Oracle® Database

Migration Assistant for Unicode Release Notes

Release 2.0

E51606-02

January 2014

ORACLE®

Oracle Database Migration Assistant for Unicode Release Notes, Release 2.0

E51606-02

Primary Author: Paul Lane

Contributor: The Oracle Database 12*c* documentation is dedicated to Mark Townsend, who was an inspiration to all who worked on this release.

Contributor: Sergiusz Wolicki, Weiran Zhang

Contents

Preface

These notes include additional information for users of the Oracle Database Migration Assistant for Unicode.

Audience

These notes are intended for database administrators who perform the following tasks:

- Migrate character sets to Unicode
- Verify that there are no character set data problems in a current Unicode database

To use these notes, you must be familiar with Oracle databases.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see these Oracle resources:

- *Oracle Database Globalization Support Guide*
- *Oracle Database Migration Assistant for Unicode Guide*

1

DMU Release Notes

This document contains important information that was not included in the Oracle Database Migration Assistant for Unicode product-specific documentation for this release.

The product name Oracle Database Migration Assistant for Unicode is often abbreviated as DMU throughout this document, in other Oracle documentation and on Oracle Web sites.

This document may be updated after it is released. To check for updates to this document and to view other DMU documentation, refer to the Documentation section on the Oracle Technology Network (OTN) DMU Web site:

http://www.oracle.com/technetwork/database/database-technologies/globaliza
tion/dmu/documentation/index.html

This document contains the following topics:

- Changes Between Releases 2.0 and 1.2
- Supported Configurations
- Installation Instructions
- Requirements
- Known Issues and Limitations
- Important Security Considerations

Changes Between Releases 2.0 and 1.2

The DMU repository schema has been updated in release 2.0. If you have an old repository installed with the 1.2 release, you will need to uninstall and reinstall the repository using DMU 2.0.

New Features in Release 2.0

The following changes are for Release 2.0:

- DMU supports the migration of Oracle Pluggable Databases (PDBs) in Database 12c. If you are using the new PDB feature to consolidate databases with different database character sets, please note that every PDB must have a database character set that is compatible with that of the container database (CDB) which the PDB is being plugged into. Compatible means that the character set has to be the same or the PDB's character set must be a binary subset of the CDB's character set and both have to be single-byte or both have to be multibyte. Our recommended best practices approach for such consolidation is to use the Unicode character set

AL32UTF8 for the new CDB and its PDBs. AL32UTF8 provides a uniform superset character set that can support character data in any languages, thus allowing maximum compatibility among databases with different legacy character sets to be consolidated.

To consolidate databases with different character sets:

1. Create a CDB with the database character set AL32UTF8 and the national character set AL16UTF16. In case most databases to be consolidated use the national character set UTF8, use UTF8 instead of AL16UTF16.

2. For each non-CDB to be consolidated:

   a. Upgrade it to Oracle Database 12c, if it uses Oracle Database release before Oracle Database 12c.

   b. Migrate its database character set to AL32UTF8 using the DMU.

   c. Migrate its national character set to the national character set of the CDB (AL16UTF16 or UTF8). Contact Oracle Support to find out how to do this.

   d. Use the upgraded and migrated non-CDB to create a new PDB. See *Oracle Database Administrator's Guide* for information about creating a PDB using a non-CDB.

If you have already consolidated your databases using a non-Unicode character set and need to migrate your existing PDBs to Unicode, you do so with DMU 2.0:

1. Create or identify an AL32UTF8 CDB into which the migrated PDBs are to be plugged.

2. For each PDB to be migrated:

   a. Use the DMU to scan the PDB and resolve any reported convertibility issues while it is still plugged into the original non-Unicode CDB.

   b. Unplug the PDB to be migrated and plug it into the target AL32UTF8 CDB (this will put the PDB into restricted mode due to the character set incompatibility).

   c. Use the DMU to convert the PDB to Unicode.

   d. Restart the converted PDB in unrestricted mode.

This approach will allow for an efficient and predictable consolidation process which reduces the downtime window requirement.

