est possible d'importer et d'exporter des fichiers :

- de type texte, que les champs soient séparés par un délimiteur donné ou par une tabulation par exemple
- au format JMP
- ou au format SPSS (.sav)

Le module SAS/Access® to PC Files Format élargit la liste des types de fichiers disponibles, à Excel, Lotus, MS Access et DBase.

Ces informations sont valides pour les versions 8 et 9 de SAS.

2. Les deux modes d'utilisation de ces procédures

Les procédures sont utilisables soit via programmation soit via un assistant.

La programmation est aisée. Au minimum, le fichier à lire - ou à créer - et la table à lire - ou à créer - sont requis.

Exemple :

```
proc import datafile='c:\stockage\supplyinfo.xls' out=supplyinfo;
run;
```

- SAS reconnaît le type de fichier à lire, Excel ici en l'occurrence, en se basant sur l'extension (xls).

```
proc export data=sashelp.class outfile='c:\stockage\class.csv';
run;
```

- Un fichier de type texte est créé. Le délimiteur utilisé est la virgule (délimiteur par défaut pour les fichiers d'extension csv).

Les assistants permettent de s'affranchir de la connaissance de la syntaxe des deux procédures. Ils sont accessibles via les menus SAS : Fichier > Importer données... (File > Import Data...) et Fichier > Exporter données... (File > Export Data...). La dernière étape permet de créer un programme SAS contenant la syntaxe générée par l'assistant.

Il est également intéressant de noter que dans le cas de traitement de fichiers de type texte, SAS génère automatiquement du code, une étape data, à partir de la procédure IMPORT ou EXPORT. Pour s'en rendre compte, il suffit d'étudier le journal. Ce code peut être rappelé dans la fenêtre éditeur en utilisant la touche F4, pour être personnalisé si besoin, au niveau des instructions file/infile, informat et format, input/put.

3. L'importation de fichiers externes

Comme indiqué précédemment, le cas de fichiers Excel est traité plus spécifiquement ici. Nous allons détailler ici plusieurs options et instructions intéressantes de la procédure IMPORT.

L'option REPLACE.

Cette option permet d'autoriser le remplacement d'une table existante.

Exemple :

```
proc import datafile='c:\stockage\supplyinfo.xls' out=suppliers
replace;
run;
```

Information complémentaire :

- si cette option n'est pas spécifiée et que vous exécutez deux fois la procédure, vous recevrez ce message d'erreur : ERROR: Import cancelled. The dataset SUPPLIERS is being used and cannot be replaced.

L'instruction SHEET.

Elle permet d'indiquer le nom de la feuille à lire.

Exemple :

```
proc import datafile='c:\stockage\supplyinfo.xls' out=suppliers
replace;
sheet='suppliers';
run;
```



Informations complémentaires :

- Si cette option n'est pas spécifiée, c'est la première par ordre alphabétique qui est traitée (avec les versions les plus récentes d'Excel).
- La longueur maximale d'une feuille est de 31 caractères (32 avec le signe \$).
- Si la feuille Excel contient une apostrophe, il faut suivre une nomenclature particulière. Ainsi une feuille nommée "Données de l'entreprise" sera lue avec :
`sheet='Données de l'entreprise$'n;`

L'instruction GETNAMES

Elle permet d'indiquer à SAS si les noms des variables sont disponibles dans le fichier à lire.

Si le fichier Excel contient effectivement ces noms des variables en première ligne, ils seront automatiquement utilisés dans la table SAS créée. Si le fichier Excel n'en contient pas, il faut le préciser en utilisant l'instruction `getnames=no`. Les noms des variables seront alors par défaut F1, F2, etc.

```
proc import datafile='c:\stockage\supplyinfo.xls' out=suppliers
replace;
sheet='suppliers';
getnames=no;
run;
```

Information complémentaire :

- Par défaut, l'option `getnames` est positionnée à `yes`.

L'instruction DBSASLABEL. Nouveauté SP3

Par défaut, le libellé de chaque variable est identique au nom de la variable SAS.

L'instruction `DBSASLABEL` permet de modifier ce comportement par défaut. En la positionnant à `NONE`, les libellés des variables seront 'à blanc'.

Exemple :

```
proc import datafile='c:\stockage\supplyinfo.xls' out=suppliers
replace;
dbsaslabel=none;
sheet='suppliers';
run;
```

Information complémentaire :

- La valeur par défaut de cette option est `COMPAT`.

L'option DBMS

Elle permet de spécifier la version du fichier Excel à lire. La valeur `EXCEL` par exemple, permet de lire toutes les versions d'Excel, de la 4.0 à 2002. Globalement, il faut juste prêter attention à ne plus utiliser les valeurs `EXCEL4` et `EXCEL5`, qui ne permettent pas de lire les fichiers Excel dont la version est plus récente.

