The SAS® LOCALE Option
Win an international passport for your SAS code
Xiaojin Qin, Covance Pharmaceutical R&D Co., Ltd., Beijing, China

ABSTRACT
Want to win an international passport for a round-the-world journey? Start with your SAS code!

SAS internationalization is the step that generalizes your project to be language independent. This allows you to write code once, and then have it run in different cultural environments with the appropriate cultural interpretation; for example, write your code so it uses formats that are familiar to you in China, yet when your French clients run the code it will use formats familiar to them. LOCALE, one of the internationalization features of SAS, can help your code be a world traveler. This paper will show the use of LOCALE in our SAS sessions as a major part of making the round-the-world journey a reality.

You will see how a minor change can make a major difference.

INTRODUCTION

Overview of Locale Concepts
The term internationalization is not new for us anymore. Nowadays, more and more companies and organizations are operating in more than one country. In the pharmaceutical industry, most large clinical trials, particularly Phase III trials, are conducted at several clinical research centers in different geographic locations. We already have corporate standards for our trials; we also need to look at standards to ensure the programmers and managers around the world can each participate in a way in which everyone is comfortable. It’s a question for the business as well as for the programmer.

Localization involves changing not only the language interaction, but also other relevant changes such as the display of numbers, dates, currency, and so on. Other types of data, such as sounds and images, might require localization if they are culturally sensitive. In this paper, we will focus on the topic of LOCALE, one of the internationalization features of SAS which can help our code win the international passport.

LOCALE Option
The LOCALE= system option is used to specify the locale, which reflects the local conventions, language, and culture in a geographical region.

The setting of the LOCALE option affects the following SAS components:
• All national language (NL) formats and informats. The NL formats and informats are locale-sensitive and are described later in this paper.
• Functions SASMSG and SASMSGL used in the string externalization mechanism.
• Other options such as DFLANG, DATESTYLE, PAPERSIZE, MAPEBCDIC2ASCII, ODSLNGCHG, LOGLANGCHG, LSWLANG, LOCALEDATA, and TIMEZONE.

LOCALE= Values
LOCALE= values can be specified by using the SAS name or the POSIX name. For some locales, you can also specify a country name or region to provide more specific locale information. For example, valid locales for China are Chinese_China and zh_CN.

Codebox 1: Example of Locale=Values.
Refer to “Values for the LOCALE= System Option” in the SAS 9.4® National Language Support (NLS): Reference Guide for a complete list of locale values supported by the SAS System.
The LOCALE option does not store values in the SAS registry. Run PROC OPTIONS to display the default value of the LOCALE option.

```
19         proc options group=LANGUAGECONTROL;
20         run;
```

Table 1: Example of displaying default Locale=Values.

<table>
<thead>
<tr>
<th>Setting the Locale Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>The option LOCALE can be set using any of the following methods:</td>
</tr>
<tr>
<td>• Within SAS Options, Tools ’ Options ’ System ’ Environment control ’ Language control.</td>
</tr>
<tr>
<td>• Within the configuration file SASv9.cfg, by altering the following line: -LOCALE &quot;EN_GB&quot;</td>
</tr>
<tr>
<td>• Within SAS code, use an options statement. It’s the trick of the magic! We’ll see how it works in the scenario part.</td>
</tr>
<tr>
<td>• SETLOCALE() function.</td>
</tr>
</tbody>
</table>

This paper mainly discusses using options within SAS code to set LOCALE given its flexibility. For example your startup configuration is set to Chinese_china so all the appropriate culturally sensitive setting will adhere to the Chinese standard. Your project will eventually be transferred to a client in France so you want to be sure none of the Chinese settings are passed through. You could change your configuration file and restart SAS to see if the project no longer reflects the Chinese standard, however changing configuration files should generally be avoided unless the change is necessary. Since this is not a necessary change, you can simply set the LOCALE= option and rerun your code without having to restart SAS. Once you verify the results are as you expect you can once again set the LOCALE= option, this time back to Chinese_china; once again you do not need to restart SAS.

As stated earlier, the setting of the LOCALE option affects other SAS components. Example below shows that the NL formats is decided by the value of LOCALE.

```
*************************************************************/;
19         proc options group=LANGUAGECONTROL;
20         run;
```
Codebox 2: Example of NL format

SETLOCALE Function

SETLOCALE() is a SAS function that can be used for the following purpose:

1. Setting SAS Locale: SETLOCALE(sas_locale)

   sas_locale : specifies a SAS locale name by using the SAS name or the posix name. You can also specify
   the locale alias.

