



SUNGARD HIGHER EDUCATION

SUMMIT 2008

Banner 8 Character Set Conversion

Presented by: Colin Abbott

McGill University

Scott Harden

University of Illinois

April 15, 2008

Course ID S-0202

Session Rules of Etiquette

- Please turn off your cell phone/pager
- If you must leave the session early, please do so as discreetly as possible
- Please avoid side conversation during the session

Thank you for your cooperation!



Introduction

- **Why are we changing our character set?**
- **Defining terminology important in understanding character set conversion**
- **Compare and contrast the pros and cons of the three conversion methods:**
 - **The csalter method**
 - **The exp/imp method**
 - **The datapump method**
- **Questions and Answers**



Goal

- To share what we learned in converting the Banner database to AL32UTF8 for the Banner 8.0 Beta test
- After attending this session you should have the tools necessary to determine the right conversion method for your Banner database, as well as the steps to perform the conversion.



Additional Materials

- We have created a supplemental white paper on the conversion. It includes more details and sample output from each step along with some scripts to resolve convertible data in the data dictionary.
- We have included scripts for the “Datapump with the paresh process”
- We suggest when you tackle the conversion to use the white paper in conjunction with this presentation, as well as the recommended Metalink notes.





*Performance Matters!
Ron O'Connell, theater professor
Founder, All the World's a Stage,
a showcase for original student work.*

SUNGARD HIGHER EDUCATION

SUMMIT 2008

Why Are We Changing Our Character Set?

Defining terminology important in the conversion process.

Why Are We Changing Our Character Set?

- Required for Banner 8.0 to support the globalization of Banner
- Starting with Oracle 11g, Oracle will start to consolidate supported character sets and slowly start to move away from single byte to Unicode (Metalink note 306411.1)
- Unicode character sets are becoming the standard





*Performance Matters!
Ron O'Connell, theater professor
Founder, All the World's a Stage,
a showcase for original student work.*

SUNGARD HIGHER EDUCATION

SUMMIT 2008

Terminology

Defining terminology important in the conversion process.

Terminology

- **What is my character set?**
- **Single-byte vs. Multi-byte**
- **NLS_LENGTH_SEMANTICS**
- **Lossy Data**
- **Truncation Data**
- **Convertible Data**



What is my character set?

- **select value from NLS_DATABASE_PARAMETERS where parameter='NLS_CHARACTERSET';**
- **In the alert log on instance startup**
- **Based on a survey on boracle most Banner sites are WE8ISO8859P1 a few are US7ASCII or WE8MSWIN1252**



Single-byte vs. Multi-Byte

- The following query in a AL32UTF8 database
`select ascii('é'), ascii('ê') from dual;`
Produced
50089 50090
- The following query in an WE8ISO8859P1 database
`select ascii('é'), ascii('ê') from dual;`
Produced
233 234
- This illustrates that accented characters have different code points in the different character sets



Single-byte vs. Multi-Byte

- The following query in WE8ISO8859P1
select dump ('abcéeê') from dual;
Produced
Typ=96 Len=5: 97,98,99,233,234
- The following query in AL32UTF8
select dump ('abcéeê') from dual;
Produced
Typ=96 Len=7: 97,98,99,195,169,195,170
- Note: the length of 7 makes sense since “abc”= 3 bytes
and the two accented characters are 2 bytes each for 4,
which makes a total of 7



NLS_LENGTH_SEMANTICS

- Is an `init.ora` parameter
- Default value is `BYTE`
- you can define a column in these ways
 - `Spriden_first_name varchar2(60 BYTE)`
 - `Spriden_first_name varchar2(60 CHAR)`
 - `Spriden_first_name varchar2(60)`
- When you don't explicitly specify `BYTE` or `CHAR` the value of `NLS_LENGTH_SEMANTICS` is used
- Is a SunGard requirement for Banner 8.0



Lossy Data (a term from the csscan reports)