		Versions d'EXCEL				
		Excel 2002	Excel 2000	Excel 97	Excel 5.0	Excel 4.0

Valeur de l'option DBMS=	XLS	oui	oui	oui	oui	non
	EXCEL	oui	oui	oui	oui	oui
	EXCEL2002	oui	oui	oui	oui	oui
	EXCEL2000	oui	oui	oui	oui	oui
	EXCEL97	oui	oui	oui	oui	oui
	EXCEL5	non	non	non	oui	oui
	EXCEL4	non	non	non	oui	oui

Informations complémentaires :

• Il existe une dernière valeur possible : `DBMS=EXCELCS`. Il s'agit de la seule valeur valide en environnement Unix. Les valeurs présentées dans le tableau ci-dessus fonctionnent exclusivement sur Windows.

• Il faut savoir qu'en fonction de la valeur de `DBMS` choisie, les options disponibles dans la procédure `import` sont différentes.

Nous allons maintenant illustrer l'utilisation de cette option en prenant un exemple de traitement de colonne d'un fichier Excel contenant à la fois des valeurs numériques et alphanumériques.

Voici un exemple de fichier Excel, qui a la particularité de contenir à la fois des valeurs numériques et alphanumériques.

	A
1	val1
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	a
12	b
13	10
14	11
15	

Avec la procédure `IMPORT` et l'option `DBMS=EXCEL`, par défaut, seules les valeurs numériques sont restituées.

	val1
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	
12	10
13	11

Avec l'option `DBMS=XLS`, même les valeurs alphanumériques sont prises en compte.

	val1
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	a
11	b
12	10
13	11



En effet, le mode d'accès à Excel est complètement différent selon la valeur de DBMS utilisée.

Avec DBMS=EXCEL, par défaut, seules les huit premières lignes sont scannées et utilisées pour déterminer le type de la variable par l'API SAS. Ici, SAS décidera donc qu'il s'agit d'une variable numérique. Les valeurs a et b sont donc incorrectes et restituées comme manquantes.

Le paramètre TypeGuessRows peut être modifié dans la base de registre Windows, afin de scanner plus de lignes et ainsi déterminer avec exactitude le type de la variable (alphanumérique et non numérique).

Les instructions pour lire une sélection du fichier Excel ? Nouveauté SP3

En fonction du type de fichier Excel choisi, la syntaxe est différente. Il s'agira soit de l'instruction RANGE, soit des instructions STARTROW, ENDROW, STARTCOL et ENDCOL.

1. Avec dbms=excel, nous utilisons l'instruction RANGE.

Exemple :

```
proc import datafile= 'c:\stockage\supplyinfo.xls' out=supply
replace dbms=excel;
range='suppliers$A1:C4';
run;
```

Informations complémentaires :

- Le nom de la feuille doit être spécifiée au niveau de l'option range. Si une option sheet est utilisée, elle sera ignorée. Il faut donc impérativement la renseigner comme ci-dessus (ou ce sera la 1^{ère} par ordre alphabétique qui sera traitée).
- Il faut que les noms des variables soient présents dans la plage de données spécifiée (en première ligne de cette plage) ou alors spécifier l'option getnames=no. Dans le cas contraire, SAS utilisera les données de la 1^{ère} ligne comme noms de variables. Prenons un exemple pour illustrer ces propos.

Considérons cet extrait de fichier Excel :

	A	B	C
1	SupplierID	CompanyName	ContactName
2	1	Exotic	Charlotte Cooper
3	2	Cajun	Shelley Burke
4	3	Kellys	Regina Murphy

Avec le code utilisé ci-dessus, donc avec l'option range définie ainsi

```
range='suppliers$A1:C4';
```

le résultat obtenu dans SAS est

	SupplierID	CompanyName	ContactName	ContactTitle
1	1	Exotic	Charlotte Cooper	Purchasing Manager
2	2	Cajun	Shelley Burke	Order Administrator
3	3	Kellys	Regina Murphy	Sales Representative

Si la ligne contenant les noms de variables n'est pas sélectionnée :

```
range='suppliers$A2:D4';
```

le résultat n'est pas celui attendu :

	F1	Exotic	Charlotte Cooper	Purchasing Manager
1	2	Cajun	Shelley Burke	Order Administrator
2	3	Kellys	Regina Murphy	Sales Representative

L'ajout de l'option getnames, positionnée à No :

```
range='suppliers$A2:D4';
```

```
getnames=no;
```

permet d'avoir toutes les données correctement dans la table SAS :

	F1	F2	F3	F4
1	1	Exotic	Charlotte Cooper	Purchasing Manager
2	2	Cajun	Shelley Burke	Order Administrator
3	3	Kellys	Regina Murphy	Sales Representative

2. Avec dbms=xls, nous utilisons des instructions STARTROW, ENDROW, STARTCOL et ENDCOL.

Ces instructions (disponibles uniquement avec le service pack 3) permettent de limiter la plage de données à lire, en choisissant soit une sélection de lignes, soit une sélection de colonnes, ou encore en associant ces deux types de sélections.

