ORACLE

Supplemental Migration Guide for OTM 6.4.3

January, 2024, Version 2.1

Public

Purpose statement

This document provides an overview of the migration steps needed to migrate to OTM 6.4.3. It is intended solely to help you assess work required to move to 6.4.3 and to plan your I.T. projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of contents

Migrating to OTM 6.4.3

Clients using this guide are expected to already have OTM 6.4.3 installed.

If needed, you can review the supplemental installation guide for 6.4.3 available on this note.

- Oracle Transportation Management Version 6.4.3 Supplemental Installation Guide (Doc ID 2602660.1)

Purging Data Before Migration

We recommend you purge as much of your older data before running the migration to 6.4.3. Purging older orders, shipments and other related data. This will help to reduce the time it takes for the migration to complete. Ideally, it is best to limit your operational data to two years. Anything older should be purged.

In addition to purging older shipment, order and invoice data, you should consider purging data from some of the partitioned tables in OTM if you are not already doing so. This should help to reduce the amount of data being exported. Log in to the database from the 6.4.x instance as glogowner and run the following command.

```
@purge_partitioned_tables
```

There are four options available

```
Q, MM, WW, D
```

Each should be run to help remove older data from your instance.

Note - In a test on our internal instance, the size of the export was reduced by 10%.

Cloning the 6.4.x Database

The instructions in this document cover doing a full export of the GLOGOWNER, REPORTOWNER and ARCHIVE schemas. The note mentioned below has additional instructions for clients that want restrict data from being exported from tables, like the I_TRANSMISSION and I_TRANSACTION tables from being exported. These tables still need to have their table structure exported and imported, which is also outlined in the note.

- **Cloning Oracle Transportation Management Database Schemas (Doc ID 2786512.1)**

Setting up the 6.4.x Database to Export Data

The following steps should be run on your existing 6.4.x instance.

Log into the database using sqlplus as sys and run the following commands.

```
select * from dba_directories where directory_name = 'MY_DIR';
```

```
select * from dba_directories where directory_name = 'MY_LOGDIR';
```

```
select * from dba_directories where directory_name = 'DATA_PUMP_DIR';
```

In most cases, the DATA_PUMP_DIR will already be set. The same directory can be used for the other values. If none of the parameters have a value associated with them, create a directory that the database user has read/write access to use for the data export.

Log into the database as sys and update the parameters with the directory you want to use using the following commands.

```
CREATE OR REPLACE DIRECTORY MY_DIR as '< your data_pump_dir>';

CREATE OR REPLACE DIRECTORY MY_LOGDIR as '< your data_pump_dir>';

CREATE OR REPLACE DIRECTORY DATA_PUMP_DIR as '< your data_pump_dir>';

commit;
```

Setting the Environment Variables for the 6.4.x Database Export

Go to the DATA_PUMP_DIR and run the following commands to set some variables needed for the export.

```
export ORACLE_SID=<your existing 6.4.x DB SID>

export NLS_LANG=AMERICAN_AMERICA.UTF8

export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
```

Creating the export.par file and running expdp

Create a par file, export.par, to export the data from the 6.4.x instance. This document only covers exporting the GLOGOWNER, REPORTOWNER and ARCHIVE schema.

```
Userid='sys/<sys password>@<64x SID> as sysdba'

DIRECTORY=DATA_PUMP_DIR

FILESIZE=8G

DUMPFILE=otm_full_64xt%U.dmp

LOGFILE=otm_full.explog

SCHEMAS= glogowner,reportowner,archive
```

Note - Clients on 6.4.2 should export the archive_c and archive_c_user instead of the archive schema.

```
Userid='sys/<sys password>@<64x SID> as sysdba'

DIRECTORY=DATA_PUMP_DIR

FILESIZE=8G

DUMPFILE=otm_full_64xt%U.dmp

LOGFILE=otm_full.explog

SCHEMAS= glogowner,reportowner,archive_c, archive_c_user
```

