# COMPRENDRE L'OPTION *ENCODING* DANS SAS®

L'option système *encoding* définit le jeu de caractère, ou 'character set', qui est utilisé pendant une session SAS®, pour encoder les données et les stocker dans des tables SAS® ou SGBD.

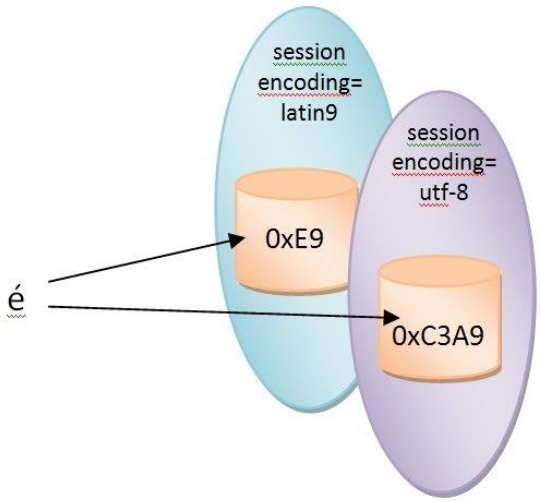
Caractéristiques :
Catégories : SAS® BASE
OS : OpenVMS, UNIX, Windows, z/OS
Version : SAS® 9.2
ENCODING System Option
Vérifié en août 2010

**Sommaire**



## *Introduction*

Voici un exemple simple, le caractère 'é' est encodé sous sa forme hexadécimale en :
- 0xE9, soit sur 1 octet, si l'*encoding* est latin9
- 0xC3A9, soit sur 2 octets, si l'*encoding* est utf-8




*Encodage du caractère 'é', en fonction de l'option encoding*

L'option système *encoding* ne concerne pas :
- La protection des mots de passe des tables SAS® : cf. ENCRYPT=, ALTER=, PW=, READ=, WRITE= cf. les options de l'étape data
- L'encodage des mots de passe : cf. PROC PWENCODE
- L'encryptage des données lors des transferts de données d'un client vers un serveur SAS® : cf. l'option système NETENCRYPT

L'option système encoding ne concerne donc pas la sécurisation des données, mais bien la manière dont sont stockées les données dans SAS®. Justement pourquoi y a-t-il plusieurs manières de les stocker ?

Historiquement, dans un souci de minimiser l'utilisation de l'espace disque, les différents systèmes de traitement des données utilisaient, et utilisent encore, leur propre jeu de caractères lié à leur langue. Ci-dessous un extrait de 4 encoding SAS® :

| Encoding SAS® | Norme sur laquelle il est basé | Nombre d'octets | Langues qui l'utilisent |
|---|---|---|---|
| latin1 | ISO 8859-1 | 1 | US,… |
| latin9 | ISO 8859-15 | 1 | France,… |
| cyrillic | ISO 8859-5 | 1 | Russie,… |
| big5 | Big5 | 2 | Chinois |

De nouvelles normes d'encodage, autour de l'*unicode*, sont apparues au début des années 1990, afin de prendre en compte tous les caractères de toutes les langues internationales :

| Encoding SAS® | Norme sur laquelle il est basé | Nombre d'octets | Langues qui l'utilisent |
|---|---|---|---|
| utf-8 | Unicode Consortium | Variable de 1 à 4 | toutes |
| utf-32 | Unicode Consortium | 4 | toutes |

*Remarques* :

- Si la question de l'encodage reste un sujet important, cela réside dans le fait que nous héritons encore de données et de systèmes applicatifs qui utilisent les jeux de caractères propres à chaque langue, alors même que les applications informatiques deviennent de plus en plus internationales. On peut très bien imaginer que dans un futur proche, toutes les données et tous les systèmes applicatifs utiliseront un type d'encodage universel des données, tel que l'unicode, auquel cas l'encodage ne deviendrait plus un sujet de discussion, car il n'y aurait plus de caractères altérés ('¿',' Â£','Ã©'…) dans les données !
- l' « Unicode » est une norme générique qui regroupe en fait 3 normes dérivées : UTF8, UTF16, UTF32.

