Unicode Migration: Chance and Challenge

Johannes Ahrends

CarajanDB

Agenda

- Me and my company
- Character conversion
- Database conversion
- Migration using Export / Import
- Migration planning
- Unicode Migration Best Practice
- Minimal Downtime Migration
- Oracle 12c

… about me

- **Oracle Professional since 1992**
  - 1992: Presales at Oracle
  - 1999: Project Manager at Herrmann & Lenz Services
  - 2005: Technical Director ADM Presales at Quest Software
  - 2011: Managing Director CarajanDB GmbH
- **2011 → Oracle ACE Award**
- **Author of some well known books (in Germany):**
  - “Oracle9i für den DBA”, “Oracle10g für den DBA”, “Oracle 11g Release 2 für den DBA”
- **Responsible for the special interest group database at DOAG (German Oracle Users Group)**
- **Hobbies:**
  - Kiting and esp. Indoor Kiting
  - Motorbike

… and my company

- **Oracle professionals with more than 25 years of experience**
- **Located near to Cologne**
- **Specialized in**
  - Oracle Database Administration
  - High Availability (RAC, Data Guard, Failsafe, etc)
  - Oracle Standard Edition
  - Oracle Migrations (HW, Unicode, Consolidation, Standard Edition)
  - Replication (Goldengate, SharePlex, Dbvisit)
  - Performance Optimization
- **Trainings and Workshops (Oracle, Toad)**

ORACLE OPEN WORLD

Character Conversion

OTN Discussion

DB Characterset Issue

This question is **Not Answered.**

I am using 12c database and DB characterset is AL32UTF8.
I am getting data in WE8MSWIN1252 from source file but I want to load in AL32UTF8 in my database. Is it possible?

Note: I dont want to change my db characterset, only on runtime i want to change the format.

