SAS® Viya® 3.4 FAQ for Processing UTF-8 Data

Troubleshooting Tips for Processing UTF-8 Data (Existing SAS Code)

What Is the Encoding of My Data Set?

PROC CONTENTS displays information about the data set header, including the encoding. Data sets in the SASHELP library are encoded as ASCII.

```
proc contents data=sashelp.class;
run;
```

```
   The CONTENTS Procedure

         Data Set Name        SASHELP.CLASS                                  Observations          19
         Member Type          DATA                                           Variables             5
         Engine               V9                                             Indexes               0
         Created              01/17/2016 20:12:45                            Observation Length    40
         Last Modified        01/17/2016 20:12:45                            Deleted Observations  0
         Protection                                                          Compressed            NO
         Data Set Type                                                       Sorted                NO
         Label                Student Data
         Data Representation  SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64
         Encoding             us-ascii  ASCII (ANSI)
```

Alternatively, you can query attributes about SAS tables from the PROC SQL DICTIONARY.TABLES view. Here is an example of PROC SQL code:

```
libname jack '/u/xxxx/vb005/jack' access=readonly;
proc sql;
   select libname, memname, encoding
   from dictionary.tables
   where libname='JACK'
   order by memname;
quit
```

See Dictionary Tables for more information.

Can I Turn Cross–Environment Data Access (CEDA) Off?

If all of the characters in the data sets are ASCII characters (the English upper– and lowercase letters A-Z, plus numbers, punctuation, space, and other characters), then you should have no problem using the data. There is a performance cost because the data set encoding does not match the session encoding, and CEDA must transcode every record.

CEDA was introduced in SAS 9. For more information, see Overview of CEDA.

If your data sets are ASCII characters, then you can change the encoding of your data set to ASCIIANY to prevent transcoding. You can use the CORRECTENCODING option in the PROC DATASETS MODIFY statement to change the encoding that is stored in the data set header. Here is an example:

```
libname myfiles "path to data sets";
proc datasets library=myfiles;
   modify olddata / correctencoding=ASCIIANY;
quit;
```

Can I Use Data Sets Created in an Earlier SAS Release?

The encoding used to create a SAS data set is saved in the data set header. For example, if you are running SAS 9 with a session encoding of WLATIN1, the WLATIN1 encoding is stored in the data set header. SAS expects the character data encoding to be the same as the SAS session encoding. When the encodings do not match, cross–environment data access (CEDA) must be used to read the data.

If the data set encoding is not UTF-8 and the message is a note or warning, the character data should be safe. For example, when the UTF-8 session reads a data set created with a WLATIN1 encoding, you see the following note in your log:

```
NOTE: Data file MYLIB.CARS.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.
```

If the data set encoding is not UTF-8 and the message is an error, transcoding fails:

```
ERROR: Some character data was lost during transcoding in the data set ZHOLD.CARS. Either the
 data contains characters that are not representable in the new encoding or truncation occurred
during transcoding.
```

This error usually means that there is not enough space in one or more character columns in the data set's observation buffer to convert the data to UTF-8.

If this error occurs, you can use the character variable padding (CVP) LIBNAME engine to create an in-memory copy of the data that has larger character columns. The CVP engine adds space to the character columns. By default, the column length is multiplied by 1.5. Use the CVP option CVPMULT= to control the amount of padding. Here is an example of using CVP with the default multiplier:

```
/* Libname for the same data set using the CVP engine.    */
/* The CVP engine widens the character columns.          */
libname mylib CVP "path to data sets";
```

CVP creates a read–only copy of the data. If you want to save a permanent copy of the data, you must create a new data set. You need a LIBNAME statement pointing to the location of your library. The following example uses PROC COPY to save the data. You must use the NOCLONE option in the PROC COPY statement. Otherwise, the attributes from the original data set are duplicated in the new copy.

```
libname zhold cvp "/tstgen/dev/tst-vb005/hstnls/tnlssio/wx6/zh_cn/zeuc";
 libname zhnew "mylib";

 proc copy in=zhold out=zhnew noclone;
```

```
    select class;
    run;

proc contents data=zhnew.class; run;
```

Table 1 UTF-8 Error Messages and Error Message Resolution

Error Message	Resolution
Data Set is UTF-8	CVP is not needed
Data Set is not UTF-8	Truncation Error: CVP is needed
Data Set is ASCII	CORRECTENCODING option can be used

CAUTION! Misuse of ASCIIANY could result in data corruption. ASCIIANY prevents transcoding. This option should be specified only if your data exclusively contains ASCII data.

Should I Create a UTF-8 Data Set?

If your data set has an integrity constraint or an index, you need to re-create the data set in UTF-8. CEDA does not read indexes.

My SAS9 Data Set Is UTF-8. Why Do I See a Message that Cross–Environment Data Access (CEDA) Was Used?

The data set contains attributes of the operating system where that data set was created. If SAS Viya is running on an operating system where the attributes are different, CEDA reads the data even if the data set encoding matches the SAS session encoding. You might see the following note in your SAS log:

```
Note: Data file MYLIB.CARS.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.
```

When a data set is opened with a DATA step or a procedure, SAS Viya examines the data set encoding. If that encoding is not UTF-8, SAS Viya transcodes the character data from the data set encoding to UTF-8.

How Do I Execute a SAS Program that Has National Characters in It if the Program Is not in UTF-8?

