

## **Quatrième PARTIE**

### **Divers changements dans le langage Sas de Base**

## Chapitre 10 Panorama des changements du Langage Sas en v9.2.3

### 10.1 Modifications globales

- Il y a un certain nombre de nouvelles options systèmes et prenons l'occasion de les découvrir par la proc options que peu de lecteurs connaissent :

```
proc options help;    *donne le help de la syntaxe ;
proc options listgroups;
```

COMMUNICATIONS	ISPF
ENVDISPLAY	REXX
ERRORHANDLING	ADABAS
ENVFILES	DATACOM
LANGUAGECONTROL	DB2
EXECMODES	IDMS
EXTFILES	IMS
SASFILES	ORACLE
INPUTCONTROL	ODSPRINT
GRAPHICS	EMAIL
LOGCONTROL	DATAQUALITY
LISTCONTROL	HELP
LOG_LISTCONTROL	META
MACRO	PDF
SORT	SECURITY
INSTALL	SVG
MEMORY	SMF
PERFORMANCE	SQL
CODEGEN	

Les options sont en effet rattachées à des groupes aux noms spécifiques portant sur un thème précis.

```
proc options group=(macro graphics); * listage des options de deux groupes;
proc options option=work;           * listage d'une option ;
```

Pour obtenir plus de détails sur **une** option, on complétera l'instruction par un *define* et *value* .

Ainsi, hormis les nouvelles options *pdf* vues à la page 152 et celles liées au *sql* ou au device *svg*, les principales nouvelles options sont

Dldmgaction= <b>repair</b>	Cette option peut porter sur les catalogues et les tables et sert en général en cas de problème de fichier endommagé avec le choix entre <i>fail</i> , <i>abort</i> , <i>prompt</i> ou <b>repair</b> (par <b>défaut</b> ). Portant sur les tables (c'est aussi un option de dataset peu connue) on pourra aussi faire le choix additionnel de nettoyer les aspects indexation et audit avec <i>dldmgaction=noindex</i> (nouveau en 9.2)
(no) sortvalidate	Si activé en enlevant le no, Sas opérera une vérification de l'ordre selon les clés spécifiées en cas de présomption marquée par l'utilisateur à l'aide de l'option (de dataset ou dans la procédure dataset ou modify) sortedby=
Lrecl= <b>256</b> <b>1.073.741.823 =</b> <b>Nouv limite 9.2.3 ?</b>	par <b>défaut</b> mais peut s'étendre de 1 à 32767 et sert définir par défaut la longueur logique d'un enregistrement en lecture ou écriture (in/file etc)
Dmspgmlinesize= <b>136</b>	Au démarrage d'une session (donc config etc) spécifier autre chose que le minimum de 136 comme longueur autorisée de ligne de code dans le l'Editeur. Maxi=960, mais les écrans sont-ils aussi wide ?
Ibufno= <b>0</b>	Le <i>nombre de buffer supplémentaires</i> allouables pour le fonctionnement des <i>index</i> peut passer de 0 à 10000 et s'exprimer aussi en unités immédiatement accolées comme le k, m, ou g pour les octets.
(no) <b>gstyle</b>	Cette option ne portera que sur les <i>grseg</i> et si elle est désactivée, le style de la destination utilisée ne s'appliquera pas à ce graphique sas classique qui apparaîtra comme avant la 9.2 (activé + java non concernés)
Fontrendering=	Comme Sas a introduit de nouvelles familles de police ttf (21 true type fontes), on aura le choix entre le défaut à <i>freetype_points</i> ou <i>host_pixels</i> lors du rendu avec certains devices Sas.

Jpegquality=	Une image jpeg se compresse habituellement pour garder les détails jusqu'à 75 (défaut en % de qualité résiduelle), mais vous pouvez choisir 100 ou descendre à 50 par ex.
Deflate=6 (no) <b>uprintcompression</b>	S'utilise avec l'option UPRINTCOMPRESSION activée par défaut et portant sur les imprimantes universelles de sas (pcl..., pdf, ps et svgz) ou les devices graphiques (sasprt..., pdf, pdfc et svgz) le facteur de compression va de 0 à 9.
(no) <b>fontembedding</b>	Provoque l'inclusion des fontes utilisées dans les fichiers créés

- L'instauration pour le batch d'un système de checkpoint et d'un mode redémarrage (sans réexécution de ce qui a déjà été exécuté. Je ne le détaille pas : cherchez *steprestart*.
- Sas supporte le protocole internet Ipv6. Sas s'est aussi mis au diapason (lecture et écriture) avec la norme ISO 8601 en matière de dates , temps et intervalles.
- Comme vu partiellement plus haut, les 21 nouvelles fontes acquises par Sas (5 latines compatibles Microsoft, 8 unicode et 8 asiatiques) seront utilisées par tous les *Universal Printers* et par bien des devices graphiques.
- De nouveaux *printers* apparaissent : pdfa, png, pngt, svg, svgt et svgz (t pour la transparence du png, a pour l'archive pdf <http://fr.wikipedia.org/wiki/PDF/A-1> , z pour compressé) . Pour déblayer la différence printer/device lire par ex : <http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/a003307061.htm>

```
proc registry listhelp startat='ods\fonts'; run;
montrera ces familles cumberland/thorndale/albany/thorndale duospace/monotype sans wt/etc.
```

Un exemple illustrera cela (adapté à partir d'une réponse de Cynthia Zender)

```
ods listing style=listing;
options gstyle;
goptions reset=all hsize=15 cm vsize=8 cm;
** Superscripts/Exposants dans une Fonte Unicode ;
title f='Albany AMT' h=8pt 'Superscript zero' f='Arial Unicode MS/unicode' '2070'x;
title2 f='Albany AMT' h=10pt 'Exposant un' f='Arial Unicode MS/unicode' '00B9'x;
title3 f='Albany AMT' h=12pt 'Superscript deux' f='Arial Unicode MS/unicode' '00B2'x;
title4 f='Albany AMT' h=14pt 'Exposant trois' f='Arial Unicode MS/unicode' '00B3'x;
title5 f='Albany AMT' h=16pt 'Superscript quatre' f='Arial Unicode MS/unicode' '2074'x;
title6 f='Albany AMT' h=20pt 'Formule exposée?'
f='Arial Unicode MS/unicode' '207D2074207A2075207C2079207E'x;
** Subscripts/Indices ;
footnote c=blue h=5 pct 'Note: Cette eau fait des bulles H'
m=(+2,-1.75) pct h=3 pct '2'
m=(+2,+1.75) pct h=8 pct 'O';
footnote3 f='Albany AMT' h=8pt 'Subscript zero' f='Arial Unicode MS/unicode' '2080'x;
footnote4 f='Albany AMT' h=10pt 'Indice un' f='Arial Unicode MS/unicode' '2081'x;
footnote5 f='Albany AMT' h=12pt 'Subscript deux' f='Arial Unicode MS/unicode' '2082'x;
footnote6 f='Albany AMT' h=14pt 'Indice trois' f='Arial Unicode MS/unicode' '2083'x;
footnote7 f='Albany AMT' h=16pt 'Subscript formula'
f='Arial Unicode MS/unicode' '208D2084208A2085208C2089208E'x;
proc gslide;run;
quit;
```

Superscript zero<sup>°</sup>  
Exposant un<sup>°</sup>  
Superscript deux<sup>°</sup>  
**Exposant trois<sup>3</sup>**  
Superscript quatre<sup>4</sup>  
Formule exposée?<sup>(4+5=9)</sup>

Note: Cette eau fait des bulles H<sub>2</sub> O

Subscript zero<sub>0</sub>  
Indice un<sub>1</sub>  
Subscript deux<sub>2</sub>  
**Indice trois<sub>3</sub>**  
Subscript formula<sub>(4+5=9)</sub>

6. Le composant objet/componentobject Java dans l'étape Data est en production. Ceci sera-t-il présenté dans le volume2 ?

<http://support.sas.com/rnd/base/datastep/dot/javaobj.html>

«**JAVA in SAS, JavaObj, a DATA Step Component Object**» de Richard de Venezia  
<http://www.devenezia.com/papers/sugi-30/241-30%20%20%28JavaObj%29.pdf>

Par contre, l'ODS *data step object* et l'ODS *layout* restent en pré-production, même si ce dernier commence à être utilisé relativement souvent.

7. Le processus de l'expression *where like* a été modifié pour capter un éventuel `_` ou `%` situé dans la chaîne de caractère, ces deux caractères étant réservés à un usage de joker.

```
data a;
  b="in_SAS "; output; b="inconnu"; output;run;
data b(where=( b like 'in^_%' escape '^'));
set a ; run;
```

L'enregistrement contenant `in_SAS` sera retenu et pas l'autre car l'option `escape` a changé le sens du `^_` tout comme il changerait le sens d'un éventuel `^%` à la recherche du caractère `%`.

8. Le rajout d'un catalogue avant ou après dans un liste de *libref* déjà existante est possible

```
Options insert=(fmtsearch=(templib));
proc options option=fmtsearch;run; => FMTSEARCH=(TEMPLIB WORK LIBRARY)
Options append=(fmtsearch=(templib)) => FMTSEARCH=(WORK LIBRARY TEMPLIB)
```

## 10.2 Modifications dans l'étape Data :

### 10.2.1 : Les instructions modifiées

**Abort** `__` | **n** | **abend** | **return** | **cancel (file)** | **nolist** ;

Deux nouveaux arguments à cette instruction qui s'utilise généralement dans un *if* ou un *select*. L'effet est en général l'arrêt de l'étape data et selon le paramètre utilisé un autre événement.

*Nolist* à mettre en fin d'instruction pour ne plus voir s'afficher sur la log la liste de toutes les variables de la table.

