

## The Power of SAS National Language Support - Embrace Multilingual Data

Chao Wang, MSD R&D (China) Co.,Ltd. Beijing

### ABSTRACT

As clinical trials take place around the globe, data in different languages may need to be handled. The usual solution for this issue is to translate the data to English, after some manipulations, results then will be translated back to the original language.

Since version 9.1.2, SAS software can provide a specific function called National Language Support, this issue would be easy to handle. In this paper, an introduction will be made on: 1) how to build the ready SAS session to import and display the mixed data with NLS system option 2) how to use the edge tool – SAS K-Function (including K-macro Function) to do data manipulations and 3) how to display any character in reports with the Unicode support. Here would use a multilingual (Chinese and English) AE dataset as an example on SAS 9.3 PC version. The related applications on the UNIX server and SAS Enterprise Guide would also be discussed.

### INTRODUCTION

When conduct a multi-national study, data often comes from various countries using different languages and scripts, especially in Asia area. The different sources data will make the clinical operation more complex and cost additional budget and time. It always needs to translate the raw data to English and save the new information in database. And analysis and reporting are based on English words. After review, then translate the English reports back to the local language. Besides the CSR, the data monitoring may require to do some analysis based on the unclear raw data. The indirect working way will give more trouble to meet the timeline. Although this action is sometimes required by the regulatory or company, less introduction and support to handle the multilingual data are also the reasons.

As we know, the data can be stored in different systems (Relational Database <Oracle, Inform or MS-SQL and etc.>, Excel Spreadsheets, or plain text files) under different platforms (Windows, UNIX). When the data are from different companies or organizations, it will make the data sets with different encoding methods.

### DATA EXCHANGE

Generally, the English data sets are always encoded by ASCII. So the data exchange through different organizations or platforms are very smooth. When meet the mixed multilingual data, the different encoding will give more challenges for programming work.

### COMMON PROBLEM

The first challenge is the dataset can't open. The following dataset (tmp.sas7bdat) will be used as an example.

**Table 1** Example Data Set

Subject Identifier for the Study	Reported Term for the Adverse Event	Body System or Organ Class	Causality	End Date/Time of Adverse Event	Start Date/Time of Adverse Event
1002	体温升高	全身性异常	RELATED	2008-03-09	2008-03-09
1003	右眼结膜炎	视觉异常	NOT RELATED		2008-03-23
1017	疲劳乏力	全身性异常	RELATED	2008-03-10	2008-03-08
1019	胃部疼痛(胃炎)	胃肠系统异常	NOT RELATED		2008-03-22
1020	经期延长	女性生殖系统异常	RELATED	2008-03-17	2008-03-08
1020	下腹痛	胃肠系统异常	RELATED	2008-03-15	2008-03-14
1022	头痛	中枢及周围神经系统	RELATED	2008-04-14	2008-04-14
1023	肌肉痛	全身性异常	RELATED	2008-04-14	2008-04-13
1027	感冒	呼吸系统异常	NOT RELATED		2008-04-23
1028	变态反应	全身性异常	NOT RELATED	2008-04-17	2008-04-13
1030	肌肉痛	全身性异常	RELATED	2008-04-13	2008-04-13
1030	咳嗽	呼吸系统异常	NOT RELATED	2008-04-13	2008-04-13
1034	恶心呕吐	胃肠系统异常	NOT RELATED	2008-04-14	2008-04-14
1034	肌肉痛	全身性异常	NOT RELATED	2008-04-15	2008-04-14
1034	头痛	中枢及周围神经系统	NOT RELATED	2008-04-17	2008-04-14
1034	疲劳乏力	全身性异常	NOT RELATED	2008-04-17	2008-04-14
1034	咳嗽	呼吸系统异常	NOT RELATED		2008-04-17
1035	头痛	中枢及周围神经系统	RELATED	2008-04-14	2008-04-14
1035	腹痛	胃肠系统异常	NOT RELATED	2008-04-15	2008-04-15
1036	疲劳乏力	全身性异常	NOT RELATED	2008-04-18	2008-04-16
1037	咳嗽	呼吸系统异常	NOT RELATED		2008-04-19
1037	感冒	呼吸系统异常	NOT RELATED		2008-04-19

The most common issue is lost the characters during the transcoding.

