

The correct NLS_LANG in a Windows Environment [ID 179133.1]

Modified: Apr 16, 2012 Type: BULLETIN Status: PUBLISHED Priority: 1

In this Document

[Purpose](#)

[Scope](#)

[Details](#)

[1. Key Concepts/terminology:](#)

[1.1 Windows and Dos Code Pages:](#)

[1.2 Fonts:](#)

[2. How to setup my NLS_LANG:](#)

[2.1 In the Registry:](#)

[2.2 As a System or User Environment Variable, in System properties:](#)

[2.3 As an Environment variable defined in the command prompt:](#)

[3. The correct NLS_LANG for my Windows Code Page:](#)

[3.1 Determine your Windows ACP code page \(used by sqlplusW.exe\):](#)

[3.2 Find the correspondent Oracle client character set:](#)

[3.3 Set it in your Registry:](#)

[4. The correct NLS_LANG for my DOS / Command Prompt OEM Code Page \(used by sqlplus.exe\) :](#)

[5. How to check the NLS_LANG:](#)

[6. List of common NLS_LANG's used in the Windows Registry:](#)

[7. List of common NLS_LANG's used in the Command Prompt \(DOS box\):](#)

[8. How Windows uses Fonts to display the different charactersets:](#)

[9. Updating/changing the NLS_LANG in the registry to the correct value:](#)

[References](#)

Applies to:

Oracle Server - Enterprise Edition
Microsoft Windows (32-bit)
Microsoft Windows Itanium (64-bit)
Microsoft Windows x64 (64-bit)

Purpose

To give an overview on how to correctly define the NLS_LANG variable on Microsoft Windows platforms.

Scope

DBA's configuring Oracle clients on any desktop/server windows platform (=excluding windows Mobile/CE).

Details

To debug display problems (you see "funny" symbols or characters become "?" or "ÃÃÃÃ¿") then please also select the existing (application) data in SQL Developer *and* test with new data in a test table inserted trough Sqldeveloper from a (Windows) client, this is a "know good client" that needs no NLS configuration. Download the latest version from <http://www.oracle.com/technetwork/developer-tools/sql-developer> . Sqldeveloper does not need a client installation, for testing purposes we even recommend to install this on a client that has no other Oracle software installed.

If *new data* inserted trough Sqldeveloper in a test table is displayed correctly in SQL Developer then you are sure the NLS_CHARACTERSET supports these characters, hence *the database side is fine and the problem is at the client side*. This new data, if shown correctly in Sqldeveloper, can then be used as a "reference" to check your client's configuration.

If existing application data is NOT correct in SQL Developer then the *inserting* client config is the first thing to debug before trying to configure anything else.

If *existing application* data is displayed correctly in SQL Developer then you are sure this data is correctly stored in the database and the

problem is pure the *selecting* client side.

If you see "squares" instead of the symbols you might need to change the font in SQLDeveloper - "Arial Unicode MS" is normally installed on every windows client and supports a wide range of characters

In sqldveloper you change the font in "Tools" , "Preferences" then expand the "Code Editor" tree and choose "Fonts". In the drop down box choose the font you want.

Note that missing characters from a font (=you see squares) has as such no relation with "missing characters" (then you see ? or ÃÄÃÄÃÄ¿ , not squares), the characters is there in the tool, the system simply does not know how to display it. But you can copy paste it for example to another tool.

1. Key Concepts/terminology:

This note covers the <clients character set> part of NLS_LANG and provide windows specific information in addition of [Document 158577.1](#) NLS_LANG Explained (How does Client-Server Character Conversion Work?).

NLS_LANG consist of: NLS_LANG=<NLS_Language>_<NLS_Territory>.<clients character set>

where:

NLS_Language specifies:

- language used for Oracle (error) messages
- day names and month names

note: NLS_LANGUAGE has NOTHING to do with inserting or selecting *data* in a certain language.

NLS_Territory specifies:

- monetary and numeric formats,
- territory and conventions for calculating week and day numbers

CLIENTS CHARACTERSET:

- defines for the Oracle client what encoding is used by the client application
- * or it matches your Windows code page (ACP for a GUI tool, CHCP value for a DOS box/CMD prompt)
- * or it set to UTF8/AL32UTF8 for an Unicode WIN32 application

4 important remarks:

* Setting the NLS_LANG to the character set of the database (NLS_CHARACTERSET) MAY be correct but IS NOT ALWAYS correct. Please DO NOT assume that NLS_LANG needs to be ALWAYS the same as the database character set. THIS IS NOT TRUE.