## ***Comment retrouver la représentation hexadécimale d'une chaîne de caractères ?***

```
data _null_;
      set sashelp.class (obs=1);
      put name=name $hex200.;
run;
```

```
Name=Alfred
416C6672656420 20
NOTE:  1 observations copiées de la table SASHELP.CLASS.
NOTE: L'étape DATA a utilisé (Durée totale du processus) :
      temps réel                    0.01 secondes
      temps processeur    0.01 secondes
```

## Comment utiliser l'option système encoding au niveau de la session SAS® ?

L'option système *encoding* est positionnée de façon implicite par une autre option système : *locale* (exemple dans le fichier sasv9.cfg : **-locale 'French'**). Elle peut aussi être définie directement (exemple dans le fichier sasv9.cfg : **-encoding latin9**). Ainsi, l'option locale a un effet direct sur quatre options. En voici l'illustration pour deux valeurs :

| Locale= | Définition implicite des 4 options | | | |
|---|---|---|---|---|
| | ENCODING= (unix) | DFLANG= | DATESTYLE= | PAPERSIZE= |
| fr_FR<br>alias : French | latin9 | French | DMY | A4 |
| en_US<br>alias : English | latin1 | English | DMY | A4 |

*Remarques* :

- Généralement les options *encoding* / *locale* sont positionnées dans les fichiers de configuration spécifiques à chaque langue : exemple le fichier !SASROOT/nls/fr/sasv9.cfg positionne l'option locale sur « *French_France* », et implicitement l'option encoding sur *latin9.* Ces options peuvent être redéfinies au niveau d'un serveur SAS® spécifique, comme un Workspace Server. La Usage Note 18639 précise comment démarrer SAS® sous différentes langues.
- S'il y a incompatibilité entre la définition d'*encoding* et de *locale*, c'est la définition de l'encoding qui prime.
- L'option *encoding* peut être définie uniquement au démarrage de la session SAS® (*locale* est modifiable durant la session SAS®)
- Depuis la version 9.2 de SAS®, l'unicode UTF8 est systématiquement installé, mais il n'est pas activé par défaut.
- L'option *locale* est généralement choisie au démarrage de SAS®, en fonction de l'installation/configuration de SAS®.
  Exemple de commandes sous UNIX permettant de configurer SAS® en UTF8 :

```
cd /usr/local/SAS/SASFoundation/9.2
rm sas
ln -s bin/sas_u8 sas
# lien symbolique vers sas_u8
# sas_u8 fait reference au sasv9.cfg, qui contient les 2 lignes
# -DBCS
# -ENCODING UTF-8
```

## Comment connaître la valeur de l'option système encoding ?

```
proc options group=languagecontrol;
run;
```

```
DATESTYLE=DMY      Identify sequence of month, day and year when ANYDATE informat data is ambiguous
DFLANG=FRENCH      Language for EURDF date/time formats and informats
PAPERSIZE=LETTER   Size of paper to print on
TRANTAB=(lat1lat1,lat1lat1,wlt1_ucs,wlt1_lcs,wlt1_ccl,,,)
                   Names of translate tables
DBCSLANG=NONE      Specifies the double-byte character set (DBCS) language to use
DBCSTYPE=WINDOWS   Specifies a double-byte character set (DBCS) encoding method
NODBCS             Do not Process double byte character sets
ENCODING=WLATIN1   Specifies default encoding for processing external data.
LOCALE=FRENCH_FRANCE
                   Specifies the current locale for the SAS session.
NONLSCOMPATMODE    Uses the user specified encoding to process character data
```

## ***Comment traiter des données dont l'encodage est différent de la session SAS® en cours ?***

A des fins de conversion, ou pour des besoins spécifiques, il est possible de se substituer localement l'option système *encoding*. Exemple de cas :

- pour importer un fichier texte d'un encodage différent de la session SAS® en cours
- pour écrire dans une table SAS®, dans un encodage différent de la session SAS® en cours

```
filename extfile 'c:\temp\currency.txt';
data currency;
      infile extfile encoding=wlatin2;
      input code $;
run;
/*la table currency est créée en wlatin1 (encoding de la session)
tandis que le fichier extfile doit être interprété non pas
selon l'encoding de la session, mais en wlatin2*/
```

```
data cur_warabic (encoding=warabic);
      set currency;
run;
/*la table cur_warabic est créée en warabic (cette option est à spécifier
pour que la table soit encodée dans un encodage différent de la session
SAS)*/
```

*Remarques* :

