Overview

The DBCSTAB procedure produces conversion tables for the double-byte character sets that SAS supports.

Procedure Syntax

PROC DBCSTAB TABLE=table-name
   <BASETYPE=base-type> <CATALOG=libref.>catalog-name>
   <DATA=libref.>table-name> <DBCSLANG=language>
   <DESC='description'> <FORCE> <VERIFY>;

PROC DBCSTAB Statement

PROC DBCSTAB TABLE=table-name
   <option(s)>;

Required Arguments

TABLE=table-name
   specifies the name of the double-byte code table to produce. This table name becomes an entry of type DBCSTAB in the catalog that is specified with the CATALOG= option. By default, the catalog name is SASUSER.DBCS.

Alias: NAME=, N=
Options

**BASETYPE=base-type**  
Specifies a base type for the double-byte code table conversion. If you use this option, you reduce the number of tables that are produced.  
If you specify BASETYPE=, all double-byte codes are first converted to the base code, and then converted to the required code. If you have \( n \) codes, there are \( n(n-1) \) conversions that must be made.  
**Alias:** BTYPE=

**CATALOG=<libref.>catalog-name**  
Specifies the name of the catalog in which the table is to be stored. If the catalog does not exist, it is created.  
**Default:** SASUSER.DBCS

**DATA=<libref.>table-name**  
Specifies the data for producing the double-byte code table. Several double-byte character variables are required to produce the table. Use variable names that are equivalent to the value of the DBCSTYPE system option and are recognized by the KCVT function.

**DBCSLANG=language**  
Specifies the language that the double-byte code table uses. The value of this option should match the value of the DBCSLANG system option.  
**Alias:** DBLANG

**DESC='description'**  
Species a text string to put in the DESCRIPTION field for the entry.

**FORCE**  
Produces the conversion tables even if errors are present.

**VERIFY**  
Checks the data range of the input table per code. This option is used to check for invalid double-byte code.

---

Examples

**Example 1: Producing Japanese Conversion Tables**

**Procedure features:**

PROC DBCSTAB statement options:

```
  TABLE=
  DATA=
  DBLANG=
  BASETYPE=
  VERIFY
```

**Program**

```
data ja_jpn;
  length ibm jis euc pcibm $2.;
```
ibm='4040'x;
jis='2121'x;
euc='a1a1'x;
pcibm='8140'x;
run;

proc dbcstab
  table=japanese
  data=ja_jpn
  dblang=japanese
  basetype=jis
  verify;
run;

NOTE: Base table for JIS created.
NOTE: IBM table for JIS created.
NOTE: PCIBM table for JIS created.
NOTE: EUC table for JIS created.
NOTE: Base table for IBM created.
NOTE: JIS table for IBM created.
NOTE: Base table for PCIBM created.
NOTE: JIS table for PCIBM created.
NOTE: Base table for EUC created.
NOTE: JIS table for EUC created.
NOTE: 10 DBCS tables are generated. Each table has 1 DBCS characters.
NOTE: Each table is 2 bytes in size.
NOTE: Required table memory size is 612.
NOTE: There were 1 observations read from the dataset WORK.JA_JPN.