2. Customize single locale elements: SETLOCALE(key, value)

   key: specifies a SAS locale element key.
   value: specifies a value for the locale element.

3. Customize single locale elements: SETLOCALE(category_name, sas_locale)

   category_name: values can be the following:
   • LC_TIME
   • LC_MONETARY
   • LC_NUMERIC
   • LC_ALL

Codebox 3: Example of SETLOCALE function changes the value of the specified key, DATE_YEAR_FORMAT

HOW LOCALE WORKS IN SAS

Locale interacts with other components of SAS: below are some key concepts you need to know before you
look into how it works.

Locale with National Language Support (NLS)

National Language Support (NLS) is a set of features to ensure SAS applications conform to local language
conventions. Software applications that incorporate NLS can avoid dependencies on language-specific or
cultural-specific conventions for software features such as:

• character classifications
• character comparison rules
• code sets
• date and time formatting
• interface
• message-text language
• numeric and monetary formatting
The NL formats are locale-sensitive and should return values that reflect the locale being set. Formats such as international date and datetime formats, time zone formats, currency representation and numbers are supported by SAS.

<table>
<thead>
<tr>
<th>English Format</th>
<th>International Format</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>NLDATE.</td>
<td>10</td>
<td>200</td>
<td>10</td>
</tr>
<tr>
<td>DATETIME</td>
<td>NLDATM.</td>
<td>10</td>
<td>200</td>
<td>10</td>
</tr>
<tr>
<td>DOWNAME</td>
<td>NLDATENAME</td>
<td>4</td>
<td>200</td>
<td>10</td>
</tr>
<tr>
<td>NONNAME</td>
<td>NLDATENAME</td>
<td>4</td>
<td>200</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LOCALE=</th>
<th>Currency</th>
<th>National Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>English, United States</td>
<td>U.S. dollars</td>
<td>$12,345.00</td>
</tr>
<tr>
<td>French, Canada</td>
<td>Canadian dollars</td>
<td>12,345.00</td>
</tr>
<tr>
<td>French, France</td>
<td>French euros</td>
<td>12,345.00€</td>
</tr>
</tbody>
</table>

Table 2: Example of NL formats.

When setting LOCALE to zh_CN (Chinese_China), then the number “12345” is formatted as “¥12,345.00” by NL format NLMNY15.2.

```sas
10.InputStream.;
19 Options locale=zh_CN;
20
21 data _null_;
22   x=12345;
23   put "¥" x nlmny15.2;
24 run;
```

Codebox 4: Example of getting a RMB symbol in SAS output

Locale with Encoding

Encoding is a set of characters that have each been assigned a unique number. The ENCODING option sets the encoding that SAS uses for processing external data. FILENAME, INFILE, or FILE statements with an encoding option allows you to tell SAS which encoding should be used to read or write external files. The Output Delivery System (ODS) will create output using the encoding that matches the ENCODING system option.

When the ENCODING option is set, the TRANTAB option will be set to match the ENCODING system option. Transcoding is the process of converting data from one encoding to another. Transcoding is often necessary when you move data between operating environments that use different ENCODING. The transport formats that are set by the TRANTAB option are used by PROCs CPORT and CIMPORT to transfer SAS data files. They are also used by PROCs UPLOAD and DOWNLOAD for transferring files and catalogs, resubmitting code to the server, and returning logs and listings to the client.

As we can see from the table 1 above, when LOCALE is set, the DFLANG, TRANTAB and ENCODING options are also updated to match the locale you selected.

<table>
<thead>
<tr>
<th>POSIX Locale</th>
<th>Windows Encoding</th>
<th>UNIX Encoding</th>
<th>z/OS Encoding</th>
<th>DFLANG=</th>
<th>DATESTYLE=</th>
<th>PAPERSIZE=</th>
</tr>
</thead>
<tbody>
<tr>
<td>af_ZA</td>
<td>watin1</td>
<td>latin1</td>
<td>open-1047</td>
<td>English</td>
<td>YMD</td>
<td>A4</td>
</tr>
<tr>
<td>ar_AE</td>
<td>warabic</td>
<td>arabic</td>
<td>open-425</td>
<td>English</td>
<td>DMY</td>
<td>A4</td>
</tr>
</tbody>
</table>