- Pour une conversion de données, utiliser une simple étape DATA (cf. 2ème exemple ci-dessus), ou une PROC COPY (avec l'option NOCLONE)
- autre possibilité pour les tables SAS® : les options INENCODING= et OUTENCODING=se positionnant au niveau de l'instruction libname
- lorsque SAS® doit convertir des données d'un encodage vers un autre (ce qui est fait implicitement dans les deux exemples ci-dessus, mais également lorsque SAS® lit une table d'un encodage différent de celui de la session), la fonctionnalité CEDA ('**Cross Environment Data Access**') est utilisée. La note suivante le rappelle systématiquement dans le journal :

***'...Cross Environment Data Access will be used, which might require additional CPU resources and might reduce performance...'***

## *Comment traiter les données en provenance de tables SGBD, via un module SAS/Access® to ?*

Principe :

- le client RDBMS a en charge la conversion des données entre le client RDBMS et le serveur RDBMS
- l'encodage doit être identique entre celui de la session SAS® et le client RDBMS

Prenons un exemple avec le module SAS/Access to Oracle :

- configuration sur le serveur Oracle : NLS_LANG=french_france.we8mswin1252
- encoding de la session SAS® : utf-8
- le client Oracle doit être configuré en utf-8. Il faut ajouter dans le fichier **sasenv_local** :

  **NLS_LANG=french_france.UTF8**
  **export NLS_LANG**

## *Comment connaître l'attribut encoding d'une table SAS® ?*

```
proc datasets lib=work;
      contents data=currency;
quit;
```

```
                                The DATASETS Procedure

          Data Set Name        WORK.CURRENCY                     Observations          1
          Member Type          DATA                              Variables             1
          Engine               V9                                Indexes               0
          Created              vendredi 13 août 2010 00 h 09     Observation Length    8
          Last Modified        vendredi 13 août 2010 00 h 09     Deleted Observations  0
          Protection                                             Compressed            NO
          Data Set Type                                          Sorted                NO
          Label
          Data Representation  WINDOWS_32
          Encoding             wlatin1  Western (Windows)
```

*Remarques* :

- Jusqu'à la version 8 de SAS®, l'attribut encoding a une valeur indéterminée. Donc pour traiter des tables SAS® créées sous SAS® v8, il faut impérativement connaitre l'encodage de la table (et utiliser l'option encoding au niveau de l'étape data si besoin).
- A partir de la version 9 de SAS®, l'attribut étant explicitement positionné au niveau de la table, toute session SAS® qui lit une table d'un encodage différent utilise implicitement CEDA.

## Comment modifier l'attribut *encoding* d'une table SAS® ?

```
proc datasets library=work;
      modify currency/correctencoding="latin9";
quit;
```

*Remarques* :

- Cette action n'affecte pas l'encodage des données, mais uniquement la valeur de l'attribut encoding

- De ce fait, il faut garder à l'esprit qu'il peut y avoir une divergence entre l'attribut *encoding* et l'encodage réel des données.

## ***Recommandations dans le cadre d'une session SAS® en UTF8***

- Dans le cadre d'une migration ou d'une conversion de données encodées sur 1 octet (exemple : latin1), vers un serveur SAS® en UTF8, la longueur des variables doit naturellement augmenter, du fait que UTF8 stocke les données sur 1 à 4 octets. Comme l'encodage est de longueur variable, il n'est pas possible de prévoir précisément la longueur maximale d'une variable, il faut donc spécifier le moteur CVP lors de la déclaration de la bibliothèque. Exemple de conversion :

```
$ ./sas_u8 -nodms

libname src cvp "/local/users/sas" cvpmultiplier=2 ;
/*valeur par défaut du facteur de multiplication cvpmultiplier=1,5*/
libname tgt "/local/users/sas";
data tgt.utf8;
      set src.latin1 (encoding=latin1);
run;
/*la conversion peut également être faite avec la PROC COPY, mais en
rajoutant l'option NOCLONE*/
```

- Il est nécessaire d'utiliser les K-functions en lieu et place des fonctions « classiques » de SAS : les données étant stockées sur plus d'un octet, le traitement des chaînes de caractère est différent, et l'usage des k-functions (ou DBCS string functions) est nécessaire. A chaque fonction standard, on retrouve généralement la fonction équivalente, préfixée d'un K :
    - SUBSTR => KSUBSTR
    - TRIM => KTRIM
    - SCAN => KSCAN…

## ***Quelques cas d'anomalies d'encodage rencontrées***

Deux types d'anomalies peuvent être rencontrés :
- En consultant les données, certains caractères altérés apparaissent : '¿',' Â£','Ã©'…
- Des erreurs liées à l'encodage apparaissent dans le journal SAS®

