

# **DATA ENCODING: ALL CHARACTERS FOR ALL COUNTRIES**

**DONNA DUTTON, SAS INSTITUTE INC., CARY, NC, USA**

**PAPER DH03**



# SO YOU WANT TO COMBINE DATA FROM DIFFERENT COUNTRIES?

## THE CHALLENGES:

### MANY OPERATING SYSTEMS

- Windows (different versions, 32-bit vs 64-bit)
- Unix (different families of unix)

### MANY COUNTRIES IN MANY DIFFERENT LANGUAGES

- Western Europe
- UK, USA, Australia
- Eastern Europe
- Japan, China, Korea
- India, Africa, the Middle East
- Anywhere in the World!

## **CHOICES:**

### **YOU MUST CHOOSE AN OPERATING SYSTEM**

- You have to run the analysis on all the data somewhere...

### **YOU MUST CHOOSE AN ENCODING**

- A SAS data set is stored with a single encoding.
- It must support all characters found in any language in any text column in the data.

## CHOICES:

Choosing an operating system and converting to it is easy.  
So easy that we aren't going to talk about it today!

- See the paper in the Proceedings
- PROC CONTENTS displays the Operating System Data Representation
  - WINDOWS\_64
  - HP\_UX\_64, RS\_6000\_AIX\_64, SOLARIS\_64, HP\_IA64
  - SOLARIS\_X86\_64, LINUX\_X86\_64, ALPHA\_TRU64, LINUX\_IA64

**BUT WHAT IS AN **ENCODING** AND HOW DO I CHOOSE ONE?**

## WHAT IS ENCODING?

- **Character data** – stored as a series of bytes, 8 bits per byte
- **Bytes contain binary data** – a number between 0 and 255
- **Character Set** – the list of characters which must be represented
- **Coded Character Set** – the mapping of these characters to numbers
- **Encoding**
  - computer instructions of how to read/write the bits and map them to the number defined by the associated coded character set.  
(e.g. The wlatin1/UTF-8 encoding was used to create the data set.)
  - the organization of the bits within the bytes of the character string  
(e.g. My data set has an encoding of UTF-8/UTF-16/UTF-32.)

## EXAMPLES OF ENCODINGS

- **Wlatin1, Wlatin9**
  - English/Western Europe
- **Wlatin2**
  - Central Europe (Bosnian, Croatian, Czech, Hungarian, Polish, Romanian, Slovak)
- **Shift-JIS**
  - Japanese
- **Big5, EUC-CN**
  - Chinese
- **Cyrillic**
  - Russian, Bulgarian and others

# THE WLATIN1 ENCODING

- **History: 1960's ASCII character set**
  - 0-127: upper/lower case English characters, digits 0-9, some special and control characters (e.g. \$, \*)
- **128-255 “Extended ASCII”**
  - Characters common to most Western European languages (e.g. ñ, ø, ü, á)
  - Additional common special characters (e.g. ©, ¢, ¡)
- **All 256 characters are stored in a single byte**
- **Single Byte Character Set (SBCS)**

# THE SHIFT-JIS ENCODING

- **Japanese Industrial Standards**
  - **ASCII characters – 1 byte**
    - same as standard ASCII, except:  $\backslash \Rightarrow \text{¥}$ ,  $\sim \Rightarrow \text{¯}$  (overline)
  - **Katakana characters – 1 byte**
  - **Other Japanese characters – 2 bytes**
- **Double Byte Character Set (DBCS)**

# THE UTF-8 ENCODING

- **Universal character set Transformation Format-8-bit.**
- **Unicode character set – attempts to represent all characters in all languages**
  - More than 120,000 characters covering 129 scripts and multiple symbol sets!
  - 1<sup>st</sup> 128 chars the same as ASCII
  - Other character may require up to 4 bytes:
    - ñ, ø, ü, á, ©, ¿, ¡ and most “extended” ASCII characters require 2 bytes
    - Most Japanese, Chinese and Korean characters require 2-4 bytes
- **The most commonly used encoding for Unicode**
- **Multi-Byte Character Set (MBCS)**