*Cancel* alors en interactif, seul le programme soumis est arrêté avec un message sur la log et le reste du code n'est pas affecté. On rajoutera le mot clé `file` si cet abort est dans un fichier tel que `l'autoexec.sas` ou celui appelé par un `%include`.

**By** **descending** | *variable(s)* | **notsorted** | **groupformat** ;

Cette instruction se soumet aux nouvelles spécifications de l'option `sortseq=linguistic` du `proc sort` (voir plus loin).

#### **Data**

Trois arguments optionnels nouveaux en 9.2...

`/nesting` affichera un message dans la log explicitant les niveaux d'ouverture et fermeture de boucles

`/stack=n` fixera un maximum au nombre de *link* imbriqués

`/nolist` comme dans abort, on pourra éviter l'affichage de la longue liste de variables.

```

data b /nesting nolist;
set sashelp.class;
select ;
    when (sex="M") do;
        select;
            when(10<age<12) put 'message trop jeune';
            when(age<16) abort cancel;
            otherwise ;
        end;
    end;
    when (sex="F") do;
        select;
            when(10<age<12) put 'message trop jeune';
            when(age<16) abort cancel;
            otherwise ;
        end;
    otherwise;
end;
run;

```

### Infile

Possibilité de détecter une combinaison de caractères comme séparateurs de champs grâce à l'option *dlmstr=* et d'en éliminer les blanc de fin avec l'option *dlmsopt='T'*.

```

data a;
infile cards dlmstr="à# " dlmsopt='T' ;
informat a :15. b :10. ;
input a b c $ ;
put _all_;
cards;
123456 à# 1456789 à# SPIROU
7889456 à# 89514561à#ZORGBLU
;
run;

```

### Filename

L'instruction filename a bénéficié d'enrichissements (notamment sur l'authentification de domaine, les accès par mot de passe et à un serveur proxy, temps d'attente) en ce qui concerne les méthodes d'accès *ftp*, *url*, *webdav* et la nouvelle *sftp*(secure file transfer protocol).

Sous ftp, avec des enregistrements de longueur variable, il vous sera possible de définir le *termstr=""* avec *crlf*, *lf* (défaut) ou *null* (0x00) pour indiquer le(s) caractère(s) invisible(s) terminant l'enregistrement.

### File

On y retrouve le *termstr*.

Il n'est pas impossible que dans le contexte de l'usage de la nouvelle option *encoding="nomvalide"*, vous soyez obligé d'utiliser l'option *ignoredoseof* pour outrepasser une reconnaissance de fin de fichier de type ^Z insérée dans le flot d'informations lues.

### Merge

```

data bercy;
merge dep1-dep4 dep2: dep33-dep37;
by impots;
run;

```

Ces appels abrégés sont maintenant autorisés.

Cependant, gardez en tête que l'opération merge sur de nombreuses tables a, du fait de la complexité du programme complémentaire dans l'étape data, parfois intérêt à être isolée des modifications qui suivent.

## Set

```
data class1 class2 class3 class4;
set sashelp.class;
if _n_ < 6 then output class1;
if 5 < _n_ < 11 then output class2;
if 16 > _n_ > 10 then output class3;
if 15 < _n_ < 20 then output class4;
run;
data regroupe;
set class: indsname=orig;
length origin $ 10;
origin=scan(orig,2);
run;
proc freq data=regroupe;
tables origin;
run;
data rassemble;
set class1-class4;
run;
```

Il ressort de ce programme deux nouveautés :

- possibilité d'appeler, de façon abrégée, une séquence de table comme cela était possible pour les variables ici *class1-class4*
- possibilité de créer une variable indicatrice d'origine de table tout en utilisant un appel synthétique du genre racine: ici *class:*

Procédure FREQ				
origin	Fréquence	Pourcentage	Fréquence cumulée	Pctage cumulé
CLASS1	5	26.32	5	26.32
CLASS2	5	26.32	10	52.63
CLASS3	5	26.32	15	78.95
CLASS4	4	21.05	19	100.00

En effet, l'option *indsname=* sert à donner un nom temporaire récupérant l'information sur la source de chaque observation lue. Il suffit ensuite de piocher dans le second mot de la chaîne (libref.nomdetable) pour récupérer le nom de cette table.

### 10.2.2 : Nouveautés en informats et formats

#### Les formats

```
data _null_;
da="21dec1949"d;
put da e8601da.;           → 1949-12-21
run;
```

Tous les formats et informats ISO 8601 ont été intégrés dans SAS Voir  
<http://support.sas.com/documentation/cdl/en/lrdict/63026/HTML/default/a003169814.htm>  
<http://support.sas.com/documentation/cdl/en/lrdict/63026/HTML/default/a001263753.htm>

Une petite modif au format **datew.** rend possible `put da date11.;` → 21-DEC-1949

**Bestdw.p.** w par défaut 12 [1-32] p pour précision par défaut 3 [1-(w-1)]

```
data best; input nb;
put nb best12.
@15 nb d10.4
@26 nb bestd14.
@40 nb bestd11.3;
cards;
12345           12345           12345.0           12345           12345
1234.5          1234.5           1234.5          1234.5000000    1234.50000
123.45         123.45           123.45000      123.4500000    123.45000
12.345         12.345           12.34500       12.3450000     12.34500
1.2345         1.2345           1.23450        1.2345000      1.23450
.12345         0.12345          0.12345        0.1234500      0.12345
1.23456        1.23456          1.23456        1.2345600      1.23456
; run;
```

**Percentw.d** w par défaut 6 [4-32] d [0-31] et inférieur à w

```
data pct;
input nb;
put nb
@10 nb percent10.
@21 nb percentn10.
@32 nb percentn9.2;
cards;
0.1          0.1          10%          10%          10.00%
1.2          1.2          120%         120%         120.00%
0.8          0.8          80%          80%          80.00%
0.005        0.005         1%           1%           0.50%
-0.5         -0.5         ( 50%)       -50%         -50.00%
;run;
```

Apparition des formats informatiques **sizekw.d** ou **sizekbw.d** et de l'informat **sizekmgw.d**

```
data b ;
input valkb :sizekmg5. @6 valM :sizekmg5.1 valG :sizekmg5.1;
octet=256;
kocet=1024;* 2**10;
mega=2**20;
giga=2**30;
put valkb valm valg /          == > 1024 2621440 78920024064
   octet sizek6.3 +2 kocet sizekb6.2/  == > 0.25K 1.00KB
   mega sizekmg6.2 + 2 giga sizekmg10.2 ; == > 1.00MB 1.00GB
cards;
1K          2.5M          73.5G
;run;
```

### Les informats

**Trailsgnw.** W 6 par défaut [1-32] ignorera la décimale si virgule et acceptera le signe (négatif) en queue

```
data lectfr;
input nb trailsgn7.;
put nb ;
cards;
1.5-          -1.5
3.5-          -3.5
5+            5
1,23          123
;run;
```

En matière de NLS (national language support), nous avons l'arrivée de l'arabe, du perse et pour l'euro, **nlmngbpw.d** qui ravira les anglais (**nlmngbpw.d.** = le format); **nlmnieurw.d** permettra l'informat pour l'euro et **nlmnieurw.d** correspondra au format pour l'euro.

**nlbestw.** fera équivalence du format numérique Sas bien connu **bestw.** en version locale.

**\$Base64xw.** est aussi un nouvel informat car on rencontre de telles informations dans l'XML et les email etc.

```
motpasse="secret"; put motpasse $Base64X64. ; → c2VjcmV0
```

### 10.2.3 : Les fonctions

<http://support.sas.com/documentation/cdl/en/lrdict/63026/HTML/default/a000245860.htm>

#### Les fonctions caractères – findw, char, first, catq, countw, transtrn, whichc, call sortc --

- La nouvelle fonction **findw**(chaîne,mot<,delimiteur, modifieur, startpos>) servira à trouver la position d'un mot isolé (borné par délimiteur ou début ou fin de chaîne) dans une chaîne de caractères.

En windows, le délimiteur= une liste de caractères (propre à Sas) blank ! \$ % & ( ) \* + , - . / ; < ^ |

```
data _null_;
a="Et moi, et moi, et moi, j'y pense    et puis j'oublie";
m11=" et moi";
m12="et moi";
m13=" et moi,";
m2=" j";
m3="moi";
b=findw(a,m3); c=indexw(a,m3); put 'pas comme indexw ' b= c;
b=findw(a,m11,',');put '1le blanc compte et le délimiteur font le mot ' b= 'ème position';
b=findw(a,m11,',',9);put '2même recherche à partir de la position 9 + donc de g à dr ' b=;
b=findw(a,m12,',','i'); put '3modifieur i=ignore case ' b=;
b=findw(a,m11,',','e'); put '4modif e=nombre de mots explorés pour le trouver ' b= 'mots';
b=findw(a,m13,',','r'); put '5modif i + r=suppr délim tête et fin du mot ' b=;
b=findw(a,m3,-23); put '6recherche dr à g car -, à partir de position 23 ' b=;
b=findw(a,m2,',','p e');put '7modif e + p=rajout ponctuation dans liste délim ' b= 'mots';
b=findw(a,m11,',','p b');put '8modif p + b=recherche de dr(right) à g ' b= 'ème position';
b=findw(a,m11,',','p e b'); put '9ce qui fait ' b= 'mots';
run;
```

```
pas comme indexw b=4 c=0
1le blanc compte et le délimiteur font le mot b=8 ème position
2même recherche à partir de la position 9 + donc de g à dr b=16
3modifieur i=ignore case b=1
4modif e=nombre de mots explorés pour le trouver b=2 mots
5modif i + r=suppr délim tête et fin du mot b=8
6recherche dr à g car -, à partir de position 23 b=20
7modif e + p=rajout ponctuation dans liste délim b=4 mots
8modif p + b=recherche de dr(right) à g b=16 ème position
9ce qui fait b=4 mots
```

Pour extraire facilement un seul caractère d'une chaîne, prenez la fonction **char**(chaîne,valeur).

```
data b;
a="Anticonstitutionnellement";
b=char(a,24);c=lengthc(b);          d=substr(a,24,1);e=lengthc(d);
f=substrn(a,26,1);g=lengthc(f);    h=char(a,26);i=lengthc(h);
run;
proc print noobs;run;
```

a	b	c	d	e	f	g	h	i
Anticonstitutionnellement	n	1	n	25		25		1

- La fonction **first**(chaîne) retournera le premier caractère de toute chaîne de caractères.

```
data b;
a="Barack";b=" Hussein"; c="Obama";
initial=first(a)||first(b)||first(c);d=lengthc(initial);
ca=cat(a,b,c);e=lengthc(ca);
sa=substr(a,1,1)||substr(b,1,1)||substr(c,1,1);f=lengthc(sa);
run;
proc print noobs;run;
```

a	b	c	initial	d	ca	e	sa	f
Barack	Hussein	Obama	B O	3	Barack HusseinObama	200	B O	19



- La fonction **catq**(*modifieur*,<*délimiteur*>, *item1*, *item2*...) concatènera les items en tenant compte de la définition éventuelle de délimiteurs et des indications de fonctionnement issue des modifieurs.