Dans tous les cas, mais plus particulièrement dans le premier, il est parfois difficile de faire un diagnostic rapide sur l'origine de l'anomalie : il est souvent nécessaire de vérifier toute la chaîne d'alimentation des données, jusqu'à leur affichage dans le client SAS® (SAS® Enterprise Guide®, SAS® Data Integration Studio, SAS® Olap Cube Studio, SAS® Web Report Studio…). Pour une table SAS®, il faut tenir compte des différents niveaux de configuration suivante :
- L'attribut *encoding* de la table SAS®
- L'encodage réel des données de la table SAS® (a priori cela ne devrait pas diverger de l'attribut *encoding* de la table SAS®, si toutes les étapes de constitution et d'alimentation de la table ont été respectées)
- L'option système *encoding* paramétrée au niveau du serveur SAS®
- L'option système *encoding* paramétrée au niveau du client SAS®
- Les polices installées au niveau de l'OS du serveur SAS®
- Les polices installées au niveau de l'OS du client SAS®

Cas 1 : impossibilité de mise à jour d'une table

```
107  proc sql;
108       update wrt.utf8 set val="1";
ERROR: File WRT.UTF8 cannot be updated because its encoding does not match the session
encoding or the file is in a format native to another host, such as WINDOWS_32.
```

⇨ Une table d'encoding ou créée sur un système d'exploitation différent ne peut être modifiée, elle peut uniquement être consultée.

Cas 2 : erreur d'écriture due à un problème d'encoding ou de troncature de données

```
116  data tgt.utf8 (encoding=utf8);
117       set src.wlatin1;
118  run;

NOTE: Data file TGT.UTF8.DATA is in a format that is native to another host, or the file
encoding does not match the session encoding. Cross Environment Data Access will be used,
which might require additional CPU resources and might reduce performance.
ERROR: Some character data was lost during transcoding in the dataset TGT.UTF8. Either the
data contains characters that are not representable in the new encoding or truncation
occurred
during transcoding.
```

⇨ Lors d'une conversion d'un encoding vers un autre, il faut bien sûr s'assurer que les caractères rencontrés trouvent une correspondance dans l'encoding cible, mais surtout que la longueur des variables soit suffisante. En l'occurrence, dans le cas présent la libname tgt ne contenait pas l'option CVP.

Cas 3 : problème de configuration du client SAS® Enterprise Guide® en 9.2

Certains clients SAS® comme SAS® Enterprise Guide® 4.2 peuvent avoir un *encoding* différent de celui du serveur SAS®. Les conséquences peuvent être :

- L'impossibilité de mettre à jour une table
- Une restitution différente de certains caractères par rapport à ce qui est réellement stocké.

⇨ Ceci est dû au fait que le workspace server démarre la session SAS® avec la valeur de l'option locale du client, et non avec celle du serveur. Dans ce cas, il faut soit choisir la bonne langue dans SAS® Enterprise Guide® 4.2 pour être en phase avec celle du serveur, soit ajouter l'option **–encoding latin1** (si le serveur est en latin1) dans le fichier sasv9_usermods.cfg du workspace server. Ceci est documenté dans la Usage Note 35644.

Cas 4 : problème d'affichage de caractères Unicode dans SAS® Foundation

Le SAS® Display Manager System ne supporte pas complètement Unicode, en particulier dans l'interface fenêtrée UNIX.

⇨ SAS® préconise d'utiliser le plus possible un client comme SAS® Enterprise Guide®.

## ***Conclusion***

Les problèmes d'encodage des données deviennent assez courants du fait qu'un système décisionnel est de plus en plus amené à traiter des données d'origine internationale et des systèmes hétérogènes. Ils peuvent être à l'origine d'un défaut de la qualité de données.

En cas de doute sur le contenu d'une table SAS®, le recours ultime est d'ouvrir la table dans une session SAS® de même *encoding* que son propre attribut *encoding* (sinon CEDA convertit les données), puis de consulter le contenu des variables en hexadécimal.

Pour toutes informations, vous avez la possibilité de vous rendre sur la page de documentation du guide de référence SAS® 9.2 NLS :
http://support.sas.com/documentation/cdl/en/nlsref/61893/HTML/default/viewer.htm

Un dossier sur le traitement de données multilingues sur un serveur SAS® Unicode est disponible à cette adresse : http://support.sas.com/kb/33/443.html

Philippe LAFFIN
Consultant Support Clients SAS France