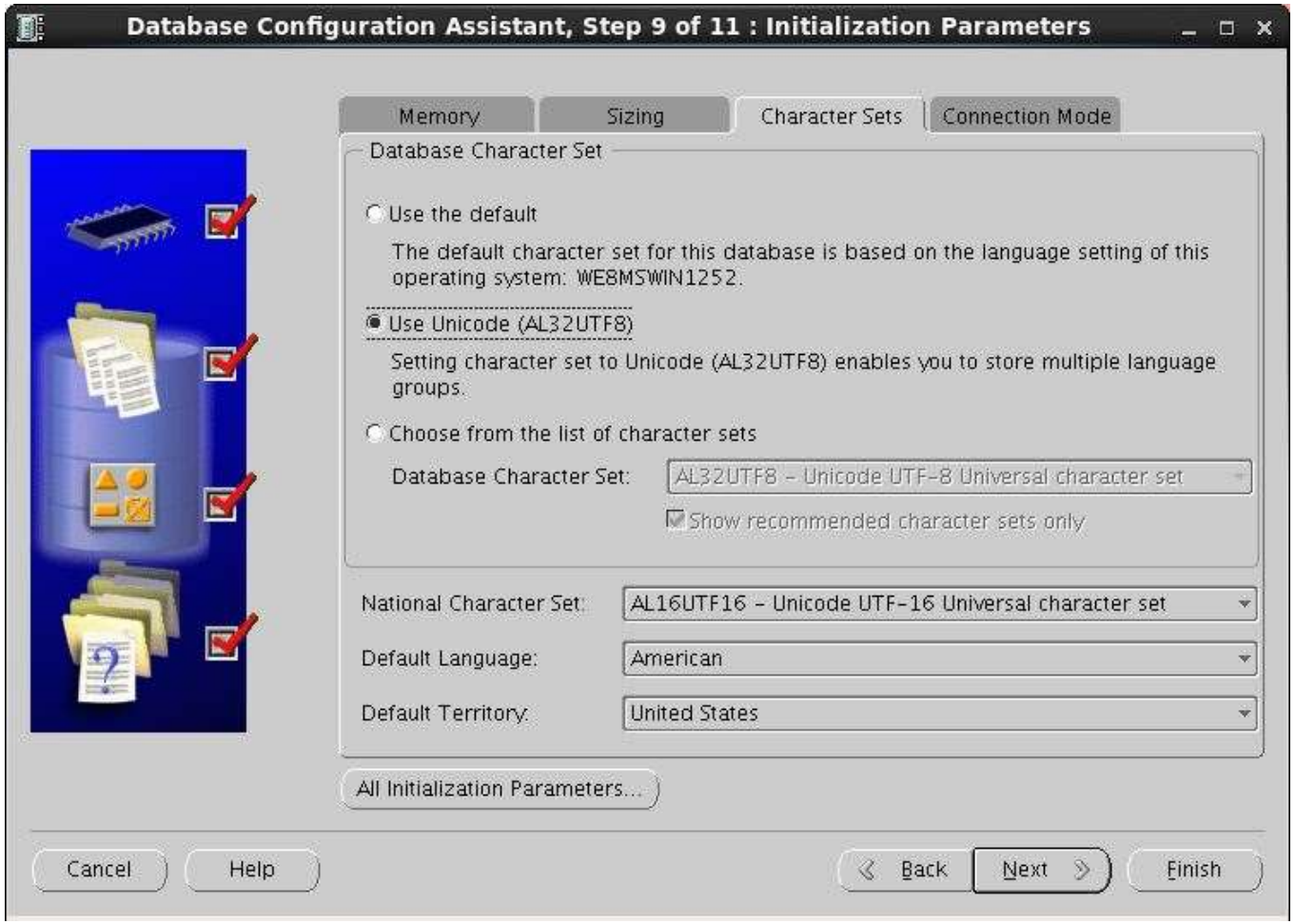
Thanks,

93 Views Tags:

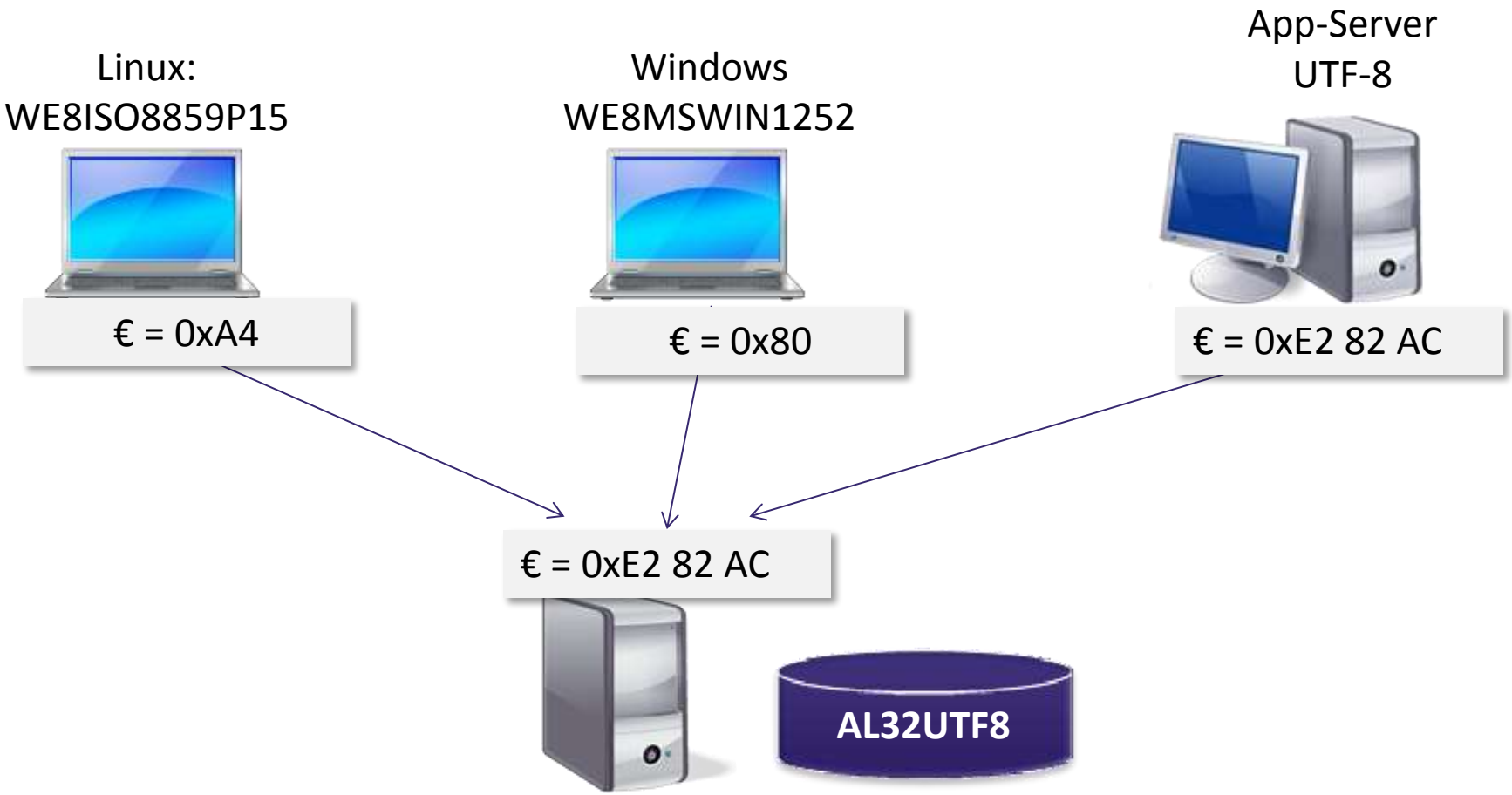
Character Conversion

Character set	A	Ö	ä	ß	€
US7ASCII	0x41	N/A	N/A	N/A	N/A
WE8PC850	0x41	0x99	0x84	0xE1	N/A
WE8ISO8859P1	0x41	0xD6	0xE4	0xDF	N/A
WE8ISO8859P15	0x41	0xD6	0xE4	0xDF	0xA4
WE8MSWIN1252	0x41	0xD6	0xE4	0xDF	0x80
AL32UTF8	0x41	0xC3 96	0xC3 A4	0xC3 9F	0xE2 82 AC