Table 3: Example of Default Values for the ENCODING, DFLANG, DATESTYLE, and PAPERSIZE System Options Based on the LOCALE= System Option

The locale you set also has a common encoding that is used most often on the platform where the SAS System is running. When the LOCALE option is set, the ENCODING option is set to match this common locale encoding and also sets the TRANTAB option to support the locale encoding. However, if the ENCODING system option is also specified at startup, the encoding in the ENCODING option will take precedence over the startup locale encoding and will become the new SAS encoding for the session.
If neither the LOCALE= nor the ENCODING= option is set, a default value is set. Values for the ENCODING= system option depend on the operating environment. The default encoding varies according to the operating environment and the locale. The common encoding methods are listed here:

- EBCDIC (Extended Binary Coded Decimal Interchange Code).
- ISO (International Organization for Standardization) 646 family.
- ISO 8859 family and Windows family.
- Unicode (Unicode can accommodate basically all of the world’s languages). There are three Unicode encoding forms: UTF-8, UTF-16 and UTF-32.

Locale with Format
You can use the PROC FORMAT statement to create a Locale-Specific Format Catalog.

The example below demonstrates how to create a format in two languages, English and Chinese, and then shows how to access the English and Chinese format catalogs to print a data set in the two languages.

1. Create the AGE format.

```sql
/*no locale information*/;
proc format lib=work.formats;
value age
  low - 5 = 'baby'
  6 - 12 = 'child'
  13 - 15 = 'teen'
  16 - 30 = 'youth'
  31 - 50 = 'midlife'
  51 - high = 'older'
;
run;

/* Use the English format catalog*/
options locale=en_US;
proc print data=datatst;
format agegrp age.;
run;

/* Use the Chinese format catalog*/
options locale=zh_CN;
proc print data=datatst;
format agegrp age.;
run;
```

2. Add the locale-specific format catalogs to the format search path.
```sql
options fmtsearch = (work/locale);
```

3. Run the above code and get the following results.

```
Obs  sex age agegrp
   1   M   6  baby
   2   F   6  child
   3   M  12  child
   4   F  13  teen
   5   M  16  teen
   6   F  16  youth
   7   M  30  youth
   8   F  36  midlife
   9   M  51   older
  10   F 100   older
```

```
Obs  sex age agegrp
   1   M   5   婴儿
   2   F   6   小孩
   3   M  12  小孩
   4   F  13  少年
   5   M  16  少年
   6   F  16  青年
   7   M  30  青年
   8   F  36  中年
   9   M  51  老年
  10   F 100  老年
```

Table 4: Results from above two formats.
Locale with PROC SORT Statement

In the PROC SORT statement, we can use SORTSEQ= `<collating-sequence>` to achieve a locale appropriate collating sequence.

`Collating-sequence` can be a collating-sequence, a translation table, an encoding, or the keyword LINGUISTIC. Only one collating sequence can be specified. When you specify LINGUISTIC as the value for the SORTSEQ option, a linguistic collation will be invocated. This means the rules and default collating-sequence options based on the language specified in the current locale setting are used. The implementation is provided by the International Components for Unicode (ICU) library. It produces results that are largely compatible with the Unicode Collation Algorithms (UCA). The examples below show the use of PROC SORT with a SORTSEQ= option; the first example uses the collating sequence with the locale that is currently in effect, the second shows using a locale specific collating sequence.

```sas
/* Sorting based on the current SAS Locale session */
PROC SORT data=mydata
   SORTSEQ=LINGUISTIC;
run;

/* Sorting for a specific locale */
PROC SORT data=mydata
   SORTSEQ=LINGUISTIC (LOCALE=fr_FR);
run;
```

Codebox 6: Example of LINGUISTIC sort.

Locale with LOCALEDATA Procedure

The LOCALEDATA procedure enables you to manipulate LOCALE data, e.g. NL formats and informats. Using this procedure, you can view, open and modify LOCALE data and then store the customized locale data into the SAS registry. The following example shows changing some locale specify attributes; this is followed by a short explanation of the steps.

```sas
proc localedat;
   /* Load locale definition from locale DB */
   LOAD SASLOCALE;
   /* Change the elements value */
   MODIFY key=QTR1 value='QTR1';
   MODIFY key=QTR2 value='QTR2';
   MODIFY key=QTR3 value='QTR3';
   MODIFY key=QTR4 value='QTR4';
   MODIFY key=DATE_SHORT_FORMAT value='%Y/%m/%d';
   MODIFY key=DATE_YEAR_FORMAT value='Year(%Y)';
   /* Store the locale definition into registry, this will override the existed definition*/
   SAVE REGISTRY / _ALL_ syntax=SAS;
   /* View the locale definition from registry */
   CONTENTS _ALL_
   quit;
```

Codebox 7: Example of proc localedat.