If the SAS program is stored in an external file in an encoding that is different from UTF-8, you can read it by using the ENCODING= option. Here is an example:

```
filename in 'your SAS pgm' encoding='Shift-JIS';
%inc in;
```

How Can I Create an External File in Another Encoding?

When you write data to an external file, the data is written out in the session encoding unless you specify an appropriate ENCODING= option. In the following example, a subset of the contacts data set is created, and then the output is written to an external file with an encoding of Shift-JIS:

```
/* Create a subset with Japanese data from the main table */
proc sql;
```

```
   create table japan as
   select * from mylib.contacts where country_e = 'Japan';
quit;
/* Write the output to an external file with an appropriate encoding */
filename out 'external-file' encoding='Shift-JIS'
data _null_;
   set WORK.japan;
   file out;
   put @1 name @31 first @62 street @133 zip @144 city;
run;
```

How Can SAS Read a Data File that Has National Characters in It if the Data File Is not in UTF-8?

When SAS reads data from an external file, it expects the file encoding to match the session encoding. If your external file contains only ASCII characters, there should be no problem in reading the file. If the file contains any characters that are not ASCII (for example, national characters in another encoding), use the ENCODING= option in the FILENAME statement.

The FILE, FILENAME, and INFILE statements support the ENCODING= option, which enables users to dynamically change the encoding for processing external data. SAS reads and writes external files using the current session encoding. The system assumes that the external file is in the same encoding as the session encoding. If your session encoding is UTF-8, and you are creating a new SAS data set by reading an external file, SAS assumes that the file's encoding is also UTF-8. If it is not, the data could be written to the new SAS data set incorrectly unless you specify an appropriate ENCODING= option. Here is an example:

```
filename in 'external-file' encoding='Shift-JIS';
data mylib.contacts;
   infile in;
   length name $ 30 first $ 30 street $ 60 zip $ 10 city $ 30;
   input name first street zip city;
run;
```

Note: The encoding names that are used with the ENCODING= option are documented in the *SAS Viya National Language Support: Reference Guide*.

What String Functions Should I Use in My SAS Code?

String functions and CALL routines manipulate characters and strings. For example, INDEX, LENGTH, SUBSTR, SCAN, LOWCASE, and UPCASE are string functions. These functions are byte-based offset. These functions might not return the expected result in a UTF-8 or DBCS environment with non-ASCII characters. The name in this example has six characters, Mátyás.

```
name="Mátyás" ;
len=length(name) ;
put len= ;
put name= $hex20. ;
```

With a UTF-8 SAS session, the length function returns `len=8`. In UTF-8 encoding, a character can be encoded on 1, 2, 3, or 4 bytes and the string function processes the byte as a unit.

```
   len=8
   name=4DC3A17479C3A173
```

In this example, the small letter *a* with an acute accent is encoded on 2 bytes, C3A1, and is represented in this array:

byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7	byte 8
4D	C3	A1	74	79	C3	A1	73
M	á		t	y	á		s

To avoid problems, you can use the equivalent K functions. The K functions are character–oriented while the regular functions are byte–oriented. In the example, the KLENGTH statement returns 6, which is the number of characters.

How Do I Convert CHAR Columns to the VARCHAR Data Type?

SAS Viya supports the VARCHAR data type for character data. Variables created using the VARCHAR data type vary in width and use character semantics, Variables created using the CHAR data type are fixed-width and use byte semantics. When a VARCHAR variable is passed to a traditional string function or to a K function, SAS assumes that the length represents the number of characters. An offset for a VARCHAR variable always represents a character position.

You can convert CHAR variables to VARCHAR using the CVP engine. The CVP engine converts variables defined as CHAR to a VARCHAR data type when you specify `CVPVARCHAR=YES`.

Here is an example that creates two variables. One variable, ch1, is converted to VARCHAR and the other variable, ch2, is excluded from this conversion by using the `TRANSCODE=NO` attribute:

```
libname baselib 'c:\temp\testdata';
libname cvplib cvp 'c:\temp\testdata' cvpvarchar=yes;
data baselib.latin1;
   length ch1 $ 10 ch2 $ 20;
   attrib ch1 transcode=yes;
   attrib ch2 transcode=no;
   ch1='hello';
   ch2='hello world';
run;
proc contents data=cvplib.latin1;
run;
```

The data can be saved to a CAS table:

```
libname casdata cas sessref=mysess;
data casdata.utf8;
   set cvplib.latin1;
run;

proc contents data=casdata.utf8;
run;
```

How Can I View English Messages in My SAS Log?

If you use a LOCALE option value other than English, you might see log messages displayed in another language. By default, the Unicode server uses the log messages that match the language of the LOCALE option that is set at start-up. If you always want to see English log messages, add the following option to your command line:

```
-loglangeng
```

Note: The default is NOLOGLANGCHG. LOGLANGCHG must be set at start-up.

How Can I View ODS Output in English?

By default, output generated by SAS Viya is displayed in the language of the current locale if SAS provides translations for that language. If you want your output to display in English, set the following option:

```
-noodslangchg
```

Note: The default is ODSLANGCHG. ODSLANGCHG must be set at start-up.

How Does Batch Mode Affect Viewing My Log File?

If you run SAS in batch mode, you must use a UTF-8 editor to view the SAS log file. Otherwise, you might see meaningless characters.