- **Lossy data indicates that a code point in your source database can not be translated to a code point in your destination database.**
- **This usually occurs when the destination database's character set is not a superset of the source character set**
- **We saw this the most in our WE8ISO8859P1 database because users saved special windows characters. i.e. bullets and smart quotes**
- **You can eliminate Lossy data by taking an intermediate step of converting your character set from WE8ISO8859P1 to WE8MSWIN1252**



Truncation Data (a term from the csscan reports)

- Truncation Data is data that when expanded from single byte to multi byte would no longer fit in the column length and therefore the data would be truncated on import.
- Truncation data is resolved by setting `NLS_LENGTH_SEMANTICS = CHAR`



Convertible Data (a term from the csscan reports)

- **Convertible data is data that Oracle is capable of converting from one character set to another. To accomplish conversion this data needs to be unloaded and then reloaded.**



Changeless Data (a term from the csscan reports)

- Data that will keep the same code points in the converted character set





*Performance Matters!
Ron O'Connell, theater professor
Founder, All the World's a Stage,
a showcase for original student work.*

SUNGARD HIGHER EDUCATION

SUMMIT 2008

Compare the 3 Conversion Methods

Compare the 3 conversion methods

- **csalter**
 - Good if you have a small amount of convertible data
 - Good in that you don't need to create a new database
 - Unfortunately it is very messy and error prone and requires meticulous attention to detail.
- **exp/imp**
 - Good if exp/imp is your normal Oracle upgrade method
 - Good if you have a small database < 100G
 - Requires a new database to be created
- **DataPump method**
 - Fastest method for large databases
 - Good if you have a lot of convertible data
 - Requires a new database to be created





*Performance Matters!
Ron O'Connell, theater professor
Founder, All the World's a Stage,
a showcase for original student work.*

SUNGARD HIGHER EDUCATION

SUMMIT 2008

csalter Method

Focus on changing from WE8ISO8859P1 to AL32UTF8 mostly
based on concepts from Metalink note 260192.1.

Overview – csscan

- 25 step process documented in detail in our whitepaper and Metalink note 260192.1
- At McGill on a Sun-Fire-V890 500G database total conversion process took 71 hours!!
- Install csscan using script `$ORACLE_HOME/rdbms/admin/csminst.sql`
- csscan is a command line utility that will scan your character set and report any problems with your conversion.
- csscan will indicate how much your database will expand by (McGill's expanded by 851.60M)
- csscan must be run for csalter to work and the report must indicate that it is safe to convert.
- csscan generates 3 files and populates tables in the csmig schema.



Overview – csscan

- **Basic concept is to:**
 1. Run csscan
 2. Resolve any lossy characters
 3. Have no truncation issues
 4. Save/Export any object that has convertible data and truncate/drop it from the database
 5. If csscan produces errors goto 1.
 6. Convert character set with csalter
 7. Import/recreate any object that was dropped



Overview – csscan reports

- **Reports generated**
 - .out file – screen capture of the scan process
 - .txt file – Summary of the scan process
 - .err file – detailed list of table rows that have a conversion problem
- **A successful scan the .txt file will display**
 - All character type data in the data dictionary remain the same in the new character set
 - All character type application data remain the same in the new character set
 - The data dictionary can be safely migrated using the CSALTER script



Overview - running csalter

- **Set parameters (note original value to set them back later)**
 - **CLUSTER_DATABASE=FALSE**
 - **PARALLEL_SERVER =FALSE**
 - **JOB_QUEUE_PROCESSES=0**
 - **AQ_TM_PROCESSES=0**
- **Shutdown immediate**
- **Startup restrict**
- **Spool switch.log**
- **@@?/rdbms/admin/csalter.plb**
- **Script will prompt you for confirmation, enter Y**
- **Reset init.ora parameters**
- **Shutdown / startup**
- **Verify v\$NLS_PARAMETERS**
- **Note: You may want to do a cold backup before and after the conversion.**



Tips!