* The character set defined with the NLS_LANG parameter does NOT CHANGE your client's character set, it is used to let Oracle know what character set you are USING on the client side, so Oracle can do the proper conversion.

You cannot just set NLS_LANG to the character set you WANT. If you need Hebrew support (for example) on an Cyrillic windows then that Windows system need to be changed FIRST to have an 1255 ACP (see point 3) and then , AFTER this, the NLS_LANG needs to be set to the Windows configuration.

Just setting the nls_lang to Hebrew will NOT allow you to retrieve/store Hebrew.

* Another myth is that if you don't set the NLS_LANG on the client it uses the NLS_LANG of the server. This is also NOT true! The character set part of the NLS_LANG parameter is never inherited from the server. Please also see: [Document 241047.1](#) The Priority of NLS Parameters Explained.

* Note that NLS_LANGUAGE and NLS_TERRITORY have nothing to do with the ability to store (insert) or retrieve (select) characters from a certain language in/from a database. A NLS_LANG set to JAPANESE_JAPAN.WE8MSWIN1252 will not allow you to store Japanese as WE8MSWIN1252 doesn't know Japanese characters.

For <NLS_Language> and <NLS_Territory> part of NLS_LANG see [Document 241047.1](#) The Priority of NLS Parameters Explained.

1.1 Windows and Dos Code Pages:

On Windows systems, the encoding scheme (=Character set) is specified by a Code Page. Code Pages are defined to support specific languages or groups of languages which share common writing systems.

From Oracle point of view the terms Code Page and Character set mean the same.

Note that in non Chinese-Japanese-Korean environments, the Windows GUI and DOS command prompt do not use the same code page (!).

As a result windows uses 2 different charactersets for the ANSI (sqlplusw.exe) and to OEM (dos box - sqlplus.exe) environments.

To illustrate the issue:

```
open notepad, type some "accented characters" like"éçèà", save that file as
c:\test.txt
(make sure that the encoding save as setting of notepad is set to ANSI)
open a dos box
Start - Run - cmd - ok
and type

C:\Documents and Settings\user>edit c:\test.txt

you see other characters in edit than in notepad
```

Here you see the SAME file in "ANSI" encoding (notepad) displayed by the OEM (dos box) environment of edit. You see clearly that windows do not use the same character set for the ANSI and CMD / DOS box environments. For more questions about this, please contact Microsoft.

Please *DO* make the difference between the GUI sqlplusW.exe and the "DOS box/CMD window" sqlplus.exe

1.2 Fonts:

A font is a collection of glyphs (from "hieroglyphs") that share common appearances (typeface, character size). A font is used by the operating system to convert a numeric value into a graphical representation on screen.

A font does not necessarily contain a graphical representation for all numeric values defined in the code page you are using. That's why you get sometimes black squares on the screen if you change fonts and the new that font has no representation for a certain symbol.

The Windows "Character Set Map" utility can be used to see which glyphs are part of a certain font.

On Windows 2000:

```
Start -> Programs -> Accessories -> System Tools -> Character Map
or
Start -> Run...
Type "charmap", and click "ok"
```

A font also implements a particular code page or set of code pages.

For example, the Arial font implements the code pages 1252, 1250, 1251, 1253, 1254, 1257.

For more in-depth info on fonts see point 8 in this note.

2. How to setup my NLS_LANG:

To specify the locale behaviour of your client Oracle software, you have to set your NLS_LANG parameter. It sets the language, territory and also the character set of your client.

For a short overview, it uses the following format:

NLS_LANG=<NLS_Language>_<NLS_Territory>.<clients character set>

2.1 In the Registry:

By default any Oracle installation on windows uses the registry to define the settings.

The value set by default by the Oracle Universal Installer is the depending on the ACP value of the Windows OS at the time of installation.

There will be a NLS_LANG registry subkey for each of your Oracle Homes, you can easily modify this subkey with the Windows Registry Editor:

Start -> Run...