# **SAS'S CHOICES FOR HOSTED SOLUTIONS:**

## **SAS Solutions OnDemand Hosted Solutions**

- **SAS® DRUG DEVELOPMENT**
  - **Operating System: Linux**
  - **Encoding: UTF-8 or Shift-JIS**
- **SAS® CLINICAL TRIAL DATA TRANSPARENCY**
  - **Operating System: Linux**
  - **Encoding: UTF-8 only**

## YOUR CHOICES FOR NON-HOSTED SOLUTIONS:

- **Operating System:**
  - **Windows**
  - **Any Unix platform**
- **Encoding:**
  - **Your choice, but must include all characters in any language represented in your data**

NOTE: Use of the IBM Mainframe Operating Systems and EBCDIC encodings is outside the scope of this paper.

## CONVERTING BETWEEN ENCODINGS - TRANSCODING

- **Easiest - CROSS-ENVIRONMENT DATA ACCESS (CEDA)**
  - Advantages – automatic
    - Use a ordinary LIBNAME statement.
    - Get a message in the SAS log:

NOTE: Data file WINLAT.TEST\_ENCODING\_SHORT\_1.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.

# CONVERTING BETWEEN ENCODINGS - TRANSCODING

## CROSS-ENVIRONMENT DATA ACCESS (CEDA)

- Disadvantages
  - **Minor, possibly**

|                                   |                                       |
|-----------------------------------|---------------------------------------|
| Increased CPU requirements        | No Indexes or Constraints             |
| No data set modification in place | No DATA Step Views or Stored Programs |
| No data set audit trails          | No data sets older than V7            |

- **Major**
  - Transcoded on every read
  - No support for reading SAS catalogs (Formats, Macros, Source entries, etc)
  - **May experience truncation of character variables!**

## EASIEST TRANSCODING - CROSS-ENVIRONMENT DATA ACCESS (CEDA)

```
40 data linutf8.test_encoding_short2;  
41 set winlat.test_encoding_short_1;
```

NOTE: Data file WINLAT.TEST\_ENCODING\_SHORT\_1.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.

```
42 run;
```

ERROR: Some character data was lost during transcoding in the dataset  
INLAT.TEST\_ENCODING\_SHORT\_1. Either the data contains characters that are not representable  
in the new encoding or truncation occurred during transcoding.

NOTE: The DATA step has been abnormally terminated.

NOTE: The SAS System stopped processing this step because of errors.

WARNING: The data set WINLAT.TEST\_ENCODING\_SHORT2 may be incomplete. When this step was  
stopped there were 0 observations and 1 variables.

WARNING: Data set WINLAT.TEST\_ENCODING\_SHORT2 was not replaced because this step was stopped.

## WHY DID OUR EXAMPLE FAIL?

- Data set test\_encoding\_short\_1 (WINDOWS\_64, WLATIN1) copied to Linux
  - Variable WORD \$4
  - Some values in the WLATIN1 data set:
    - doña, brød, über, será
- UTF-8 encoding requires 2 bytes to represent ñ, ø, ü, á
- If we only have 4 bytes to work with, we can't represent the “extended ASCII” characters
- We need more space when we READ the characters!

