



MythBusters Globalization Support

Avoid Data Corruption

Christian Gohmann
Data Eng, Mgmt & Governance Associate Manager

Accenture Technology

Who am I?



Christian Gohmann

- Data Eng, Mgmt & Governance Associate Manager
- Instructor (O-AI, O-AI-DBA, O-BR)
- Tool Owner of db*BACKUP
- Working with Oracle since 2006
- Focused on High Availability Solutions, Migration Projects, Backup & Recovery and Cloud Technologies (OCI, ExaCC)
- Blog: <https://www.christian-gohmann.de>
- Twitter/X: CGohmannDE





400+ technical experts helping peers globally

The **Oracle ACE Program** recognizes and rewards community members for their technical and community contributions to the Oracle community



3 membership tiers



For more details on Oracle ACE Program:
ace.oracle.com



Nominate
yourself or someone you know:
ace.oracle.com/nominate

Connect:  aceprogram_ww@oracle.com

 Facebook.com/OracleACEs

 [@oracleace](https://twitter.com/oracleace)



SYMPOSIUM⁴²

Created by the community, to support the community

Sharing of reliable knowledge

Supporting the various user groups and individuals



@sym_42



<https://sym42.org/>

Agenda

MythBusters Globalization Support

- 01** Fundamentals
- 02** Configuration
- 03** Conversion
- 04** Tools
- 05** Debugging
- 06** Migration



Fundamentals

Globalization Support

- Store, process and retrieve data in native languages
 - Allows the development of multilingual applications
- Former known as **National Language Support** (NLS)
 - Subset of the Globalization Support
- Implemented with the **Oracle NLS Runtime Library** (NLSRTL)
 - Functions for text and character processing and manipulations
- Locale-specific data is stored in `$ORACLE_HOME/nls/data`
 - Use `ORA_NLS10` environment variable to choose a different location
 - Loaded as independent set at runtime



Language Support

BUSTED

“Language Support means the ability to store text of a specific language”

- Refers to **the additional language-dependent functionalities**
 - Display dates using translated month names
 - Sort text according to cultural conventions
 - Translated error messages and user interfaces
- Only available for a subset of the languages
 - Parameter: NLS_LANGUAGE

```
SQL> ALTER SESSION SET nls_language = 'FRENCH';
```

Territory Support

BUSTED

“Territory support allows the database to store characters of different languages”

- Has nothing to do with the ability to store characters in the database
 - **Support of cultural conventions**, specific to geographical locations
 - Default local time format
 - Date format
- Numeric and monetary conventions
- Parameter: NLS_TERRITORY

```
SQL> ALTER SESSION SET nls_territory = 'FRANCE';
```

Character Set Support 1/2

CONFIRMED

"A character set maps a numeric code to a characters visual representation."

- Two character sets are defined at database creation

```
SQL> CREATE DATABASE TEST (  
    ...  
    CHARACTER SET WE8MSWIN2152  
    NATIONAL CHARACTER SET AL16UTF16  
);
```

- National Character Set** to store Unicode
 - Limited to NCHAR, NVARCHAR2 and NCLOB data types
- Oracle naming convention (except UTF8 and UTFE)
 - <Region><Bit length><Standard Character Set Name>[S|C]**
 - Optional: S → Server, C → Client
 - Example: **WE8ISO8859P15**



Character Set Support 2/2

BUSTED

"Windows character sets are limited to the Windows platform"

- Windows character sets like WE8MSWIN1252 can be used on Unix
 - EBCDIC character sets cannot be used on ASCII platforms and vice-versa



Some character sets are client-only.



Calendar Systems

- Support of eight different calendar systems

Name	Name
Gregorian	Japanese Imperial
ROC Official (Republic of China)	Thai Buddha
Persian	English Hijrah
Arabic Hijrah	Ethiopian

- Switch to a different calendar

```
SQL> ALTER SESSION SET nls_calendar = 'Persian';
```

- NLS Calendar Utility** can be used to register calendar modifications (plain text files *.nlt)

```
$> lxcgen
```

Unicode Support

- Support of the Unicode standard since Oracle 7

Oracle Version	Unicode Version	Oracle Version	Unicode Version
11g	5.0	12c	6.2 / 7.0
18c	9.0	19c	9.0
21c	12.1	23ai	15.0

- Two ways to store Unicode characters in the database
 - Create a database with a Unicode character set (e.g. AL32UTF8)
 - Support multilingual data in specific columns (NCHAR, NVARCHAR, NCLOB)
- Oracle recommends the usage of AL32UTF8 (default since Oracle 12c Release 2)
 - In a CDB at least on CDB\$ROOT level to support any character set on PDB level
- Database Migration Assistant for Unicode (DMU)