Choose the database Characterset



Character Conversion



Example (1)

- **Client: DOS-Box:**

```
SQL> insert into umlaute values (1,'Test ü ß');

1 row created.

SQL> commit;

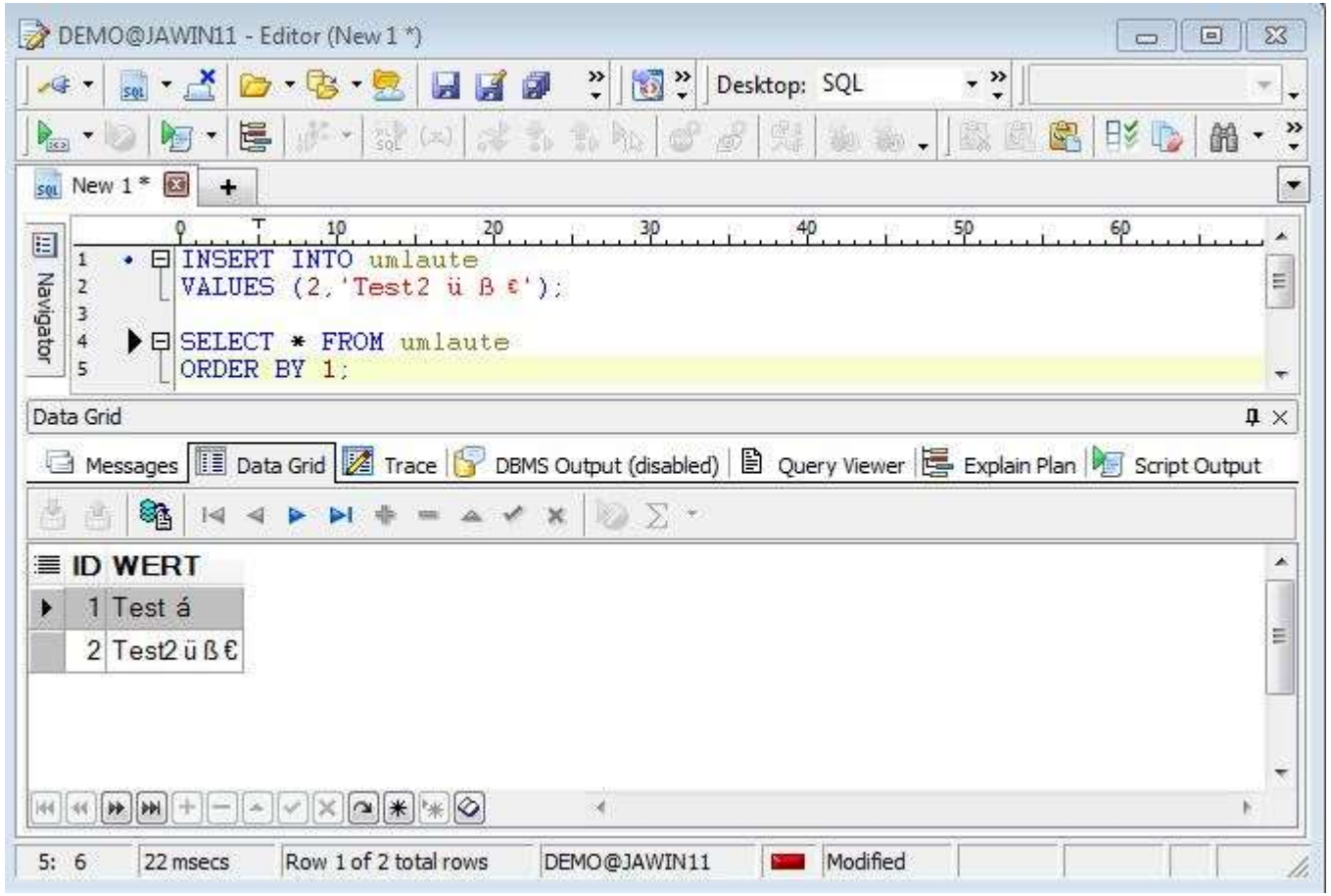
Commit complete.

SQL> select * from umlaute;

        ID VALUE
---------- --------------------
         1 Test ü ß
```