**NOTE:** Data file TMP.AE.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.  
**WARNING:** Some character data was lost during transcoding in the dataset TMP.AE. Either the data contains characters that are not representable in the new encoding or truncation occurred during transcoding.

**ERROR:** Some character data was lost during transcoding in the dataset TMP.AE. Either the data contains characters that are not representable in the new encoding or truncation occurred during transcoding.

The WARNING information is occurred when open the dataset and the ERROR message is occurred when manipulate the datasets.

These issues are always pop-up when using the SAS 9.3 (English, Default). To resolve it, the **SAS 9.3 (Unicode Support)** is recommended. **SAS 9.3 (English with DBCS)** needs to update the **Locale** Option in the configure file (sasv9.cfg, the default is support Japanese). These two new SAS icons can be found through the steps below,

**Windows -> Start -> Programs -> SAS -> Additional Languages.**

**Table 2** Inappropriate Display by SAS 9.3 (English)

SUBJID	AETERM	AEBODSYS	AEREL	AEENDTI	AESTDTI
1002	101AEy 8	全身性异常	RELATED	2008-03-09	2008-03-09
1003	右眼结膜	视觉异常	NOT RELATED		2008-03-23

Another issue may meet is the inappropriate display. As Table 2, the data set may be opened by SAS 9.3 (English) without ERROR/WARNING messages. But the data can't be manipulate (like where SUBJID = "1002") due to the incorrect encoding.

Even use SAS 9.3 (Unicode Support) to open the data, another type of inappropriate display may be met as below.

**Table 3** Inappropriate Display by SAS 9.3 (Unicode Support)

Subject Identifier for the Study	Reported Term for the Adverse Event	Body System or Organ Class	Causality	End Date/Time of Adverse Event	Start Date/Time of Adverse Event
1002	体温升高	全身性异常	RELATED	2008-03-09	2008-03-09
1003	右眼结膜炎	异常	NOT RELATED		2008-03-23
1017	疲劳乏力	全身性异常	RELATED	2008-03-10	2008-03-08
1019	胃部疼痛(胃炎)	胃肠道异常	NOT RELATED		2008-03-22
1020	腰腿痛	女性生殖系统异常	RELATED	2008-03-17	2008-03-08
1020	下腿疼痛	胃肠道异常	RELATED	2008-03-15	2008-03-14
1022	头痛	中枢及外周神经系统异常	RELATED	2008-04-14	2008-04-14
1023	肌肉痛	全身性异常	RELATED	2008-04-14	2008-04-13
1027	感冒	呼吸系统异常	NOT RELATED		2008-04-23
1028	反应	全身性异常	NOT RELATED	2008-04-17	2008-04-13
1030	咳嗽	呼吸系统异常	NOT RELATED	2008-04-13	2008-04-13
1030	肌肉痛	全身性异常	RELATED	2008-04-13	2008-04-13
1034	心呕吐	胃肠道异常	NOT RELATED	2008-04-14	2008-04-14
1034	肌肉痛	全身性异常	NOT RELATED	2008-04-15	2008-04-14
1034	疲劳乏力	全身性异常	NOT RELATED	2008-04-17	2008-04-14
1034	头痛	中枢及外周神经系统异常	NOT RELATED	2008-04-17	2008-04-14
1034	咳嗽	呼吸系统异常	NOT RELATED		2008-04-17
1035	头痛	中枢及外周神经系统异常	RELATED	2008-04-14	2008-04-14
1035	腰痛	胃肠道异常	NOT RELATED	2008-04-15	2008-04-15
1036	疲劳乏力	全身性异常	NOT RELATED	2008-04-18	2008-04-16
1037	感冒	呼吸系统异常	NOT RELATED		2008-04-19
1037	咳嗽	呼吸系统异常	NOT RELATED		2008-04-19

As highlighted in Table 3, several Chinese Characters can't display correctly. This issue is always caused by the system setting. For Window 7 (English), the default language for non-Unicode programs is **English (United States)**. When change it to **Chinese (Simplified, PRC)**, the data will correct display in SAS Explorer as Table 1. This option can be found through,

**Start -> Control Panel -> Region and Language -> Administrative.**

But sometimes, there are still some flaws for some words. Such as

1034	头痛	中枢及外周神经系统
1034	疲劳乏力	全身性异常
1034	咳嗽	呼吸系统异常
1035	头痛	中枢及外周神经系统

There should be a best solution for this issue and make the Chinese characters display more clearly and be usable for programming as English words.