Enhancements of 23ai

- Support for Unicode 15.0
- New territory TURKIYE
- New linguistic sort ICELANDIC_IST
- New procedures in UTL_I18N to support detecting the most likely character set and language based on the input data
 - DETECT_CHARSET, DETECT_LANGUAGE, DETECT_LANGUAGE_CHARSET
- Improvements of DBMS_DST to optimize time zone data upgrades
- New parameter TIME_AT_DBTIMEZONE can be used to control which time zone should be used for time-dependent operations
 - OFF: database host system (default),
 - DATABASE: database
 - USER_SQL: PDB, only for SYSDATE and SYSTIMESTAMP



Configuration

General

- A new session initializes the NLS environment from the NLS instance parameters
 - If specific client-side NLS settings are given, an ALTER SESSION is executed
- Check **V\$NLS_VALID_VALUES** for valid values of NLS settings
- Direct connection from the server acts like a client connection
- Set NLS_% parameter on instance level requires restart of instance

```
SQL> ALTER SYSTEM SET nls_date_format = 'YYYY-MM-DD HH24:MI' SCOPE=SPFILE;
```

Client 1/3

BUSTED

“NLS_LANG must match the character set of the database server”

- NLS_LANG only represents the **character set that the client use**

```
$> export NLS_LANG=<Language>_<Territory>.<Client Character Set>
```

- Example: German Windows client → GERMAN_GERMANY.WE8MSWIN1252
- Example: Unix client with Unicode locale → AMERICAN_AMERICA.UTF8
- It can be the same, but it is not a must
- It is required for proper conversion
- If client and server character sets match, then no validation is enforced



Client 2/3

BUSTED

“If NLS_LANG is not set it uses the language, territory and character set of the database server”

- If not set it defaults to the hardcoded value **US7ASCII**

```
SQL> ! echo $NLS_LANG
```

```
SQL> SELECT client_charset, network_service_banner
       FROM v$session_connect_info
       WHERE sid = SYS_CONTEXT('USERENV', 'SID');
```

```
CLIENT_CHARSET NETWORK_SERVICE_BANNER
```

```
-----
US7ASCII      Oracle Bequeath NT Protocol Adapter for Linux: Version 12.1.0.1.0 - Production
US7ASCII      Authentication service for Linux: Version 12.1.0.1.0 - Production
US7ASCII      Encryption service for Linux: Version 12.1.0.1.0 - Production
US7ASCII      Crypto-checksumming service for Linux: Version 12.1.0.1.0 - Production
```

- Windows registry key: HKLM\SOFTWARE\ORACLE\KEY_<Oracle Home Name>



Client 3/3

BUSTED

“Setting NLS_LANG changes the character set of the client”

- Setting NLS_LANG changes nothing on the client
- The character set can only be changed with operating system tools

```
# Change Locale on Linux
$> export LANG=de_DE.CP1252
$> locale
LANG=de_DE.CP1252
LC_CTYPE="de_DE.CP1252"
...
LC_IDENTIFICATION="de_DE.CP1252"
```

Miscellaneous

- Different tools or input files require specific NLS_LANG settings

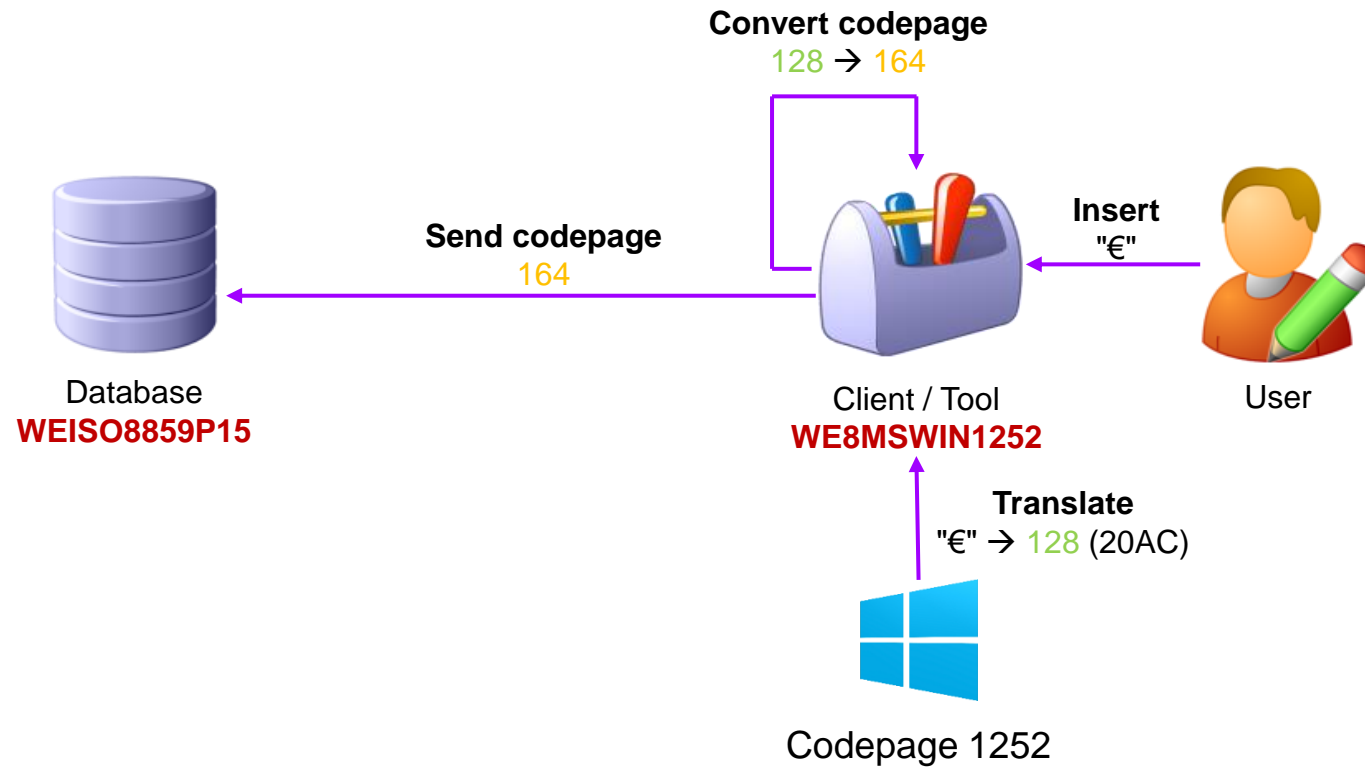
Tool	NLS_LANG	Comment
SQL Loader / SQL Developer	Character set of the text/flat file	
UTL_FILE	Does not use it	Runs in the database. Assume encoding of the input file is in the character set of the database. Otherwise use UTL_RAW.CONVERT
Export	Same as the database character set	No character conversion
Import	Character set of the source database	Check with <code>imp .. show=yes</code>
Data Pump	Not affected by the NLS_LANG setting. Only used for the parameter file.	
SQLcl	Does not evaluate NLS_LANG.	Because SQLcl is a Java program, it evaluates environment variable LANG (except Windows)