Example (2)

- Windows Tool (Toad)



Who's correct?

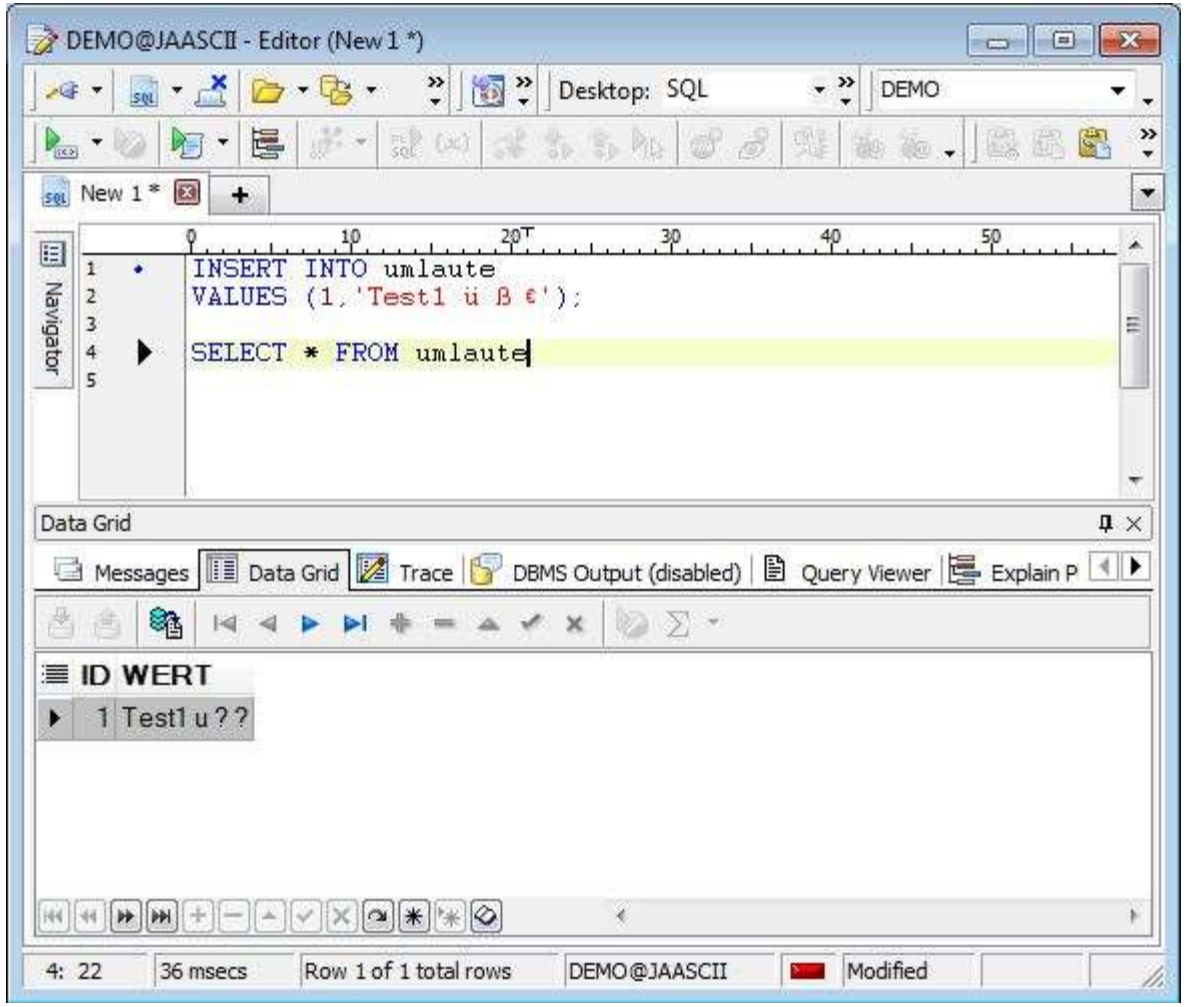
```
SELECT s.SID, s.username, s.PROGRAM,
           i.client_charset, i.client_version
  FROM v$session s, v$session_connect_info i
 WHERE i.SID=s.SID
   AND s.PROGRAM IN ('Toad.exe','sqlplus.exe')
   AND s.username='DEMO'
   AND i.network_service_banner LIKE 'Windows NT%';

   SID USERNAME   PROGRAM          CLIENT_CHARSET  CLIENT_VERSION
------ ---------- ---------------- --------------- --------------
   331 DEMO       Toad.exe         UTF16           11.2.0.3.0
   482 DEMO       Toad.exe         UTF16           11.2.0.3.0
   484 DEMO       sqlplus.exe      WE8MSWIN1252    11.2.0.3.0
```

