

Reconocer el set de caracteres de un export de BBDD sin su log

Posted by admin in Bases de datos, November 6, 2008 @1:10 pm

Estos días he tenido diversos problemas por el set de caracteres de unos exports.

Esta información me ha sido muy útil para solucionarlos:

[quote]Introduction:

One of the most common NLS-related problems reported to Oracle Worldwide Customer Support is the loss or changing of characters after an export and import. This is almost always due to NLS_LANG being set to the incorrect character set during export.

Role of NLS_LANG During Export/Import:

This is explained in detail in Note 15095.1 but, in brief, export and import are client tools and will work under the character set specified by NLS_LANG. If, for instance, the database is created with a character set of WE8DEC and NLS_LANG is set to AMERICAN_AMERICA.WE8PC850 then the ascii values of the stored characters in the database will be translated to the values for the same characters in the WE8PC850 character set. The character set defined by NLS_LANG during the export is stored in the export dump file and is used to ensure that the data is translated correctly to the character set defined by NLS_LANG for the import.

Potential Problems:

If NLS_LANG is not set, for instance, export will be done under US7ASCII, the default character set. If the database was built with character set WE8DEC the characters stored in the database will be converted to US7ASCII and any 8-bit characters, having no equivalent in US7ASCII, will be stripped out.

The same problem will be seen if the character set defined by NLS_LANG is not a superset of the one being translated from (ie: the database character set on export or the export file character set on import).

Identifying the Export Character Set:

When investigating problems like these it is useful to check the character set used for the export. As said above, this is held in the export dump file. It can be seen by doing a hex dump of the export file as follows (in Unix):

```
cat expdat.dmp | od -x | head
```

This will produce output similar to:

```
0000000 0300 0145 5850 4f52 543a 5630 372e 3033
0000020 2e30 330a 4454 534f 0a52 5441 424c 4553
0000040 0a31 3032 340a 300a 0020 2020 2020 2020
0000060 2020 2020 2020 2020 2020 2020 2020 2020
*
0000140 2020 2020 2020 2020 4d6f 6e20 4e6f 7620
0000160 3130 2031 343a 3031 3a33 3620 3139 3937
0000200 0a54 4142 4c45 2022 454d 5022 0a43 5245
0000220 4154 4520 5441 424c 4520 2245 4d50 2220
```

The second and the third byte in the file define the character set used for the export.

In the example above, the second byte is 0x00 and the third byte is 0x01, yielding 0x0001 as the character set ID. This shows that NLS_LANG was set to US7ASCII during the export. The new Oracle8 functions NLS_CHARSET_NAME and NLS_CHARSET_ID can be used to map character set IDs to character set names. The mapping is also given in Note 13971.1.

Note that the 16-bit value is stored in the EXP platform endian.

Most unix platforms are big-endian (Sparc, PowerPc, PARisc, RS/6000, SGI R4000 systems), i.e. the most significant byte is showed first.
(like above example -> if the file begins with 03xx -> big endian)

On little-endian platforms, (platforms running on Intel/AMD x86 and Alpha mainly) the output will be slightly different as below:

```
00000000 0003 4501 5058 524f 3a54 2e37 3330
etc.
```

Here the most significant byte is showed *last* (!)

(if the file begins with xx03 -> little endian)

The values for the most commonly used character sets are below:

Name ID

```
US7ASCII 0x0001
WE8DEC 0x0002
WE8ISO8859P1 0x001f
EE8ISO8859P2 0x0020
```

SE8ISO8859P3 0x0021
NE8ISO8850P4 0x0022
CL8ISO8859P5 0x0023
AR8ISO8859P6 0x0024
EL8ISO8859P7 0x0025
IW8ISO8859P8 0x0026
WE8ISO8859P9 0x0027
WE8ISO8859P15 0x002e
TH8TISASCII 0x0029
US8PC437 0x0004
WE8ROMAN8 0x0005
WE8PC850 0x000a
EE8PC852 0x0096
RU8PC855 0x009B
TR8PC857 0x009C
WE8PC858 0x001c
WE8PC860 0x00A0
IS8PC861 0x00A1
N8PC865 0x00BE
RU8PC866 0x0098
EE8MSWIN1250 0x00aa
CL8MSWIN1251 0x00ab
WE8MSWIN1252 0x00b2
EL8MSWIN1253 0x00ae
TR8MSWIN1254 0x00b1
IW8MSWIN1255 0x00af
AR8MSWIN1256 0x0230
BLT8MSWIN1257 0x00b3
ZHT16MSWIN950 0x0363
ZHS16GBK 0x0354
ZHT16HKSCS 0x0364
JA16EUC 0x033e
JA16SJIS 0x0340
ZHT16BIG5 0x0361
AL24UTFFSS 0x0366
UTF8 0x0367
AL32UTF8 0x0369

```
select nls_charset_id(value) nls_charset_id, value  
from v$nls_valid_values  
where parameter = 'CHARACTERSET'  
order by nls_charset_id(value);
```

Gives the nls_charset_id in DECIMAL, so you need to convert it to HEX first.

Alternative you can open the character set definition using Locale Builder (9i and up), this will also show the character set ID in DECIMAL in the first screen (note that there is also an ISO ID that is NOT used here in the exp file)

Note 223706.1 Using Locale Builder to view the definition of character sets

Warning: User modifications of export dump files are not supported
===== by Oracle. The character set information is also held in other
places in the export dump file and modifying only the two bytes
may lead to problems with imported data.

NOT WORKING any more with Oracle 9i R2 (9.2) and up,
due to changes in the import /export tools
but you can use a 8i exp against a 9i db for example, more
info is in Note 132904.1 Compatibility Matrix for Export chr(38) Import
Between Different Oracle Versions

In some cases it can be useful to modify the character set information
held in the dump file. This should not be taken lightly since the character
set information is also held in other places.

We **STRONGLY** advice you to log a NLS tar **FIRST** to get confirmation that this is
a solution for your problem before starting to change this header.

If, after careful consideration of other options and verification by support,
you do decide to edit the character set simply use a binary file editor to do so.

You could for example use a freeware Hex Editor like

* XVI32 <http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm>

* HxD <http://mh-nexus.de/en/hxd/>

In case of multiple dump files from a single export, you need to modify each
and every export dump file. Otherwise, you will error out with -
IMP-00008: unrecognized statement in the export file:
when opening the second file.

[/quote]

Fuentes:

ligarius.wordpress.com

infor.uva.es