

## **Data conversion issues in v6-v8 which is of special interest for customers using languages other than English.**

### **Understanding data conversion**

Different hosts or platforms on which the SAS System runs use different standards for encoding characters. As a result, character conversion (or mapping) is required when moving data across platforms. The SAS System provides a number of ways of transporting data and applications across hosts. However, the processes and translation tables that are involved differ depending on the mechanisms involved. These are explained in the following sections.

### **Data conversion within the SAS System**

To ensure portability of data and applications the SAS system provides basically two data conversion mechanisms:

- ❑ Transport-format translation tables use an intermediate transport format when transporting files from one host to another.
- ❑ Host-to-host translation tables translate characters directly from the source platform's encoding standard to the target platform's encoding standard.

These tables are stored in SAS catalog entries with an entry type of TRANTAB. Both of these usually need to be customized to accommodate character sets other than US English. SAS will search for the translation tables in the following order:

- ❑ SASUSER.PROFILE
- ❑ SASHELP.HOST *in Version 7 and later versions additionally in*
- ❑ SASHELP.LOCALE.

### **Host-to-Host Translation Tables**

Each entry contains two translation tables. The first table is for importing translation, while the second table is for exporting translation. For example, the EBCDIC/ASCII-OEM translation entry (\_0000A0) on OS/390 contains an import table for the ASCII-OEM to EBCDIC translation and it contains an export table for the EBCDIC to ASCII-OEM translation.

The SAS System provides the following TRANTAB entries for direct, host-to-host character translation:

- |                        |          |
|------------------------|----------|
| ❑ EBCDIC/ASCII-ISO     | _0000030 |
| ❑ EBCDIC/ASCII-ANSI    | _0000060 |
| ❑ EBCDIC/ASCII-OEM     | _00000A0 |
| ❑ ASCII-ISO/ASCII-ANSI | _0000050 |
| ❑ ASCII-ISO/ASCII-OEM  | _0000090 |
| ❑ ASCII-ANSI/ASCII-OEM | _00000C0 |
| ❑ MAC/ASCII-ISO        | _0000110 |
| ❑ MAC/EBCDIC           | _0000120 |
| ❑ MAC/ASCII-ANSI       | _0000140 |
| ❑ MAC/ASCII-OEM        | _0000180 |

*In Version 6* host-to-host translation tables are *only used with the REMOTE engine* to provide access to remote data. *In Version 7 and later versions* the UPLOAD and DOWNLOAD procedures (part of SAS/CONNECT software) also use this method if the following conditions are met:

- ❑ both the client and server sessions are Version 7 or later

- ❑ you are transferring a SAS data set.

No translation occurs if both the remote and local sides have the same machine representations. This prevents any unnecessary translation from occurring. If translation is needed, the receiving side (the remote session in an upload, the client session in a download) translates the data directly into its native representation.

### Transport-format translation tables

The tables specified in the TRANTAB= system option control character translation for your SAS session or job and all file transfers. Translation occurs twice for every transmission. The data is translated from local to transport format, and then the receiving side translates from transport format to local format using the following TRANTAB entries:

- ❑ SASXPT - controls local-to-transport format translation
- ❑ SASLCL - controls transport-to-local format translation

*In Version 6* transport-format translation tables are used with the procedures UPLOAD, DOWNLOAD, CPORT, and CIMPORT. As stated above, *in Version 7 and later versions* the procedures UPLOAD and DOWNLOAD directly translate the data into its native representation when doing a *data set transfer* and when both the client and server sessions are *Version 7 or later*.

In all other cases, UPLOAD/DOWNLOAD uses the local-to-transport, transport-to-local routines for translation. This would include:

- ❑ if either or both the client/server sessions are Version 6 (or earlier)
- ❑ if you are transferring external files or catalogs
- ❑ for moving the rsubmitted code to the remote session and returning log/list lines to the client
- ❑ for subsetting data with WHERE clauses and SQL views.

### TRANTAB= system option and TRANTAB statement

The TRANTAB= system option specifies a translation table for your SAS session or job and includes all file transfers. The TRANTAB system option can be set per session or permanently by adding the statements to your CONFIG file. *In Version 7 and later versions* the options are written to the SAS registry after having run Locale Setup (see: Building Customized Translation Tables).

The TRANTAB statement specifies the name of the translation table to apply to the character data in the SAS file you export. It is an enhancement of the TRANSLATE= option that translates specified characters from one EBCDIC (or ASCII) value to another. It should only be used in specific cases (see below).

#### *Trantab statement with PROC CPORT*

This example shows how to apply a customized translation table to the transport file before PROC CPORT exports it.