German Umlaut in US7ASCII

- US7ASCII doesn‘t allow 8 bit characters !?
- What if I‘m telling the database: „here is a character and you don‘t have to convert it!“?

Toad



… and Putty?

```
insert into umlaute
values (2,'Test2 ü ß Ö');

1 row created.

SQL> commit;

Commit complete.

SQL> select * from umlaute
  2  ;

        ID WERT
---------- --------------------
         1 Test1 u ? ?
         2 Test2 ü ß Ö
```

Why?

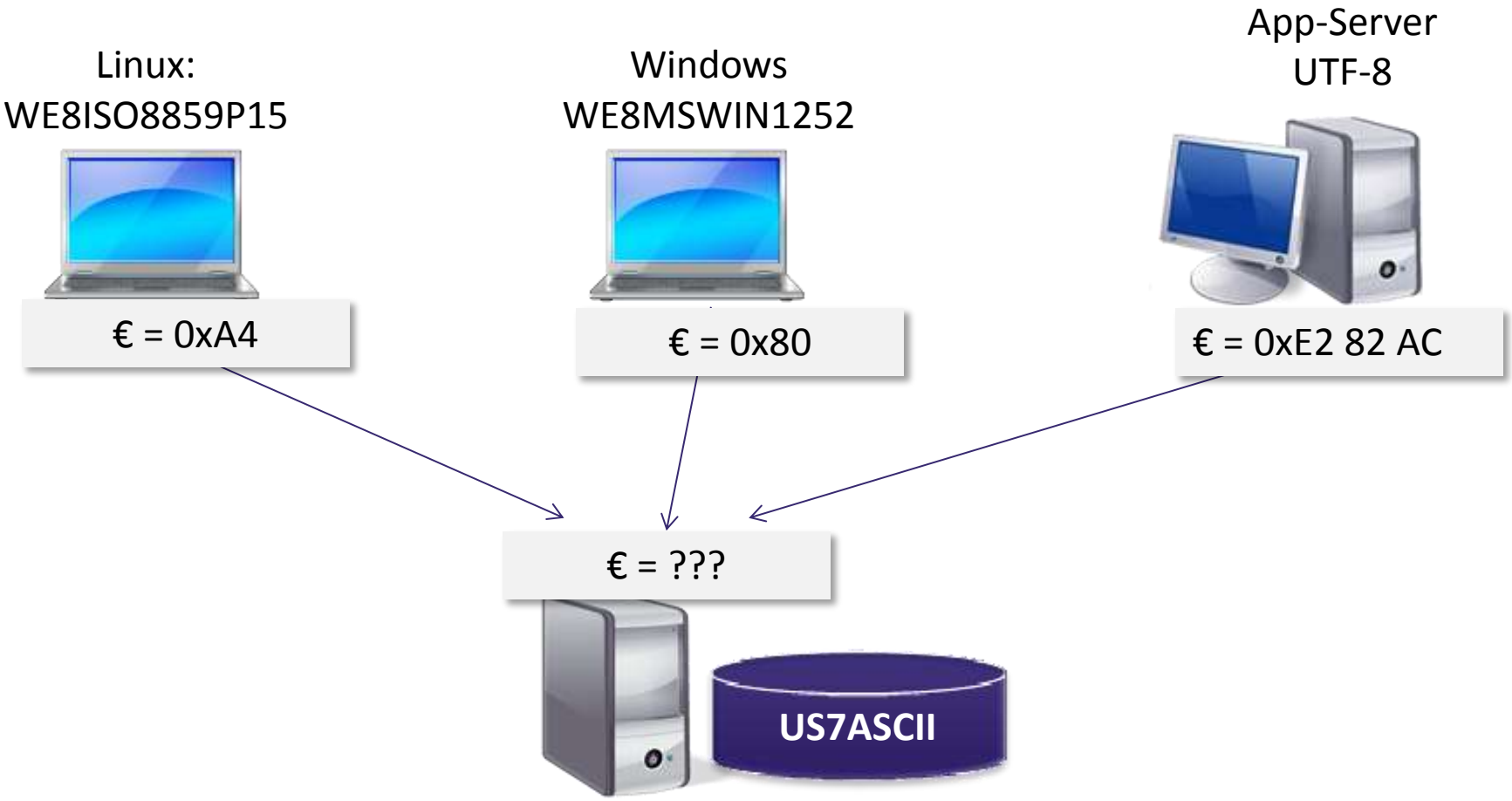
```
$ export NLS_LANG=GERMAN_GERMANY.US7ASCII
```

