

INTERNATIONALISATION DES PROGRAMMES SAS – PARTIE 1

Vous avez des utilisateurs SAS partout dans le monde ? Vous manipulez des données hétéroclites, dans différents langages, utilisant différents jeux de caractères, et provenant de divers systèmes ? SAS propose de nombreuses options, formats et fonctions qui vous permettent d'exploiter ces informations quelle que soit leur source, de publier des résultats et rapports adaptés à la culture et la langue du public visé, et adapter vos applications SAS à vos utilisateurs.

Caractéristiques :

Catégories : Base SAS®
OS : Windows, Unix, z/OS
Version : SAS® 9.4
Vérifié en septembre 2014

Dans cette première partie, nous allons voir comment « Internationaliser » votre environnement et vos programmes SAS, et ainsi les rendre flexibles vis-à-vis des différentes langues et spécificités régionales que vous pouvez rencontrer. Nous verrons aussi comment « Localiser » certains aspects d'un programme, pour qu'il soit utilisé facilement par des utilisateurs de différents pays, ou parlant différentes langues.

Dans la seconde partie de ce document, nous irons un peu plus loin dans la localisation d'une application, en abordant l'externalisation des chaînes de caractères, en vue de leur traduction en fonction de l'utilisateur.

Sommaire

1.	Introduction : I18N et L10N	2
2.	Internationalisation – les concepts	2
2.1.	Encodage de session	2
2.2.	Le « Locale ».....	3
2.3.	Les fuseaux horaires (SAS 9.4)	4
3.	Les formats localisés	5
3.1.	Dates	5
3.2.	Heures / Datetimes.....	5
3.2.1.	Heures et Fuseaux horaires (SAS 9.4)	5
3.3.	Nombres.....	7
3.4.	Devises	7
3.5.	Formats personnalisés.....	7
4.	Manipulation des chaînes de caractères.....	9
5.	Trier des données	10
6.	Changer de langue (SAS 9.4)	11
7.	Liens utiles.....	13
8.	Conclusion	13

1. INTRODUCTION : I18N ET L10N

Dans une organisation globalisée, les applications informatiques doivent pouvoir s'adapter aux différents utilisateurs qui vont s'en servir, et aux personnes qui recevront les résultats produits. Cela concerne la langue utilisée, mais également des caractéristiques comme le format des nombres, dates et heures, la devise, voire éventuellement la disposition des éléments dans une page, ou un format d'adresse.

On parle alors de « **Software Globalization** », ou G11N, que l'on peut séparer en deux processus principaux : Internationalisation et Localisation.

Par **Internationalisation** (I18N), on désigne le processus qui consiste à concevoir une application de telle manière qu'elle pourra par la suite être adaptée à différentes langues ou régions.

La **Localisation** (L10N) consiste à modifier (ou paramétrer) une application pour une région géographique ou un groupe d'utilisateurs partageant la même langue et les mêmes habitudes spécifiques à leur culture. Cela concerne notamment la traduction de l'interface.

2. INTERNATIONALISATION – LES CONCEPTS

2.1. Encodage de session

Un **Encodage** est un ensemble de caractères – lettres, chiffres, marques de ponctuation, symboles, idéogrammes, et caractères de contrôle – représentés par un code hexadécimal (appelé *code point*) qui sera la représentation interne de ces caractères. Par exemple, *IBM-1047* est un encodage de la famille EBCDIC utilisée sur mainframe; *Windows Cyrillic* est un encodage de la famille Windows; et *Latin1* est un encodage de la famille ISO 8859.