Configuration

Character Conversion

- Normally conversion is done **at client side** (OCI driver)
 - 8.0.4 onwards
- If character set is not known at client side, server side will do conversion
 - 8.1.6 onwards
- If server and client-side **character sets are the same, no validation** will be performed
 - Increased performance
 - **Chance to write data garbage if the “real” client character set is different**

Character Lifecycle



Special Characters

Character	Description
□	The used font has no graphical representation for the codepage
¿	Replacement character that is used for unknown characters in the character set

- Some character sets offers **mappings from unknown characters to known characters**
 - E.g. é → e, ä → a

Tools

SQL*Plus 1/2

CONFIRMED

*"The codepage of running SQL*Plus in a DOS box differs from the regular operating system codepage"*

- SQL*Plus in a DOS box does not use the regular Windows codepage
- Instead it uses the **OEM environment** (e.g. codepage 850)

```
$> chcp  
Active Codepage: 850
```

- Character set part of NLS_LANG must be set to **WE8PC850**
- Alternatively change codepage using chcp and set different font

```
$> chcp 1252
```



The PowerShell is using the default codepage of the operating system (e.g. 1252).

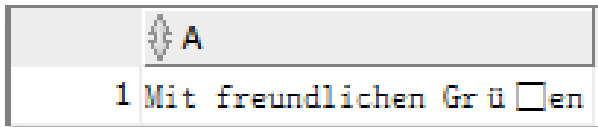


SQL*Plus 2/2

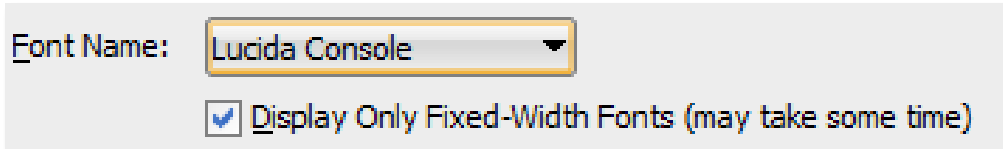
- Until Oracle 10g an ANSI version of SQL*Plus (sqlplusw.exe) exists
 - Uses the configured Windows codepage (e.g. 1252)
 - Officially replaced by the SQL Developer
- On Windows only: SQL*Plus is not **UTF8/Unicode aware**

SQL Developer 1/2

- SQL Developer **needs no client side NLS configuration**
 - Unicode client
- Download from Oracle Technology Network (OTN)
 - <https://www.oracle.com/database/sqldeveloper/technologies/download/>
- Change font if □ characters appear in the output
 - Reason: The used font cannot display the character