Ces modifieurs sont :

- a placera des quotes autour de chaque chaîne concaténée
- l simples quotes 2 doubles quotes
- d traitera l'argument suivant comme le délimiteur de chaîne
- s supprimera les quotes de début et de fin
- m ajoutera des délimiteurs après le premier item même si l'item a une valeur manquante

Si on rencontre un délimiteur dans une chaîne, celle-ci est automatiquement mise entre quotes.

```
data b;
a="Nicolas"; b="Sarközy" ;c=" de Nagy-Bocsa"; d="Napoléon"; e="le Petiot";
f=1;g=2;h="I";i="II";j="III";k="ème";kk=" ";
length bb cc dd ee ff gg $ 35;
aa=catq(" ",a,b,c); aaa=lengthc(aa);put "<" aa +(-1) '>' +2 aaa=;
ee=catq("as",a,c); put "<" ee +(-1) '>';
bb=catq(" ",a,g,k); put "<" bb +(-1) '>';
cc=catq("a",d,h); put "<" cc +(-1) '>';
dd=catq(" ",d,h,i); put "<" dd +(-1) '>'; *dd=catq("",d,h,i);
ff=catq("m",kk,e); put "<" ff +(-1) '>';
gg=catq("d","+",f,g); put "<" gg +(-1) '>';
run;
<Nicolas Sarközy " de Nagy-Bocsa"> aaa=200
<"Nicolas" "de Nagy-Bocsa">
<Nicolas 2 ème>
<"Napoléon" "I">
<Napoléon I II>
<" " "le Petiot">
<1+2>
```

- La fonction **countw**(*chaîne*,<*delimiteurs*><,'*modifieurs*'>) servira à compter les mots dans la chaîne et retournera 0 si non trouvé. La liste des délimiteurs est toujours la même.

La liste des modifieurs varie quelque peu par rapport à celle de findw (voir <http://support.sas.com/documentation/cdl/en/lrdict/63026/HTML/default/a002977495.htm> ).

```
data b;
a="Le vif zéphyr<jubile> sur les kumquats 'du(clown) gracieux'. Phrase-_test";
b=countw(a); put '1 nombre de mots tous délimiteurs ' b= ;
c=countw(a,' '); put '2 nombre de mots isolés par un blanc ' c=;
d=countw(a,"()",'s'); put '3 n mots s=inclure char espacement comme b,'
'tabulations et remplissage par - + etc ' d=;
e=countw(a,""); put "4 mots sép par ' " e=;
f=countw(a,"()",'q'); put '5 mots sép () et q=ignorer les délimiteurs '
'dans sous-chaines entre quotes ' f= ;
g=countw(a,"()"); put '6 seulement les parenthèses mais partout ' g=;
h=countw("ah oui",' ',); put " essai avec touche ==> " h=;
run;
```

```
1 nombre de mots tous délimiteurs b=12
2 nombre de mots isolés par un blanc c=9
3 n mots s=inclure char espacement comme b,tabulations et remplissage par - +
etc d=10
4 mots sép par ' e=3
5 mots sép () et q=ignorer les délimiteurs dans sous-chaines entre quotes f=1
6 seulement les parenthèses mais partout g=3
7 essai avec touche ==> h=2
```

- Des changements ont eu lieu pour la nouvelle fonction **transtrn**(source, cible,remplacement) : une chaîne vide est bien remplacée, alors que dans la fonction **tranwrđ**, ce qui est substitué est un blanc.

```

data petrole;
input phrase $50.;
if _n_=1 then do;slate='*' || translate(phrase, 'ea', 'ee');put slate=;
                ouf='*' || tranwrđ(phrase, 'ee', 'ea');put ouf=;end;
if _n_=2 then do;
sword= '*' || tranwrđ(phrase, 'BP', 'Total') || '*'; put 'sword=' @10 sword;
string='*' || transtrn(phrase, 'BP', 'Total') || '*'; put 'string=' @10 string;
vide= '*' || transtrn(phrase, 'BP', trimn('')) || '*';put 'vide=' @10 vide;
blanc= '*' || tranwrđ(phrase, 'BP', trimn('')) || '*';put 'blanc=' @10 blanc;end;
cards;
Buster Keeton est en combustion,yeeh!
BP est la meilleure compagnie pétrolière.Signé BP!
run;

slate=*Bustar Kaaton ast an combustion,yaah!
ouf=*Buster Keaton est en combustion,yeah!
sword= *Total est la meilleure compagnie pétrolière.Signé Total!*
string= *Total est la meilleure compagnie pétrolière.Signé Total!*
vide= * est la meilleure compagnie pétrolière.Signé !*
blanc= * est la meilleure compagnie pétrolière.Signé !*

```

- Enfin, les anciennes fonctions **whichC**(argument, var1, var2 ...) et **whichN** (idem) sont documentées

Et permettent de chercher l'argument dans les valeurs des variables consécutives. C pour une recherche caractère et N pour une recherche de valeur numérique.

\*SAS-L Joe Matisse recherche la plus petite valeur du vecteur et obtient son rang (indice de l'array);

```

data test;
array _v[*] var1-var4;
input _v[*];
low = min(of _v[*]);
lowVar = vname(_v[whichN(low,of _v[*])]);
datalines;
20 30 15 40
17 50 25 30
;;;

```

- Call sortn**(<of> var1, var2 ...) et **Call sortc**(<of> varc1,varc2...)

Tri croissant de variables de même type et de même longueur par call sortc caractère et call sortn numérique

<pre> data c; length a1-a4 \$8; a1="Truand";a2="Brute";a3="minable";a4="Bon"; n1=18950;n5=-2.3;n2=.;n3=3;n4=1; array nn(*) n1-n5; call sortc(of a1-a4);call sortn(n4,n5); put "char=" a1 a2 a3 a4;put n4= n5=; call sortn(of nn(*)); put nn(*) 10.2; run; </pre>	<pre> char=Bon Brute Truand minable n4=-2.3 n5=1 . -2.30 1.00 3.00 18950.00 </pre>
--	--

- La fonction **scan**(chaîne, compteur < ,charliste, modifieurs>) a un nouveau paramètre optionnel *modifieurs*.

*Charliste* correspond habituellement aux délimiteurs blank ! \$ % & ( ) \* + , - . / ; < ^ |

Les **modifieurs** sont dans la liste suivante : a, b, c, d, f, g, h, i, l, m, n, o, p, q, r, s, t, u, w et x

Ainsi "**mo**" signifierait les délimiteurs multiples et les délimiteurs en début et en fin sont détectés au lieu d'être habituellement délaissés et **once** les arguments de la charliste et des modifieurs ne sont exécuté qu'une seule fois.

Pour les étudier tous, aller en :

<http://support.sas.com/documentation/cdl/en/lrdict/63026/HTML/default/a000214639.htm>

## Les fonctions numériques ou statistiques

- Contrairement à la fonction **nmiss** qui ne peut porter que sur des variables numériques, la nouvelle fonction **cmiss**(arg1, arg2, arg3...) assurera la détection des valeurs manquantes aussi bien numériques que caractères.

<pre>data test; length var1 \$2 yy 8 var2 8 zz \$1 xx 8 tt \$8; input var1 yy var2 zz xx tt; cards; 1 2 3 4 5 6 . . . . . 7 . . . . . 8 . 9 n . z . 3 . . . . ;</pre>	<pre>data subset; set test; array nn[*] _numeric_; array cc[*] _character_; if cmiss(of _all_) &lt; dim(nn) + dim(cc); run; NOTE: 5 observation(s) lue(s) dans la table WORK.TEST. NOTE: La table WORK.SUBSET a 4 observation(s) et 6 variable(s).</pre>
---	--

- Une longue liste de fonctions ou routine call dans le domaine math/stat a été incluse dans cette version.

Sans nous étendre sur elles, en voici la liste :

-Allcomb allperm arcosh arsinh artanh euclid gcd (+grand commun diviseur d'entiers) lcm lcomb lexcomb (etc dans le domaine lexicographique) lfact loglpx lperm lpnorm msplint

-Le fonction **divide**(x,y), évitant une division par zéro, est enfin documentée. Découvrez y le sens du **.I** (infini), du **.M** (infini négatif) et du **.\_**

-**Sumabs**(arg1,arg2...) serait une fonction sommant les valeur absolues d'arguments non-manquants

<pre>data _null_; v1=162; v2=.25; v3=456; v4=.A; v5=-1; e=sumabs(of v1-v5);put e=; f=sum(of v1-v5);put f=; g=sumabs(25,.,.Z,56,-1); put g=; h=sum(25,.,.Z,56,-1); put h=; run;</pre>	<pre>e=619.25 f=617.25 g=82 h=80</pre>
--	--

## Des fonctions spécialisées

- La nouvelle fonction **geodist**(lat1,long1,lat2,long2,<options>) permettra de calculer une distance en M miles, K kilomètres, R radiants ou D degrés.