Let's assume that you are downloading data with Swedish characters from OS/390 (EBCDIC) to Windows (ASCII-ANSI). For data set labels, variable labels, and variable values, you want EBCDIC '5B'x (\$/Å) to map to ASCII-ANSI 'C5'x (Å). For format names in data set headers, you want EBCDIC '5B'x (\$/Å) to map to an ASCII-ANSI '24'x (\$).

The following statements should do what is desired:

```
proc trantab table=ebcdic; /* standard 1-to-1
                           EBCDIC table */
list;
rep '5B'x '8B'x; /* make 5Bx go to 8Bx
                 in Phase 1 */
list;
save table=swecprt;
quit;
```

```
proc cport d=your.dataset file=your_export_file;
  trantab name=swecprt type=dataset;
run;
```

In "Phase 1," PROC CPORT uses the SWECPRT trantab, causing the '5B'x to be translated to a '8B'x. The '8B'x code point was determined by looking at SASXPT on OS/390 for the character cell that contains an 'C5'. That is, '8B'x is the value that is translated to 'C5'x in "Phase 2," when the default export table (SASXPT) is applied.

The full program for all of the Swedish national and special characters looks like this:

```
proc trantab table=ebcdic; /*      Phase 1      Phase 2      */

  rep '4F'x '5A'x;          /* (!) '4F'x -> '5A'x -> '21'x */
  rep '5A'x '45'x;          /* (å) '5A'x -> '45'x -> 'A4'x */
  rep '5B'x '8B'x;          /* (Å) '5B'x -> '8B'x -> 'C5'x */
  rep 'D0'x '6A'x;          /* (ä) 'D0'x -> '6A'x -> 'E5'x */
  rep '7B'x '8A'x;          /* (Ä) '7B'x -> '8A'x -> 'C4'x */
  rep 'C0'x 'BC'x;          /* (ä) 'C0'x -> 'BC'x -> 'E4'x */
  rep '7C'x 'AE'x;          /* (Ö) '7C'x -> 'AE'x -> 'D6'x */
  rep '6A'x 'EC'x;          /* (ö) '6A'x -> 'EC'x -> 'F6'x */
  rep 'E0'x '8F'x;          /* (É) 'E0'x -> '8F'x -> 'C9'x */
  rep '79'x 'CB'x;          /* (é) '79'x -> 'CB'x -> 'E9'x */
  rep 'A1'x 'FC'x;          /* (ü) 'A1'x -> 'FC'x -> 'FC'x */

  save table=swecprt; list; quit;
```

#### *Trantab statement with PROC UPLOAD/ DOWNLOAD*

In *Version 7 and later versions* you can achieve the same effect with PROC UP-/DOWNLOAD.

```
proc download d=your.dataset out=your_export_file;
  trantab=swecprt; run;
```

### **Building Customized Translation Tables**

In *Version 6* the TRABASE program, which builds transport-format and character-operations tables for a number of languages and operating systems, is part of the SAS sample library. It does not create tables for all possible combinations, but it can easily be adapted for specific needs. Starting with *Version 6.11* the NLSSETUP Application (also shipped as part of the SAS sample library) helps a user prepare SAS to be used with a language other than English. With its point-and-click interface it allows to create both transport-format and host-to-host translation tables for a great number of languages. (Please refer to the online help of the NLSSETUP Application for further detail).

In *Version 7* or later the Locale Setup Window (LSW) installs tables appropriate for the text representation of a given locale. To access the Locale Setup window, select **Solutions Accessories Locale Setup** from the Explorer window menu. LSW also stores information into the USER part of the SAS Registry and with the proper access permissions into the SYSTEM part of the SAS Registry. (Please refer to the online help of the Locale Setup Window for further detail).

### **Recommendations for Users**

#### *Transferring data sets*

With PROC CPORT and CIMPORT

Make sure that you have set the appropriate TRANTAB= system option before exporting a data set with PROC CPORT.

Example: Suppose you have data that contain Spanish characters and you want to use PROC CPORT to export that data from OS/390 (EBCDIC) and later import on UNIX. Let's further assume that you have customized transport-format trantabs by using one of the methods described above. To specify that SAS should use these trantabs instead of the default, (SASXPT and SASLCL) transport-format trantabs, you specify the following OPTIONS statement on OS/390:

```
options trantab=(spaeti);
```