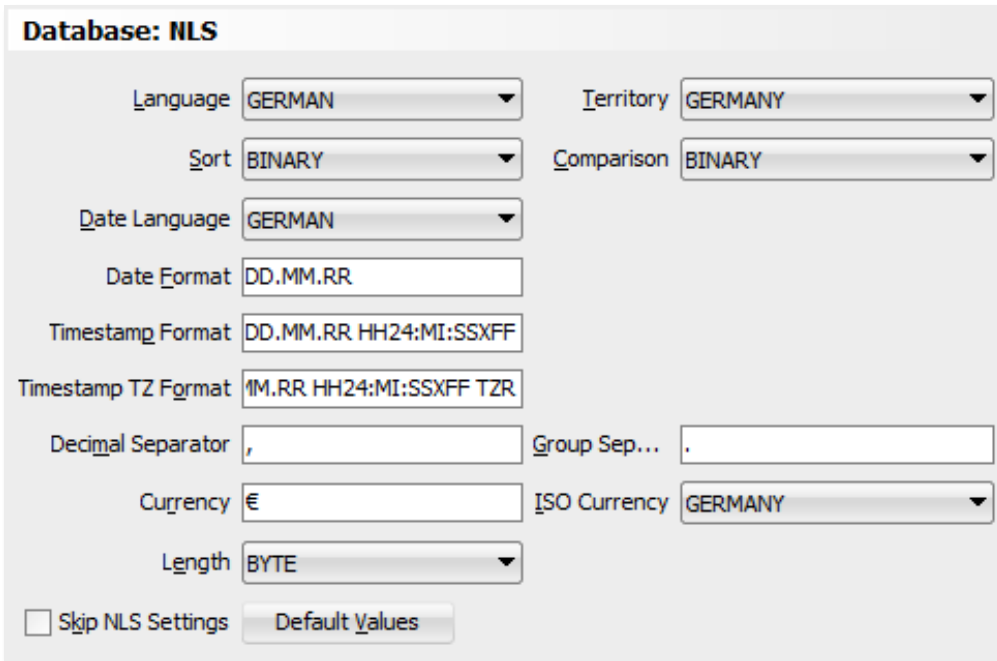


- Navigate to **Tools > Preferences > Code-Editor > Fonts**



SQL Developer 2/2

- Change specific NLS settings of the SQL Developer session
 - Tools > Preferences > Database > NLS**



The screenshot shows the 'Database: NLS' preferences dialog box. It contains the following settings:

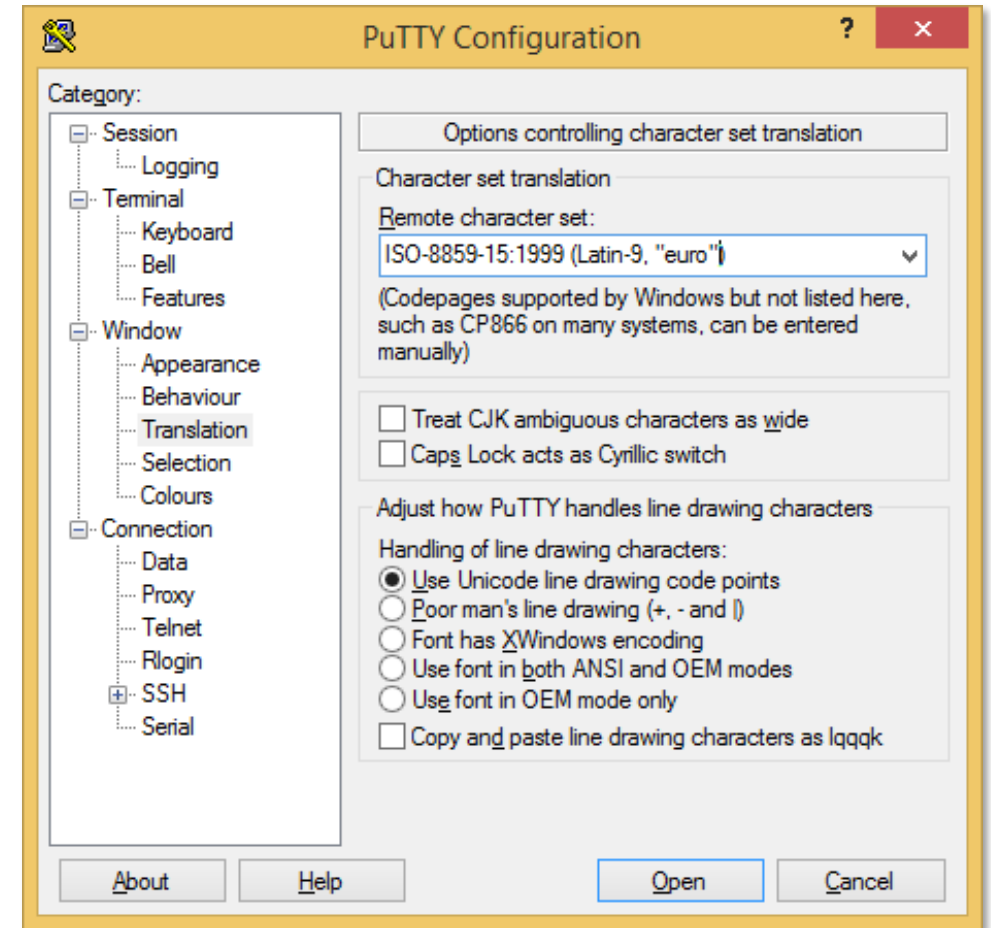
Language	GERMAN	Territory	GERMANY
Sort	BINARY	Comparison	BINARY
Date Language	GERMAN		
Date Format	DD.MM.RR		
Timestamp Format	DD.MM.RR HH24:MI:SSXFF		
Timestamp TZ Format	MM.RR HH24:MI:SSXFF TZR		
Decimal Separator	,	Group Sep...	.
Currency	€	ISO Currency	GERMANY
Length	BYTE		

At the bottom, there is a checkbox for 'Skip NLS Settings' which is unchecked, and a 'Default Values' button.