Exemple :

```
proc import datafile= 'c:\stockage\supplyinfo.xls' out=test
replace dbms=xls;
sheet='suppliers';
startrow=5;
endrow=10;
startcol=A;
endcol=C;
run;
```

Informations complémentaires :

- Au niveau des options startrow et endrow, des numéros de lignes sont attendus.
- Pour les options startcol et endcol, ce sont les noms des colonnes Excel (A, B, C, etc).
- Contrairement au cas précédent (lors de l'utilisation de l'option range), la ligne contenant le nom des variables ne doit pas systématiquement se trouver dans la plage définie. Les noms de variables seront recherchés dans la 1^{ère} ligne du fichier Excel. S'ils sont sur une autre ligne, l'option namerow permet de spécifier laquelle.



L'exportation de tables SAS

1. L'instruction SHEET Nouveauté SAS®9

Depuis la version SAS®9 (auparavant, l'instruction sheet n'existait pas), il est possible de créer plusieurs feuilles au sein du même classeur Excel.

Il faut exécuter autant de procédures export que le nombre de feuilles à créer.

La table à exporter est à spécifier au niveau de l'option data, et le nom de la feuille à créer au niveau de l'instruction sheet.

Exemple :

```
proc export data=don_janvier outfile='c:\stockage\donnees.xls' ;  
sheet='Rapport Janvier';  
run;
```

```
proc export data=don_fevrier outfile='c:\stockage\donnees.xls' ;  
sheet='Rapport Février';  
run;
```



Informations complémentaires :

- Si un espace ou un tiret est présent dans le nom de la feuille, il est remplacé automatiquement par le caractère _.
- Pour remplacer une feuille déjà existante, l'ajout de l'option replace est nécessaire.

2. L'exportation des libellés des variables Nouveauté SAS®9

En fait, la procédure Export ne permet pas d'exporter les libellés des variables SAS. C'est une question qui nous est souvent posée. Il semble donc intéressant de sortir du cadre de cet article, consacré aux procédures Import et Export, pour vous signaler que c'est néanmoins possible. Cette solution utilise toujours le module SAS/Access® to PC Files Format, via une instruction libname. L'idée consiste à créer une bibliothèque définie sur un fichier Excel et l'option dlabel.

Exemple :

```
libname expexcel excel 'c:\stockage\labels.xls';  
data expexcel.donnees (DBLABEL=YES);  
set matable;  
run;
```

Informations complémentaires :

- Cette fonctionnalité sera ajoutée au niveau de la procédure export dans la version SAS® 9.2.

- Pour pouvoir ouvrir le fichier Excel, il faut au préalable désallouer la bibliothèque (`libname expexcel clear;`).

Spécificité et nouveauté au niveau des procédures import et export en environnement UNIX. Nouveauté SP3

Lors de l'utilisation de ces procédures sur Unix, un composant était indispensable : le "PC Files server" qui devait être installé sur un PC, pour faire le lien entre le serveur Unix et le monde Windows. Ce n'est plus le cas à compter de SAS® 9.1.3 SP3, à la condition de disposer d'un accès aux fichiers Windows depuis le serveur Unix. Cet accès peut être mis en place en utilisant un montage de disque réseau (utilisation du logiciel SAMBA par exemple).

Attention, cette nouvelle fonctionnalité n'est pour l'instant disponible que pour les procédures import et export. L'instruction libname n'est donc pas concernée.

La documentation sur les nouveautés apportées au niveau des procédures import et export dans le service pack 3 est accessible uniquement dans ce document :

<http://support.sas.com/techsup/technote/ts745.pdf> et sera accessible dans la documentation officielle de SAS à compter du service pack 4.

Karine Chrillesen



Interaction entre SAS et le système

A partir de SAS, il est possible d'interagir avec le système d'exploitation. Cet article présente un certain nombre d'instructions SAS possibles pour exécuter des commandes système. Tous les exemples ont été exécutés sur un environnement Windows, mais peuvent être adaptés pour Unix.

Filename pipe

SAS dispose de plusieurs types de filename, dont le filename "pipe". Un filename permet de lire le contenu d'une ressource, que ce soit un fichier, un catalogue SAS, un élément d'un serveur ftp/http ou autre. L'option nommée "pipe" permet d'accéder en temps réel au résultat de l'exécution d'une commande système.