## BEST SOLUTION

The best solution is creating a customized SAS icon to launch the suitable Session accordingly. Before show this

approach, brief discussion about encoding and transcoding is necessary.

Encoding and transcoding are twins like SAS FORMAT/INFORMAT. Encoding map each character in a character set to a unique numeric representation, which results in a table of all code points as Figure 1. The character set includes the set of characters which are specific to a particular languages, special characters (such as punctuation marks and computer control characters) and unaccented Latin characters A–Z and the digits 0–9.

The encoding map can be treated as an ordered set with numeric index. These numeric (hexadecimal) indexes can be recognized across the system and the value of a character in different encoding map is always different.

Figure 1 Encoding map of Windows Latin 1 (ANSI) (part)

	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0		0	@	P	`	p	NOT USED	NOT USED	NBSP	°	À	Ð	à	ð
	32	48	64	80	96	112	128	144	160	176	192	208	224	240
1	!	1	A	Q	a	q	NOT USED	'	;	±	Á	Ñ	á	ñ
	33	49	65	81	97	113	129	145	161	177	193	209	225	241
2	"	2	B	R	b	r	,	'	ç	²	Â	Ò	â	ò
	34	50	66	82	98	114	130	146	162	178	194	210	226	242

Source: <http://msdn.microsoft.com/en-us/library/cc195054.aspx>

The common Encoding methods are defined by the standards organizations, such as International Organization for Standardization (ISO), American National Standards Institute (ANSI) and Unicode Consortium. Such as

- ASCII WLatin1/ Latin1: The default encoding of SAS 9.3 English (WIN7/UNIX)
- ISO GB 2312-80: Simplified Chinese
- UNICODE UTF-8/UTF-16: Support almost all the language characters in the world

Transcoding is the process of converting data from one encoding method to another. It's very necessary for data exchange.

- The default system encoding in different region and platform are different.
- Data exchange across organization and region are always required

According above discussion, it's better to know the encoding and transcoding method of SAS session and dataset.

### SAS Session

The SAS Session encoding method can be got by reading the option in SAS Configure File or using the code below,

```
proc options option = encoding;
run;
```

And the encoding method can be found in the log file. To change the encoding method, the Option **ENCODING** or **LOCALE** should be updated in the SAS Configure File (\*.cfg). If meet the East Asia languages (Chinese, Japanese, Korea and etc.), the DBCS should also be applied. The DBCS stands for Double-Byte Character Sets. It has a counterpart Single-Byte Character Sets. Because East Asian languages have thousands of different characters, double (two or more) bytes of information are needed to represent each character.

**ENCODING** Option: Specifies the default character-set encoding for the SAS session. The syntax is,

**-ENCODING= ASCIIANY | EBCDICANY | encoding-value (UNIX and Windows)**

**DBCS** Option: Recognizes double-byte character sets (DBCS). SAS Default is SBCS. The syntax is,

**-DBCS | -NODBCS (UNIX and Windows)**

**DBCSLANG** Option: Specifies a double-byte character set (DBCS) language. The syntax is,

**-DBCSLANG language** (CHINESE, JAPANESE, KOREAN and TAIWANESE)

**DBCSTYPE** Option: Specifies the encoding method to use for a double-byte character set (DBCS). DBCS encoding methods vary according to the computer hardware manufacturer and the standards organization. The syntax is,

**-DBCSTYPE encoding-method** (PCMS < Microsoft PC encoding method>, HP15 < Hewlett Packard encoding method> and etc.)

**LOCALE** Option: Specifies a set of attributes in a SAS session that reflect the language, local conventions, and culture for a geographical region. The syntax is,

**-LOCALE locale-name** (zh\_CN, en\_US and etc.)

When LOCALE= is used, the value of the following system options are modified unless explicit values have been specified: ENCODING, DATESTYLE, DFLANG and PAPER SIZE.