On peut classer les encodages dans 2 catégories :

- *Single-byte character set* (**SBCS**) : chaque caractère est représenté sur un octet. Certains *code point* sont réservés pour représenter les caractères de contrôle du système (retour chariot, tabulation, etc...). Un encodage SBCS est donc capable de représenter 256 caractères différents. C'est généralement suffisant pour gérer les langues européennes, même si différentes langues nécessiteront différents encodages.
- *Multi-byte character set* (**MBCS**) : l'encodage nécessite un ou plusieurs octets pour représenter chaque caractère. Ce type d'encodage est généralement utilisé pour les langues asiatiques, ou pour gérer plusieurs jeux de caractères dans une même session – comme l'encodage *UTF-8*, l'un des plus commun, qui peut utiliser jusqu'à 4 octets pour chaque caractère. Les encodages *double-byte character set* (**DBCS**) sont une sous-catégorie des encodages MBCS, avec un codage sur 2 octets.

Dans les programmes SAS, les caractères encodés sur plusieurs octets nécessitent en général un traitement particulier. Nous verrons cela en détail dans le paragraphe « Manipulation des chaînes de caractères ».

Les langues qui devront être supportées par votre application vont déterminer l'encodage à utiliser pour votre session SAS. Les fichiers de données SAS ont leur propre encodage – de même que les sources de données externes, fichiers texte ou bases de données – et il y aura une **transcodification** des caractères vers l'encodage de la session SAS, si nécessaire. Cet encodage de session devra donc être capable de supporter les caractères à traiter. Par exemple, si vous devez traiter des données avec des caractères japonais, anglais et français, UTF-8 sera un bon choix pour votre session SAS, cet encodage étant capable de représenter les caractères de ces trois langues.

Voici quelques exemples d'encodage disponible avec le Système SAS, sous Windows. La liste exhaustive se trouve dans la documentation [SAS 9.4® National Language Support \(NLS\): Reference Guide](#) (voir Annexes)

Description	ENCODING=	Type d'encodage
US – Western Europe	wlatin1	SBCS
Central Europe	wlatin2	SBCS
Cyrillic	wcyrillic	SBCS
EBCDIC cp1147-French	EBCDIC1147	SBCS
Japanese	shift-jis	DBCS
UTF-8 Unicode	utf-8	MBCS

L'option ENCODING= ne peut être positionnée qu'au démarrage de la session SAS, et ne peut pas être modifiée en cours de session.

2.2. Le « Locale »

Un « *Locale* » est un ensemble de conventions concernant l'écriture de texte (sens d'écriture, guillemets, ordre de tri) et de différentes unités (valeurs numériques, format de date ou d'heure, sigle monétaire) dans une langue donnée.

Ces conventions sont liées, non seulement à la langue utilisée, mais également à la région géographique. Ainsi, plusieurs *locales* peuvent être associées à une langue. Par exemple, la population francophone de France utilisera le *locale* « fr_FR », la population francophone du Canada utilisera le *locale* « fr_CA ».

Ci-dessous quelques exemples de *locales* :

LOCALE= (Notation abrégée)	LOCALE= (Notation longue)	Alias
fr_BE	French_Belgium	
fr_CA	French_Canada	
fr_FR	French_France	French fr
fr_LU	French_Luxembourg	
fr_CH	French_Switzerland	
en_US	English_UnitedStates	English en
en_GB	English_UnitedKingdom	
en_CA	English_Canada	

L'option LOCALE= influe sur les composants SAS suivants :

- les formats et informats NL. Pour plus de détail, voir le paragraphe « Les formats localisés », ci-dessous.
- les fonctions SASMSG() et SASMSGF, utilisés dans l'externalisation des chaînes de caractères (paragraphe abordé dans la 2nde partie de cet article).
- les options SAS DFLANG=, DATESTYLE=, PAPERSIZE=, MAPEBCDIC2ASCII=, ODSLANGCHG=, LOGLANGCHG=, LSWLANG=, LOCALEDATA=, et TIMEZONE=.

Il est possible de modifier un *locale* existant, par exemple pour adapter un format localisé à un cas particulier. Pour plus d'information sur cette possibilité, voir la procédure LOCALEDATA dans la documentation [SAS 9.4® National Language Support \(NLS\): Reference Guide](#).

L'option LOCALE= peut être positionnée au démarrage de la session, mais également en cours de session via l'instruction OPTIONS.