Prenons un exemple simple qui liste le contenu d'un répertoire. L'option "pipe" va nous permettre de gérer la liste de fichiers aisément.

```
filename liste pipe 'dir /b "C:\TEST" ;
data liste;
  length fichier $132;
  infile liste truncover;
  input fichier 1-132;
run;
ods html ;
proc print;run;
ods html close ;
```

Obs	fichier
1	class sas7bdat
2	cours_stp.zip
3	exemple.html
4	prdsal2 sas7bdat
5	prdsal3 sas7bdat
6	prdsale sas7boot
7	sas.gif
8	seaint.sss

Ce filename peut être utilisé pour réaliser une copie de fichier, et valider son exécution.

```
filename copy pipe 'copy c:\test\sas.gif c:\test\sas2.gif ;
data liste;
  length rc $132;
  infile copy truncover;
  input rc 1-132;
  ok="copi" ; /* test sur le mot copi */
  if index(rc,ok)>1 then do ;
    put "NOTE: Le fichier a été copié" ;
  end ;
else do ;
  put "ERROR: Problème lors de la copie." ;
  erreur= "ERROR: "||rc ;
  put erreur ;
end ;
run;
```

D'autres exemples sont disponibles sur le site de SAS sur l'utilisation de ce filename.

Les fonctions de base SAS

Plusieurs fonctions SAS peuvent être utilisées pour interagir avec le système d'exploitation. En voici quelques-unes.

sysget :

La fonction sysget est utile pour récupérer la valeur d'une variable d'environnement de la machine. Par exemple, la syntaxe suivante permet de récupérer l'emplacement du répertoire temporaire du système :

```
data _null_;
  TMP=sysget("TEMP");
run ;
```

sysmask :

Cette instruction est utilisée dans le cadre du mode client-serveur (plus particulièrement MpConnect), mais nous pouvons aussi nous en servir pour lancer l'exécution de commandes système. Il faut noter que SYSTASK soumet les commandes systèmes indépendamment de SAS : la commande n'a aucune relation avec SAS, et s'exécutera en parallèle de la session SAS où elle a été soumise.

```
SYSTASK COMMAND "xcopy c:\test c:\test\temp /s /e /i /r /q";
```

Vous pouvez lister les commandes SYSTASK soumises en utilisant la fonction LISTTASK.

```
LISTTASK _ALL_ ;
```

system** :

Dans une étape data, elle permet d'exécuter une commande système, comme le permet le filename pipe. Néanmoins, elle ne récupère pas le résultat complet de la commande mais seulement un code retour système.

```
data _null_;
  tmp=system("dir /b c:\test");
  put tmp=;
run ;
```

Syntaxe :

```
retour_code=system("commande système");
```

- Si le code retour est négatif, la commande ne s'est pas exécutée correctement.
- Si le code retour est égal à 0, la commande s'est exécutée avec succès.

Remarque :

Cette fonction peut aussi être utilisée avec l'instruction "call" :

```
data _null_;
  call system("dir /b c:\test");
run ;
```

**Commande X**:**

La commande "X" est aussi disponible pour exécuter des commandes système. Son fonctionnement est sensiblement le même que l'instruction "system".

Voici un exemple de son utilisation :

```
options noxwait ;
data _null_;
  X "mkdir c:\test\TEMP";
run;
```

** : Pour les commandes X et SYSTEM, il est nécessaire d'ajouter une option spécifique à l'environnement de SAS afin que ces commandes s'exécutent de façon non interactive. En effet, celles-ci lancent des sous-processus du système pour exécuter les requêtes et sortent temporairement de la session SAS. Cette sous-tâche ne se fermera pas automatiquement sans l'ajout de l'option NOXWAIT. NOXWAIT spécifie qu'aucune action de l'utilisateur n'est nécessaire pour que l'environnement SAS reprenne la main dans l'exécution du programme.

Pour la déclarer, il suffit de soumettre le code suivant :

```
options noxwait ;
```

Pour en savoir plus sur ces instructions, vous pouvez consulter :

- l'aide en ligne
- ou la SAS OnlineDoc® qui donne une vision par système d'exploitation.

Elle est disponible à l'adresse :

<http://support.sas.com/onlinedoc/913/docMainpage.jsp>.

Pascal Nicolas

Migration de SAS® Enterprise Miner™ 5.1 vers 5.2

SAS® Enterprise Miner™ 5.2 est disponible depuis le kit d'installation SAS® 9.1.3 dont la date du CD "The SAS® Software Navigator" est postérieure au 11 Octobre 2005.

Son installation nécessite au préalable, la mise en place du Service Pack 3 pour SAS® Foundation (sur la machine hébergeant le serveur d'espace de travail).