- Changes will be **applied immediately**

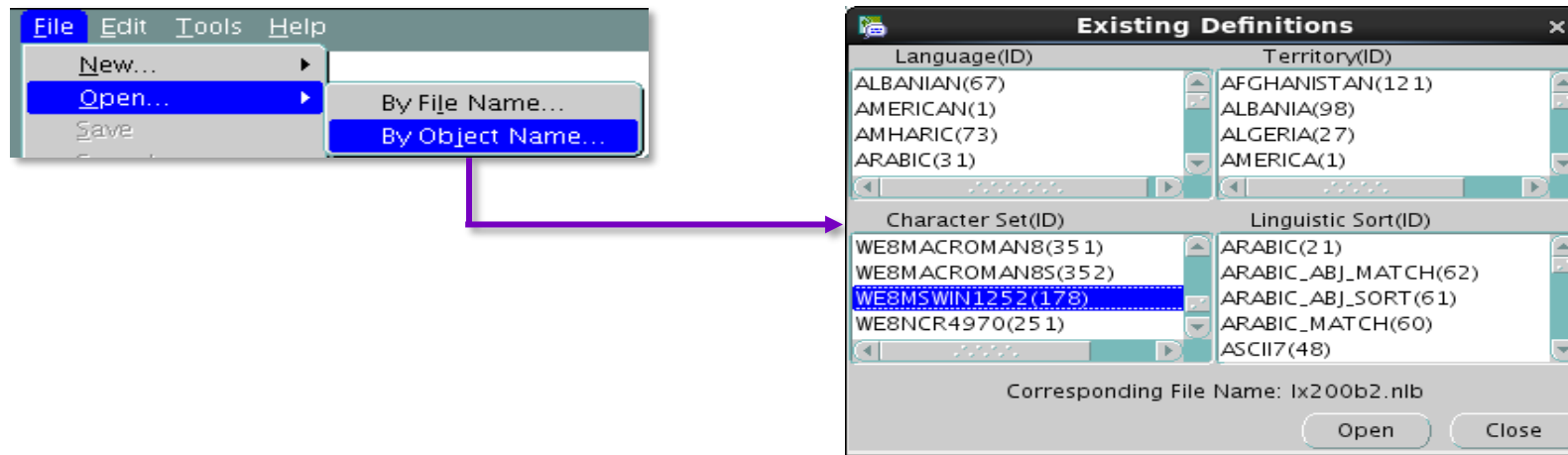
Telnet/SSH Client

- Responsible for conversion of the Unix locale to the client's environment
 - Translation settings **must match the encoding of the Unix Shell**
- Example: Connection to an **en_EN.ISO8859-15** Unix Shell with PuTTY



Locale Builder 1/2

- **View, modify or create** Language, Territory, Character Set or Linguistic Sort definitions
- Available since Oracle 9i
 - Windows: %ORACLE_HOME%\nls\lbuilder\lbuilder.exe
 - Unix: \$ORACLE_HOME/nls/lbuilder/lbuilder
- Open NLS locale data definitions using file or object name



Locale Builder 2/2

Oracle Locale Builder - Ix200b2.nlb

General | Type Specification | Charac... | Lower... | Upper... | Class... | Repla... | Displ...

Character Set Name:

Character Set ID:

ISO Character Set ID:

Base Character Set ID:

Filename: Ix200b2.nlb | Category: Character Set | Name: WE8MSWIN1252 | Status: Viewing

Oracle Locale Builder - Ix200b2.nlb

General | Type Specification | Charac... | Lower... | Upper... | Class... | Repla... | Displ...

LocalChar Value	Glyph	Unicode Value
0x39	9	\u0039
0x3a	:	\u003a
0x3b	;	\u003b
0x3c	<	\u003c
0x3d	=	\u003d
0x3e	>	\u003e
0x3f	?	\u003f
0x40	@	\u0040
0x41	A	\u0041
0x42	B	\u0042
0x43	C	\u0043
0x44	D	\u0044
0x45	E	\u0045

LocalChar Value	Unicode Value
0x41	\u0041

Filename: Ix200b2.nlb | Category: Character Set | Name: WE8MSWIN1252 | Status: Viewing

Oracle Locale Builder - Ix200b2.nlb

General | Type Specification | Charac... | Lower... | Upper... | Class... | Repla... | Displ...