Run the expdp command from your DATA_PUMP_DIR

```
expdp parfile=export.par
```

Compressing the .dmp files

Once the export is completed, compress the files and copy them to the new database or data pump directory you intend to use for your 6.4.3 database install.

```
tar -zcvf export64x.tar.gz <directory where the .dmp files were generated>
```

Preparing the New 6.4.3 Database

Update tnsnames.ora on the Existing 6.4.x Instance

Add the database entry for the new 6.4.3 database to the exiting 6.4.x installs tnsnames.ora file inside the Oracle Client Home on one of the application servers. This new tnsnames.ora entry is needed to run the create_gc3_tablespaces.sql and create_glog_users.sql script from the existing 6.4.x install.

```
cd $ORACLE_HOME/network/admin

cp tnsnames.ora tnsnames.ora_backup

vi tnsames.ora
```

Checking the Tablespace on your existing 6.4.x Instance

Check the tablespace being used on your existing 6.4.x instance you exported the data from and ensure the new database for 6.4.3 has more tablespace than what you are using now. Use the following query to check what your existing 6.4.x database is using.

```
SELECT TABLESPACE_NAME, SUM(BYTES)/1024/1024/1000 AS "Size (Gigs)" FROM DBA_DATA_FILES GROUP BY
TABLESPACE_NAME;
```

This should display the tablespace name and the number of GB's it is using. This will help you determine the size you will need to use when creating the tablespaces on the new 6.4.3 database. Be sure and create the tablespaces on the new database with extra space to allow for changes during the migration and additional future growth.

Creating the Tablespaces and Users on the New 6.4.3 Database

From the existing 6.4.x install log into the new 6.4.3 database using sqlplus as sys and run the create_gc3_tablespaces.sql and the create_glog_users.sql.

```
sqlplus sys/<sys password>@<new 6.4.3 DBSID> as sysdba

@create_gc3_tablespaces.sql

@create_glog_users.sql
```

When running the **create_glog_users.sql** script be sure and use the same passwords for users like glogdba, glogload, and that you used when you ran the 6.4.3 installer.

```
GLOGDBA_PASSWORD=

GLOGLOAD_PASSWORD=
```

You should use the expected password format for 6.4.3 and above, which is each password should be at least 8 characters and should include one number. The following special characters should not be used.

- # [ ] $ % ? { } ` \ " ' |

Setting up the New 6.4.3 Database to Import Data

While logged into the new 6.4.3 database using sqlplus as sys and run the following queries.

```
select * from dba_directories where directory_name = 'MY_DIR';
select * from dba_directories where directory_name = 'MY_LOGDIR';
select * from dba_directories where directory_name = 'DATA_PUMP_DIR';
```

In most cases, the DATA_PUMP_DIR will already be set. The same directory can be used for the other values. If none of the parameters have a value associated with them, create a directory that the database user has read/write access to use for the data import.

```
CREATE OR REPLACE DIRECTORY MY_DIR as '< your data_pump_dir>';
CREATE OR REPLACE DIRECTORY MY_LOGDIR as '< your data_pump_dir>';
CREATE OR REPLACE DIRECTORY DATA_PUMP_DIR as '< your data_pump_dir>';
commit;
```

Log in the database server for the new 6.4.3 database and go to the DATA_PUMP_DIR

Copy the export file from the 6.4.x database instance to the DATA_PUMP_DIR for the new 6.4.3 database and unzip the contents.

```
tar -zxvf export64x.tar.gz
```

Setting the Environment Variables for the New 6.4.3 Database Import

Run the following commands before importing the data to set the required environmental variables.

```
export ORACLE_SID=<your new 6.4.3 instance>
export NLS_LANG=AMERICAN_AMERICA.UTF8
export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
```