Dans cet article, la migration de SAS® Enterprise Miner™ 5.1 vers de SAS® Enterprise Miner™ 5.2 sera détaillée pour une installation sur plusieurs machines, dans l'environnement Windows. Les étapes à suivre dans le cas d'une migration sur une seule machine, seront présentées ensuite.

Installation sur plusieurs machines

Si vous travaillez avec SAS® Enterprise Miner™ 5.1 sur deux machines différentes, c'est-à-dire en architecture 2 tiers et que vous souhaitez migrer en SAS® Enterprise Miner™ 5.2, suivez les instructions suivantes :

Sur le serveur

1. Pour des raisons de sécurité, il est vivement conseillé de sauvegarder une copie du référentiel de métadonnées, grâce à la macro %omabackup existante.

Vous trouverez des informations très complètes sur cette macro à l'adresse suivante :

<http://support.sas.com/rnd/eai/openmeta/v9/setup/backup.html>

Exemple de syntaxe (architecture une machine) :

```
options metaserver='localhost'
  metaport=8561
  metaprotocol=bridge
  metauser='<domain\userid>'
  metapass='<password>'
  metarepository='Foundation';
```

```
%omabakup(destinationpath="c:\sas\eminer\lev1\sasbackup",
  serverstartpath="c:\sas\eminer\lev1\sasmain",
  rposmgrpath="metadataserver\rposmgr");
run;
```

Remarque : Le répertoire "destinationpath" est à créer avant de lancer la macro.

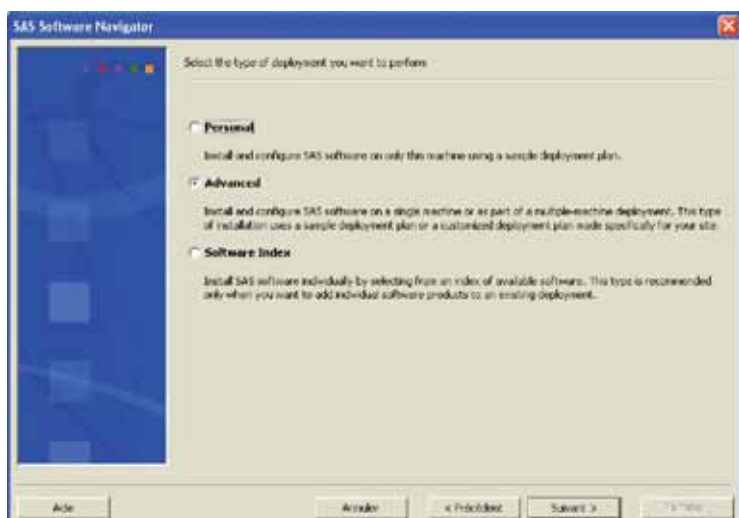
2. Arrêter les services suivants : Object Spawner, serveur de métadonnées et plateforme partagée de SAS® Enterprise Miner™ 5.1.