Character translation depends on which platforms are involved. If you wanted to translate characters between the two 8-bit ASCII extensions of OS/2 and Windows, you would need to use another set of transport-format trantabs. TRABASE and NLSSETUP use specific naming conventions for these:

```
EBCDIC          <-> OEM (PC-ASCII) : <language>eta, <language>ate
EBCDIC          <-> MAC           : <language>etm, <language>mte
EBCDIC          <-> ISO           : <language>eti, <language>ite
EBCDIC          <-> WINDOWS       : <language>etw, <language>wte
ISO             <-> OEM (PC-ASCII) : <language>ita, <language>ati
ISO             <-> MAC (Apple)    : <language>itm, <language>mti
ISO             <-> WINDOWS       : <language>itw, <language>wti
OEM (PC-ASCII) <-> MAC           : <language>atm, <language>mta
OEM (PC-ASCII) <-> WINDOWS       : <language>atw, <language>wta
MAC             <-> WINDOWS       : <language>mtw, <language>wtm
```

where the **eta/ate**, **ita/ati**, **eti/ite**, **itm/mti**, **atm/mta**, **etm/mte**, and **wti/itw** suffixes are the identifiers for the machine-to-machine translation. For example, **eta** specifies EBCDIC to ASCII-OEM, and **ate** is the reverse translation (ASCII-OEM to EBCDIC).

EBCDIC encoding is used by OS/390, CMS, and VSE hosts. OEM (PC ASCII) encoding is used by OS/2 and Windows hosts in OEM mode. ISO encoding is used by UNIX and VMS hosts. Windows hosts use WINDOWS encoding. Apple Macintosh hosts use MAC encoding.

<language> represents one of the following language codes:

```
ara Arabic
bel Byelorussian
csy Czech
dan Danish/Norwegian
eng English - Britain/Ireland
fre French
ger German - Austria/Germany
grk Greek
heb Hebrew
hun Hungarian
ita Italian
pol Polish
ptb Portuguese - Brazil
ptg Portuguese - Portugal
rus Russian/Bulgarian
sky Slovakian
slv Slovenian
spa Spanish - Latin America/Spain
swe Swedish/Finnish
swi French - Switzerland/Belgium; German - Switzerland; Italian -
    Switzerland
tur Turkish
ukr Ukrainian
```

You can rename the TRANTAB entries according to your needs. Such renaming is useful in cases where two or more countries' languages share the same table. For example, because Danish and Norwegian users both make use of the DANxxx tables (where xxx represents the machine-to-machine transcoding identifier), Norwegian users may want to rename the tables to NORxxx. The same process is true for any other set of languages that share tables.

In the Spanish example above SPAETI does the necessary host-to-transport format translation. When you use PROC CIMPORT on UNIX to import the transport file, you do not need to specify a trantab, because SPAETI has already translated all EBCDIC characters to their correct ASCII-ISO counterparts. *In Version 7 or later versions* the Locale Setup Window the Locale Setup window automatically sets up to export to and import from the Windows operating environment, in other words the Windows (ASCII-ANSI) encoding is used as common denominator for data transfer between all platforms. The Locale Setup Window also stores the associated TRANTAB option for a given locale in the SAS registry, e.g.

```
options trantab=(E273WLT1,WLT1E273);
```

for German on OS/390.

**Note:** Naming conventions for the Local-to-Transport and Transport-to-Local entries are usually specified with a four-letter internal SAS code that represents a particular encoding, e.g. e273, wlt1, lat1, p850, and arom for German. (Please refer to the online help of the Locale Setup Window for further detail).

For example, Local-to-Transport and Transport-to-Local entries for German between EBCDIC and ASCII-ANSI encodings are as follows:

- E273WLT1 EBCDIC to ASCII-ANSI translation table
- WLT1E273 ASCII-ANSI to EBCDIC translation table.

You do not need to specify the TRANTAB= system option for PROC CPORT and CIMPORT, it will be used implicitly. **To ensure seamless data transfer, apply your customizations on all platforms in question.**

With PROC UPLOAD and DOWNLOAD

In Version 6 style translation, translation occurs twice for every transmission. The data is translated from local to transport format, and then the receiving host translates from transport format to local format. Two translations occur for all data that is transferred. This means as with PROC CPORT and CIMPORT transport-format translation tables have to be customized (see above) and the appropriate TRANTAB= system option has to be set. This has to be done on the remote platform.

When both the local host and the remote host are *Version 7 or later*, the translation rules have changed. First, no translation occurs if both the remote and local hosts have the same machine representations. This prevents any unnecessary translation from occurring. If translation is needed, the receiving host translates the data directly into its native representation. In this case, the host-to-host trantabs are used for the direct conversion. So, when both the local and remote hosts are running Version 7 or Version 8, the data is translated only one time when translation is necessary and is not translated when both hosts have the same machine architecture.

With PROC UPLOAD/DOWNLOAD and PROC CPORT/CIMPORT

Only use *the TRANTAB statement* if you need to do context-sensitive character mappings as described above. You might need to customize the trantabs yourself or ask SAS Technical Support for assistance.