For performing scanning and cleansing operations on a PDB, you can connect using the DMU as any user with SYSDBA privilege in the local PDB. For performing conversion operations on a PDB, you must connect using the DMU as either the SYS user or a common user with SYSDBA privilege in both the local PDB and the CDB.

- The `CSREPAIR` script has also been enhanced to support correcting the character sets of PDBs to match with the stored database contents when no data conversion is required. To run CSREPAIR on a PDB, you need to connect using either the SYS user or a common user with SYSDBA privilege in both the local PDB and the CDB.

- DMU 2.0 offers enhanced bulk cleansing features to facilitate the cleansing of data convertibility issues with similar causes or symptoms. In addition to the ability to perform bulk migration to character length semantics, the DMU now supports pattern-based cleansing which enables the batch replacement or removal of occurrences of a byte or character sequence in column values. This can be particularly useful in dealing with offending bytes or characters that appear in

multiple database objects. For columns containing data with convertibility issues but are insignificant to your application, you can now set the "Allow Conversion of Data with Issues" property in bulk to instruct the DMU to convert the columns despite the reported issues.

■ This release of the DMU also introduces the support for migrating PeopleSoft databases to Unicode. When the connected database is detected to be a PeopleSoft instance, the DMU will execute the PeopleSoft-specific migration logic transparently as part of the migration workflow. The pre-requisite is that the databases need to be for PeopleSoft Application versions 9.0 or later and PeopleTools versions 8.48 or later.

■ The conversion error-handling mechanism has been enhanced to include options to automatically skip errors related to materialized view refreshes and index rebuilds and export the failing SQL statements to external scripts for subsequent resolution. All steps in the conversion phases have been made resumable, including ALTER DATABASE CHARACTER SET which may require a database restart to re-sync the in-memory character set information with the data dictionary.

■ The "Copy data using CREATE TABLE AS SELECT" conversion method can now be applied for tables with user-named LOB segments.

■ Performance optimizations have been implemented to improve the scalability and conversion time on databases involving large numbers of character length semantics columns.

■ The validation mode conversion feasibility check has been redesigned to more clearly convey the readiness status for converting invalid columns to Unicode. All relevant errors and warnings will be presented in the validation status panel before the user attempts to convert columns containing data in a different character set to Unicode.

Supported Configurations

The latest support information for Oracle Database Migration Assistant for Unicode is available on the OTN DMU Web site at:

http://www.oracle.com/technetwork/database/database-technologies/globalization/dmu/learnmore/index.html

in the document titled Supported Configurations.

Installation Instructions

The installation instructions for Oracle Database Migration Assistant for Unicode are available on the OTN DMU Web site at:

http://www.oracle.com/technetwork/database/database-technologies/globalization/dmu/learnmore/index.html

in the document titled Getting Started.

Requirements

This section describes the following types of requirements:

■ General Database Requirements

■ Database Convertibility Requirements

- [Database Space Requirements](#)

General Database Requirements

The database must meet certain requirements to be supported by the DMU. These requirements are:

- The database character set must be ASCII-based, therefore, databases running on the EBCDIC-based platforms IBM z/OS and Fujitsu BS2000 are not supported.

- The package `SYS.DBMS_DUMA_INTERNAL` must be installed in the database.

  The script `?/rdbms/admin/prvtdumi.plb` to create the package is available as part of the database installation. You must create the package manually by running the script from the Oracle home of the database. See Installation Instructions for details.

- Oracle Database Vault must be disabled before starting the migration process, because the DMU has not been certified to work with it enabled.

- The database must be opened in read/write mode.

Database Convertibility Requirements

Additional requirements pertain to databases that the DMU should convert. Without meeting these requirements, the DMU can still be used for scanning and cleansing the database. The requirements are:

- All database objects, including auxiliary objects created by standard PL/SQL packages, such as `DBMS_RULE`, `DBMS_DATA_MINING`, or `DBMS_WM`, must be named using only characters from the ASCII character set. In other words, the data dictionary of the database cannot contain non-ASCII characters except in a few selected tables.

  For more details, see *Oracle Database Migration Assistant for Unicode Guide*, chapter 5, section "Migrating Data Dictionary Contents".