3. Insérer le CD "The SAS® Software Navigator" (datant du 11 Octobre 2005).

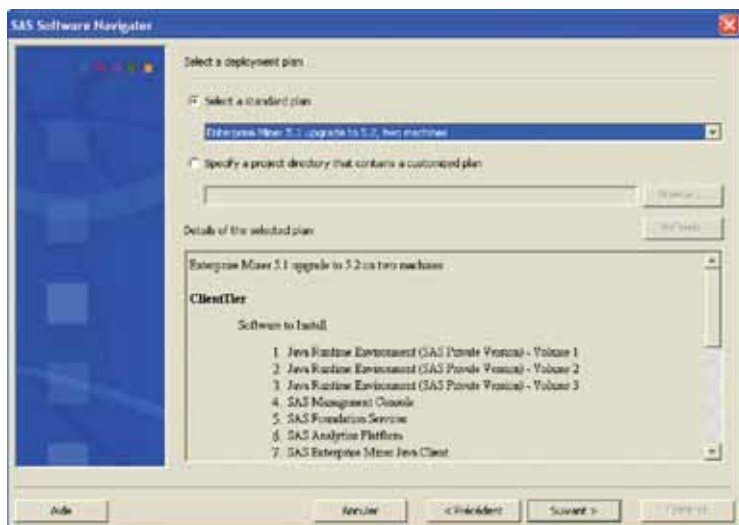
RETOUR SOMMAIRE



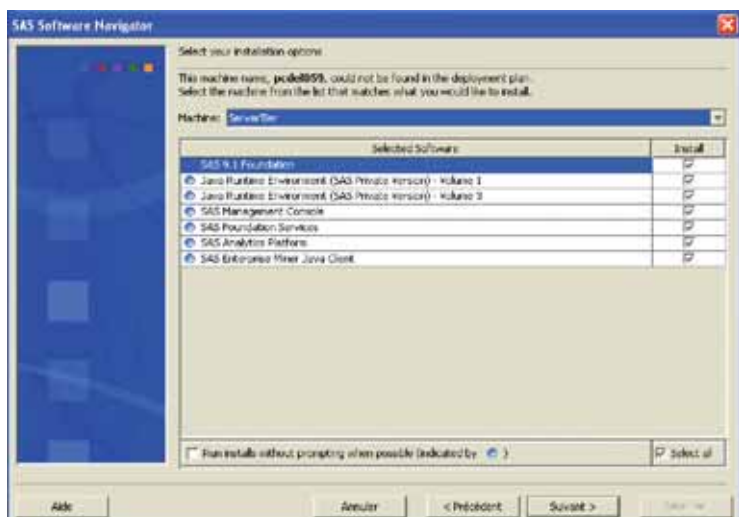
4. Choisir "Install and configure SAS Software", spécifier le chemin où se trouve votre fichier d'installation de données (SID) et choisir une installation avancée :



5. Choisir le plan standard :
"Enterprise Miner 5.1 upgrade to 5.2 on two machines"



6. Sélectionner l'option "ServerTier" pour cette machine :

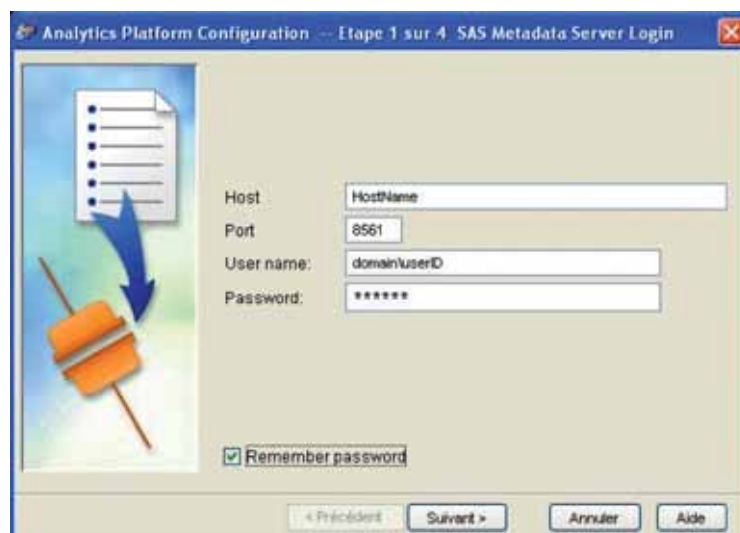


7. Installer les composants spécifiés dans ce plan pour le serveur L'assistant d'installation détecte si une instance de SAS est déjà installée et présente alors les choix suivants : "Add Components to SAS" ou "Install a new instance of SAS". Choisir le choix par défaut 'Add Components to SAS'.

8. Lorsque l'installation est terminée, vous devez télécharger le correctif suivant pour la plateforme analytique (ce correctif est spécifique à Windows, il n'est pas requis sur UNIX) :
http://ftp.sas.com/techsup/download/hotfix/dmine_cl52.html

9. Redémarrer le serveur de métadonnées et l'Object Spawner
Note : La plateforme partagée de SAS® Enterprise Miner™ 5.1 a été remplacée par la plateforme analytique.

10. Configurer la plateforme analytique :
Démarrer > Programmes > SAS > Analytics Platform > AP Server Advanced Configuration.
Spécifier le nom du serveur de métadonnées et le numéro de port (par défaut : 8561), votre userID et votre mot de passe. Ne pas oublier d'inclure le nom du domaine devant votre userID (domain\userID) :



Remarques :

- Ce sont les mêmes paramètres que dans la configuration avancée de SAS® Enterprise Miner™ 5.1 :

Démarrer > Programmes > SAS > Enterprise Miner > EM 5.1 Advanced configuration

- Vous trouverez des informations très complètes sur la plateforme analytique à l'adresse suivante :

<http://support.sas.com/documentation/onlinedoc/apcore>

11. Démarrer la plateforme analytique :
Démarrer > Programmes > SAS > Analytics Platform > Start AP Server

Le serveur est alors en attente d'un client.