\*Washington étant 39°91'N 77°02'W la latitude est positive ==> 39.91  
et la longitude est négative en rapport Méridien Greenwich ==> -77.02;

```
data _null_;
input nom :$21. lat1 long1 lat2 long2 ;
dist=geodist(lat1,long1,lat2,long2,'K');
put nom $char25. dist 10.4 ' Kilomètres';
cards;
Teheran-Jerusalem 35.44 51.30 31.47 35.12
Washington-Moscou 39.91 -77.02 55.45 37.35
Buenos_Aires-Brasilia -36.30 -60.00 -15.47 -47.55
;
run;
Teheran-Jerusalem 1565.2504 Kilomètres
Washington-Moscou 7757.9106 Kilomètres
Buenos_Aires-Brasilia 2617.6705 Kilomètres
```

- La fonction **unicode**(<instr>(<Unicode type>)) servira à traduire une chaîne unicode en caractères Sas dans l'encodage courant de la session et **unicodec**(...) fera l'inverse  

```
data a;a="é";unia=unicodec(a,('utf8'));put unia=;run; → unia=Ã©
```

Il en existe d'autres à explorer dans le manuel NLS.

## Des fonctions liées aux dates

- Intéressante est la nouvelle fonction **nwkdome**(*nième occurrence, jour-semaine, mois, année*) qui permet de calculer par exemple la date du passage de la collecte des encombrants : tous les premiers jeudi du mois ou celle de la paye car fixée le dernier vendredi du mois  
Dans cette fonction, on part d'un jour-semaine selon la définition de Sas : le dimanche = le chiffre 1.

```
data _null_;
encombrant=nwkdome(1,5,8,2010);put encombrant eurdfwvx.;
paie=nwkdome(5,6,12,2010);put paie eurdfwvx.;
run;
                Jeudi 5 août 2010
                Vendredi 31 décembre 2010
```

- J'avoue avoir peu compris l'utilité des fonctions d'intervalle **intget**(date1,date2,date3) **Intshift**('intervalle') et **intfit**(date1,date2,'d' ou 'dt' ou 'obs')

<pre>data _null_; d1='1jan2010'd; d2='28feb2010'd; d3='31dec2010'd; d4='1jan2011'd; d12=intfit(d1,d2,'d');put d12=; d13=intfit(d1,d3,'d');put d13=; d14=intfit(d1,d4,'d');put d14=; sh=intshift('week'); put sh=;run;</pre>	<pre>d12=DAY58.52 d13=WEEK52.69 d14=YEAR sh=DAY</pre>
---	---

- quelques modifications ont été faites à :

Un 5<sup>ème</sup> argument optionnel pour **INTCK**(interval<multiple><.shift-index>, start-from, increment, <alignment>) 'D' discrete ou 'C' continuous pour une valeur d'alignement dans l'intervalle pour la position des dates SAS

Intck compte le nombre de semaines (à la mode SAS commençant le dimanche week) ou (grâce au nouveau mot-clé iso 8601 **weekv** démarrant de façon standard le lundi auquel on a ajouté la spécification du 2<sup>ème</sup> jour weekv.2 donc le mardi.

Sas a -dans les deux cas- évalués ce nombre d'intervalles selon que la nouvelle spécification additionnelle était Continuous ou Discrete en 25 semaines ou 26 semaines.

Le 1<sup>er</sup> janvier était un vendredi et le 1<sup>er</sup> juillet était un jeudi

```
data _null_;
d1=intck('weekv.2','01jan2010'd,'01jul2010'd,'C');
d2=intck('weekv.2','01jan2010'd,'01jul2010'd,'D');
d3=intck('week','01jan2010'd,'01jul2010'd,'C');
d4=intck('week','01jan2010'd,'01jul2010'd,'D');
put d1 +2 d2 +2 d3 +2 d4 ;
run;
                == > 25 26 25 26
```

La base de calcul dans la fonction **datdif**(début,fin,base) s'est enrichie et propose maintenant '30/360' 'ACT/ACT' 'ACT/360' 'ACT/365'

**Les fonctions informatives**

<http://support.sas.com/documentation/cdl/en/sqlproc/62086/HTML/default/a001385596.htm>

La version 9.2 a créé de nouvelles vues de travail sashelp.vdest, vfunc et vformat.

```
proc sql ; create view listfunc as select * from dictionary.functions;
quit;
```

	Source du format	Nom de la fonction	Args mini pour la fonction	Args maxi pour la fonction	Type de fonction	Attributs de l'argument	Type de mise en oeuvre
	source	fncname	minarg	maxarg	fnctype	fnargs	fnprod
850	B	VERIFY	2	0	N	146	X
851	B	VFORMAT	1	1	C	3	B

Une interrogation particulière retournerait les fonctions cmiss et nmiss.

```
proc sql ;select * from dictionary.functions
where fncname like '_MISS'; quit;
```

Pour le format, ceci avait déjà été introduit dans la SP4 de la version 9.1.3

```
proc sql ; create view listformat as select * from dictionary.formats;
quit;
```

	fmtname	fmttype	source	minw	mind	maxw	maxd	defw	defd
	Nom du format	Type du format	Source du format	Largeur minimum	Largeur décimale minimum	Largeur maximum	Largeur décimale maximum	Largeur par défaut	Largeur décimale par défaut
981	YYMMDD	I	B	6	0	32	31	6	0
982	YYMMDDDB	F	B	2	0	10	9	8	0

Ce qui se lit ↓ c'est un informat du Sas/Base etc :

158	BZ	I	B	1	0	32	31	1	0
-----	----	---	---	---	---	----	----	---	---

En consultant cette longue liste, on peut se rappeler que cette fonction sert à lire des blancs comme des zéros.

```
proc sql;
select * from dictionary.formats where fmtname like 'COM%';quit;
```

vous informera sur des détails des in/formats de type Comma...

<pre>ods listing ; proc sql; select * from dictionary.destinations; quit; ods listing close; ods html file="d:\tabsas\vdest.htm"; ods rtf file="d:\tabsas\vdest.rtf"; ods tagsets.Excelxp file="d:\tabsas\vdest.xml" style=ocean; proc sql; select * from dictionary.destinations; quit; ods all close;</pre>	<table border="1"> <thead> <tr> <th>Destination</th> <th>Style</th> </tr> </thead> <tbody> <tr> <td>LISTING</td> <td>Listing</td> </tr> <tr> <td>LISTING_GRAPHICS_EDITOR</td> <td>Default</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Destination</th> <th>Style</th> </tr> </thead> <tbody> <tr> <td>HTML</td> <td>Default</td> </tr> <tr> <td>RTF</td> <td>Rtf</td> </tr> <tr> <td>TAGSETS.EXCELXP</td> <td>Ocean</td> </tr> </tbody> </table> <p>Bref une information sur les destinations ouvertes et les templates de styles utilisés.</p>	Destination	Style	LISTING	Listing	LISTING_GRAPHICS_EDITOR	Default	Destination	Style	HTML	Default	RTF	Rtf	TAGSETS.EXCELXP	Ocean
Destination	Style														
LISTING	Listing														
LISTING_GRAPHICS_EDITOR	Default														
Destination	Style														
HTML	Default														
RTF	Rtf														
TAGSETS.EXCELXP	Ocean														

<pre>libname library "d:\data"; data work.resultats(keep=var valcol valformat); set a.classe; array __vnum [*] _numeric_; length var \$ 25 valcol \$ 25 valformat \$ 25; do i = 1 to dim(__vnum); var = vname(__vnum[i]); valcol = put(__vnum[i],best25.-1); valformat= putn(__vnum[i],vformat(__vnum[i])); output; end; run;</pre>	<table border="1"> <thead> <tr> <th></th> <th>var</th> <th>valcol</th> <th>valformat</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Age</td> <td>14</td> <td>groupé n</td> </tr> <tr> <td>2</td> <td>taille</td> <td>69</td> <td>69</td> </tr> <tr> <td>3</td> <td>poidsplus</td> <td>112.5</td> <td>112.5</td> </tr> <tr> <td>4</td> <td>j</td> <td>15</td> <td>15</td> </tr> <tr> <td>5</td> <td>m</td> <td>1</td> <td>1</td> </tr> <tr> <td>6</td> <td>daten</td> <td>12068</td> <td>12068</td> </tr> <tr> <td>7</td> <td>datef</td> <td>12068</td> <td>15/01/1993</td> </tr> <tr> <td>8</td> <td>Age</td> <td>13</td> <td>groupé n</td> </tr> <tr> <td>9</td> <td>taille</td> <td>56.5</td> <td>56.5</td> </tr> </tbody> </table>		var	valcol	valformat	1	Age	14	groupé n	2	taille	69	69	3	poidsplus	112.5	112.5	4	j	15	15	5	m	1	1	6	daten	12068	12068	7	datef	12068	15/01/1993	8	Age	13	groupé n	9	taille	56.5	56.5
	var	valcol	valformat																																						
1	Age	14	groupé n																																						
2	taille	69	69																																						
3	poidsplus	112.5	112.5																																						
4	j	15	15																																						
5	m	1	1																																						
6	daten	12068	12068																																						
7	datef	12068	15/01/1993																																						
8	Age	13	groupé n																																						
9	taille	56.5	56.5																																						

## 10.3 Modifications dans les procédures de base (hormis graphique et ods graphics =volume 2)

### 10.3.1 Proc Append :

- Avec l'option *nowarn*, le message d'avertissement apparaissant à l'utilisation de *force* est supprimé.