Running the impdp command on the New 6.4.3 Database

Use the impdp command to import the data.

```
impdp directory=DATA_PUMP_DIR DUMPFILE=<filename1>.dmp LOGFILE=otm_full_<new 643 DB SID>.implog
SCHEMAS=glogowner,reportowner,archive
```

If multiple .dmp files were generated during the export you can the additional .dmp files into the import command.

```
impdp directory=DATA_PUMP_DIR DUMPFILE=<filename1>.dmp,<filename2>.dmp LOGFILE=otm_full_<new 643 DB SID>.implog SCHEMAS=glogowner,reportowner,archive
```

When prompted for the user name and password use the following syntax.

```
sys/<sys password>@<new DB SID> as sysdba
```

After the import completes you should see something similar to this.

```
Processing object type SCHEMA_EXPORT/TABLE/CONSTRAINT/REF_CONSTRAINT
Processing object type SCHEMA_EXPORT/TABLE/TRIGGER
Processing object type SCHEMA_EXPORT/VIEW/TRIGGER
Processing object type SCHEMA_EXPORT/EVENT/TRIGGER
Processing object type SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Processing object type SCHEMA_EXPORT/STATISTICS/MARKER
Processing object type SCHEMA_EXPORT/JOB
Processing object type SCHEMA_EXPORT/TABLE/POST_INSTANCE/PROCACT_INSTANCE
Processing object type SCHEMA_EXPORT/TABLE/POST_INSTANCE/PROCDEPOBJ
Processing object type SCHEMA_EXPORT/POST_SCHEMA/PROCACT_SCHEMA
Job "SYS"."SYS_IMPORT_SCHEMA_02" completed with 3 error(s) at Thu Jan 11 20:54:21 2024 elapsed 0 00:39:41
```

The errors mentioned in the logs should show these.

```
ORA-31684: Object type USER:"ARCHIVE" already exists
ORA-31684: Object type USER:"GLOGOWNER" already exists
ORA-31684: Object type USER:"REPORTOWNER" already exists
```

If you are importing the data from a 6.4.2 instance, you should see errors related to the ARCHIVE_C and ARCHIVE_C_USER listed instead of ARCHIVE.

Restart the new 6.4.3 database before continuing.

Run the post import steps from Cloning Oracle Transportation Management Database Schemas (Doc ID 2786512.1)

Clients on version 6.4.x should be sure and follow the instructions mentioned on the following note.

- Cloning Oracle Transportation Management Database Schemas (Doc ID 2786512.1)

Avoiding Known Issues When Migrating to OTM 6.4.3 (Doc ID 2361277.1)

Review the information available on the following note.

- Avoiding Known Issues When Migrating to OTM 6.4.3 (Doc ID 2361277.1)

Download the following patches from MOS - 27333834, 26870178 and 27391599.

Unzip each of the patches and install them in this order.

```
java -jar otm643_quickpatch_27333834.jar -d $GLOG_HOME
java -jar otm643_quickpatch_26870178.jar -d $GLOG_HOME
java -jar otm643_quickpatch_27391599.jar -d $GLOG_HOME
```

Note – Do not follow any of the post installations steps mentioned when installing these patches. Any DB changes made by these patches will made during the migration to 6.4.3.

Review the Migration Steps in the OTM 6.4.3 Migration Guide

Edition enabled Database Schemas

Be sure and remove any customizations that point to objects in the glogowner, reportowner or other OTM schemas. Any custom view, table, trigger, package, procedure, function, type, VPD policy or synonym that was not part of the OTM installation should be dropped and re-implemented after the migration to 6.4.3 is complete. Leaving customizations in place has been known to cause failures during the migration process. The following statement is from the 6.4.3 Migration Guide.

