CHAPTER

20

Double Byte Character Sets

Definition

Double Byte Character Sets
are foreign-language character sets that require more than one byte of information to express each character.

Note: Depending on the performance needs of your site, your system administrator has the option of installing Double-Byte Character Set extensions. See your system administrator for more information about installing these features. △

Background

Encoding is the process of converting text data into a numbering system that computers recognize. The most widely used English language encoding systems for computers are called ASCII and EBCDIC. ASCII and EBCDIC encoding systems convert characters of the Latin alphabet into computer representation. Other encoding systems convert Asian pictographic characters into computer representation.

Because of the relatively small number of characters that are required to produce the characters of the Roman alphabet, one byte of information is adequate to represent each character (Single Byte Character Sets). Many Asian languages require thousands of characters, and two bytes of information are needed to represent each character. This is the origin of the term "Double Byte Character Set" (DBCS).

Each Asian language usually has more than one DBCS encoding system, due to nonstandardization between computer manufacturers. The SAS System has features designed to process the DBCS encoding information that is unique to each manufacturer for the major Asian languages, which include Japanese, Korean, simplified Chinese (used in mainland China and Singapore), and traditional or complex Chinese (used in Taiwan and Hong Kong).

What You Can Do with a DBCS Language and SAS Software

With the proper hardware and software extensions, you can use SAS to

- □ display any of the major Asian languages in the English version of the SAS System
- □ import data from Asian language computers and move the data from one application or platform to another (which may require SAS ACCESS or other SAS products)
- □ convert standard Asian date and time notation to SAS date values, SAS time values, and SAS datetime values
- □ create data input screens that contain or accept a mixture of European and Asian languages
- □ create various types of output, such as reports and graphs, and data sets with Asian language characters in them.

Limitations

- □ SAS cannot translate one Asian language to another.
- □ SAS cannot display words in traditional Asian right-to-left format. Presentation must be left to right.
- □ Variable names cannot consist of Asian characters. They must conform to SAS naming conventions. (However, variable values, variable labels, and data set labels can use Asian characters.)

Using a DBCS Language with SAS

What You'll Need to Use DBCS Languages with the SAS System

You must have the following in order to display data sets containing DBCS characters on your computer system:

- □ system support for local functions or for multiple code pages
- □ DBCS fonts corresponding to the languages you intend to use.

If you need to generate your own Asian language characters for use with SAS software, you'll need a computer that supports DBCS. These computers are available on a limited basis in the US and Europe. These Asian language computer systems use various methods of creating the characters. In one popular method, the user types the phonic pronunciation of the character, often using English letters. The computer then pops up a menu of like-sounding characters and prompts the user to select one of them.

If you intend to use Asian language characters in your SAS session, you'll need to become familiar with turning on the DBCS capability and specifying which language and computer platform to use. See the *SAS Language Reference: Dictionary* for more information on these system options:

System Option	Description
DBCS	Turns on the capability to handle DBCS characters
DBCSLANG	Specifies the language to use
DBCSTYPE	Specifies the manufacturer's encoding system to use

Most of the time, you'll use all three of these system options together. Here is an example of how they are used in a SAS configuration file:

```
options dbcs             /*turn on DBCS*/

        dbcstype=IBM       /*specify the IBM  mainframe
                             environment(encoding scheme)*/
        dbcslang=JAPANESE; /*specify the Japanese language
                             (coded character set)*/
```

When You Can Use DBCS Features

Once you've set up your SAS session to recognize a specific DBCS language and computer platform, you'll be able to work with your specified language in these general areas:

- □ the DATA step and batch-oriented procedures
- □ windowing and interactive capabilities
- □ cross-system connectivity and compatibility
- □ access to databases
- □ graphics.

In a DATA step and in batch-oriented procedures, you can use DBCS wherever a text string within quotation marks is allowed. Variable values, variable labels, and data set labels can all be in DBCS. DBCS can also be used as input data and with range and label specifications in the FORMAT procedure.