- **With my environment variable I'm telling the database → this client is using a US7ASCII character set.**
- **If the database is US7ASCII as well there is no need for conversion!**

This is real life

- **Database for healthy cooking recipes**
  - Recipes for clients
  - VARCHAR2(4000) for the text of the recipes
- **Oracle 11g 11.2.0.1 on Oracle Linux**
- **Database US7ASCII**
- **Client 1 = Java Application on Linux → Unicode**
  - But NLS_LANG=GERMAN_GERMANY.US7ASCII
- **Client 2 = Java Application on Windows → WE8MSWIN1252**
  - But NLS_LANG=GERMAN_GERMANY.US7ASCII

A real life example



ORACLE OPEN WORLD

Database Conversion

Database Conversion routines

- CSALTER
- ALTER DATABASE CHARACTER SET …
- DMU
- (Data Pump) Export / Import

ORACLE OPEN WORLD

Migration using Export / Import

Export the orginal Database

- Don‘t use a full export
- Preferable method: Schema export!

```
DUMPFILE="export.dmp"
LOGFILE="exp_export.log"
DIRECTORY=DATA_PUMP_DIR
FLASHBACK_TIME="TO_TIMESTAMP('2012-10-13 09:00:00','YYYY-MM-DD HH24:MI:SS')"
COMPRESSION=NONE
CONTENT=ALL
SCHEMAS=('BASIS', 'DEMO')
```

- The reason: you clean up your database and get rid of stuff you haven‘t used for year (e.g. scott)

Import data into new database

- Copy the dump file to the target
- Or: Use a database link

```
DUMPFILE="export.dmp"
LOGFILE="imp_export.log"
DIRECTORY=DATA_PUMP_DIR
STREAMS_CONFIGURATION=n
TABLE_EXISTS_ACTION=SKIP
SKIP_UNUSABLE_INDEXES=y
CONTENT=ALL
PARTITION_OPTIONS=none
```