To make the SAS support Chinese & English mixed data, there are two ways:

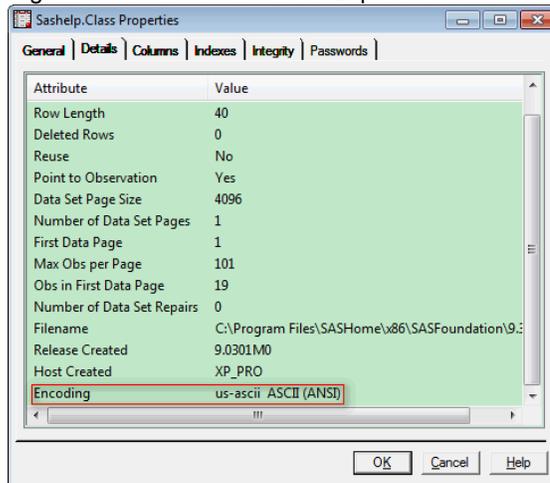
- **-DBCS**
- **-LOCALE "zh\_CN"**

Or

- **-DBCS**
- **-DBCSLANG CHINESE**
- **-DBCSTYPE PCMS**

To make the data more readable, the external data encoding method is also need to know. Here are three different ways to find this information:

1. Right Click the dataset in SAS Explorer and find this information through the **Properties** menu like below.

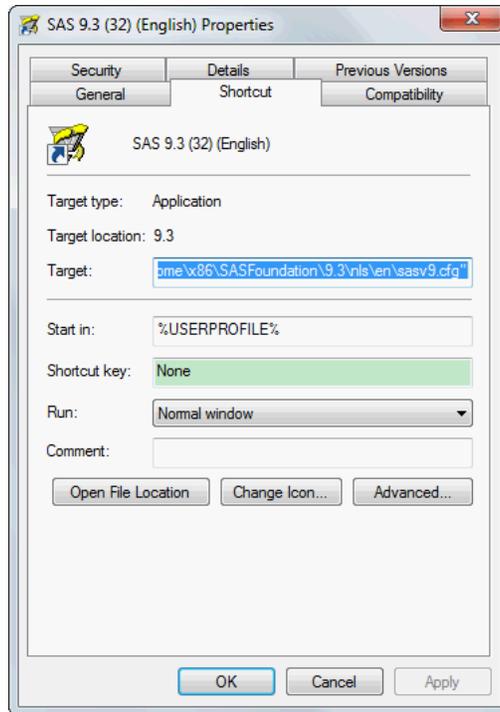


2. Use SAS PROC CONTENTS and find the encoding method in the output
3. Use SAS File I/O Functions

```
%LET DSID=%SYSFUNC(open(sashelp.class,i));  
%PUT %SYSFUNC(ATTRC(&DSID,ENCODING));
```

After get the encoding method for the external data, the ENCODING option can be updated according.

These Options can be updated in SAS Configure File. The default File is under the *SASFoundation* root folder and always named as **sasv9.cfg**. For **SAS 9.3 (Unicode Support)** and **SAS 9.3 (English with DBCS)**, the configure files are under the different subfolder of NLS (... \SASFoundation\9.3\NLS). To be more customized, a new configure file can be created and saved in some location. For example, to read the Chinese and English mixed data (encoding = "euc\_cn"), we create a new SAS Icon (by copy of the default SAS English) and named the updated configure file as **sasce.cfg**, then change the Icon Properties as below,



The default Target statement is "C:\Program Files\SASHome\x86\SASFoundation\9.3\sas.exe" -CONFIG "**C:\Program Files\SASHome\x86\SASFoundation\9.3\win\en\sasv9.cfg**".

Updates the bold strings with the new configure file location. Now, the mixed data can be best read.

## BEST PRACTICE

By changing the Options, the most suitable environment may be built for multi-lingual data. But some information may be still lost during the Transcoding process due to the reasons below,

- Encodings can conflict with another.
- Characters in one encoding might not be present in another encoding.

So during the clinical trial, from Database design to Data analysis and reporting, the data encoding and transcoding issue should be carefully treated. Instead of keeping the data in separate databases, it is more efficient to maintain a central repository that can support a global set of languages (such as encoding with UTF-8/UTF-16). This will help company to store, process, and deliver multilingual data more efficient.

## DATA MANIPULATION

When data can be correct loaded in the SAS, then the manipulation techniques are coming.