Type "regedit", and click "ok"

Edit the following registry entry:

for version 7:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
```

For version 8 , 8i and 9i:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOMEx\
```

where "x" is the unique number identifying the Oracle home. HOME0 is the first installation

For version 10g and up:

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_<oracle_home_name>
```

There you have an entry with as name NLS_LANG

For 32 bit software installed on x64 windows platforms the 32 bit compatibility location is used

```
HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\ORACLE\KEY_<oracle_home_name>
```

Note that:

* for 32bit installs on an x64 OS there will be no

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_<oracle_home_name>
```

seen Windows redirects all to the "32 bit compatibility location"

* Oracle does NOT certify the 32bit Database on a 64bit OS. You can install the 32bit client software, but NOT the 32bit Database software.

[note 1173433.1](#) How to Install Oracle 10.2.0.5 on MS Windows 7 / Windows 2008R2

* there [Bug 8277395](#): WINDOWS 2008 X64 10.2.0.4 INSTALLATION DOESN'T SET NLS_LANG REGISTRY , a windows x64 bit install will not set the NLS_LANG during install

When starting an Oracle tools, like sqlplusw.exe (or sqlplus.exe) , it will read the content of the oracle.key file located in the same directory to determine which registry tree will be used, therefore which NLS_LANG subkey will be used.

Note: Some people are confused by finding a NLS_LANG set to "NA" in HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE when no version 7 was installed.

This is used for backwards compatibility, ignore this or it can be even removed. For 8.0, 8i and 9i you need to set the NLS_LANG in HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOMEx\

2.2 As a System or User Environment Variable, in System properties:

Although the Registry is the primary repository for settings on Windows, it is not the only place where parameters can be set.

Even if not at all recommended, you can set the NLS_LANG as a System or User Environment Variable in the System properties. This setting will be used for ALL Oracle homes.

To check and modify them:

```
Right-click the 'My Computer' icon -> 'Properties'  
Select the 'Advanced' Tab -> Click on 'Environment Variables'
```

The 'User Variables' list contains the settings for the specific OS user currently logged on and the 'System variables' system-wide variables for all users.

Since these environment variables take precedence of the parameters already set in your Registry, you should not set Oracle parameters at this location unless you have a very good reason.

Particularly note the "ORACLE_HOME" parameter that is set on Unix but NOT on windows.

2.3 As an Environment variable defined in the command prompt:

If you set the NLS_LANG as an environment variable in a Command prompt, be aware that it will overwrite the current NLS_LANG setting in the Registry and also the System Properties.

In an MS-DOS command prompt, use the set command, for example:

```
C:\> set NLS_LANG=American_America.WE8PC850
```

3. The correct NLS_LANG for my Windows Code Page:

The NLS_LANG should reflect the used Windows setting, if the windows setting is not what you want then you need to change the Windows Setting.

The ACP (Ansi Code Page) and the default OEM (dos box / CMD window code page) is defined by the "default locale" setting of windows, so if you have a UK Windows 2000 client and you want to input Cyrillic (Russian) you need to change the ACP (by changing the "default locale") in order to be able to input Russian.

See [Document 199926.1](#) How to change the ANSI Code Page (ACP) on Windows.

Not all languages have a ANSI encoding, sqlplus on Windows only supports ANSI (and OEM) charactersets.

The following URL provides a list of the default code pages for a language for Windows xp and later versions: <http://msdn.microsoft.com/en-gb/goglobal/bb896001.aspx>

OEM = the command line codepage (dos box), ANSI = the gui codepage

An ANSI codepage of 0 in this list means that the language is ONLY supported by Microsoft windows trough a Unicode layer and that there is no ANSI code page for that language

An OEM codepage of 1 in this list means that the language is supported by Microsoft windows trough a Unicode layer and that there is no OEM code page for that language

If you want to see / insert languages with a 0 or 1 on a windows client then you need to use a Unicode client. Sqlplws.exe and sqlplus.exe are NOT Unicode clients (but they can connect to an Unicode database) and CANNOT be used for these languages.

You can use UTF8/AL32UTF8 as Oracle client character set (=NLS_LANG) on Windows NT, 2000 and XP but you will be limited to use only client programs that explicitly support this configuration.

Programs relying on ANSI Win32 API, like SQL*Plus (sqlplusW.exe), older Oracle Forms , etc. cannot work with an NLS_LANG set to UTF8 / AL32UTF8.

Examples of Unicode program are

* Oracle Forms in version 5 and later on NT 4.0. [Document 105809.1](#) Character Set Support for Developer Tools

* SQL Developer which can be downloaded from <http://www.oracle.com/technetwork/developer-tools/sql-developer> If your data is visible but looks like a square block with some number in it then the used font cannot display this language you change the font in SQL Developer in "Tools -> Preferences", Expand the Code Editor node and select Fonts, then choose for example "Arial Unicode MS".