Successful?

```
Processing object type SCHEMA_EXPORT/SEQUENCE/SEQUENCE
Processing object type SCHEMA_EXPORT/TABLE/TABLE
ORA-39083: Object type TABLE:"DEMO"."AUFTRAEGE" failed to create with error:
ORA-00439: feature not enabled: Partitioning
Failing sql is:
CREATE TABLE "DEMO"."AUFTRAEGE" ("AUFID" NUMBER(10,0) NOT NULL ENABLE, "PERSID" NUMBER(10,0) NOT NULL ENABLE, "AUFDATUM"
DATE, "LIEFERDATUM" DATE, "AUFSTATUS" CHAR(1 BYTE) NOT NULL ENABLE) PCTFREE 10 PCTUSED 0 INITRANS 1 MAXTRANS 255 NOCOMPRESS
LOGGING  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 BUFFER_POOL DEFAULT FLASH_CACHE
DEF
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "DEMO"."PRODUKTE"                           [illegible]5 MB     374 rows
. . imported "DEMO"."POSITIONEN"                         714.[illegible] KB   29780 rows
. . imported "BASIS"."NACHNAMEN"                         177.2 KB   11552 rows
. . imported "BASIS"."ORTE"                              [illegible]0 KB    6104 rows
. . imported "BASIS"."VORNAMEN"                          178.2 KB    8856 rows
. . imported "DEMO"."ADRESSEN"                           106.7 KB    2000 rows
. . imported "DEMO"."PERSONEN"                           39.68 KB    1000 rows
. . imported "DEMO"."TELEFONE"                           108.1 KB    4000 rows
. . imported "BASIS"."STRASSEN"                          20.24 KB     640 rows
. . imported "DEMO"."BUNDESLAENDER"                      5.531 KB      16 rows
. . imported "DEMO"."PRODUKTGRUPPEN"                     10.61 KB     156 rows
ORA-02374: conversion error loading table "DEMO"."STATUS"
ORA-12899: value too large for column STATUSID (actual: 2, maximum: 1)

ORA-02372: data for row: STATUSID : 0X'C4'


. . imported "DEMO"."STATUS"                             5.835 KB       5 out of 6 rows
```

NO!

ORACLE OPEN WORLD

How to plan a migration

Check the current length semantic

- **Table definition**
  - ```
    CREATE TABLE status (
        statusid     CHAR(1),
        beschreibung VARCHAR2(50));
    ```
- **What does „1“ or „50“ stands for?**
  - CHAR(1) = 1 BYTE or 1 CHAR?
  - VARCHAR2(50) = 50 BYTE or 50 Characters?
- **Preferrable:**
  - ```
    CREATE TABLE status (
        statusid     CHAR(1 CHAR),
        beschreibung VARCHAR2(50 CHAR));
    ```
- **Or as an alternative:**
  - ```
    ALTER SESSION SET NLS_LENGTH_SEMANTICS='CHAR';
    ```
    ```

Using Length semantics or not

- **Using length semantic explicitly**

```
CREATE TABLE STATUS (
  STATUSID           CHAR(1 BYTE)    NOT NULL,
  KURZBESCHREIBUNG   VARCHAR2(10 BYTE),
  BESCHREIBUNG       VARCHAR2(255 BYTE)
) TABLESPACE USERS;
```

- **Using length semantic implicitly**

```
CREATE TABLE STATUS (
  STATUSID           CHAR(1)    NOT NULL,
  KURZBESCHREIBUNG   VARCHAR2(10),
  BESCHREIBUNG       VARCHAR2(255)
) TABLESPACE USERS;
```

Limitation

- **Maximum length of data types up to 11g:**
  - CHAR = 2000 **BYTE**
  - VARCHAR2 = 4000 **BYTE**
- **Maximum lenght of object names = 30 BYTE**
- **Total size of indexed columns approx. 75% of a block → 6k with 8k block size**
- **But what if your Unicode conversion requires more byte / characters?**
  - VARCHAR2(4000) → CLOB
  - Be careful because this requires additional space

Why CLOB is a problem with Unicode Databases

```
SELECT SUM (bytes) / 1024 / 1024 MByte
  FROM dba_segments
 WHERE owner = 'DEMO'
   AND segment_name NOT LIKE 'BIN$%';
```

```
Single Byte Database (e.g. WE8ISO8859P15)

     MBYTE
----------
   187,625
```

```
Unicode Database (AL32UTF8):

     MBYTE
----------
  356,1875
```

Why?

```
SELECT segment_name, tablespace_name, bytes
  FROM dba_segments
 WHERE owner = 'DEMO' AND segment_name NOT LIKE 'BIN$%'
 ORDER BY bytes DESC;
```

Single Byte Database:

```
SEGMENT_NAME                   TABLESPACE_NAME                        BYTES
------------------------------ ------------------------------ ----------
SYS_LOB0000051856C00006$$      USERS                           191889408
POSITIONEN                     USERS                              983040
…
```

Unicode Database:

```
SEGMENT_NAME                   TABLESPACE_NAME                        BYTES
------------------------------ ------------------------------ ----------
SYS_LOB0000014229C00006$$      USERS                           369098752
POSITIONEN                     USERS                              983040
…
```

Conversion from VARCHAR2 → CLOB

„Oracle recommends using Unicode for all new system deployments."

"Data in CLOB columns is stored in a format that is compatible with UCS-2 when the database character set is multi byte, such as UTF8 or AL32UTF8. This means that the storage space required for an English document doubles when the data is converted."

ORACLE OPEN WORLD

Unicode Migration Best Practice

Best Practice

- **Export object definition without length semantics**
  - e.g. Toad → Generate Schema Script
- **Change the definition is necessary**
  - Change from VARCHAR2 → CLOB, if 4000 characters is not sufficient
  - Change from NVARCHAR2 → VARCHAR2
  - Change object names (hopefully not required!)
- **Create schema objects on the target with length semantic 'CHAR'**
- **Disable Foreign Key Constraints**
- **Export Data (CONTENT=DATA_ONLY)**
- **Import Data (TABLE_EXISTS_ACTION=APPEND)**
- **Enable Foreign Key Constraints with NOVALIDATE**

… and what about the US7ASCII DB?

- **Exporting the database as it is will remove all umlaute.**
- **Workaround**
  - Start the database in restricted mode:

```
SQL> STARTUP RESTRICT
```

  - Switch the database character set in the data dictionary:

```
SQL> ALTER DATABASE
     CHARACTERSET INTERNAL_USE WE8ISO8859P15;
```

  - Export schemas which were using this character set
  - Switch the database character set again:

```
SQL> ALTER DATABASE
     CHARACTERSET INTERNAL_USE AL32UTF8;
```

  - Export schemas which were using this character set

ORACLE OPEN WORLD

Minimal Downtime Migration

Minimal Downtime Migration

- **Request a „NO DDL“ Period**
  - There should be no structural changes (DDL) during the migration
- **Export and import the schema objects like with any „normal“ migration**
- **Create a replication environment using:**
  - DELL SharePlex for Oracle
  - Oracle Goldengate
  - Dbvisit Replicate

Start the replication

- Capture / Mine all changes with the replication solution
- No Apply / Post on the target yet!!!
- Data Pump Export / Import with Flashback SCN
- Reconcile of the Apply / Post Process to this SCN
- Start Apply / Post Process
- Wait …
- When databases are in sync ➔ Reverse the replication for failback

Which replication solution to use

- **GoldenGate**
  - Entire database replication
  - Nearly no restrictions on data types or DDL
- **Dbvisit Replicat**
  - Cheap
  - Easy setup and maintenance
- **DELL SharePlex for Oracle:**
  - Compare and Repair included
  - Takes care on open transactions
  - Data Type conversion (e.g. NVARCHAR2 → VARCHAR2)

ORACLE OPEN WORLD

… and Oracle 12c

MAX_STRING_SIZE

- **MAX_STRING_SIZE=STANDARD**
  - VARCHAR2(4000 BYTE)
  - NVARCHAR2(4000 BYTE)
  - RAW (2000 BYTE)
- **MAX_STRING_SIZE=EXTENDED**
  - VARCHAR2(32767 BYTE)
  - NVARCHAR2(32767 BYTE)
  - RAW (32767 BYTE)

MAX_STRING_SIZE

- **STANDARD → EXTENDED** **Okay**
- **EXTENDED → STANDARD** **No!**

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP UPGRADE -- CDBsh
SQL> ALTER DATABASE OPEN MIGRATE; -- PDB
SQL> ALTER SYSTEM SET MAX_STRING_SIZE=EXTENDED;
SQL> @?/rdbms/admin/utl32k.sql
```

Restriction

- **VARCHAR2 > 4000 = CLOB**
- **Default CLOB UCS-2 Character set**
- **VARCHAR2 converted to CLOB → „normal“ Database Character set(z.B. AL32UTF8)**
- **But be aware:**
  - CLOB is not included in the size of the table segment

Conclusion

- **Critical parameters for Unicode conversion:**
  - Entirely used VARCHAR2(4000) columns
  - „Dirty characters“ (no character conversion)
  - CLOB length
- **Preferred practice: Data Pump Export / Import**
  - Pre create objects with length semantic „Char“
  - Be aware: precreation of objects has an impact on the import performance (up to 30%)
  - Single schema export and Import
- **Use Replication for Minimal Downtime Migration**
  - Downtime for application switch and some clean ups (e.g. sequences, triggers)
  - Failback

EMEA Oracle User Group

CARAJANDB

I am a proud member of an EMEA Oracle User Group

Are you a member yet?

Meet us at
south upper lobby of Moscone South

www.iouc.org

Questions?

Johannes Ahrends

www.carajandb.com

Johannes.ahrends@carajandb.com