Note: Variable names cannot be in DBCS. They must conform to SAS naming conventions. △

In WHERE expression processing, you can search for embedded DBCS text.

Using a DBCS with SAS on a Mainframe

Another type of DBCS encoding scheme exists on mainframe systems, which combine DBCS support with the 3270-style data stream. Each DBCS character string is surrounded by escape codes called *shift out/shift in*, or SO/SI. These codes originated from the need for the old-style printers to shift out from the EBCDIC character set, to the DBCS character set. The major manufacturers have different encoding schemes for SO/SI; some manufacturers pad DBCS code with one byte of shift code information while others pad the DBCS code with two bytes of shift code information. These differences can cause problems in reading DBCS information on mainframes.

PCs, minicomputers, and workstations do not have SO/SI but have their own types of DBCS encoding schemes that differ from manufacturer to manufacturer. SAS has several formats and informats that can handle DBCS on SO/SI systems:

Keyword	Language Element	Description
$KANJI	informat	Removes SO/SI from Japanese Kanji DBCS
$KANJIX	informat	Adds SO/SI to Japanese Kanji DBCS
$KANJI	format	Adds SO/SI to Japanese Kanji DBCS
$KANJIX	format	Removes SO/SI from Japanese Kanji DBCS

Converting Data from One DBCS Encoding Scheme to Another

Normally, DBCS data that is generated on one computer system is incompatible with data generated on another computer system, due to differences between manufacturers, mentioned earlier. SAS has features that allow conversion from one DBCS source to another, as shown in the following table.

Language Element	Type	Use
KVCT	function	Converts DBCS data from one operating environment to another
CPORT	procedure	Moves files from one environment to another
CIMPORT	procedure	Imports a transport file created by CPORT

Avoiding Problems with Split DBCS Character Strings

- When working with DBCS characters, review your data to make sure that SAS recognizes the entire character string when data is imported or converted or used in a DATA or a PROC step.
- On mainframe systems that employ shift out/shift in escape codes, DBCS character strings may become truncated during conversion across operating environments.
- There is a possibility that DBCS character strings can be split when working with the PRINT, REPORT, TABULATE, and FREQ procedures. If undesirable splitting occurs, you may have to add spaces on either side of your DBCS character string to force the split to occur in a better place. The SPLIT= option may also be used with PROC REPORT and PROC PRINT to force string splitting in a better location.

DATA Step Functions Designed to Handle DBCS

Several DATA step functions, all beginning with the letter K, have been developed for working with DBCS:

KCOMPRESS
removes specific characters from a character string.

KCOUNT
returns the number of double-byte characters in a string.

KINDEX
searches a character expression for a string of characters.

KLEFT
left aligns a SAS character expression by removing unnecessary heading DBCS blanks and SO/SI.

KLENGTH
returns the length of an argument.

KLOWCASE
converts all letters in an argument to lowercase.

KREVERSE
reverses a character expression.

KRIGHT
right aligns a character expression by trimming trailing DBCS blanks and SO/SI.

KSCAN
selects a given word from an expression.

KSTRCAT
concatenates two or more character strings.

KSUBSTR
extracts a substring from an argument.

KSUBSTRB
extracts a substring from an argument based on byte position.

KTRANSLATE
replaces specific characters in a character expression.

KTRIM
removes trailing DBCS blanks and SO/SI from character expressions.

KTRUNCATE
truncates a numeric value to a specified length.

KUPCASE
converts all single-byte letters in an argument to uppercase.

KUPDATE
inserts, deletes and replaces character value constants.

KVERIFY
returns the position of the first character that is unique to an expression.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS Language Reference: Concepts*, Cary, NC: SAS Institute Inc., 1999. 554 pages.

SAS Language Reference: Concepts

ISBN 1–58025–441–1

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, November 1999

The Institute is a private company devoted to the support and further development of its software and related services.