> As part of the 6.4.3 Release, all Oracle Transportation Management Database Schemas are going to be enabled with the database feature Edition Based Redefinition. This is necessary to provide future support for zero downtime patching. The OTM release 6.4.3 uses a new database edition "OTM643" as a default edition instead of the database default edition ORA$BASE. All of the OTM schemas have been editioned enabled except for GLOGOAQ schema, which contains Oracle Advanced Queue related objects.
>
> All of the OTM tables have been converted into Editioned Views. Original tables have been renamed to the table name with the "_T" suffix to facilitate this conversion. After a successful migration to 6.4.3, OTM will use the Editioning view seamlessly. Customer-defined queries, saved queries, and reports will not be affected by this change; they will seamlessly use Editioned Views instead of tables.
>
> **Note:** Any view, trigger, package, procedure, function, type, VPD policy or synonym will be dropped during 6.4.3 migration if the Oracle Transportation Management does not provide it and if it exists on the OTM database schema. In addition, configurations done to the OTM-provided view, trigger, package, procedure, function, type, VPD policy, or synonym will be lost and it will be replaced with the OTM definition. You must store copies of such objects on the file system and recreate them after migrating to 6.4.3. Database object on the customer-defined database schema may be affected if it references any OTM database object.
>
> Any future configurations should use the Editioned View name in the customer-defined code for referencing the OTM table. You may not add non-edition objects which depend on the edition object provided by the Oracle Transportation Management.
>
> Refer to "Using Edition-Based Redefinition" in Oracle Database Development guide.

Checking required and recommended database settings

Check the value of the following database parameters. They should both be set to 3000.

```
select name, value from v$parameter where name in ('open_cursors','processes');
```

Note - Use the following query to check the values of all of the required and recommended parameter settings.

```
select name, value from gv$parameter where name in
('cursor_sharing','db_16k_cache_size','db_block_size','log_buffer','job_queue_processes','optimi
zer_index_cost_adj','optimizer_index_caching','query_rewrite_enabled','query_rewrite_integrity',
'processes','open_cursors','session_cached_cursors','statistics_level');
```

These are the values for the required (*) and recommended settings.

```
MEMORY_TARGET = 8G
cursor_sharing = FORCE
*db_16k_cache_size = 104857600
db_block_size = 8192
log_buffer = 163840
job_queue_processes = 4
optimizer_index_cost_adj = 50
optimizer_index_caching = 50
query_rewrite_enabled   =  true
query_rewrite_integrity    = trusted
*processes = 3000
*open_cursors = 3000
session_cached_cursors=100
statistics_level = TYPICAL
```

Running the pre_migration_check.sh script

On the OTM 6.4.3 instance go to the $GLOG_HOME/glog/oracle/script8 directory and run the pre_migration_check.sh script. Be sure and fix any errors that are reported.

```
./pre_migration_check.sh
```

The following possible pre_migration_check errors you may encounter.

Synonym Errors

One error that may be reported is a problem with a synonym used by OTM. The following example shows a problem with a synonym for REPORTOWNER related to the PLAN_TABLE.

```
OWNER              OBJECT_NAME                    OBJECT_TYPE
-------------- ------------------- ------------------------------
REPORTOWNER        PLAN_TABLE                     SYNONYM
```

For errors related to the synonyms you can take the following action.

Log into the database using sqlplus as sys and run the following:

```
drop public synonym <OBJECT_NAME>

drop public synonym plan_table
```

Log into the database using sqlplus as <OWNER> and run the following.

```
drop synonym <OBJECT_NAME>

drop synonym plan_table
```