* iSqlplus.

[Document 231231.1](#) Quick setup of iSQL*Plus 9.2 as Unicode (UTF8) client on windows.

[Document 281847.1](#) How do I configure or test iSQL*Plus 10i?

Contact Microsoft if you have questions about writing a Windows Unicode application.

An intro can be found on MSDN [http://msdn.microsoft.com/en-us/library/dd374081\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/dd374081(VS.85).aspx)

Note: Microsoft Windows uses UTF16 as Unicode encoding, but the NLS_LANG need to be set to UTF8/AL32UTF8 not AL16UTF16. The Oracle client libraries will do then the conversion (which is a byte shift operation and is very fast) from Windows UTF16 to Oracle's AL32UTF8. For specific instructions on adaptors like ODBC, JDBC, etc please check the relevant documentation. Only in some precompiler situations UTF16 is used on the Oracle side.

For tools like sqlloader you need to set the NLS_LANG to the characterset of the FILE you loading.

[Document 227330.1](#) Character Sets & Conversion - Frequently Asked Questions

18. What is the best way to load non-US7ASCII characters using SQL*Loader?

For Export / Import please see: [Document 227332.1](#) NLS considerations in Import/Export

For Honk Kong HKSCS see [Document 787371.1](#) Oracle Database Server support for HKCSC 1999, 2001 and 2004 character sets.

3.1 Determine your Windows ACP code page (used by sqlplusW.exe):

You'll find its value in the registry:

Start -> Run...

Type "regedit", and click "ok"

Browse the following registry entry:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Nls\CodePage\
```

There you have (all the way below) an entry with as name ACP . The value of ACP is your current GUI Codepage, see the table in point 3.2 for the mapping to the oracle name.

Since there are many registry entries with very similar names, please make sure that you are looking at the right place in the registry.

Again, if you need to change the "ACP" please see: [Document 199926.1](#) How to change the ANSI Code Page (ACP) on Windows

Do NOT simply change it in the registry, this will NOT work and provoke very confusing results.

3.2 Find the correspondent Oracle client character set:

Find the Oracle client character set in the table below based on the ACP you found in point 3.1.

Note that there is only ONE CORRECT value for a given ACP

Not all scripts / languages have ANSI support. The following languages have NO ANSI support in windows:

Armenian, Divehi, Georgian, Gujarati, Hindi, Kannada, Oriya, Konkani, Marathi, Punjabi, Sanskrit, Syriac, Tamil and Telugu. (there are more...

this list is just a example, see point 3.1 in this note)

```
ANSI CodePage (ACP) - Oracle Client character set (3rd part of NLS_LANG)
```

```
1250 - EE8MSWIN1250
1251 - CL8MSWIN1251
1252 - WE8MSWIN1252
1253 - EL8MSWIN1253
1254 - TR8MSWIN1254
1255 - IW8MSWIN1255
1256 - AR8MSWIN1256
1257 - BLT8MSWIN1257
1258 - VN8MSWIN1258
874 - TH8TISASCII
932 - JA16SJIS
936 - ZHS16GBK
949 - KO16MSWIN949
950 - ZHT16MSWIN950 - except for Hong kong.
950 - (Hong Kong) ZHT16HKSCS31 / ZHT16HKSCS . Only possible on Windows 2000 and XP, not on Vista and 7,
see Document 787371.1 Oracle Database Server support for HKCSC 1999, 2001 and 2004 character sets.
```

This is the characterset used by the GUI sqlplus (sqlplusW.exe/ plus80W.exe / plus33W.exe) that you start trough the windows start menu.

Please *DO* make the difference between the GUI sqlplusW.exe and the "DOS mode" sqlplus.exe (see point 4 in this note)

3.3 Set it in your Registry:

Use the Windows Registry Editor to set up the NLS_LANG in your Oracle Home with the value you have just find above. Section 2.1 gives you more details on how to use the Registry Editor for that purpose.

4. The correct NLS_LANG for my DOS / Command Prompt OEM Code Page (used by sqlplus.exe) :

MS-DOS mode uses, with a few exceptions like CJK, a different code page (called OEM code page) than Windows GUI (ANSI code page).

There is no euro support in the default Command Prompt OEM Code Pages, see: [Document 68790.1](#) RDBMS Support for the Euro Currency Symbol

Meaning that before using an Oracle command line tool such as SQL*Plus (sqlplus.exe/ plus80.exe / plus33.exe) en svrmgrl in a command prompt then you need to MANUALLY SET the NLS_LANG parameter as an environment variable with the set DOS command BEFORE using the tool.

For Japanese, Korean, Simplified Chinese, and Traditional Chinese the MS-DOS OEM code page (CJK) is identical to the ANSI code page meaning that, in this particular case, there is no need to set the NLS_LANG parameter in MS-DOS mode.

Not all scripts / languages have OEM support. The following languages have NO OEM support in windows:
Armenian, Divehi, Georgian, Gujarati, Hindi, Kannada, Oriya, Konkani, Marathi, Punjabi, Sanskrit, Syriac, Tamil and Telugu. (there are more... this list is just a example, see point 3.1 in this note)

In all other cases, you need to set it in order to overwrite the NLS_LANG registry key already matching the ANSI code page. The new "MS-DOS dedicated" NLS_LANG needs to match the MS-DOS OEM code page that could be retrieved by typing chcp in a Command Prompt:

```
C:\> chcp
Active code page: 437
C:\> set NLS_LANG=American_America.US8PC437
C:\> sqlplus.exe /nolog
```

If the NLS_LANG parameter for the MS-DOS mode session is not set appropriately, error messages and data can be corrupted due to incorrect character set conversion.

Use the following list to find the Oracle character set that fits to your MS-DOS code page in use on your locale system:

```
MS-DOS code page Oracle Client character set (3rd part of NLS_LANG)
437 US8PC437
720 AR8DOS720
737 EL8PC737
850 WE8PC850
852 EE8PC852
857 TR8PC857
858 WE8PC858
861 IS8PC861
862 IW8PC1507
865 N8PC865
866 RU8PC866
```

For tools like sqlloader you need to set the NLS_LANG to the characterset of the FILE you loading.

[Document 227330.1](#) Character Sets & Conversion - Frequently Asked Questions

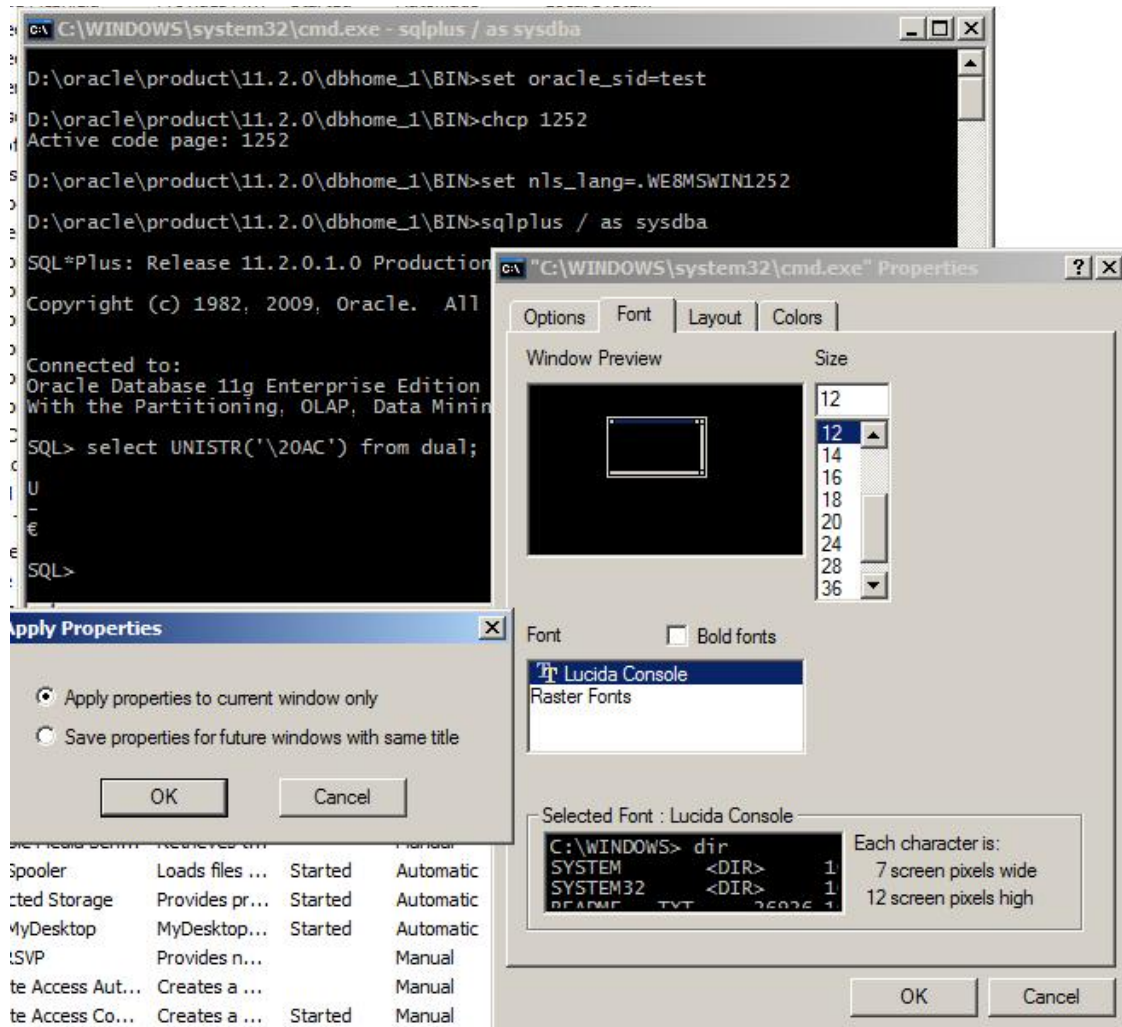
18. What is the best way to load non-US7ASCII characters using SQL*Loader?

For Export / Import please see: [Document 227332.1](#) NLS considerations in Import/Export

Note that it is possible to configure the DOS / Command Prompt to use the ANSI encodings. To use ANSI chartersets in the console window, you must use "Lucida Console" because the "raster fonts" only know the OEM codepages.

Right-click on the title bar, choose Properties, go to the Fonts tab, and select Lucida Console. Click on "ok" and select "Save properties for future windows with same title".

Once this is done one can do for example "chcp 1252" to switch to the WE8MSWIN1252 characterset. Note that "chcp" is NOT persistent, so this needs to be done every time or the shortcut needs to be adapted. If you open a new DOS/CMD window it will use by default the OEM codepage defined for this Windows system.



The "Unicode" "chcp 65001" and "chcp 65000" values are NOT supported with sqlplus.exe.

5. How to check the NLS_LANG:

To check the NLS_LANG, you need to open a command prompt and to run sqlplus in command line mode.

First, check if it's set in the environment:

```
SQL> host echo %NLS_LANG%
```

If this reports just %NLS_LANG% back, the variable is not set in the environment. If it's set it reports something like ENGLISH_UNITED KINGDOM.WE8PC850

If NLS_LANG is not set in the environment, you should check the value in the registry:

```
SQL> @.[%NLS_LANG%].
```

If you get something like:

```
unable to open file ".[ENGLISH_UNITED KINGDOM.WE8MSWIN1252]."
```

the "file name" between the '[' is the value of the registry parameter. (This is NOT an error but just a "trick" to get the NLS_LANG value)

If you get this as result:

```
unable to open file "[%NLS_LANG%]."
```

then the parameter NLS_LANG is also not set in the registry.

Note: the @:[%NLS_LANG%]. "trick" reports the NLS_LANG known by the sqlplus executable, it will not read the registry itself. But then you are not sure if the variable is set in the environment or in the registry. That's the reason of checking with the host commando first.

6. List of common NLS_LANG's used in the Windows Registry:

note: this is the correct setting for the GUI sqlplus version, (sqlplusW.exe/ plus80W.exe / plus33W.exe)

If you are testing with "special" characters please DO use the gui and not the "dos box" sqlplus.exe !

Not all scripts / languages have ANSI support. The following languages have NO ANSI support in windows:

Armenian, Divehi, Georgian, Gujarati, Hindi, Kannada, Oriya, Konkani, Marathi, Punjabi, Sanskrit, Syriac, Tamil and Telugu. (there are more... this list is just a example, see point 3.1 in this note)

Operating System Locale	Default NLS_LANG Value
Arabic (U.A.E.)	ARABIC_UNITED ARAB EMIRATES.AR8MSWIN1256
Bulgarian	BULGARIAN_BULGARIA.CL8MSWIN1251
Catalan	CATALAN_CATALONIA.WE8MSWIN1252
Chinese (PRC)	SIMPLIFIED CHINESE_CHINA.ZHS16GBK
Chinese (Taiwan)	TRADITIONAL CHINESE_TAIWAN.ZHT16MSWIN950
Chinese (Hong Kong HKCS)	TRADITIONAL CHINESE_HONG KONG.ZHT16HKSCS
Chinese (Hong Kong HKCS31)	TRADITIONAL CHINESE_HONG KONG.ZHT16HKSCS31 (new in 10gR2) HKSCS is only possible on Windows 2000 and XP, not on Vista and 7, see Document 787371.1 Oracle Database Server support for HKCSC 1999, 2001 and 2004 character sets.
Croatian	CROATIAN_CROATIA.EE8MSWIN1250
Czech	CZECH_CZECH REPUBLIC.EE8MSWIN1250
Danish	DANISH_DENMARK.WE8MSWIN1252
Dutch (Belgium)	DUTCH_BELGIUM.WE8MSWIN1252
Dutch (Netherlands)	DUTCH_THE NETHERLANDS.WE8MSWIN1252
English (United Kingdom)	ENGLISH_UNITED KINGDOM.WE8MSWIN1252
English (United States)	AMERICAN_AMERICA.WE8MSWIN1252
Estonian	ESTONIAN_ESTONIA.BLT8MSWIN1257
Finnish	FINNISH_FINLAND.WE8MSWIN1252
French (Canada)	CANADIAN FRENCH_CANADA.WE8MSWIN1252
French (France)	FRENCH_FRANCE.WE8MSWIN1252
German (Germany)	GERMAN_GERMANY.WE8MSWIN1252
Greek	GREEK_GREECE.EL8MSWIN1253
Hebrew	HEBREW_ISRAEL.IW8MSWIN1255
Hungarian	HUNGARIAN_HUNGARY.EE8MSWIN1250
Icelandic	ICELANDIC_ICELAND.WE8MSWIN1252
Indonesian	INDONESIAN_INDONESIA.WE8MSWIN1252
Italian	ITALIAN_ITALY.WE8MSWIN1252