Uppercase Value	Uppercase Glyph	Lowercase Value	Lowercase Glyph
0x3a	:	0x3a	:
0x3b	;	0x3b	;
0x3c	<	0x3c	<
0x3d	=	0x3d	=
0x3e	>	0x3e	>
0x3f	?	0x3f	?
0x40	@	0x40	@
0x41	A	0x61	a
0x42	B	0x62	b
0x43	C	0x63	c
0x44	D	0x64	d
0x45	E	0x65	e
0x46	F	0x66	f
0x47	G	0x67	g

Uppercase Value	Lowercase Value
0x41	0x61

Filename: Ix200b2.nlb | Category: Character Set | Name: WE8MSWIN1252 | Status: Viewing

Oracle Locale Builder - Ix200b2.nlb

General | Type Specification | Charac... | Lower... | Upper... | Class... | Repla... | Displ...

Default Replacement Character:

Default Multibyte Replacement Character:

Unicode Value	Unicode Glyph	LocalChar Value	LocalChar Glyph
\u0100	Ā	0x41	A
\u0101	ā	0x61	a
\u0102	Ă	0x41	A
\u0103	ă	0x61	a
\u0104	Ą	0x41	A
\u0105	ą	0x61	a
\u0106	Ć	0x43	C
\u0107	ć	0x63	c
\u0108	Ĉ	0x43	C
\u0109	ĉ	0x63	c
\u010a	Ċ	0x43	C
\u010b	ċ	0x63	c

Unicode Value	LocalChar Value
\u0102	0x41

Filename: Ix200b2.nlb | Category: Character Set | Name: WE8MSWIN1252 | Status: Viewing

Oracle Locale Builder - Ix200b2.nlb

File | Edit | Tools | Help

Type S... | Chara... | Lower... | Upper... | Class... | Repla... | Displ... | Multi... | Prev...

```

1 <!--
2 # Copyright (c) 1996 - 2005 by Oracle Corporation. All Rights Reserved. \
3 # $
4 #
5 # NAME
6 # Ix200b2.nlb
7 # DESCRIPTION
8 # Character set definition for WE8MSWIN1252
9 # NOTES
10 #
11 -->
12 <!DOCTYPE NLSDATA SYSTEM "lx.dtd">
13 <NLSDATA>
14 <CHARACTER_SET>
15 <VERSION> 3.0.0.0.</VERSION>
16 <Name> WE8MSWIN1252 </Name>
17 <Id> 178 </Id>
18 <iso_charset> 31 </iso_charset>
19 <flags> ASCII_BASED,BYTE_UNIQUE </flags>
20 <default_replacement_char> 0xbf </default_replacement_char>
21 <character_data>
22 <P> <F> 0x00 </F> <T> 0x0000 </T> </P>
23 <P> <F> 0x01 </F> <T> 0x0001 </T> </P>
24 <P> <F> 0x02 </F> <T> 0x0002 </T> </P>
25 <P> <F> 0x03 </F> <T> 0x0003 </T> </P>
26 <P> <F> 0x04 </F> <T> 0x0004 </T> </P>

```

Filename: Ix200b2.nlb | Category: Character Set | Name: WE8MSWIN1252 | Status: Viewing



Debugging

General 1/2

- Check data with a Unicode client like SQL Developer
- Check NLS settings on different levels

```
SQL> SELECT * FROM nls_database_parameters;  
SQL> SELECT * FROM nls_instance_parameters;  
SQL> SELECT * FROM nls_session_parameters; -- Current session
```

- Client character set is visible in the V\$SESSION_CONNECT_INFO view
 - Available since 11g Release 1

```
SQL> SELECT client_charset, network_service_banner  
       FROM v$session_connect_info  
       WHERE sid = SYS_CONTEXT('USERENV', 'SID');
```

General 2/2

- Get NLS_LANG value set in Registry only

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

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

```
SP2-0310: Unable to open file ".[GERMAN_GERMANY.WE8MSWIN1252]..sql"
```

Character Set Health Check

- Use **CSSCAN** utility to search for **incompatible characters** (< 12c)
 - Requires a schema called CSMIG to store internal data

```
SQL> @?/rdbms/admin/csminst.sql
```

- Run CSSCAN as SYSDBA
 - Set FROMCHAR and TOCHAR to the same value

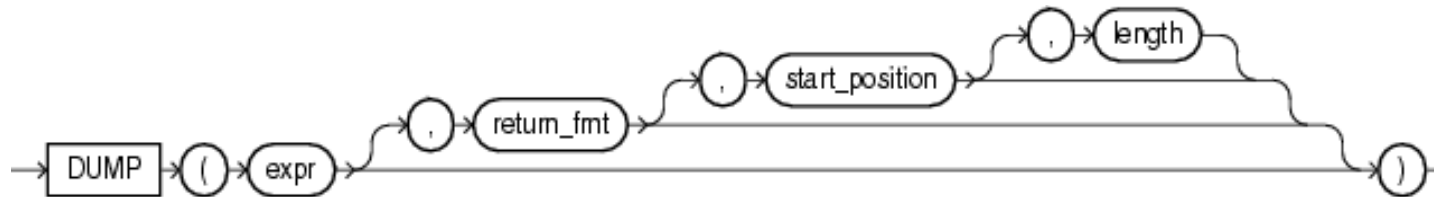
```
$> csscan FULL=Y FROMCHAR=WE8ISO8859P1 TOCHAR=WE8ISO8859P1  
      ARRAY=1024000 LOG=cs_healthcheck CAPTURE=Y PROCESS=4
```