## SAS OPTIONS

In previous part, the environment control Options (DBCS/LOCAL/ENCODING) have been introduced. These environment Options should be assigned before SAS Session launched.

Meanwhile, the **ENCODING= Data Set Option** is also very important when read and write data. It can override the encoding to use for reading or writing a SAS data set.

- When read data from a data set, SAS transcodes the data from the specified encoding to the session encoding.

- When write data to a data set, SAS transcodes the data from the session encoding to the specified encoding.

When work with the datasets which are over two different encodings, by specifying the option **ENCODING=ANY**, SAS can create data set that contains mixed encodings, and suppress transcoding for either input or output processing.

```
data mixed (encoding=any);
  set work.encoding1 work.encoding2;
run;
```

For example, the tmp1 dataset can transcode to UTF-8. The UTF-8 is a recommend encoding method when handle the multilingual data, since this encoding map include almost all language characters across the whole world.

```
data tmp2 (encoding="UTF-8");
  set tmp1;
run;
```

Another important Option is **SORTSEQ = Option in the PROC SORT statement**. When the SORT Procedure is executed, the data sequence is determined by the sort order (higher, lower, or equal to) of a particular character in relation to other characters. The relation is called as collating sequence. The default collating sequence is binary collation, and the order is defined in the code page of each encoding map. For SBCS, ASCII (most popular) and EBCDIC (always on IBM Mainframe) are two widely used encoding method for data processing.

Currently, this Option support the user supplied (encoding names) or the following western aliases (ASCII, EBCDIC, DANISH, FINNISH, ITALIAN, NATIONAL, POLISH, REVERSE, SPANISH and SWEDISH). For example, tmp1 data can be sorted by the Chinese characters (AEBODSYS AETERM) by specified EUC-CN in the option.

```
proc sort data = tmp1 out = tmp3 sortseq = "euc-cn";
  by AEBODSYS AETERM SUBJID AESTDTI AEENDTI;
run;
```

Besides these two Options, **VALIDVARNAME= System Option** (VALIDVARNAME=V7 | UPCASE | ANY) is a very useful Option. It makes the variable name more flexible. For other Option information, please find the details in SAS NLS Reference Guide.

## SAS K-FUNCTION

As we know, there are many situations to handle the string by SAS Functions or CALL Routines during the data processing. And as above description, the SBCS and DBCS are saved in different ways in the system. So when use the function to handle the mixed string or mixed data, the KFUNCTION needs to show up.

In this section, SAS provides 3 levels of Internationalization (I18N) compatibility for each SAS string function.

- I18N Level 0: This function is designed for SBCS data. Do not use this function to process DBCS data.
- I18N Level 1: This function should be avoided, if possible. Since it not works well for DBCS encodings under certain circumstances. (Function: FIND, REPEAT, SUBSTRN and etc. )
- I18N Level 2: This function can be used for both SBCS and DBCS data. (Called as K-Functions)

Most of the K-Functions are **character-based** offset-length instead of byte-based offset-length. It will help the program more readable. Also, most K functions require strings encoded in current SAS session encoding.

Here are the frequently-used string functions and its counterparts.

- Compare: COMPARE vs. KCOMPARE
- Connect: CAT vs. KSTRCAT
- Compress: COMRESS vs. KCOMPRESS
- Count: COUNT vs. KCOUNT
- Index: INDEX vs. KINDEX
- Length: LENGTH vs. KLENGTH
- Sub-string: SCAN vs. KSACN; SUBSTR vs. KSUBSTR; TRIM vs. KTRIM

- Align: LEFT vs. KLEFT; RIGHT vs. KRIGHT

And SAS also provides the related K-Macro Functions. Such as, %KCOMPRESS, %KINDEX, %KLEFT/ %QKLEFT, %KLENGTH, %KSCAN, %QKSCAN, %KSUBSTR / %QKSUBSTR and %KUPCASE / %QUPCASE.

## SAS FORMAT/INFORMAT

Besides the K-Functions, there are also some international FORMATS/INFORMATs for multi-lingual data handing. Most of them are localized related, such as Date/Time and Currency. There are also some Formats for DBCS handing, but only support the Japanese.