Japanese	JAPANESE_JAPAN.JA16SJIS
Korean	KOREAN_KOREA.KO16MSWIN949
Latvian	LATVIAN_LATVIA.BLT8MSWIN1257
Lithuanian	LITHUANIAN_LITHUANIA.BLT8MSWIN1257
Norwegian	NORWEGIAN_NORWAY.WE8MSWIN1252
Polish	POLISH_POLAND.EE8MSWIN1250
Portuguese (Brazil)	BRAZILIAN PORTUGUESE_BRAZIL.WE8MSWIN1252
Portuguese (Portugal)	PORTUGUESE_PORTUGAL.WE8MSWIN1252
Romanian	ROMANIAN_ROMANIA.EE8MSWIN1250
Russian	RUSSIAN_CIS.CL8MSWIN1251
Slovak	SLOVAK_SLOVAKIA.EE8MSWIN1250
Spanish (Mexico)	MEXICAN SPANISH_MEXICO.WE8MSWIN1252
Spanish (Spain)	SPANISH_SPAIN.WE8MSWIN1252
Spanish (Venezuela)	LATIN AMERICAN SPANISH_VENEZUELA.WE8MSWIN1252

Swedish	SWEDISH_SWEDEN.WE8MSWIN1252
Thai	THAI_THAILAND.TH8TISASCII
Turkish	TURKISH_TURKEY.TR8MSWIN1254
Ukrainian	UKRAINIAN_UKRAINE.CL8MSWIN1251
Vietnamese	VIETNAMESE_VIETNAM.VN8MSWIN1258

*Note that NLS_LANGUAGE and NLS_TERRITORY have nothing to do with the ability to store (insert) or retrieve (select) characters from a certain language in/from a database. Setting on an 1252 (US/West European windows client) the NLS_LANG set to JAPANESE_JAPAN.WE8MSWIN1252 will not allow you to store Japanese as WE8MSWIN1252 doesn't know Japanese characters. On a Japanese windows client using AMERICAN_AMERICA.JA16SJIS WILL allow you to store or retrieve Japanese (if the NLS_CHARACTERSET also supports Japanese)

7. List of common NLS_LANG's used in the Command Prompt (DOS box):

note: this is the correct setting for the DOS BOX sqlplus version (sqlplus.exe/ plus80.exe / plus33.exe)

Not all scripts / languages have OEM support. The following languages have NO OEM support in windows:

Armenian, Divehi, Georgian, Gujarati, Hindi, Kannada, Oriya, Konkani, Marathi, Punjabi, Sanskrit, Syriac, Tamil and Telugu. (there are more... this list is just a example, see point 3.1 in this note)

Operating System Locale	Oracle Client character set (3rd part of NLS_LANG)
-------------------------	--

Arabic (limited support in DOS)	AR8DOS720
Catalan	WE8PC850
Chinese (PRC)	ZHS16GBK
Chinese (Taiwan)	ZHT16MSWIN950
Czech	EE8PC852
Danish	WE8PC850
Dutch	WE8PC850
English (United Kingdom)	WE8PC850
English (United States)	US8PC437
Finnish	WE8PC850
French	WE8PC850
German	WE8PC850
Greek	EL8PC737
Hebrew	IW8PC1507
Hungarian	EE8PC852
Italian	WE8PC850
Japanese	JA16SJIS
Korean	KO16MSWIN949
Norwegian	WE8PC850
Polish	EE8PC852
Portuguese	WE8PC850
Romanian	EE8PC852
Russian	RU8PC866
Slovak	EE8PC852
Slovenian	EE8PC852
Spanish	WE8PC850
Swedish	WE8PC850
Turkish	TR8PC857

8. How Windows uses Fonts to display the different charactersets:

We assume you have an UTF8 database with correctly stored UTF8 codepoints.

On Windows there are two kinds of tools / applications:

1) A fully Unicode enabled application which accepts Unicode codepoints and which can render them. It's the application that needs to deal with the Unicode, Windows provides the Unicode API but the GUI system itself is NOT Unicode "by default".