2.3. Les fuseaux horaires (SAS 9.4)

Une nouvelle option a été introduite en version 9.4 du Système SAS, TIMEZONE=. Cette option permet de définir le fuseau horaire de référence pour la session.

Plusieurs notations sont supportées :

TIMEZONE=	Description
Europe/Paris	Identifiant de fuseau horaire. C'est la notation recommandée, car elle prend en compte le passage heure d'hiver / heure d'été
CET	Nom standard (Central European Time)
CEST	Nom standard (Central European Summer Time)
SASNC+1:00	Nom défini par l'utilisateur

La liste complète des fuseaux horaires et de leurs notations se trouve dans la documentation, dont le lien est en annexe.

L'option TIMEZONE= influe sur les composants SAS suivants :

- les heures enregistrées dans les logs.
- les heures de création et de modification des données SAS.
- le résultat des fonctions DATE(), DATETIME(), TIME(), TODAY(), TZONEOFF(), TZONEID(), TZONENAME(), TZONES2U(), TZONEU2S(), TZONESTTNAME(), TZONEDSTNAME(), TZONESTTOFF(), TZONEDSTOFF().
- l'affichage des formats B8601DXw. E8601DXw., B8601LXw., E8601LXw., B8601TXw., E8601TXw., NLDATMZw., NLDATMTZw., et NLDATMWZw.

L'option TIMEZONE = peut être positionnée au démarrage de la session, mais également en cours de session via l'instruction OPTIONS.

3. LES FORMATS LOCALISÉS

La façon d'écrire les dates, heures et nombres diffère en fonction de l'endroit du monde où l'on se trouve. Si l'information donnée est la même, son format d'affichage peut être différent, et si l'on n'y prend pas garde, cela peut entraîner des confusions. Par exemple, la chaîne de caractères '15.125' désignera le nombre « quinze mille cent vingt-cinq » en Allemagne, alors qu'il sera compris comme « quinze virgule cent vingt-cinq » aux Etats-Unis.

Il existe dans le Système SAS un certain nombre de formats, dont l'affichage est fonction du *locale* en cours de la session SAS.

3.1. Dates

Le cas des dates est emblématique des différences de conventions géographiques et culturelles. Comment lit-on la date '04/02/12' ?

- En France, il s'agira du 4 février 2012.
- Aux Etats-Unis, ce sera le 2 avril 2012.
- Au Japon, on comprendra le 12 février 2004.

Les formats et informats prenant en compte le *locale* de la session ont un nom commençant par NL. Voici quelques exemples tirés de la documentation :

Nom	Description	option locale=fr_FR;	option locale=en_US;
NLDATE	Date	26 juillet 2014	July 26, 2014
NLDATEL (9.4)	Date (long)	26 juillet 2014	July 26, 2014
NLDATEM (9.4)	Date (moyen)	26 juil. 2014	Jul 23, 2014
NLDATES (9.4)	Date (court)	26/07/2014	07/26/2014
NLDATEW	Jour de la semaine et date	samedi 26 juillet 2014	Saturday, July 26, 2014
NLDATEYW	Année et semaine	Week 30 2014	Week 29 2014

3.2. Heures / Datetimes

La principale différence que l'on retrouve dans la notation des heures, est l'utilisation d'un format sur 12 heures (U.S.) ou 24 heures (plus ou moins le reste du monde...)

Exemples :

Nom	Description	option locale=fr_FR;	option locale=en_US;
NLTIME	Heure	21 h 48	21:48:23
NLTIMAP	Heure (A.M./P.M.)	9:48:23 PM	9:48:23 PM
NLDATM	Date et heure	26 juillet 2014 21 h 48	26Jul2014:21:48:23
NLDATML (9.4)	Date et heure (long)	26 juillet 2014 21:48:23	July 26, 2014 09:48:23 PM
NLDATMM (9.4)	Date et heure (moy.)	26 juil. 2014 21:48:23	Jul 26, 2014 09:48:23 PM
NLDATMS (9.4)	Date et heure (court)	26/07/2014 21:48:23	07/26/2014 21:48:23

3.2.1. Heures et Fuseaux horaires (SAS 9.4)

Avec l'introduction du support des fuseaux horaires, SAS supporte également le standard international ISO 8601, qui spécifie une notation claire et non-ambiguë des dates et heures, incluant le fuseau horaire.