Like SBCS, the DBCS character can be used in the define of PROC FORMAT/INFORMAT. Here is an example.

```
proc format;
  value $CO "皮疹" = 1
           "头痛" = 2
           "头晕" = 3;
run;

data tmp2;
  set tmp1;
  format NAME $20.;
  keep SUBJID NAME;
  NAME = '身体适 || AETERM;
run;

data ex2;
  set ex1;
  format NAME1 $20.;
  order = input (put (NAME1, $CO.), best.);
  NAME1 = KSUBSTR (NAME, 5);
  if NAME1 in ("皮疹" "头痛" "头晕");
run;
```

In the above code, the mixed lingual-data can be treated identical in the programming process.

## OUTPUT WITH MULTILINGUAL DATA

After the manipulation, the final data/result should be outputted in the documents. The documents formats may include but not limited to RTF, PDF, PS and TXT. By using different technique, the multilingual result can be displayed correctly in different formats.

## OUTPUT WITH ODS

SAS ODS (Output Delivery System) is designed to overcome the limitations of traditional SAS line-printer (monospace fonts) and make it easier for us to format the output. The ODS Statements and ODS PRINTER (which includes Printer, PCL, PostScript and PDF) provides greater flexibility with the best possible to present result.

By using the SAS ODS OUTPUT, the multilingual characters can be directly displayed as below.

### <ODS PDF, ODS GRAPH> <Two Figures>

For the multilingual characters (especially for the special symbol, such as alpha, beta and etc.) which are shown in the Title, AXIS Labels or other positions, they can be displayed by using the ODS ESCAPECHAR Statement (This function is available since SAS 9.2). The syntax is

```
ods escapechar='^';
^{unicode <value>}
```

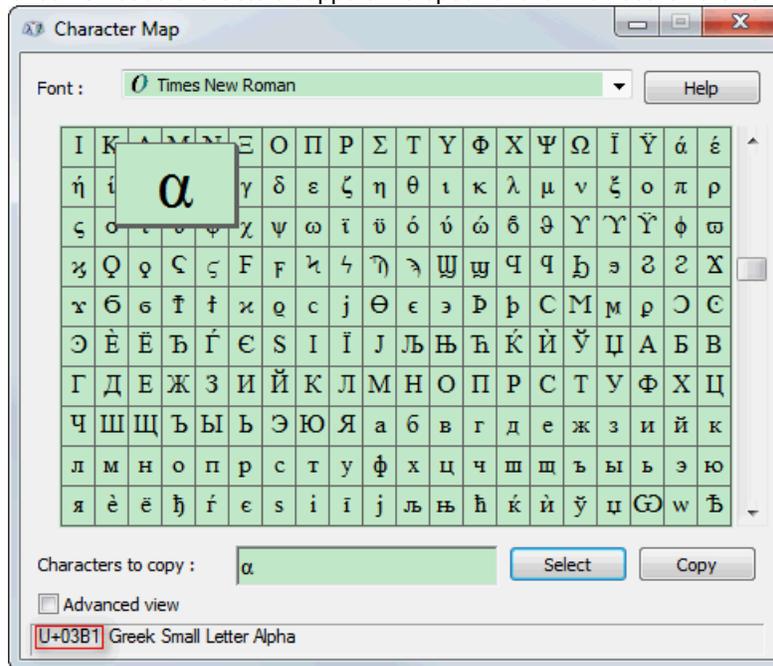
As the title of Figure 1, the SAS code is below,

```
title 'Unicode Alpha is ^{unicode 03B1}';
```

By using the four-digit Unicode Value, the symbol can be incorporated in the output. The related Unicode value can be got by different ways.

- Web Tools: Google, Fileformat.INFO (<http://www.fileformat.info/info/unicode/char/search.htm>) and etc.
- Windows Tool: Character Map (Win7: Start/All Programs/Accessories/System Tools/Character Map). As the snapshot below, the Unicode value of **α** is **03B1**.

It's a good practice to check Unicode characters support in a specific font when search the value.



## OUTPUT WITH RTF CODE

Sometimes, the industry needs highly customized report-ready tables. This type of table can't be created easily by SAS PROC or ODS PRINTER. But it can be completed by output the prepared data with RTF language into Rich Text Format or Microsoft Word (DOC) Format.

As mentioned before, the multilingual data correctly display is relied on the related FONT. When using the RTF language to output the result. The related FONT set should be specified in the initialization part.