### 10.3.2 Proc Cimport :

Si vous disposez d'un fichier de transport (du genre cport et non xport) créé avant la 9.2, ce fichier ne contient en interne aucune info sur son encodage. Vous devez, pour pouvoir l'importer correctement avec Cimport, aller à la pêche pour les informations suivantes

l'environnement de la session (-source-) par ex Windows

- la version de Sas
- le nom du fichier de transport
- l'encodage utilisé pour les caractères par ex wlatin1
- le caractère national de ces caractères par ex en\_US

Si maintenant dans votre session ( -cible-), vous travaillez sous Sas UTF-8, vous pourrez importer cet ancien fichier de transport, s'il a déjà été créé en utf-8 en utilisant la nouvelle option `Isfileutf8=yes`

```
proc cimport isfileutf8=yes infile=importin library=target memtype=data;
```

pour préciser le caractère utf-8 de l'encodage du fichier de transport.

Dans la plupart des autres cas d'impossibilité, Sas invitera à démarrer avec une autre définition de l'option locale, car ceci entraîne d'autres définitions par défaut de *encoding=*, *dflang=*, *datestyle=* et *papersize=*.

### 10.3.3 Proc Contents :

- `COLLATE`, `CASECOLLATE`, `IGNORECASE` et `VARNUM` sont les options du *order=*  
`proc contents data=p.p_1343_maj out=a order=collate ;run;`

**Varnum** imprimait déjà la liste de variables dans l'ordre de leur création.

**Ignorecase** imprimait la liste dans l'ordre alphabétique quelque soit la casse.

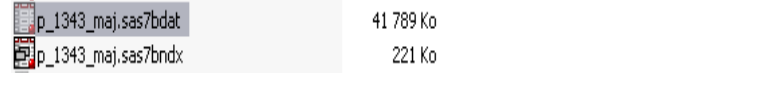
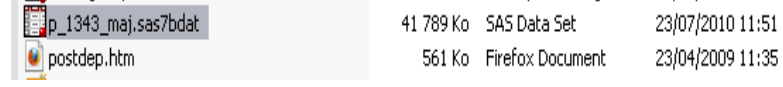
**Collate** imprimera cette liste dans l'ordre alphabétique en commençant par toutes les majuscules et seulement ensuite les minuscules, tandis que **casecollate** imprimera une liste alphabétique même s'il y a eu des saisies variées de noms mixtes, du numérique et des noms de longueurs différentes.

### 10.3.4 Proc Corr :

- `ID var` ; permet l'ajout de variables illustratives en vue d'identifier les observations dans les scatter plots/nuages de points

### 10.3.5 Proc Datasets :

- `REBUILD` permet la *reconstruction* des index et contraintes d'intégrité pour les fichiers à partir de la version 7, ou pour leur *suppression*. L'instruction (pour sa suppression) s'écrit ainsi :

<pre>proc datasets library=p; modify p_1343_maj; index create ident1;quit;</pre>	
<pre>proc datasets library=p; rebuild p_1343_maj /noindex; quit;</pre>	

Le `dldmgaction=noindex` vu page 176 aura laissé la table en mode input seul. Le rebuild rétablira l'état normal.

- Renommer une table Sas *etc* peut aussi se faire hors procédure avec la nouvelle fonction `rename(anciennom,nouveaunom,'type')` grâce au choix du type adéquat (*data catalog view* ou *file*)

```
data _null_ ; rc=rename ("d:\data\f.sas7bcat", "d:\data\formats.sas7bcat", 'file') ; run;
```

\* correction du nom physique d'un fichier f.sas7bdat en formats.sas7bcat ;

```
data _null_ ; rc=rename ('work.gseg.gmap1.grseg', 'gmap', 'catalog');put rc=; run;
```

\* correction en cours de travail du nom de graphique Sas gmap1 ici dans un catalogue gseg de la work en gmap;

- `NOPRINT`

### 10.3.6 Proc Freq :

- Le tableau croisé des fréquences possède maintenant un template propre (vu page 125)
- En ayant recours à l'ODS graphics, la procédure freq peut produire sous réserve de la spécification d'un /plot=

ODS Graph Name	Plot description	Tables statement options
CumFreqPlot	Cumulative frequency plot	One-Way table request
DeviationPlot	Deviation plot	CHISQ and one-way table request
FreqPlot	Frequency plot	Any table request
KappaPlot	Kappa plot	AGREE (h x r x r table)
ORPlot	Odds ratio plot	MEASURES or RELRISK (hx2x2 table)
WtKappaPlot	Weighted kappa plot	AGREE 5h x r xr table, r > 2)

- Une série de petites additions/options statistiques ont été implantées
- La présentation de la sortie du proc freq a été modifiée et les cellules sont empilées

```
ods listing close;
ods html file="d:\tabsas\stackedcell.htm";
proc freq data=sashelp.prdsale;
tables COUNTRY*REGION*DIVISION*QUARTER*PRODUCT*PRODTYPE;
run;
ods html close;
ods listing;
```

### 10.3.7 Proc Print :

- Le saut d'une ligne blanche toutes les *n* observations est permis grâce à l'option *blankline=*.

```
proc print data=sashelp.class blankline=5;run;
```

- L'option *sumlabel* s'appliquera en cas d'utilisation de l'instruction *sum*.

```
proc sort data=sashelp.cars out=cars;by origin make;run;
proc print data=cars noobs sumlabel; by origin make;
var model invoice;sum invoice;
label origin='Continent' make='Fabriquant';
run;
```

sans sumlabel		avec sumlabel	
Continent=USA Fabriquant=Saturn		Continent=USA Fabriquant=Saturn	
Model	Invoice	Model	Invoice
VUE	\$19,238	VUE	\$19,238
Ion1 4dr	\$10,319	Ion1 4dr	\$10,319
Ion2 4dr	\$13,393	Ion2 4dr	\$13,393
Ion3 4dr	\$14,811	Ion3 4dr	\$14,811
Ion2 quad coupe 2dr	\$13,904	Ion2 quad coupe 2dr	\$13,904
Ion3 quad coupe 2dr	\$15,299	Ion3 quad coupe 2dr	\$15,299
L300-2 4dr	\$19,801	L300-2 4dr	\$19,801
L300 2	\$21,779	L300 2	\$21,779
-----	-----	-----	-----
Make	\$128,544	Fabriquant	\$128,544
Origin	\$3814553	Continent	\$3814553
	=====		=====
	12846292		12846292

### 10.3.8 Proc Rank :

- Nouvelle option pour le calcul des rangs/score pour les données liées (tied) *dense* (autre high, low ou le défaut *mean*)

### 10.3.9 Proc Report :

- La statistique **mode** est maintenant autorisée dans les procédures *means*, *report* et *tabulate* et **probt** devient un alias de *pvt*.
- Bypageno=n pour redéfinir le numéro de page entre les ruptures de by
- Addition de l'option **spanrows** à la procédure permettant pour les destinations pdf, rtf et markup de créer une case globale pour toute modalité chapeautant une distribution dépendante :

```
ods rtf file="d:\temp\rtf_output_j.rtf" style=journal;
proc report data=test1 nowd spanrows;
  title ' SPANROWS avec le style JOURNAL ';
  column var_c var_n ;
  define var_c / order style(column)={vjust=m};
  define var_n / sum;
run;
ods _all_ close;
```

avec SPANROWS avec le style ocean		sans SPANROWS avec le style ocean	
<i>var_c</i>	<i>var_n</i>	<i>var_c</i>	<i>var_n</i>
Charles	1.52	Charles	1.52
	1.35		1.35
Louis	1.52	Louis	1.52
	1.35		1.35

- De même qu'en *proc tabulate*, on pourra utiliser ces attributs de style :

```
ods html file='border.html' style=sasweb;
ods pdf file='border.pdf';
ods rtf file='border.rtf';

proc report data=sashelp.class(obs=2) nowd
  style(report)={borderbottomcolor=cyan borderbottomwidth=5
    bordertopcolor=yellow bordertopwidth=5
    borderleftcolor=magenta borderleftwidth=5
    borderrightcolor=black borderrightwidth=5};

title "1) REPORT: borderbottomcolor= and borderbottomwidth";
column name age height;
define name / style(column)={borderbottomcolor=red borderbottomwidth=5
  bordertopcolor=green bordertopwidth=5
  borderleftcolor=blue borderleftwidth=5
  borderrightcolor=pink borderrightwidth=5};

run;
ods _all_ close;
```

#### 1) REPORT: borderbottomcolor= and borderbottomwidth

Name	Age	Height
Alfred	14	69
Alice	13	56.5



### 10.3.10 Proc Sort :

- L'option *presorted* est disponible quand il y a forte supposition d'existence d'un tri préalable. Elle évitera un nouveau tri. Voir aussi *sortvalidate* p 176.
- Depuis quelques années, Sas, face à l'internationalisation de sa distribution, a intégré progressivement les problèmes liés aux séquences de tri et a adopté la norme ICU (International Components for Unicode) et son algorithme UCA alias chez Sas : *linguistic*.

Outre le détail présenté ci après, vous pouvez jetez un œil sur <http://www.repole.com/dinosaur/linguistic.html> et approfondir la référence détaillée sur ces questions en lisant ce document [http://support.sas.com/resources/papers/linguistic\\_collation.pdf](http://support.sas.com/resources/papers/linguistic_collation.pdf)

Cette nouveauté entraine aussi la dévaluation des TRANTAB et le classement de seulement 256 caractères. Pour rappel, le tri ascii met en premier les caractères majuscules. Les lettres accentuées sont classées après A\_Z. C'est ainsi qu'apparaît dans le proc sort l'option *sortseq=linguistic* qui combiné avec le contenu de l'option système *locale*, permet d'appliquer une séquence non anglo-saxonne. Chaque langue aura son préreglage.