Le Temps universel coordonné ou UTC¹ (en anglais : « Coordinated Universal Time ») sert de référence. Le fuseau horaire en France métropolitaine est défini par UTC+1 en heure d'hiver, et UTC+2 en heure d'été.

¹ Coordinated Universal Time a été abrégé en UTC, au lieu de CUT correspondant à l'acronyme en anglais ou de TUC correspondant à l'acronyme en français. En effet, si les experts de l'Union internationale des télécommunications étaient d'accord pour définir une abréviation commune à toutes les langues, ils étaient divisés sur le choix de la langue. Finalement, c'est le compromis UTC, nécessitant un effort des deux parties, qui fut choisi.

Voici un tableau récapitulatif des formats permettant de faire apparaître le fuseau horaire. Pour les exemples, nous avons utilisé les dates et heures suivantes :

- datetime='26jul2014:21:48:23'dt;
- time='21:48:23't;

TZ='Europe/Paris' locale=fr_FR :

Nom	Description	Affichage
B8601DX	Conversion d'un datetime UTC vers un datetime locale	20140726T234823+0200
B8601LX	Affichage d'un datetime local (pas de conversion) selon ISO 8601	20140726T214823+0200
B8601TX	Conversion d'une heure UTC vers une heure locale	234823+0200
E8601DX	Conversion d'un datetime UTC vers un datetime locale (étendu)	2014-07-26T23:48:23+02:00
E8601LX	Affichage d'un datetime local (pas de conversion) (étendu)	2014-07-26T21:48:23+02:00
E8601TX	Conversion d'une heure UTC vers une heure locale (étendu)	23:48:23+02:00
NLDATMZ	Affichage d'un datetime local (pas de conversion) selon locale	26 juillet 2014 21 h 48 +0200
NLDATMTZ	Affichage de l'heure d'un datetime locale (pas de conversion) selon locale	21 h 48 +0200
NLDATMWZ	Affichage d'un datetime local (pas de conversion) avec jour de la semaine, selon locale	samedi 26 juillet 2014 21:48:23 +0200

TZ='America/Denver' locale=en_US :

Nom	Description	Affichage
B8601DX	Conversion d'un datetime UTC vers un datetime locale	20140726T154823-0600
B8601LX	Affichage d'un datetime local (pas de conversion) selon ISO 8601	20140726T154823-0600
B8601TX	Conversion d'une heure UTC vers une heure locale	154823-0600
E8601DX	Conversion d'un datetime UTC vers un datetime locale (étendu)	2014-07-26T15:48:23-06:00
E8601LX	Affichage d'un datetime local (pas de conversion) (étendu)	2014-07-26T15:48:23-06:00
E8601TX	Conversion d'une heure UTC vers une heure locale (étendu)	15:48:23-06:00
NLDATMZ	Affichage d'un datetime local (pas de conversion) selon locale	26Jul2014:21:48:23 -0600
NLDATMTZ	Affichage de l'heure d'un datetime locale (pas de conversion) selon locale	21:48:23 -0600
NLDATMWZ	Affichage d'un datetime local (pas de conversion) avec jour de la semaine, selon locale	Sat, Jul 26, 2014 09:48:23 PM -0600

3.3. Nombres

Les nombres aussi sont représentés avec des conventions propres à chaque pays. L'exemple le plus connu est le séparateur décimal, qui sera un point aux Etats-Unis et une virgule en France.