- Our lossy data is resolved in WE8ISO8859P1 by converting first to WE8MSWIN1252.
- Truncation data is resolved by changing the character semantics from BYTE to CHAR for every char and varchar2 column (Sungard FAQ 1-311A13 - Metalink Note 313175.1).
- Scripts can be created off the csmig tables to clean up the convertible data.
- Run csscan with FROMCHAR and TOCHAR both set to your current character set to identify problems before you start.
- Running some csscans is a good learning opportunity.
- If you run csscan and have a lot of convertible data then give up on this method!
- BACKUPS, BACKUPS, BACKUPS!!



Problems we encountered

- The script to convert to CHAR semantics fails on partitioned tables there is a workaround involving partition exchange (see Metalink Note:330964.1)
- Oracle 10g Advisors were broken after conversion needed to run “execute dbms_advisor.setup_repository;”
- Exports need to be done as sys or schema owners need to be exempt from FGAC policies.
- When Importing convertible data into AL32UTF8 NLS_LANG needs to be set to the character matching the export.
- NLS_LENGTH_SEMANTICS=CHAR needs to be set in the forms environment *before* you compile Banner 8 forms.





*Performance Matters!
Ron O'Connell, theater professor
Founder, All the World's a Stage,
a showcase for original student work.*

SUNGARD HIGHER EDUCATION

SUMMIT 2008

Export/Import Method

Good for sites with small databases whose normal upgrade method is exp/imp.

Exp/Imp Method

- Create a new database with AL32UTF8 character set
- Set NLS_LENGTH_SEMANTICS = CHAR in the new database
- Perform full database export
- Import schemas with no data
- Run script to convert char and varchar2 columns from BYTE to CHAR semantics (Sungard FAQ 1-311A13 - Metalink Note 313175.1)
- Import data
- Note: NLS_LANG should be set to the original character set for export and import, this will force the conversion.
- Note: We did not test this method as our databases are too large.





*Performance Matters!
Ron O'Connell, theater professor
Founder, All the World's a Stage,
a showcase for original student work.*

SUNGARD HIGHER EDUCATION

SUMMIT 2008

DataPump Method

*A good method for large database with a large amount of
convertible data*

DataPump Method

- **Requires Oracle patch 5874989 as documented in Oracle Metalink document 119119.1**
- **Also requires patch 5875568 for Datapump network import**
- **Current results: ~12 hours**
 - **428 GB Database**
 - **Export data only: 2.5 hours**
 - **Export indexes, constraints: 10 minutes**
 - **Import data only: 3.5 hours**
 - **Build indexes and constraints: 5.5 hours**
 - **Export/import stats: 20 minutes**



DataPump Method

- **Create New Database**
 - **Install Oracle 10.2.0.3**
 - **Apply patches 5874989 and 5875568**
 - **Create a new database with AL32UTF8 character set (NLS_CHARACTERSET=AL32UTF8) and CHAR semantics (NLS_LENGTH_SEMANTICS=CHAR)**
 - **Install JVM, Java packages and XMLDB as needed**



DataPump Method

- **Configure Tablespaces**

- **Extend USERS, TEMP, and UNDO tablespaces to UNLIMITED**
- **Create your Banner tablespaces/datafiles**
- **Data will compress as part of datapump export/import but some columns will increase in size, especially CLOBs, so make sure you provide plenty of space or set to unlimited**



DataPump Method

- **Configure target DB Parms for speedier Import**

- **_disable_logging = TRUE**
- **DB_BLOCK_CHECKING=FALSE**
- **DB_BLOCK_CHECKSUM=FALSE**
- **pga_aggregate_target = 300M***
- **sort_area_size = 671089***
- **shared_pool_size = 1000M***
- **sga_max_size = 2000M***
- **parallel_max_servers = 24***

*** Your mileage may vary, adjust accordingly for your environment.**



DataPump Method

- **Export from source**
 - Shutdown source, startup restricted
 - Export Data: run expdp with parfile parm (2.5 hours)
**EXCLUDE=TABLESPACE,STATISTICS,INDEX,CONSTRAINT,
REF_CONSTRAINT**
 - Export other items: run expdp with parfile parm (7 min.)
INCLUDE=INDEX,CONSTRAINT,REF_CONSTRAINT



DataPump Method

- **Extract index and constraint DDL from export file**
 - Extract DDL from export file to SQL file for indexes and constraints: run impdp with parfile parm (15 min.)
SQLFILE=XXXXXXXXX_METADATA.SQL
 - The metadata SQL file will be used later with a parallel execution shell (paresh) to speed up import process because Oracle's impdp single threads these otherwise.