Il sera parfois encore très rarement nécessaire de préciser l'option système *nobysorted* ou le *notsorted* dans le *by*, car ce type de tri s'ordonne différemment de l'usage habituel du *by*.

Des options complémentaires mise entre parenthèses sont possibles

- *strength*= 1 ce premier niveau/level porte sur les différences de bases « a » < « b »  
2 « as » < « à » < « at » introduction des différences secondaires  
3 « ai » < « Ai » < « aï » ajout de la différence majuscule/minuscule  
4 « ab » < « a-b » < « aB » introduction supplémentaire de la ponctuation  
5 ou *identical* Sas cite ici comme exemple la « *cantillation hébraïque* »

- *numeric\_collation*=

On les chiffres seront classés par valeur numérique «8 rue... » < «48 rue... »  
Off tri selon la valeur caractère «48 rue... » < «8 rue... »

- *locale*= "fr\_FR" ne sera pas entièrement identique à "fr\_BE" de même pour "nl\_BE" et "nl\_NL"

- *collate*= généralement réglé par locale, mais avec l'allemand on mettra "phonebook" et "traditional" pour l'espagnol

- *case\_first*= Upper les majuscules en premier Lower les minuscules en premier

- *alternate\_handling*= Shifted provoque la minimisation des espaces et de la ponctuation

```
data moimin;
length mois $10;
infile cards delimiter=',';
input mois info1;
cards;
janvier, 1
mars, 3
février, 2
mai, 5
avril, 4
juin, 6
juillet, 7
âoût, 8
septembre, 9
octobre, 10
novembre, 11
décembre, 12
;Run ;
```

```
data moimaj;
length mois info2 $10 ;
infile cards delimiter=',';
input mois info2;
cards;
DECEMBRE, Hiver
OCTOBRE, Automne
NOVEMBRE, Automne
SEPTEMBRE, Automne
JUILLET, Eté
AOÛT, Eté
JUIN, Eté
MAI, Printemps
AVRIL, Printemps
MARS, Printemps
FÉVRIER, Hiver
JANVIER, Hiver
;Run ;
```

<pre> proc sort data=moimaj out=maj ; by mois;run; proc sort data=moimin out=min; by mois;run; data m;merge maj min; by mois; run; proc print; run;***** ; proc sort data=moimaj out=maj1   sortseq=linguistic(strength=2); by mois;run; proc sort data=moimin out=min1   sortseq=linguistic(strength=2) ; by mois;run; data m1;merge maj1 min1; by mois; run; proc print; run; ‡ ***** ;  proc sort data=moimaj out=maj2   sortseq=linguistic(strength=1); by mois;run; proc sort data=moimin out=min2   sortseq=linguistic(strength=1) ; by mois;run; data m2;merge maj2 min2; by mois; run; proc print; run; </pre>	<table border="1"> <thead> <tr> <th>Obs</th> <th>mois</th> <th>info2</th> <th>info1</th> </tr> </thead> <tbody> <tr><td>1</td><td>AOÛT</td><td>Eté</td><td>.</td></tr> <tr><td>2</td><td>AVRIL</td><td>Printemps</td><td>.</td></tr> <tr><td>3</td><td>DECEMBRE</td><td>Hiver</td><td>.</td></tr> <tr><td>4</td><td>FÉVRIER</td><td>Hiver</td><td>.</td></tr> <tr><td>5</td><td>JANVIER</td><td>Hiver</td><td>.</td></tr> <tr><td>6</td><td>JUILLET</td><td>Eté</td><td>.</td></tr> <tr><td>7</td><td>JUIN</td><td>Eté</td><td>.</td></tr> <tr><td>8</td><td>MAI</td><td>Printemps</td><td>.</td></tr> <tr><td>9</td><td>MARS</td><td>Printemps</td><td>.</td></tr> <tr><td>10</td><td>NOVEMBRE</td><td>Automne</td><td>.</td></tr> <tr><td>11</td><td>OCTOBRE</td><td>Automne</td><td>.</td></tr> <tr><td>12</td><td>SEPTEMBRE</td><td>Automne</td><td>.</td></tr> <tr><td>13</td><td>août</td><td></td><td>8</td></tr> <tr><td>14</td><td>avril</td><td></td><td>4</td></tr> <tr><td>15</td><td>décembre</td><td></td><td>12</td></tr> <tr><td>16</td><td>février</td><td></td><td>2</td></tr> <tr><td>17</td><td>janvier</td><td></td><td>1</td></tr> <tr><td>18</td><td>juillet</td><td></td><td>7</td></tr> <tr><td>19</td><td>juin</td><td></td><td>6</td></tr> <tr><td>20</td><td>mai</td><td></td><td>5</td></tr> <tr><td>21</td><td>mars</td><td></td><td>3</td></tr> <tr><td>22</td><td>novembre</td><td></td><td>11</td></tr> <tr><td>23</td><td>octobre</td><td></td><td>10</td></tr> <tr><td>24</td><td>septembre</td><td></td><td>9</td></tr> <tr><td>25</td><td>âoût</td><td></td><td>8</td></tr> </tbody> </table> <p>Le tri classique maj/ min/ caractère spéciaux sans linguistic.</p>	Obs	mois	info2	info1	1	AOÛT	Eté	.	2	AVRIL	Printemps	.	3	DECEMBRE	Hiver	.	4	FÉVRIER	Hiver	.	5	JANVIER	Hiver	.	6	JUILLET	Eté	.	7	JUIN	Eté	.	8	MAI	Printemps	.	9	MARS	Printemps	.	10	NOVEMBRE	Automne	.	11	OCTOBRE	Automne	.	12	SEPTEMBRE	Automne	.	13	août		8	14	avril		4	15	décembre		12	16	février		2	17	janvier		1	18	juillet		7	19	juin		6	20	mai		5	21	mars		3	22	novembre		11	23	octobre		10	24	septembre		9	25	âoût		8
Obs	mois	info2	info1																																																																																																						
1	AOÛT	Eté	.																																																																																																						
2	AVRIL	Printemps	.																																																																																																						
3	DECEMBRE	Hiver	.																																																																																																						
4	FÉVRIER	Hiver	.																																																																																																						
5	JANVIER	Hiver	.																																																																																																						
6	JUILLET	Eté	.																																																																																																						
7	JUIN	Eté	.																																																																																																						
8	MAI	Printemps	.																																																																																																						
9	MARS	Printemps	.																																																																																																						
10	NOVEMBRE	Automne	.																																																																																																						
11	OCTOBRE	Automne	.																																																																																																						
12	SEPTEMBRE	Automne	.																																																																																																						
13	août		8																																																																																																						
14	avril		4																																																																																																						
15	décembre		12																																																																																																						
16	février		2																																																																																																						
17	janvier		1																																																																																																						
18	juillet		7																																																																																																						
19	juin		6																																																																																																						
20	mai		5																																																																																																						
21	mars		3																																																																																																						
22	novembre		11																																																																																																						
23	octobre		10																																																																																																						
24	septembre		9																																																																																																						
25	âoût		8																																																																																																						

<table border="1"> <thead> <tr> <th></th> <th>mois</th> <th>info2</th> <th>info1</th> </tr> </thead> <tbody> <tr><td>1</td><td>août</td><td>Eté</td><td>8</td></tr> <tr><td>2</td><td>âoût</td><td></td><td>8</td></tr> <tr><td>3</td><td>avril</td><td>Printemps</td><td>4</td></tr> <tr><td>4</td><td>DECEMBRE</td><td>Hiver</td><td>.</td></tr> <tr><td>5</td><td>décembre</td><td></td><td>12</td></tr> <tr><td>6</td><td>février</td><td>Hiver</td><td>2</td></tr> <tr><td>7</td><td>janvier</td><td>Hiver</td><td>1</td></tr> <tr><td>8</td><td>juillet</td><td>Eté</td><td>7</td></tr> <tr><td>9</td><td>juin</td><td>Eté</td><td>6</td></tr> <tr><td>10</td><td>mai</td><td>Printemps</td><td>5</td></tr> <tr><td>11</td><td>mars</td><td>Printemps</td><td>3</td></tr> <tr><td>12</td><td>novembre</td><td>Automne</td><td>11</td></tr> <tr><td>13</td><td>octobre</td><td>Automne</td><td>10</td></tr> <tr><td>14</td><td>septembre</td><td>Automne</td><td>9</td></tr> </tbody> </table> <p>Le strength=2 a maintenu les différences secondaires par rapport aux caractères de base. F/f ont été neutralisés et confondus car le é était présent dans les 2 cas en seconde position, tandis que D/d du 12<sup>ème</sup> mois est resté distinct vu la nouvelle différence E/é. De même pour le 8<sup>ème</sup>.</p>		mois	info2	info1	1	août	Eté	8	2	âoût		8	3	avril	Printemps	4	4	DECEMBRE	Hiver	.	5	décembre		12	6	février	Hiver	2	7	janvier	Hiver	1	8	juillet	Eté	7	9	juin	Eté	6	10	mai	Printemps	5	11	mars	Printemps	3	12	novembre	Automne	11	13	octobre	Automne	10	14	septembre	Automne	9	<table border="1"> <thead> <tr> <th></th> <th>mois</th> <th>info2</th> <th>info1</th> </tr> </thead> <tbody> <tr><td>1</td><td>âoût</td><td>Eté</td><td>8</td></tr> <tr><td>2</td><td>août</td><td>Eté</td><td>8</td></tr> <tr><td>3</td><td>avril</td><td>Printemps</td><td>4</td></tr> <tr><td>4</td><td>décembre</td><td>Hiver</td><td>12</td></tr> <tr><td>5</td><td>février</td><td>Hiver</td><td>2</td></tr> <tr><td>6</td><td>janvier</td><td>Hiver</td><td>1</td></tr> <tr><td>7</td><td>juillet</td><td>Eté</td><td>7</td></tr> <tr><td>8</td><td>juin</td><td>Eté</td><td>6</td></tr> <tr><td>9</td><td>mai</td><td>Printemps</td><td>5</td></tr> <tr><td>10</td><td>mars</td><td>Printemps</td><td>3</td></tr> <tr><td>11</td><td>novembre</td><td>Automne</td><td>11</td></tr> <tr><td>12</td><td>octobre</td><td>Automne</td><td>10</td></tr> <tr><td>13</td><td>septembre</td><td>Automne</td><td>9</td></tr> </tbody> </table> <p>Le strength=primary a provoqué la neutralisation maj/min ; cependant le second enregistrement du 8ème mois est resté distinct mais a été bien complété.</p>		mois	info2	info1	1	âoût	Eté	8	2	août	Eté	8	3	avril	Printemps	4	4	décembre	Hiver	12	5	février	Hiver	2	6	janvier	Hiver	1	7	juillet	Eté	7	8	juin	Eté	6	9	mai	Printemps	5	10	mars	Printemps	3	11	novembre	Automne	11	12	octobre	Automne	10	13	septembre	Automne	9
	mois	info2	info1																																																																																																																		
1	août	Eté	8																																																																																																																		
2	âoût		8																																																																																																																		
3	avril	Printemps	4																																																																																																																		
4	DECEMBRE	Hiver	.																																																																																																																		
5	décembre		12																																																																																																																		
6	février	Hiver	2																																																																																																																		
7	janvier	Hiver	1																																																																																																																		
8	juillet	Eté	7																																																																																																																		
9	juin	Eté	6																																																																																																																		
10	mai	Printemps	5																																																																																																																		
11	mars	Printemps	3																																																																																																																		
12	novembre	Automne	11																																																																																																																		
13	octobre	Automne	10																																																																																																																		
14	septembre	Automne	9																																																																																																																		
	mois	info2	info1																																																																																																																		
1	âoût	Eté	8																																																																																																																		
2	août	Eté	8																																																																																																																		
3	avril	Printemps	4																																																																																																																		
4	décembre	Hiver	12																																																																																																																		
5	février	Hiver	2																																																																																																																		
6	janvier	Hiver	1																																																																																																																		
7	juillet	Eté	7																																																																																																																		
8	juin	Eté	6																																																																																																																		
9	mai	Printemps	5																																																																																																																		
10	mars	Printemps	3																																																																																																																		
11	novembre	Automne	11																																																																																																																		
12	octobre	Automne	10																																																																																																																		
13	septembre	Automne	9																																																																																																																		