Ci-dessous, les formats localisés de nombre :

Nom	Description	locale=fr_FR;	locale=en_US;
NLBEST	« Meilleure » notation	-123456,789	-123456.789
NLNUM	Format numérique local	-123 456,789	-123,456.789
NLNUMI	Format numérique international	-123,456.789	-123,456.789
NLPCT	Pourcentage, format local	-12,32%	-12.32%
NLPCTI	Pourcentage, format international	-12.32%	-12.32%
NLPCTN	Pourcentage, format local, signe négatif -	-12,32%	-12.32%
NLPCTP	Pourcentage, format local, signe négatif ()	(12,32%)	(12.32%)
NLPVALUE	Valeur-p, format local	0,12	0.12

3.4. Devises

Le formatage des devises suit des règles similaires à celui des nombres, avec l'ajout du sigle ou abréviation de la devise. Il s'agit d'un simple formatage, **il n'y a pas de conversion de devises**.

Un grand nombre de devises sont prises en compte, les formats correspondant sont de la forme NLMNLxxx ou NLMNIxxx, où xxx est le code international de la devise. Voici ci-dessous quelques exemples :

Nom	Description	locale=fr_FR;	locale=en_US;
NLMNLEUR	Euro, sigle local	-123 456,78 €	(€123,456.78)
NLMNIEUR	Euro, sigle international	-123 456,78 EUR	(EUR123,456.78)
NLMNLUSD	Dollar US, sigle local	-123 456,78 US\$	(US\$123,456.78)
NLMNIUSD	Dollar US, sigle international	-123 456,78 USD	(USD123,456.78)
NLMNLGBP	Livre britannique, sigle local	-123 456,78 £	(£123,456.78)
NLMNIGBP	Livre britannique, sigle international	-123 456,78 GBP	(GBP123,456.78)
NLMNY	Monnaie locale, sigle local	-123 456,78 €	(\$123,456.78)
NLMNYI	Monnaie locale, sigle international	-123 456,78 EUR	(USD123,456.78)

3.5. Formats personnalisés

La procédure FORMAT accepte une instruction LOCALE, qui permet de définir un format spécifique à utiliser avec un *locale* donné.

Dans l'exemple ci-dessous, un format `perf.` générique est créé d'abord dans le catalogue `work.formats`. Ensuite, une version localisée est créée, dans le catalogue `work.formats_fr_fr` :

```
proc format lib=work.formats;
  value perf 0 = 'None'
            1 - 4 = 'Some'
            5 - 19 = 'Good'
            20 - high = 'Champ' ;
run;

options locale=fr_FR;
proc format lib=work.formats locale;
  value perf 0 = 'Aucun'
            1 - 4 = 'Peu'
            5 - 19 = 'Pas mal'
            20 - high = 'Champion' ;
run;
```

Pour accéder aux catalogues de formats spécifique à un *locale*, l'option FMTSEARCH doit utiliser l'instruction */locale* :

```
options fmtsearch=(work/locale);

options locale=en_US;
title "Locale is English, Use the Default Format Catalog";
proc print data=sashelp.baseball(keep=name nHome);
    format nHome perf. ;
run;

options locale=fr_FR;
title 'Locale fr_FR, le catalogue de format français est utilisé';
proc print data=sashelp.baseball(keep=name nHome);
    format nHome perf. ;
run;
```


4. MANIPULATION DES CHAINES DE CARACTÈRES

Dans ce paragraphe, nous allons nous intéresser de plus près à la façon dont les caractères sont représentés dans un encodage donné, et comment en tenir compte lorsque nous allons travailler avec des chaînes de caractères – par exemple pour rechercher un caractère, extraire un mot, etc...

Voici un exemple simple, dans une session SAS UTF-8 :

```
data _null_ ;
  str= "ッヂ" ;
  s=substr(str,1,1);
  l=length(str);
  put str= "(" str $hex12. ")" /s= "(" s $hex12. ")" /l= ;
run ;
```

Log :

```
str=ッヂ (E38383E38382)
s=□ (E32020202020)
l=6
```

Les caractères « ッヂ » sont représentés en UTF-8 par 3 octets chacun, 'E3 83 83'x et 'E3 83 82'x. La fonction SUBSTR() renvoie le premier octet, 'E3'x, qui ne correspond à aucun caractère dans le codage UTF-8 (les octets suivants – '20'x – sont des espaces). La fonction LENGTH() renvoie le nombre d'octets, 6.