DataPump Method

- **Import data into new Database**
 - Import from first export file that excluded indexes, constraints, and statistics: run imdp with parfile (3.5 hours)



DataPump Method

- **Set up parallel execution shell (paresh) environment**
 - Unzip paresh.zip and copy components to your UNIX system. Includes documentation of all these steps plus a readme file documenting the limitations of datapump and the techniques employed in the paresh script to overcome those limitations.
 - Parse metadata SQL file into components (indexes, PK constraints, FK constraints, etc.) using parse_sqlfile.ksh
 - Run install_paresh.ksh to build external tables on each of the component pieces that were parsed out of the metadata SQL file and loads each external table into a paresh specific table.



DataPump Method

- **Execute paresh steps**
 - Execute paresh2 PK to build primary keys (1.5 hours)
 - Excute paresh2 Indexes to build indexes (4 hours)
 - Execute paresh2 FK to build foreign key constraints (10 min.)
 - Build partitioned indexes using build_fgbtrnd_indexes.sql altered for your partitioned tables



DataPump Method

- **Export/import stats**
 - Export the statistics from your production database and import into the new database following the procedure documented in `import_stats.txt` which uses `dbms_stats.export_schema_stats` and `dbms_stats.import_schema_stats` (20 min.)



DataPump Method

- **Set System Permissions**
 - All of the grants on sys objects are ignored by datapump
- Extract system privileges from source and run into new database (10 min.)
 - GRANT SELECT ON SYS.DBA_POLICIES TO BAN_DEFAULT_M;
 - Grant execute on Dbms_locks to bansso, baninst1;
 - Grant execute on Dbms_pipes to bansso, baninst1;
 - May be others in your environment.



DataPump Method

- **Reset temporary settings**
 - Reset USER, TEMP and UNDO space from UNLIMITED to your normal settings
 - Reset your database parameters set for the import back to your normal settings remembering to include **NLS_LENGTH_SEMANTICS=CHAR**



DataPump Method

- **Compare and Switch to new Database**
 - Compare databases using your favorite tool to ensure all came over as expected
 - Either begin using the new database under it's current name or rename the databases as appropriate



Summary

- Banner 8 requires the AL32UTF8 character set and CHAR semantics in order to be supported.
- Datapump with the parash process is the fastest method for the required conversion for large databases > 100 GB.
- Exp/Imp may be sufficient for databases < 100 GB.
- Using the csscan/csalter method is not recommended due to complexity however it can be an interesting exercise.
- It may be possible to do the character set conversion well ahead of the Banner 8 upgrade but conclusive tests were not available when this presentation was created.



Further Reading

- **Oracle globalization manual**
- **Unicode** - <http://www.unicode.org/>
- **Globalize with Character semantics**
 - <http://www.oracle.com/technology/oramag/oracle/03-mar/o23sql.html>
- **Metalink notes:**
 - **Note:119119.1 - AL32UTF8 / UTF8 (Unicode) Database Character Set Implications**
 - **Note:144808.1 - Examples and limits of BYTE and CHAR semantics usage**



Questions & Answers

Thank You!

Colin Abbot

colin.abbott@mcgill.ca

Scott Harden

sharden@uillinois.edu

Please complete the online class evaluation form

Course ID S-0202

SunGard, the SunGard logo, Banner, Campus Pipeline, Luminis, PowerCAMPUS, Matrix, and Plus are trademarks or registered trademarks of SunGard Data Systems Inc. or its subsidiaries in the U.S. and other countries. Third-party names and marks referenced herein are trademarks or registered trademarks of their respective owners.

© 2008 SunGard. All rights reserved.