Looking at the PROC localdata example we see:

1. LOAD SASLOCALE | REGISTRY.
   Specifies which source of locale data will be loaded for manipulation. If you do not specify the LOAD statement, then PROC LOCALEDATA will copy locale data from the current SAS session.

2. MODIFY key=key-name value=key-value | category=category-name value=locale.
   Customizes the locale data by locale element key or by locale category.

3. CONTENTS_ALL | key-name | category-name.
   Specifies the element’s value that will be displayed.

4. SAVE REGISTRY<__ALL__ | key-name | category-name Syntax=SAS>.
Stores the customized locale data into SAS registry.

SAMPLE SCENARIO

Generating Multilingual Reports

As stated earlier, different coding methods may be used in different platforms or operating environments. An ideal approach is to use an encoding that deals with all language groups (for example: Western, and Eastern European and Japanese) together and generate a single output data set. This encoding is UTF-8, a form of Unicode; a universal encoding that can handle characters from all possible languages.

With the SAS® Unicode1 server, a SAS session encoding of UTF-*, it is possible to store, process, and deliver multilingual data.

If LOCALELANGCHG is on, the language switching feature is enabled and the setting of the LOCALE= option determines the language used for procedure output and user interface.

```
options locale=en_US;
title 'Systolic and Diastolic Blood Pressure';
title "LOCALE=%SYSFUNC(getpxlocale())";
ods select BasicMeasures Quantiles;
proc univariate data=BPressure;
  var Systolic Diastolic;
run;
```

```
options locale=ja_JP;
title 'Systolic and Diastolic Blood Pressure';
title "LOCALE=%SYSFUNC(getpxlocale())";
ods select BasicMeasures Quantiles;
proc univariate data=BPressure;
  var Systolic Diastolic;
run;
```

Codebox 8: Example of generating multilingual reports

This example shows the magic of “a minor change can make a major difference”.

<table>
<thead>
<tr>
<th>LOCALE=en_US</th>
<th>LOCALE=ja_JP</th>
</tr>
</thead>
<tbody>
<tr>
<td>The UNIVARIATE Procedure</td>
<td>The UNIVARIATE Procedure</td>
</tr>
<tr>
<td>Variable: Systolic</td>
<td>变数: Systolic</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Basic Statistical Measures</th>
<th>Location</th>
<th>Variability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>121.2727</td>
<td>14.28346</td>
</tr>
<tr>
<td>Std Deviation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Median</td>
<td>120.0000</td>
<td>204.01792</td>
</tr>
<tr>
<td>Variance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>120.0000</td>
<td>65.00000</td>
</tr>
<tr>
<td>Range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interquartile Range</td>
<td>13.0000</td>
<td></td>
</tr>
</tbody>
</table>

Note: The mode displayed is the smallest of 2 modes with a count of 4.

<table>
<thead>
<tr>
<th>基本統計量</th>
<th>位置</th>
<th>ばらつき</th>
</tr>
</thead>
<tbody>
<tr>
<td>平均</td>
<td>121.2727</td>
<td>14.28346</td>
</tr>
<tr>
<td>中央値</td>
<td>120.0000</td>
<td>204.01792</td>
</tr>
<tr>
<td>極値</td>
<td>88.0000</td>
<td>65.00000</td>
</tr>
</tbody>
</table>

Note: 2個の頻度が等しく(度数: 4)。表では極値のうち最も小さい値を表示します。

Table 5: Output of multilingual reports.
CONCLUSION

In a world where our data and programs become international travelers we need to ensure both our programs and our data have an international passport. In this paper I have shown how the LOCALE option helps us win that passport. When working on projects that will be used by people around the corner and around the block we need to heed the saying “When in Rome do as the Romans do”.

REFERENCES


CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:
Name: Xiaojin Qin
Enterprise: Covance Pharmaceutical R&D Co., Ltd.
E-mail: xiaojin.qin@covance.com
Work Phone: (86-10) 57323125

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 registered trademarks or trademarks of their respective companies.