Constraint Errors

You may see errors similar to this.

```
OWNER           CONSTRAINT_NAME           TABLE_NAME                     STATUS       VALIDATED
--------------- ----------------- ------------------------------ ------------------ -----------
REPORTOWNER     CK_REPORT_LOG_REPORT_SENT       REPORT_LOG
       ENABLED     NOT VALIDATED

REPORTOWNER     CK_REPS_REPORT_SYSTEM_TYPE      REPORT_SYSTEM
       ENABLED     NOT VALIDATED

REPORTOWNER     CK_RPTPRM_PARAMETER_TYPE        REPORT_PARAMETER                    ENABLED
NOT VALIDATED

REPORTOWNER     CK_USE_PARAMS_AS_BIND            REPORT
ENABLED     NOT VALIDATED

GLOGOWNER       FK_VPCPD_VAT_PROVINCIAL_CNFG    VAT_PROVINCIAL_CNFG_PERSP_D        ENABLED
NOT VALIDATED

GLOGOWNER       FK_VPCPD_VAT_CODE_ORDERED_SET   VAT_PROVINCIAL_CNFG_PERSP_D        ENABLED
NOT VALIDATED

GLOGOWNER       FK_VPCPD_SRC_PROVINCE_CODE_PR   VAT_PROVINCIAL_CNFG_PERSP_D        ENABLED
NOT VALIDATED
```

The pre_migration_check.sh recommends running the validate_constraints.sql script as glogowner or reportowner depending on what OWNER was returned.

```
@validate_constraints.sql
```

Warning messages - job_queue_processes

The following may be generated when the job_queue_processes is checked.

```
Warning         : Scheduled jobs may prevent the OTM migration from completing.

Warning         : Consider disabling all enabled jobs.

Action Required : Consider setting initialization parameter, job_queue_process to zero. Restart
database.
```

Log into the database as sys and run the following command to update the job_queue_processes parameter.

```
alter system set job_queue_processes=0 scope=both;
```

Check the change by running this.

```
select name, value from v$parameter where name in ('job_queue_processes');
```

0 should be returned.

Note - You will need to restart the database after making this change.

ARCHIVE_C and ARCHIVE_C_USERS users and schema

The ARCHIVE_C and ARCHIVE_C_USER and schema are used in version 6.4.2 and above.

Log in as sys and run the following command to see if the users exist on your instance.

```
select username from dba_users where username in ('ARCHIVE_C','ARCHIVE_C_USER');
```

Note - If you are migrating from 6.4.1 or below these users shouldn't be found. If you are migrating from 6.4.2 the users and the tablespaces should already exist.

Exit sqlplus and run the following command to create the user if needed.

```
./archive_c/setup_archive_c.sh
```

Enter database connection id, SYSDBA user password, DBA user name who has privilege to alter database users (typically system), and its password, ARCHIVE_C user password , ARCHIVE_C_USER password. The ARCHIVE_C_USER password should match the password you used for this property in your installer.properties file.

```
ARCHIVE_C_USER_PASSWORD
```

Deleting obsolete columns and tables before migrating

The OTM 6.4.3 migration works best when there are no obsolete columns or tables left over from older versions of OTM. To see what columns and tables are found in your database you can log into the database as glogowner and run the following script.

```
@gen_obsolete_objects.sql;
```

The script will display on the screen the columns and tables that are already marked obsolete in previous installations or migrations. The script will also create the drop_obsolete_objects.sql to remove all of the old columns and tables that were found. In most cases, these obsolete columns and tables are not needed but you should look over the list carefully to ensure you have not used any for your own custom data needs.

To drop the columns and tables log into the database as glogowner and run the following.

```
@drop_obsolete_objects.sql;
```

Re-run recompile_invalid_objects.sql as glogowner and reportowner after running the drop_obsolete_objects.sql script.

```
@recompile_invalid_objects.sql;
```

Log into the database as sys and run the following scripts if you see a large number of synonyms as becoming invalid when you run the recompile_invalid_objects.sql.

```
@create_otm_synonyms;
@recompile_invalid_synonyms.sql
```

Re-run the pre_migration_check.sh

Re-run the pre_migration_check.sh script again verify no new errors are reported.