- Check cs_healthceck.txt for "Convertible" or "Lossy" entries
 - If no entries exist, the current data is fine



DUMP Function

- Returns datatype code, length (bytes) and internal representation
- Supports CHAR, VARCHAR2, NCHAR, NVARCHAR2, DATE, NUMBER
 - Cannot be used on CLOB or NCLOB columns
- Useful to check if a specific character is supported
 - Requires the usage of a Unicode client like SQL Developer
 - If an insert in a table succeeded the Character Set supports the character



```
SQL> SELECT DUMP(col1, 10) FROM tab1;
```

```
SQL> SELECT DUMP('Text with € sign', 10) FROM dual;
```

```
Typ=96 Len=18: 84,101,120,116,32,119,105,116,104,32,226,130,172,32,115,105,103,110
```

Migration

Superset vs. Subset

- Character set can be a Superset or a Subset of the new character set

Tool	NLS_LANG
Superset	All characters from character set A are included in character set B.
Binary Superset	Same as Superset, but the binary representations of the character are the same in character set A and B
Subset	Only a subset of the characters in character set A are included in character set B.
Binary Subset	Same as Subset, but the binary representations of the character are the same in character set A and B

Database Migration Assistant for Unicode (DMU) 1/2

- **Migrates databases from legacy character sets to Unicode**
 - A GUI guides through all steps of the migration process
- Validation mode for existing Unicode databases
 - Searches for not correctly encoded characters
- Current version is 19.1 released in August 2019
 - Documentation <https://docs.oracle.com/en/database/oracle/dmu/19.1/duman>)
- Since version 2.1.1 only **database versions 11.2.0.4 and higher** are supported
 - For older database versions, you must use version 2.1.0 (Patch 21138450)



Migration of the CDB\$ROOT container is not supported.



Database Migration Assistant for Unicode (DMU) 2/2

- Only supported tool for character set migration in 12c and higher
 - Replaces CSSCAN and CSALTER
 - Can only migrate databases to Unicode
- Download from Oracle Technology Network
 - Link: <https://www.oracle.com/database/technologies/appdev/oracle-database-migration-assistant.html>
- Requires the installation of the Java Development Kit 8

Requirements & Limitations 1/2

- Oracle Database must be 10.2.0.4 (11.2.0.4) or higher
 - Installed XML Database feature
 - DBMS_DUMA_INTERNAL package

```
SQL> @?/rdbms/admin/prvtdumi.plb
```

- ASCII-based character set
- Oracle Database Vault must be disabled
- Data Dictionary must contain ASCII characters only
- No Flashback archives can exist