- No OLAP analytical workspaces, other than predefined system workspaces and certain predefined Oracle Applications workspaces, can exist in the database.

- No flashback data archives can exist in the database.

- No data to be converted can reside in a read-only or offline tablespace.

- Neither cluster key columns nor partitioning key columns can be defined with character length semantics.

- No convertible data can be present in tables in the recycle bin.

- No convertible data can be present in a reference partitioning key column.

- No convertible data can be present in `ANYDATA`/`ANYDATASET` columns.

Database Space Requirements

The migration process requires free space in the database. The free space is required in the following areas:

- Migration repository

  Repository tables store DMU internal state information, scan results, scheduled cleansing actions, conversion plan details, and collected rowids for convertible and/or problematic rows in scanned tables. Oracle recommends that you create a

separate tablespace for the migration repository. See *Oracle Database Migration Assistant for Unicode Guide* for information about creating such a tablespace.

■ Data conversion

Data that is converted from a legacy character set to AL32UTF8 or UTF8, and which does not consist of ASCII characters only, usually expands in size, because the UTF-8 encoding of a character has, in most cases, more bytes than the legacy character set encoding of the same character. Moreover, the conversion method "Copy data using `CREATE TABLE AS SELECT`" converts data in a table while creating a copy of the table with the SQL statement `CREATE TABLE AS SELECT`. After the copy is created, the source table is dropped but for some time both tables exist simultaneously. Therefore, additional space is required to accommodate copies of tables converted using this conversion method.

To view an estimation of the amount of free space needed per tablespace to accommodate the data expansion and the temporary space for `CREATE TABLE AS SELECT`, right-click on the database node in the Navigator pane of the DMU and select Properties. On the opened Database Properties tab, select the Scanning subtab. Click on the Estimate Tablespace Extension button at the bottom of the page to calculate the minimum and maximum space requirements for each tablespace. The minimum tablespace extension is calculated by taking into account the post-conversion data size expansion and the temporary space requirement of the largest table converted using the "Copy data using `CREATE TABLE AS SELECT`" method. The maximum tablespace extension is calculated by taking into account the post-conversion data size expansion and the temporary space requirements of the first n largest tables converted using the "Copy data using `CREATE TABLE AS SELECT`" method where n is the number of conversion worker threads.

Use the reported extension information to estimate the order of magnitude of the required free space but use the autoextend feature of database data files to make sure that tablespaces can expand if required.

Known Issues and Limitations

This section describes known issues and limitations.

Creating DMU Diagnostic Packages on PDBs

The Create Diagnostic Package functionality does not work on PDBs in the database 12.1.0.1.0 release when the PDB has an incompatible character set with the CDB character set due to a known database bug (reference: Bug 17384878).

Non-ASCII Characters in PDB PL/SQL Definitions

If the PDB to be migrated contains non-ASCII characters in PL/SQL objects, triggers, and view definitions, then the DMU conversion SQL generation operation will fail with ORA-6502 in the database 12.1.0.1.0 release (reference: Bug 16488610). The workaround is to remove the non-ASCII characters from the definitions and rescan the data dictionary before generating the conversion SQL statements.

Editing ANYDATASET Columns with Collections

The cleansing editor cannot properly display `ANYDATASET` columns containing varrays or nested tables (reference: Bug 11692435).

To cleanse data in such columns, you need to update the problematic values or use larger built-in content types, depending on the reported issues. You can use the `ANYDATASET` and `ANYDATA` OCI and/or PL/SQL APIs to access, decompose, edit and rebuild `ANYDATASET` values.

> **See Also:**
>
> - *Oracle Database PL/SQL Packages and Types Reference* for information on the `ANYDATASET` and `ANYDATA` Oracle-supplied types and their methods that comprise the `ANYDATASET` PL/SQL API
>
> - *Oracle Call Interface Programmer's Guide* for information on the `OCIAnyDataSet` and `OCIAnyData` interfaces that comprise the `ANYDATASET` C API

LOB Segment Attributes

Due to RDBMS bugs #5577093, #5983283, and #6677390, LOB segments in tables converted by the conversion method "Copy data using `CREATE TABLE AS SELECT`" may lose the storage attribute `RETENTION` and get the storage attribute `PCTVERSION`. Use the SQL statement `ALTER TABLE table_name MODIFY LOB (lob_name) (RETENTION)` to restore the expected attribute.