12. SAS® Enterprise Miner™ 5.1 ne fonctionnant plus, vous devez le désinstaller en allant dans :

Démarrer > Paramètres > Panneau de configuration > Ajout/suppression de programmes

Sur les postes clients :

13. Insérer le CD "The SAS® Software Navigator"

14. Choisir "Install and configure SAS Software", spécifier le chemin où se trouve votre fichier d'installation de données (SID) et choisir une installation avancée

15. Sélectionner le plan standard "Enterprise Miner 5.1 upgrade to 5.2 on two machines"

16. Sélectionner l'option "ClientTier" pour cette machine

17. Installer les composants spécifiés dans ce plan

18. Démarrer SAS® Enterprise Miner™ 5.2

Démarrer > Programmes > SAS > Enterprise Miner 5.2 > Enterprise Miner

Dans la fenêtre de Log On, décocher "Personal Workstation", entrer votre userID (en incluant le nom de domaine), votre mot de passe et le nom du serveur hébergeant la plateforme analytique :

Installation sur une seule machine

Si vous utilisez actuellement SAS® Enterprise Miner™ 5.1 sur une seule machine, et que vous souhaitez migrer en SAS® Enterprise Miner™ 5.2, vous devez reprendre les étapes de la partie précédente dans l'ordre indiqué ci-dessous. Les étapes précédées d'un astérisque nécessitent certaines modifications :

Etape 0 : Installation du Service Pack 3 pour SAS® Foundation.

Etape 1 : Sauvegarde du référentiel de métadonnées.

Etape 2 : Arrêt des services Object Spawner et Serveur de métadonnées.

Etape 3 : Insérer le CD "The SAS® Software Navigator".

***Etape 4 :** Choisir une installation personnalisée ('Personal').

***Etape 5 :** "Enterprise Miner 5.1 upgrade to 5.2 on one machine".

Etape 7 : Installation des composants spécifiés pour ce plan.

Etape 8 : Installation du correctif pour la plateforme analytique.

Etape 9 : Redémarrage des services Object Spawner et Serveur de métadonnées.

Etape 11 : Configuration de la plateforme analytique

***Etape 18 :** Ouverture de SAS® Enterprise Miner™ 5.2.

Dans la fenêtre de Log On, sélectionner "Personal Workstation", entrer votre userID (en incluant le nom de domaine) et votre mot de passe et cliquer sur Log On.

Etape 13 : Désinstallation de SAS® Enterprise Miner™ 5.1.

Tous les projets sauvegardés avec la version 5.1 sont compatibles avec toutes les versions 5.x de SAS® Enterprise Miner™. Les serveurs et tous les clients de SAS® Enterprise Miner™ doivent migrer dans la même version pour fonctionner correctement. En effet, un SAS® Enterprise Miner™ 5.1 se connectant à un serveur d'espace de travail 5.2 (et vice versa) n'est pas supporté et peut produire des résultats incorrects.

Les deux SAS Notes suivantes regroupent les étapes de migration présentées dans cet article :

<http://support.sas.com/techsup/unotes/SN/016/016082.html>

<http://support.sas.com/techsup/unotes/SN/016/016200.html>

Vous trouverez à l'adresse suivante des informations détaillées sur les nouveautés offertes par SAS® Enterprise Miner™ 5.2 :

<http://support.sas.com/software/91x/emgui52whatsnew900.htm>

Aurélien Gilbert

RETOUR SOMMAIRE



La procédure METAOPERATE

La procédure METAOPERATE permet d'exécuter des tâches administratives associées au serveur de métadonnées et aux référentiels. Ces tâches sont également réalisables directement avec la SAS® Management Console.

Nous avons vu dans l'article "Introduction à SAS® Open Metadata Architecture" de la lettre Allo Support N°20 les actions PAUSE, RESUME, REFRESH, STOP, et STATUS. Nous allons à présent détailler les actions UNREGISTER, DELETE, PURGE et EMPTY, puis nous aborderons les tâches qui ne sont pas du ressort de cette procédure.

Les actions

Unregister

L'action UNREGISTER permet de désactiver l'accès à un référentiel. Ce qui signifie que le référentiel n'est plus visible dans la SAS® Management Console et le serveur de métadonnées ne reconnaît plus ce référentiel. Cependant, il existe toujours physiquement : c'est uniquement le lien logique qui a été supprimé. Il peut être de nouveau enregistré au niveau du serveur de métadonnées en passant par la fonction "ajout de référentiel" du plug-in "Gestionnaire des métadonnées" de la SAS® Management Console.

Avant de procéder au "désenregistrement" du référentiel, il est important de noter tous les attributs de ce référentiel car lors de son réenregistrement, si le type du référentiel est différent, le référentiel sera initialisé lors de sa création et il est possible que les définitions des métadonnées existantes soient perdues.

Cette action correspond à l'option "Supprimer l'enregistrement" (Unregister) lorsque vous effectuez un clic droit sur le référentiel dans la SAS® Management Console.