<pre> proc sort data=moimaj out=maj3   sortseq=linguistic; by mois;run; proc sort data=moimin out=min3   sortseq=linguistic; by mois;run; data m2;merge maj3 min3; by mois; run; proc print; run; </pre> <p>Voici le tri <b>linguistic</b> par défaut : Min/maj et non confusion des lettres accentuées qui restent bien rangées dans une séquence alphabétique</p>	<table border="1"> <tbody> <tr><td>août</td></tr> <tr><td>AOÛT</td></tr> <tr><td>âoût</td></tr> <tr><td>avril</td></tr> <tr><td>AVRIL</td></tr> <tr><td>DECEMBRE</td></tr> <tr><td>décembre</td></tr> <tr><td>février</td></tr> <tr><td>FÉVRIER</td></tr> <tr><td>janvier</td></tr> <tr><td>JANVIER</td></tr> <tr><td>juillet</td></tr> <tr><td>JUILLET</td></tr> </tbody> </table>	août	AOÛT	âoût	avril	AVRIL	DECEMBRE	décembre	février	FÉVRIER	janvier	JANVIER	juillet	JUILLET	<table border="1"> <tbody> <tr><td>juin</td></tr> <tr><td>JUIN</td></tr> <tr><td>mai</td></tr> <tr><td>MAI</td></tr> <tr><td>mars</td></tr> <tr><td>MARS</td></tr> <tr><td>novembre</td></tr> <tr><td>NOVEMBRE</td></tr> <tr><td>octobre</td></tr> <tr><td>OCTOBRE</td></tr> <tr><td>septembre</td></tr> <tr><td>SEPTEMBRE</td></tr> </tbody> </table>	juin	JUIN	mai	MAI	mars	MARS	novembre	NOVEMBRE	octobre	OCTOBRE	septembre	SEPTEMBRE
août																											
AOÛT																											
âoût																											
avril																											
AVRIL																											
DECEMBRE																											
décembre																											
février																											
FÉVRIER																											
janvier																											
JANVIER																											
juillet																											
JUILLET																											
juin																											
JUIN																											
mai																											
MAI																											
mars																											
MARS																											
novembre																											
NOVEMBRE																											
octobre																											
OCTOBRE																											
septembre																											
SEPTEMBRE																											

### 10.3.11 Proc Tabulate :

Outre ce qui a déjà été déjà mentionné *en proc report*, la notation abrégée est acceptée pour les listes de variables (v1-v10 et var :).

### 10.3.12 Proc Transpose :

Les enrichissements de la proc tranpose portent sur la possibilité d'utiliser de  multiples Id variables. Ce qui engendre l'apparition de l'option  delimiter= pour permettre l'insertion de caractère(s) (valide) entre les valeurs de ces Id variables dans la création du nom des colonnes. De plus, une option  suffix= est également ajoutée.

<pre> data install; input log \$5. vers :\$5. mois \$ jour \$ n;   datalines; sas  913 jul fin  73 sas  923 jul fin  01 sas  913 aug deb  72 sas  923 aug deb  02 sas  913 aug mi   69 sas  923 aug mi   05 sas  913 aug fin  67 sas  923 aug fin  07 stata10IC jul fin  07 stata11IC jul fin  30 stata10IC aug deb  06 stata11IC aug deb  31 stata10IC aug mi   04 stata11IC aug mi   33 stata10IC aug fin  01 stata11IC aug fin  36 ; </pre>	<pre> proc transpose data=install out=reste let delimiter=et suffix=M;   by log;   id  mois vers; run; proc print heading=h width=min;run; </pre> <p>Obligation d'utiliser l'option <b>let</b> qui seule autorise des valeurs multiples dans le même bygroup : autrement un message d'erreur apparaît</p> <p>ERROR: La valeur de la variable ID "auget913M" apparaît deux fois dans le même groupe BY</p>
--	---

Avec comme résultat, cette vue en fin de mois sur mon stock glissant de licences Sas et Stata entre les sous-versions.

Obs	log	_NAME_	julet913M	julet923M	auget913M	auget923M
1	sas	n	73	1	67	7
2	stata	n	.	.	.	.

Obs	julet10ICM	julet11ICM	auget10ICM	auget11ICM
1	.	.	.	.
2	7	30	1	36

### 10.3.13 Proc Univariate :

Nous reviendrons sur cette procédure dans le volume 2 pour les aspects graphiques.

Signalons cependant des instructions nouvelles (outre les *qqplot* et *probplot* déjà existantes )

**cdfplot** (fonction de distribution cumulative observée) comportant la possibilité d'y adjoindre les distributions théoriques principales (Beta, Exponentielle, Gamma, Lognormal, Normal et Weibull )

et

**ppplot** (probability-probability plot ou percent plot) avec des options analogues.

## 10.4 Nouvelles procédures de base:

### 10.4.1 Proc Javainfo :

Depuis plusieurs versions, les mises à jour automatique de Java (Sun puis Oracle) ont semé un peu de confusion, de telle sorte que Sas pour faire le point a créé :

```
proc javainfo;run;
PFS_TEMPLATE = C:\Program Files\SAS\SASFoundation\9.2\core\sasmisc\qrpfstpt.xml
java.class.path = C:\PROGRA~1\SAS\SASVER~1\9.2\eclipse\plugins\SASLAU~1.JAR
java.class.version = 49.0
java.runtime.name = Java(TM) 2 Runtime Environment, Standard Edition
java.runtime.version = 1.5.0_12-b04
java.security.auth.login.config = C:\Program
Files\SAS\SASFoundation\9.2\core\sasmisc\sas.login.config
java.security.policy = C:\Program Files\SAS\SASFoundation\9.2\core\sasmisc\sas.policy
java.specification.version = 1.5
java.system.class.loader = com.sas.app.AppClassLoader
java.vendor = Sun Microsystems Inc.
java.version = 1.5.0_12
java.vm.name = Java HotSpot(TM) Client VM
java.vm.specification.version = 1.0
java.vm.version = 1.5.0_12-b04
sas.app.class.path = C:\PROGRA~1\SAS\SASVER~1\9.2\eclipse\plugins\tkjava.jar
sas.ext.config = C:\Program Files\SAS\SASFoundation\9.2\core\sasmisc\sas.java.ext.config
sas.jre.libjvm = C:\PROGRA~1\Java\JRE15~1.0_1\bin\client\jvm.dll
tkj.app.launch.config = C:\Program Files\SAS\SASFoundation\9.2\picklist
user.country = FR
user.language = fr
```

Si vous avez un problème de conflit, grâce à Ron Fehd ... [http://www.sascommunity.org/wiki/Java\\_JRE\\_1.5](http://www.sascommunity.org/wiki/Java_JRE_1.5)

### 10.4.2 Proc Fcmp :

Il est possible à l'aide de la procédure FCMP (Function Compiler) de créer des fonctions personnelles utilisables dans des étapes data ou certaines procédures statistiques.