*Transferring catalogs and external files*

With PROC UPLOAD/DOWNLOAD and PROC CPORT/CIMPORT

Make sure that you have customized the transport-format translation tables and set the appropriate TRANTAB= system option. Use the methods above to create customized tables for the language of

your choice. In version 6 you need to specify the TRANTAB= system option explicitly per session or store it in your CONFIG file for permanent use.

In version 7 or later the Locale Setup Window sets the TRANTAB= system option to be used in the SAS registry, e.g.

```
options trantab=(E273WLT1,WLT1E273);
```

It can then be used implicitly. **To ensure seamless data transfer, apply your customizations on all platforms in question.**

#### *Accessing data sets with the REMOTE engine*

No translation occurs if both the remote and local hosts have the same machine representations. This prevents any unnecessary translation from occurring. If translation is needed, the receiving host translates the data directly into its native representation. In this case, the host-to-host trantabs are used for the direct conversion. You can customize them with the methods described above.

Example: Suppose you have a data set with Danish characters on the remote machine, and you want to print out the data on your client session.

Assign a libname with

```
LIBNAME mylib 'host-path' server=your_serverid;
```

You can now use the PRINT procedure, for instance, to print the data set from the remote host, and keep Danish characters correct:

```
proc print data=mylib.employees;
run;
```

If you need to subset your data, make sure that the TRANTAB= system option is set appropriately, or that LSW has been run accordingly, otherwise the following code may not yield any matches:

```
data test;
set mylib.employees;
where name like "%å%";
run;
```

## **CEDA**

CEDA does not replace the traditional file transport methods (PROC CPORT and PROC CIMPORT), PROC UPLOAD and PROC DOWNLOAD in SAS/CONNECT, or server processing in SAS/SHARE because of these limitations:

- ❑ CEDA features are implemented for Version 8 SAS data sets, PROC SQL views, and MDDBs (for read-only access). CEDA does not support Version 8 stored programs, catalogs, DMDBs, or FDBs, nor does it support any pre-Version 7 member types.
- ❑ CEDA does not support update access.
- ❑ CEDA does not support subsetting by means of an index.
- ❑ CEDA is available for hosts that use directory-based file structures. On OS/390, CEDA is available only for SAS data sets that reside in a UNIX System Services Directory. Bound libraries that are traditionally used on the OS/390 host do not implement CEDA.

If your application can operate within these limitations, then CEDA offers the simplest strategy for file access across a network. If your needs exceed these limitations, then you still have other methods available.

In order to create a file in a non-native (or foreign) format for a supported member type, use the OUTREP= option in the LIBNAME statement or in the DATA step.

Specified in the LIBNAME statement, the OUTREP= option applies the designated format to all files that are created in the specified library. However, specified in the DATA step, the designated host format is limited to the specified data set.

As an example, suppose that you use a Windows host to create data sets that contain academic grades. However, the data entry and processing personnel who read and write these data sets use UNIX hosts. In order to accommodate the needs of UNIX users, you designate a non-native format for the data sets. In this example, a data set CHEM.GRADES with Polish names is created in UNIX (HP\_UX) format:

```
data chem.grades (outrep=hp_ux);
  input student $ test1 test2 final;
cards;
Wałęsa 66 80 70
Żmuda 97 91 98
Gorgoń 75 90 81
Łopuszański 61 59 72
Ćmikiewicz 76 88 90
Świtoń 77 65 87
Zajączkowski 67 75 79
Żukrowski 76 88 90
Woźniak 76 84 89
Bożyk 63 79 67
Wiśniewski 75 84 91
run;
```

If the file data representation and the accessing host data representation are the same (for example, the file is represented in Windows format and the accessing host is a Windows host), then the accessing host can read the file and write to the file.

If the file data representation and the accessing host representation are different, then the accessing host can dynamically translate the data file into the accessing host native format for read-only access.

Again, the host-to-host trantabs are used for the direct conversion. You can customize them with the methods described above. **To ensure seamless data transfer, apply your customizations on all platforms in question.**

In the example above, you can run the following code on the UNIX host to print the data set, and keep Polish characters correct:

```
proc print data=chem.grades;
run;
```

The output will look like this:

```
                The SAS System  15:01 Tuesday, October 10, 2000   1
Obs      student      test1      test2      final
   1     Wałęsa         66         80         70
   2     Żmuda         97         91         98
   3     Gorgoń        75         90         81
   4     Łopuszań      61         59         72
   5     Ćmikiewi      76         88         90
   6     Świtoń        77         65         87
   7     Zajączko       67         75         79
   8     Żukrowsk     76         88         90
   9     Woźniak       76         84         89
  10     Bożyk         63         79         67
  11     Wiśniews     75         84         91
```