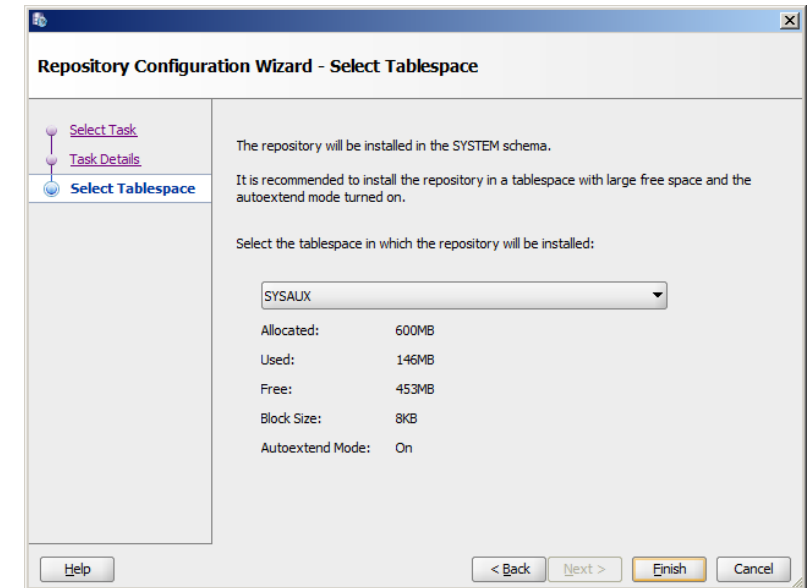
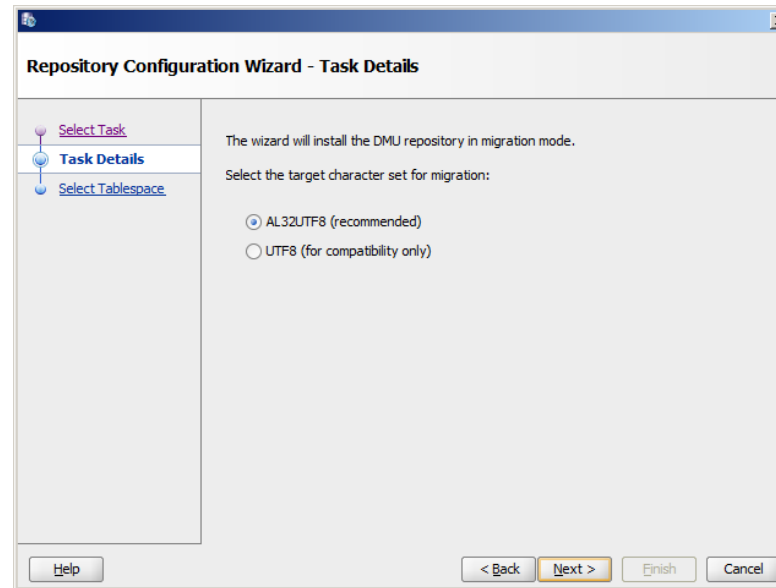
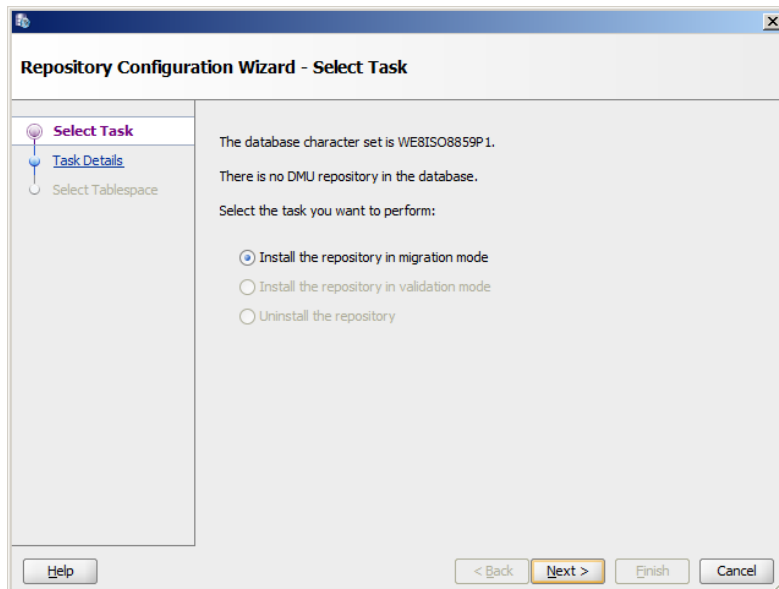


Requirements & Limitations 1/2

- Data that requires conversions ...
 - ... cannot reside in read-only or offline tablespaces
 - ... can be present in tables in the recycle bin
 - ... can be present in reference partitioning key column
- MOS Note: [The Database Migration Assistant for Unicode \(DMU\) Tool \(Doc ID 1272374.1\)](#)

Repository Configuration Wizard

- DMU creates a repository in the target database during the first connection



In a CDB environment, you can only use DMU on PDB level.



Repository Configuration Wizard

- Follow the migration workflow of the DMU to convert your database to Unicode

The screenshot displays the Database Migration Assistant for Unicode (DMU) application. The main window shows the 'Conversion Steps' section with the following tasks:

- Execute Pre-Conversion Tasks
- Convert Application Tables
- Convert System Tables
- Alter Database Character Set to AL32UTF8
- Execute Post-Conversion Tasks

The 'Step Details' section shows the SQL statements generated for the 'Execute Pre-Conversion Tasks' step:

```
ALTER SYSTEM ENABLE RESTRICTED SESSION
ALTER SYSTEM SET JOB_QUEUE_PROCESSES = 0 SCOPE=MEMORY
ALTER SYSTEM SET AQ_TM_PROCESSES = 0 SCOPE=MEMORY
ALTER TRIGGER "SYS"."AW_TRUNC_TRG" DISABLE
ALTER TRIGGER "SYS"."AW_REN_TRG" DISABLE
ALTER TRIGGER "SYS"."AW_DROP_TRG" DISABLE
ALTER TRIGGER "SYS"."XDB_FI_TRIG" DISABLE
```

A 'Database Conversion' dialog box is open in the foreground, displaying the following message:

Database Conversion

The database has been successfully migrated to Unicode.
Database conversion elapsed time: 00:00:26.202.
Please disconnect DMU and restart the database.

OK

Further Information

- **Database Globalization Support Guide 19c**
<https://docs.oracle.com/en/database/oracle/oracle-database/19/nlspg/index.html>
- **Database Globalization Support Guide 23ai**
<https://docs.oracle.com/en/database/oracle/oracle-database/23/nlspg/index.html>
- **My Oracle Support Notes - <https://support.oracle.com>**
 - 264157.1 The Correct NLS_LANG Setting in Unix Environments
 - 179133.1 The Correct NLS_LANG in a Microsoft Windows Environment
 - 179133.1 NLS_LANG Explained (How does Client-Server Character Conversion Work?)
 - 241047.1 The Priority of NLS Parameters Explained (Where To Define NLS Parameters)





Thank You