# HOW CAN WE GET MORE SPACE ON A READ OPERATION???

- Character Variable Padding - CVP
- Read-only LIBNAME engine
- Adds either a default or a specified amount of space during the READ operation
  - Default =  $1.5 \times \text{current length}$
- In our example, 4 bytes => 6 bytes, enough room for our current situation.
- Documentation: **SAS® 9.4 National Language Support (NLS): Reference Guide**

```
143 libname winlat cvp "&_sasusrws_/PhUSE_2015/win_lat/";
```

NOTE: Libref WINLAT was successfully assigned as follows:

**Engine: CVP**

Physical Name: /sddshared/SASWorkspaces/myuserid/Users/myuserid/PhUSE\_2015/win\_lat

```
144 libname linutf8 "&_sasusrws_/PhUSE_2015/lin_utf8";
```

NOTE: Libref LINUTF8 was successfully assigned as follows:

Engine: V9

Physical Name: /sddshared/SASWorkspaces/myuserid/Users/myuserid/PhUSE\_2015/lin\_utf8

```
145 data linutf8.test_encoding_enough1;
```

```
146 set winlat.test_encoding_short_1;
```

```
147 run;
```

NOTE: Data file WINLAT.TEST\_ENCODING\_SHORT\_1.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.

NOTE: There were 4 observations read from the data set WINLAT.TEST\_ENCODING\_SHORT\_1.

NOTE: The data set LINUTF8.TEST\_ENCODING\_ENOUGH1 has 4 observations and 1 variables.

## USING THE CVP ENGINE

- Need more space than that?
- **LIBNAME statement options for CVP:**
  - **CVPBYTES=** a fixed number of bytes between 0 and 32,766 is added to each character variable
  - **CVPMULTIPLIER=** each character variable's length is multiplied by a value between 1 and 5.
    - Default=1.5

**VERY IMPORTANT - monitor the log for truncation messages.**  
**If you have them, you need one of these options!**

## USING THE CVP ENGINE

Example DATA step transcoded 1 data set permanently

But what if we have **hundreds** of data sets to transcode???

- **PROC COPY**
  - Default action: create an exact copy of the input data set, preserving data representation, encoding, and other attributes – **a clone**
- **PROC COPY NOCLONE**
  - Adopt the current data representation, encoding, and other attributes.

## WORKING WITH MBCS DATA

- ‘Normal’ SAS character functions - won’t necessarily give the desired results
- K-functions – KSUBSTR, KSCAN, KINDEX, etc.
- Numeric arguments - **number of characters**, not **number of bytes**
- I18N Level indications:
  - **I18N LEVEL 0** – SBCS data only.
  - **I18N LEVEL 1** – might not work correctly with MCBS data under certain circumstances
  - **I18N LEVEL 2** – SBCS and MBCS data.

### Documentation:

**SAS® 9.4 National Language Support (NLS): Reference Guide**

## HANDLING FORMAT CATALOGS

- **CEDA Major Disadvantage**
  - No support for reading SAS catalogs (Formats, Macros, Source entries, etc.)
- **PROC CPORT => copy transport file => PROC CIMPORT**
- **SAS 9.4 M3 (Jul2015) PROC CIMPORT enhancement**
  - Can transcode transport file data
- **Earlier releases**
  - Only convert between operating system data representations
- **Encoding changes must be handled differently.**

## HANDLING FORMAT CATALOGS

- **Source System: PROC FORMAT CNTLOUT=**
  - Places format definitions into SAS data sets
- Copy SAS data sets to target system
- Transcode SAS data sets
- **Target System: PROC FORMAT CNTLIN=**
  - Creates new format definitions from transcoded data sets

## CONCLUSIONS

- Recommended encoding to support all characters for all countries: **UTF-8**
- Prevent truncation during transcoding by using the **CVP** engine
- Use I18N Level 2 functions (**K-functions**) to work with MBCS character data
- **Formats/Informats** require special handling
- See the full paper in the Proceedings for details, code examples, and recommended reading.
- **Bookmark SAS® 9.4 National Language Support (NLS): Reference Guide!!!**

Paper DH03

# **DATA ENCODING: ALL CHARACTERS FOR ALL COUNTRIES**

**DONNA DUTTON**  
**DONNA.DUTTON@SAS.COM**

**SAS INSTITUTE INC.**  
**100 SAS CAMPUS DRIVE**  
**CARY, NC 27513-2414, USA**

**CHEROKEE, TX**  
**919-531-1159**