Ceci est dû au fait que les fonctions standard sont **orientées « octet »** - ce qui fonctionne parfaitement dans une session SAS SBCS, où **1 caractère = 1 octet**.

Mais ce n'est pas ce que l'on attend ici. Nous cherchons à récupérer le premier caractère, « ッ », et le nombre de caractères, 2. Pour cela, il existe des fonctions spécifiques, appelées **K-fonctions**, qui prennent en compte l'encodage du texte pour réaliser ces opérations :

```
data _null_ ;
  str= "ッヂ" ;
  s=ksubstr(str,1,1);
  l=klength(str);
  put str= "(" str $hex12. ")" /s= "(" s $hex12. ")" /l= ;
run ;
```

Log :

```
str=ッヂ (E38383E38382)
s=ッ (E38383202020)
l=2
```

Dans une session SBCS, les K-fonctions donnent le même résultat que leur équivalent standard.

Le détail de chaque K-fonction est présenté dans la documentation [SAS 9.4® National Language Support \(NLS\): Reference Guide](#).

5. TRIER DES DONNÉES

Ceci est relativement moins connu, mais l'ordre de tri des caractères est aussi soumis à des conventions régionales et linguistiques.

La procédure SORT accepte l'option SORTSEQ=LINGUISTIC, qui précise que l'ordre de tri devra se baser sur le *locale* de la session :

```
data list;
    input name $20.;
    datalines;
Côté
Boucher
Fournier
Cotée
Legrand
Dubois
Thibeault
Martin
Vaudron
Girard
;
run;

title "Ordre de tri standard";
proc sort data=list;
    by name;
run;
proc print data=list; run;

title "Ordre de tri français";
proc sort data=list sortseq=linguistic;
    by name;
run;
proc print data=list; run;
```

Résultats :

Ordre de tri standard	
Obs.	name
1	Boucher
2	Cotée
3	Côté
4	Dubois
5	Fournier
6	Girard
7	Legrand

Ordre de tri français	
Obs.	name
1	Boucher
2	Côté
3	Cotée
4	Dubois
5	Fournier
6	Girard
7	Legrand

Vous trouverez plus de détails dans le document technique *Linguistic Collation: Everyone Can Get What They Expect* (lien en annexe).

6. CHANGER DE LANGUE (SAS 9.4)

Il est maintenant possible de changer, en cours de session SAS, la langue utilisée pour la génération des rapports, et les messages du journal SAS. Les options suivantes sont des nouveautés SAS 9.4, et sont définies au démarrage de la session :

Option	Description
ODSLANGCHG / NOODSLANGCHG	Si l'option ODSLANGCHG est active, la langue des sorties ODS est déterminée par l'option LSWLANG=, ou par l'option LOCALE=.
LOGLANGCHG / NOLOGLANGCHG	Si l'option LOGLANGCHG est active, la langue des messages de la log est déterminée par l'option LSWLANG=, ou par l'option LOCALE=.
LSWLANG=	Spécifie la langue des messages log et ODS, si les options ODSLANGCHG et LOGLANGCHG sont actives. Si LSWLANG=LOCALE est spécifié, la langue des messages sera basée sur le <i>locale</i> en cours, et sera donc dynamiquement modifiable...
LOGLANGENG / NOLOGLANGENG	Cette option permet de forcer les valeurs d'options LOGLANGCHG et LSWLANG="EN". De cette manière, on s'assure que les messages de la log seront toujours en anglais, ce qui notamment facilite les recherches dans nos bases d'incidents, en cas de problème.