Calculer l'âge exact par la formule de Bill Kreuter (proposé par [Mike Zdeb](#))

```
options cmplib = (work.func);
proc fcmp outlib=work.func.test;
function exact_age(start,end);
return (floor ((intck('month',start,end) - (day(end) < day(start))) / 12));
endsub; run;
data age_check;
do dob = '01jan2005'd to '31dec2007'd;
do dod = dob to '31dec2010'd;
age1 = floor(yrdif(dob,dod,'actual'));
age2 = exact_age(dob,dod);
same = (age1 eq age2);
output;
end; end;
format dob dod ddmmyy10.; run;
proc freq data=age_check; table same; run;
title "ages non semblables";
proc print data=age_check;
where age1 ne age2;
var dob dod age1 age2;
run; title;
```

same	Fréquence	Obs	dob	dod	age1	age2
0	174	3287	02/01/2005	02/01/2008	2	3
1	1800006	5477	03/01/2005	03/01/2008	2	3
		7666	04/01/2005	04/01/2008	2	3
		9854	05/01/2005	05/01/2008	2	3

etc.

"A Cup of Coffee and Proc FCMP: I Cannot Function Without Them" Peter [Eberhardt](#) est le document à travailler pour approfondir cette question et devenir accro !:

<http://support.sas.com/resources/papers/proceedings09/147-2009.pdf>

et le <http://support.sas.com/resources/papers/proceedings10/326-2010.pdf>

« Adding Statistical functionality to the DATA Step with PROC FCMP » de Stacey M. Christian et J. Rioux

C'est avec un bonus : aller dans Solutions/Analyses/ Editeur FCMP



Avec ce mode d'emploi

<http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/a003265389.htm>

Autre exemple :

```
%macro maxage;
  %local maxag;
  proc sql noprint;
    select max(AGE) into :MAXAG
    from sashelp.class where SEX=&gender;
  quit;
  %let maxi=&maxag;
%mend maxage;

proc fcmp outlib=work.func.retest;
  function maxAgeFor(gender $);
    maxi = .;
    rc = run_macro('maxage', gender, maxi);
    if rc = 0 then return(maxi);
    return(.);
  endsub;
run;

%let cmplib=%sysfunc(getoption(cmplib));
options cmplib = (&cmplib work.func);
data _null_;
  ans = maxAgeFor("F");
  put "âge de la plus âgée: " ans;
run;
options cmplib = (&cmplib);
```

```
MPRINT(MAXAGE):  proc sql noprint;
MPRINT(MAXAGE):  select max(AGE) into :MAXAG
from sashelp.class where SEX='F';
MPRINT(MAXAGE):  quit;
```

âge de la plus âgée: 15

On discutera certainement de la pertinence de cet exemple par rapport à d'autres manières de faire, mais il illustre l'usage d'une fonction macro !!!!

Et son fonctionnement est différent du call execute !

Une procédure a été appelée dans l'étape data !

Logique :

-&gender comporte en fait déjà des doubles quotes, car comme paramètre argument caractère dans fcmp, il les possède déjà.

-Cependant enlever les quotes à l'appel, fait considérer F comme variable.

-En testant le code retour du run\_macro, on renverra sous l'argument gender, en fait, le contenu de la variable/fonction maxi ayant été remplie par l'exécution de la macro et gardée grâce au %let !

voir

<http://support.sas.com/documentation/cdl/en/proc/61895/HTML/default/a003258539.htm>

### 10.4.3 Proc Scaproc :

Cette procédure implémente un Analyseur de Code SAS, qui capturera les informations concernant l'input, l'output et l'usage des macro-symboles. Il sera possible de diriger ces informations vers un fichier texte.

En batch

```
sas yourjob.sas -initstmt "proc scaproc; record 'votrejob.txt' ; run;"
```

Cette procédure encadrante d'un code Sas contiendra habituellement une commande record ou write.

```
RECORD filespec <ATTR> <OPENTIMES> <GRID filespec <RESOURCE "resource name">>;
```

*Attr* donnera des info additionnelles sur les variables des tables

*Opentimes* précisera les moments d'ouverture, taille et noms physique des fichiers en entrée.

*Concatmem* listera le nom de la librairie concaténée qui contient un libref (ajout 9.2.3)

```
WRITE;
```

```
filename sca "d:\tabsas\scaproctest.txt";
```

```
proc scaproc;
```

```
record sca attr opentimes;
```

```
run;
```

```
proc scaproc;
```

```
write;
```

```
run;
```

#### Contenu du fichier

```
/* JOBSPLIT: DATASET INPUT SEQ #C00002.CLASS.DATA */
/* JOBSPLIT: LIBNAME #C00002 V9 'C:\Program Files\SAS\SASFoundation\9.2\core\sashelp' */
/* JOBSPLIT: OPENTIME #C00002.CLASS.DATA DATE:19JUL2010:17:43:11.91 PHYS: SIZE:5120 */
/* JOBSPLIT: DATASET OUTPUT SEQ WORK.CLASS.DATA */
/* JOBSPLIT: LIBNAME WORK V9 'd:\My SAS Files\work\_TD580' */
/* JOBSPLIT: CATALOG INPUT #C00002.LOCALE.LATILATI.TRANTAB */
/* JOBSPLIT: LIBNAME #C00002 V9 'C:\Program Files\SAS\SASFoundation\9.2\core\sashelp' */
/* JOBSPLIT: CATALOG INPUT #C00002.LOCALE.WLT1_UCS.TRANTAB */
/* JOBSPLIT: LIBNAME #C00002 V9 'C:\Program Files\SAS\SASFoundation\9.2\core\sashelp' */
/* JOBSPLIT: CATALOG INPUT #C00002.LOCALE.WLT1_LCS.TRANTAB */
/* JOBSPLIT: LIBNAME #C00002 V9 'C:\Program Files\SAS\SASFoundation\9.2\core\sashelp' */
/* JOBSPLIT: CATALOG INPUT #C00002.LOCALE.WLT1_CCL.TRANTAB */
/* JOBSPLIT: LIBNAME #C00002 V9 'C:\Program Files\SAS\SASFoundation\9.2\core\sashelp' */
/* JOBSPLIT: FILE OUTPUT d:\tabsas\scap.html */
/* JOBSPLIT: DATASET INPUT SEQ WORK.CLASS.DATA */
/* JOBSPLIT: LIBNAME WORK V9 'd:\My SAS Files\work\_TD580' */
/* JOBSPLIT: OPENTIME WORK.CLASS.DATA DATE:19JUL2010:17:43:12.91 PHYS: SIZE:5120 */
/* JOBSPLIT: FILE OUTPUT d:\tabsas\scaproctest.txt */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Name TYPE:CHARACTER LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Sex TYPE:CHARACTER LENGTH:1 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Age TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Height TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Weight TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Name TYPE:CHARACTER LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Sex TYPE:CHARACTER LENGTH:1 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Age TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Height TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Weight TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Name TYPE:CHARACTER LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Sex TYPE:CHARACTER LENGTH:1 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Age TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Height TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR #C00002.CLASS.DATA INPUT VARIABLE:Weight TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Name TYPE:CHARACTER LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Sex TYPE:CHARACTER LENGTH:1 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Age TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Height TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA OUTPUT VARIABLE:Weight TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA INPUT VARIABLE:Name TYPE:CHARACTER LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA INPUT VARIABLE:Sex TYPE:CHARACTER LENGTH:1 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA INPUT VARIABLE:Age TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA INPUT VARIABLE:Height TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: ATTR WORK.CLASS.DATA INPUT VARIABLE:Weight TYPE:NUMERIC LENGTH:8 LABEL: FORMAT: INFORMAT: */
/* JOBSPLIT: SYMBOL SET OB */
/* JOBSPLIT: SYMBOL GET OB */
/* JOBSPLIT: SYMBOL GET SYS_IOUSEEE */
/* JOBSPLIT: SYMBOL GET SYSSUMTRACE */
/* JOBSPLIT: ELAPSED 1797 */
/* JOBSPLIT: PROCNAME DATASTEP */
/* JOBSPLIT: STEP SOURCE FOLLOWS */
```

```

%let ob=15;
data class; set sashelp.class(obs=&ob.);run;
ods html file="d:\tabsas\scap.html" style=plateau;
proc means data=class;
var age weight height;
run;
ods html close;
proc scaproc;
  write;
run;

```

/\* JOBSPLIT: END \*/

La couleur bleue signale l'aspect ods. Le mot-clé *attr* en rouge a provoqué le listage des noms de variables et le *opentimes* en vert a précisé quand j'ai ouvert des tables.

### 10.4.5 Autres :

Proc **Proto** enregistrera en batch des fonctions externes écrites en c ou c++, utilisées ensuite dans FCMP  
Proc **Soap** et Proc **Xsl** s'appliqueront à du xml (un des objets d'exploration du Volume2).

### **10.5 Une pincée de macros:**

(non encore testé)

Avant la mise à jour du manuel Macro, voici un avant-goût des nouveautés

-des nouvelles macro-variables système

```

%put 1 &sysencoding ;      1 wlatin1
%put 2 &syserrortext;     2 180-322: Instruction incorrecte ou mal utilisée
%put 3 &syshostname;      3 pc_00944
%put 4 &systcpiphostname; 4 pc_00944
%put 5 &syswarningtext;   5 FOOTNOTE1 trop long. La hauteur a été réduite à 98.78 pour
                           cent de la taille indiquée ou de la taille par défaut.

```

Les *sys...text* présentent le contenu des derniers messages de ce type envoyés sur la log en cours de session.

-des nouvelles options système : *mcompile mexecnote mexecsize mreplace minoperator*

-trois nouvelles options pour %macro : *mindelimitator et minoperator et secure*

Avec l'instruction %macro *www/secure store ;* , votre macro serait encryptée et l'usage des options *mprint* et *mlogic* ne montrent plus le source.

### **10.6 Un zeste de sql:**

( à explorer si c'est votre domaine)

Outre ce que j'ai déjà montré et comme je ne suis pas un grand usager des requêtes sur gros fichiers et donc ne suis devenu un spécialiste de l'optimisation de SQL, je me contenterai de lister les nouvelles options de l'instruction `proc sql`.

```

-constdatetime|noconstdatetime
-exitcode
-ipassthru|noipassthru
-reduceput
-remerge|noremerge

```