Here are two examples.

### English Version:

```
put "{\rtf1\ansi \deff0\deflang1033"@;
put "{\fonttbl{\f0\froman Times New Roman;}}";
put "{\f1\froman\fcharset2\fprq2 Symbol;}}";
put "{\f2\fswiss\fcharset0\fprq2 Arial;}}";
put "{\f3\fmodern\fcharset0\fprq1 SAS Monospace;}}";
put "{\f4\fmodern\fcharset0\fprq2 Courier New;}}";
put "{\f166\froman\fcharset161\fprq2 Times New Roman Greek;}}";
put "{\f266\fswiss\fcharset161\fprq2 Arial Greek;}}";
```

### Chinese Version: <support both English & Chinese>

```
put "{\rtf1\ansi\ansicpg936\uc1\deff0\deflang2052\deflangfe2052"@;
put "{\fonttbl{\f0\froman\fprq2\fcharset134\cpg936 Times;}}";
put "{\f1\froman\fcharset2\fprq2 Symbol;}}";
put "{\f2\fswiss\fcharset0\fprq2 Arial;}}";
```

```

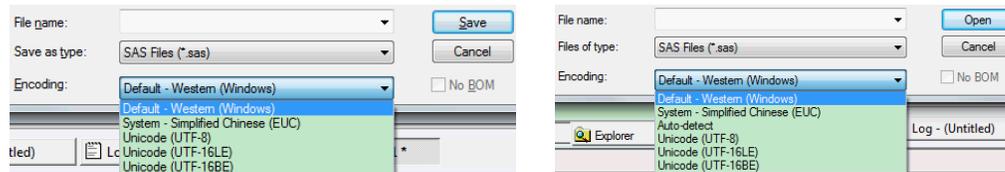
put "{\f3\fmodern\fcharset0\fprq1 SAS Monospace;}";
put "{\f4\fmodern\fcharset0\fprq2 Courier New;}";
put "{\f166\froman\fcharset161\fprq2 Times New Roman Greek;}";
put "{\f266\fswiss\fcharset161\fprq2 Arial Greek;}";

```

With regard to the special symbols, they should be coded by RTF code. Such as  $\alpha$  is `\f166\`e1`. The symbol and corresponding RTF code can be searched in the Web or Character Map.

## SAVE THE MULTILINGUAL PROGRAM

Although the multilingual characters can be directly written in SAS Enhanced Editor, the information may be lost due to inappropriate SAVE/OPEN action. Please select the right encoding method when SAVE/OPEN the SAS programs.



The file default encoding methods are Western, Unicode (UTF-8), Unicode (UTF-16LE) and Unicode (UTF-16BE). The Simplified Chinese (EUC) method is depended on the Windows system setting. The Unicode (UTF-8) is the recommend method for mixed lingual program encoding.

## CONCLUSION

SAS National Language Support is a very powerful package. In this paper, several techniques are introduced to conquer the potential problems when handing with the multilingual data from data source to output report.

- With the encoding and transcoding technique, the data can be exchanged through different formats and platforms.
- By using the specific Options (DBCS, LOCALE, ENCODING and etc.), SAS can directly read and correct display the data in SAS Explorer.
- SAS provides set of tools (SAS KFUNCTION/FORMAT/INFORMAT) to support the multilingual data manipulation.
- The multilingual data/result can be easily displayed in the industry required formats (data, pdf, txt or rtf).

## REFERENCES (HEADER 1)

- Manfred Kiefer, "Processing Multilingual Data with the SAS @9.2 Unicode Server", SAS Heidelberg
- SAS @ 9.3 National Language Support (NLS): Reference Guide, 2011, SAS Institute Inc., Cary, NC, USA

## ACKNOWLEDGMENTS

Chao would like to thank Liping Zhang, Changhong Shi and Huadan Li, who generously provided helpful comments and encouragements.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Chao Wang  
 Enterprise: MSD R&D (China) Co.,Ltd  
 Address: 2/F Building 22, Universal Business Park, No.10 Jiuxianqiao Road, Chaoyang District  
 City, State ZIP: Beijing, 100015  
 Work Phone: +86 1059873419

E-mail: [chao.wang3@merck.com](mailto:chao.wang3@merck.com)

Web: <http://www.superwang.com/>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.