Scheduled Cleansing from CHAR to VARCHAR2

When a scheduled cleansing action is defined to migrate a `CHAR` column to the `VARCHAR2` data type, the scan results may incorrectly report over type limit issues even if the post-conversion length fits within the `VARCHAR2` data type limit. If you can confirm that the post-conversion data size fits within the `VARCHAR2` data type limit in the cleansing editor, then the workaround is to set the "Allow Conversion of Data with Issues" column conversion property to "Yes" so that the conversion feasibility test on this column can be bypassed. This issue is fixed in the database 11.2.0.3 release (reference: Bug 12868420).

Column-level Character Set Tagging in Multibyte Databases

Due to a restriction in the DMU server-side data scanning function, the DMU does not allow character set tagging for character length semantics columns when the database character set is multibyte and the database version is 11.2.0.3 or older. If such tagging is necessary, please consider temporarily switching the column to byte length semantics for the duration of the migration (reference: Bug 13242969).

Editing Columns with Shift-sensitive Character Data

The cleansing editor currently does not support editing data in columns which are tagged with shift-sensitive character sets. You can still view the data details for cells in these columns using the data viewer (reference: Bug 14241789).

Replacement Characters Reported as Invalid for UTF8 Target Character Set

On database 10.2 releases, when the source character set is multibyte and the target character set is UTF8, the '?' character will be incorrectly reported as invalid in the scan results. For 11*g* releases up to 11.2.0.3, if the data being scanned contains the '?' character following any non-ASCII character, it will be incorrectly reported as invalid when the source character set is multibyte and the target character set is UTF8. If you can confirm there is no other invalid data in the column, you can set "Allow

Conversion of Data with Issues" property on this column to "Yes" so that the conversion feasibility test on this column can be bypassed (reference: Bug 14530511).

Scanning Shift-sensitive Data without Shift Characters

If the column tagged with a shift-sensitive character set contains data that does not include any shift-in/shift-out characters and the target migration character set is UTF8, the DMU scan may hang due to a known bug. You can work around the issue by adding shift characters into the input data (reference: Bug 14580879).

Editing CLOB Data in Nested Tables

The cleansing editor currently does not support editing CLOB data in nested tables. You may edit the data outside of the DMU if it contains data exceptions (reference: Bug 14585707).

Important Security Considerations

Unless you install the DMU on a host machine to which only you and appropriately authorized people have access, you need to take precautions to protect the DMU installation and the DMU configuration files. Otherwise, unauthorized access to the files could compromise security of the databases to which you connect with the DMU.

After you have uncompressed the archive file with the DMU installation, ensure that all uncompressed files and directories are writable only to you and other authorized operating system users. The DMU does not come with an installer that could set the file permissions automatically. Removing the write privilege from unauthorized users is very important because such users with access to the DMU host could modify the DMU files to make the DMU execute arbitrary SQL statements when the DMU is later started with SYSDBA credentials. Such SQL statements could compromise database security.

If you select the Save Password check box when creating a database connection, the password you specify is saved in an obfuscated form in a password file named cwallet.sso in your user directory. Because obfuscation is a reversible operation, use this feature only for passwords to test databases with no production data or only if the DMU is installed on a very well protected host. Ensure that the password file is readable only by you.

On Unix-based platforms, the file is in the directory $HOME/.dmu/. On Microsoft Windows, the file is in the directory %APPDATA%\DMU\.

This release of the DMU requires that you connect to a database specifying a database user with the SYSDBA privilege. This user will have full access to DMU repository objects. Do not grant any privileges on any of the DMU tables or PL/SQL packages to any database user, except in cases documented explicitly in the DMU documentation.

Index

D

Database Migration Assistant for Unicode
   release notes, 1-1
DMU release notes, 1-1