A fully Unicode application can only show one glyph for a given Unicode code point. So there is NO confusion possible here, this application will need to use a full Unicode font. If you have a full Unicode application, then you need to set the NLS_LANG to UTF8.

Note that there are currently NOT many applications like this and if it's not explicitly mentioned by the vendor it's most likely an ANSI application (see below). So DON'T set the NLS_LANG to UTF8 if you are not sure!

An Unicode capable client that is included in the database is iSQL*Plus. See [Document 231231.1](#) Quick setup of iSQL*Plus as Unicode client on windows. or SqlDeveloper

2) An standard ANSI application (like sqlplusw.exe) cannot use Unicode code points. So the Unicode code point stored in the database needs to be CONVERTED to a ANSI code point. This is done by setting NLS_LANG (as described in further on in this note). This allows oracle to map the Unicode point to the character set of the client (sqlplus in this case), (and here comes the tricky part) but this is NOT the same as a font.

If you want to display Arabic for example then you need to set the Windows LOCALE setting to Arabic. That way Windows knows what are valid codepoints and can use the FONT engine to DISPLAY the codepoints (this results in glyphs).

Windows passes the codepoint and the "page" to the rendering engine. This Arabic "page" defines the glyphs for the codepoints for a certain character set/codepage.

Because there are possible positions for a ANSI application, and one font contains normally glyphs for different languages this "page" is used to select from a FONT that has (for example) all the glyphs for Cyrillic, Arabic and West-European the "page" for Arabian.

So lets say you have a Arabic setup that works, you change manually the "Page" of a FONT and ask to display the glyph for ANSI codepoint XX. Now 1 of 2 things can happen:

1) There is a character defined on that position for the CHARACTERSET of that "Page", so the creator of the font has foreseen a glyph and this is displayed (but this is NOT the character expected or wanted as its stored as a different character in the database!).

2) There is NO character defined on that position for the CHARACTERSET of that "Page" so the creator of the font has NOT foreseen a glyph and you get black squares. (normally you should see a (black) square but a ? or ¿ are also possible in less quality fonts, this depends on the error handling defined in the FONT).

The above is also possible if you have a non-Unicode character set for the database.

9. Updating/changing the NLS_LANG in the registry to the correct value:

This can for example be done by importing a .reg file who will then delete the old and add the new registry value

like:

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\]
"NLS_LANG"=-
[HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\HOME0]
"NLS_LANG"=-
"NLS_LANG"="DUTCH_BELGIUM.WE8MSWIN1252"
```