L'exemple suivant permet de supprimer l'enregistrement du référentiel "Partmeta" :

PROC METAOPERATE

```
SERVER="localhost"  
PORT=8561  
USERID="sasadm"  
PASSWORD="saspw1"  
PROTOCOL=BRIDGE
```

```
Action=UNREGISTER  
Repository="Partmeta";  
RUN;
```

Comme nous l'avons vu dans l'article "Introduction à SAS® Open Metadata Architecture", les instructions de la connexion au serveur sont optionnelles dans la procédure mais sont nécessaires pour se connecter au serveur de métadonnées. Si ces instructions ne sont pas spécifiées, les valeurs prises en compte seront celles des options système (METASERVER, METAPORT, METAUSER, METAPASS et METAPROTOCOL).

Delete

Cette action supprime définitivement le référentiel. Il n'existe plus physiquement et toutes les métadonnées sont supprimées. Contrairement à l'action UNREGISTER, il n'est pas possible de récupérer les métadonnées de ce référentiel.

Il est vivement recommandé de faire une sauvegarde du référentiel au préalable.

La syntaxe est la suivante :

```
PROC METAOPERATE  
Action=DELETE  
Repository="Partmeta";  
RUN;
```

Dans cet exemple, le référentiel Partmeta est physiquement supprimé.

Purge

L'action PURGE a pour but de supprimer définitivement du référentiel tous les éléments des métadonnées qui ont été supprimés logiquement. Lorsque des enregistrements sont logiquement supprimés, cela signifie qu'ils ne sont plus disponibles mais ils sont toujours stockés physiquement dans les tables du référentiel et sont lus en mémoire lors de certains traitements sur les métadonnées. La suppression physique retire définitivement les enregistrements des métadonnées. A partir de SAS® 9.1.3 Service Pack 3, cette action n'a plus vraiment d'intérêt puisque ces éléments sont maintenant supprimés physiquement par défaut. Cependant, si vous souhaitez supprimer les éléments des métadonnées de manière logique et non physique, vous pouvez utiliser l'option de configuration du serveur `PHYSICAL_DELETE="0"`.

Pour plus d'information concernant cette option, vous pouvez vous référer au contenu des liens suivants :

<http://support.sas.com/rnd/eai/openmeta/v9/setup/specoptions.html>
<http://support.sas.com/rnd/eai/openmeta/v9/bestpractices/serveroptions.html#physicaldelete>

Dans la SAS® Management console, cette action correspond à l'option "Vider" ou "Purge".

Empty

L'action EMPTY permet de supprimer le contenu d'un référentiel. Par exemple, si le référentiel contient des définitions de bibliothèques et de tables, l'action EMPTY supprimera toutes ces définitions. Le référentiel sera vidé mais non supprimé.



L'exemple suivant permet de vider les métadonnées du référentiel Partmeta, mais en aucun cas cela supprime les données physiquement :

```
PROC METAOPERATE  
Action=EMPTY  
Repository="Partmeta";  
RUN;
```

Cette action correspond à l'option "Formater" ou "Format" lors d'un clic droit sur le référentiel au niveau de la SAS® Management Console.

Les tâches administratives non réalisables avec la procédure METAOPERATE

D'autres tâches administratives associées au serveur de métadonnées et aux référentiels peuvent être effectuées mais celles-ci se font directement à partir de la SAS® Management Console. Il n'est pas possible de les effectuer avec la procédure METAOPERATE. Ces tâches sont les suivantes :

- **Tronquer (truncate)** : Supprime tous les objets des métadonnées du référentiel sélectionné, mais ne supprime pas les conteneurs d'objets, ni l'enregistrement du référentiel.
- **Initialiser le référentiel (Initialise Repository)** : Redéfinit les paramètres initiaux du référentiel, y compris les groupes d'utilisateurs et les droits d'accès, les modèles de contrôle d'accès et les modèles de ressources installés. Cette fonction n'est pas disponible pour les référentiels projet.
- **Contrôle (Audit)** : Affiche la fenêtre Contrôle. Cette fenêtre permet d'activer le contrôle pour le référentiel sélectionné, d'indiquer le type de contrôle à effectuer, et de préciser où stocker le journal de contrôle.

Vous trouverez un complément d'information sur ces actions dans la partie "Maintaining the Active Server" du chapitre 3 "Managing Metadata" du document "SAS® Management Console : User's Guide".

http://support.sas.com/documentation/onlinedoc/91pdf/sasdoc_912/mgmtconsole_ug_8186.pdf

Clarisse Audoucet

Directeurs de la publication :

Philippe Hoffmann
Pierre Lorenzetti

Comité de rédaction :

Géraldine Deschamps
Karine Chrillesen
Pascal Nicolas
Aurélie Gilbert
Clarisse Audoucet

Comité de relecture :

Fabrice Giraud
Mouloud Dey
Philippe Hoffmann

Réalisation :

aressyassociés

+33 (0) 1 41 38 85 53 - 04/06 - 2480

RETOUR SOMMAIRE