Les valeurs valides pour LSWLANG= sont :

LOCALE
EN | English
FR | French
IT | Italian
DE | German
ES | Spanish
ZH | ChineseS
ZT | ChineseT
JA | Japanese
KO | Korean
PL | Polish
RU | Russian

Si la session SAS est lancée avec le support de l'Unicode, les options par défaut sont les suivantes :

```
-ODSLANGCHG  
-LOGLANGCHG  
-LSWLANG=LOCALE  
-NOLOGLANGENG
```

Ainsi, la langue de sortie de l'ODS et les messages de la log (pour ceux qui sont traduits) dépendront du *locale* courant. Voici un exemple :

```
options locale=fr_FR; /* puis locale=en_US */  
title1 "LSWLANG=%SYSFUNC(getoption(LSWLANG)) " ;  
title2 "LOCALE=%SYSFUNC(getpxlocale()) " ;  
footnote "%sysfunc(datetime()),nldatm." ;  
  
proc freq data=sashelp.class;  
    table age / measures ;  
run;  
  
data _null_ ;  
    put i= ;  
run;
```

Résultat :

The screenshot shows the SAS Results Viewer interface. On the left is a tree view of the results, and on the right is the main output window. The output window displays two frequency tables for the same data, one in French and one in English.

Table 1: Procédure FREQ (LOCALE=fr_FR)

Age	Fréquence	Pourcentage	Fréquence cumulée	Pctage cumulé
11	2	10.53	2	10.53
12	5	26.32	7	36.84
13	3	15.79	10	52.63
14	4	21.05	14	73.68
15	4	21.05	18	94.74
16	1	5.26	19	100.00

Table 2: The FREQ Procedure (LOCALE=en_US)

Age	Frequency	Percent	Cumulative Frequency	Cumulative Percent
11	2	10.53	2	10.53
12	5	26.32	7	36.84
13	3	15.79	10	52.63
14	4	21.05	14	73.68

Et dans la log :

```
NOTE: Variable i non initialisée.  
i=.  
  
[...]  
  
NOTE: Variable i is uninitialized.  
i=.
```

7. LIENS UTILES

Documentation Base SAS (9.4, 9.3, 9.2, 9.1.3)

<http://support.sas.com/documentation/onlinedoc/base/index.html>

SAS 9.4 National Language Support (NLS): Reference Guide, Second Edition

<http://support.sas.com/documentation/cdl/en/nlsref/67110/HTML/default/titlepage.htm>

Time Zone IDs and Time Zone Names – Region: Europe

<https://support.sas.com/documentation/cdl/en/lesysoptsref/68023/HTML/default/viwer.htm#n0luzv1jyk8awyn1ntfqnh7v34gc.htm>

Dictionary of Functions for NLS: Functions by Category

<http://support.sas.com/documentation/cdl/en/nlsref/67110/HTML/default/p0ydidiqibf0vbn1lj11cv9men9.htm>

Internationalization Compatibility for SAS String Functions

<http://support.sas.com/documentation/cdl/en/nlsref/67110/HTML/default/p1pca7vwjwucin178l8qddjn0gi.htm>

LSWLANG System Option

<http://support.sas.com/documentation/cdl/en/nlsref/67110/HTML/default/p12fwzqu1sw32gn1m2usn1s4jml3.htm>

UTF-8

<http://fr.wikipedia.org/wiki/UTF-8>

Linguistic Collation: Everyone Can Get What They Expect

Sensible Sorting for Global Business Success

http://support.sas.com/resources/papers/linguistic_collation.pdf

Temps universel coordonné (UTC)

http://fr.wikipedia.org/wiki/Temps_universel_coordonn%C3%A9

[Article en français] Comprendre l'option encoding dans SAS

http://www.sas.com/offices/europe/france/services/support/articles/US2010_Q4_option_encoding_sas.html

8. CONCLUSION

Dans cette première partie, nous avons vu un certains nombre de concepts qui vous permettront d'adapter votre application SAS à un environnement international. Nous avons également passé en revue les éléments de langage SAS qui prennent en compte le contexte linguistique et géographique lors de l'exécution du programme.

Rendez-vous très bientôt pour le second article de cette série, où nous aborderons la localisation de vos programmes, c'est-à-dire la traduction des chaînes de caractères pour chacun de vos utilisateurs.

Ludovic BREYSSE

Consultant Support Clients SAS France