Make a Restore Point for the Database

In order to see if ARCHIVELOG is enabled on the new 6.4.3 database, log into the database via sqlplus as sys and run the following query.

```
select log_mode from v$database;
```

You will likely see the following, which means archiving is not enabled.

```
LOG_MODE

------------------------------------

NOARCHIVELOG
```

While logged in as sys via sqlplus run these commands to enable archiving.

```
shutdown immediate;

startup mount;

alter database archivelog;

alter database open;

create restore point BEFORE_643 guarantee flashback database;
```

Note - You may want to increase your DB_RECOVERY_FILE_DEST_SIZE to allow for enough space to track all of the changes being made during the migration.

```
ALTER SYSTEM SET DB_RECOVERY_FILE_DEST_SIZE = 25G SCOPE=BOTH SID='*';
```

The size you need depends on the size of the database you are migrating. Follow up with your internal DBA resources for sizing recommendations and required drive space.

Running dbpatch.sh

Per the migration guide, you can run dbpatch.sh in interactive or silent mode by using either of following command strings.

```
./dbpatch.sh

./dbpatch.sh otmdb glogpw glogdbapw reportpw archivec_pwd ../../config sysdbapw <version migrating from> system systempw newdirxmlpw faadminpw glogoaqpw otmSystempw
```

Regardless of the method you use it is a good idea to check the passwords of the users. Verify the passwords for each of the users needed to run the dbpatch.sh script before running the script.

```
sqlplus glogowner/glogpw@<otmdb>

sqlplus glogdba/glogdbapw@<otmdb>

sqlplus reportowner/reportpw@<otmdb>

sqlplus archive_c/archivec_pwd@<otmdb>

sqlplus sys/sysdbapw@<otmdb> as sysdba

sqlplus system/systempw@<otmdb>
```

The remaining users are new, but the otmSystempw you use should be the same password you used when you installed OTM 6.4.3. Check the installer.properties you used for the following value.

```
OTM_SYSTEM_PASSWORD=
```

The following is an example of the logs you may see generated in your instance. Note there are several pre_migration script logs since the process was run several times.

```
pre_migration_check_<DB SID>_<YYYMMDD>_<HHMM>.log
archivec_setup_<DB SID>_<YYYMMDD>_<HHMM>.log
pre_migration_check_<DB SID>_<YYYMMDD>_<HHMM>.log
pre_migration_check_<DB SID>_<YYYMMDD>_<HHMM>.log
sm.log
csv_precheck_<DB SID>_<YYYMMDD>_<HHMM>.log
update_password_<DB SID>_<YYYMMDD>.log
import_content_<DB SID>_<YYYMMDD>_<HHMM>.log
update_content_<DB SID>_<YYYMMDD>_<HHMM>.log
dbpatch_641_<DB SID>_<YYYMMDD>_<HHMM>.log
642_update_password_<DB SID>_<YYYMMDD>.log
642_update_content_<DB SID>_<YYYMMDD>_<HHMM>.log
642_import_content_<DB SID>_<YYYMMDD>_<HHMM>.log
642_dbpatch_641_<DB SID>_<YYYMMDD>_<HHMM>.log
642_csvone_gtm_AUTO_ASSIGN_TYPE_CRIT_MAP.log
642_csvone_gtm_AUTO_ASSIGN_CRITERIA.log
642_csvone_AUTO_ASSIGN_TYPE_CRIT_MAP.log
642_csvone_AUTO_ASSIGN_CRITERIA.log
```

Additional logs will be generated in the $GLOG_HOME/glog/oracle/script8.642

```
update_password_<DB SID>_<YYYMMDD>.log
import_content_<DB SID>_<YYYMMDD>_<HHMM>.log
update_content_<DB SID>_<YYYMMDD>_<HHMM>.log
csvone_AUTO_ASSIGN_CRITERIA.log
csvone_AUTO_ASSIGN_TYPE_CRIT_MAP.log
csvone_gtm_AUTO_ASSIGN_CRITERIA.log
csvone_gtm_AUTO_ASSIGN_TYPE_CRIT_MAP.log
dbpatch_641_<DB SID>_<YYYMMDD>_<HHMM>.log
```

Review the logs for possible errors. The key phrases to look for are the following.

```
"ORA-", "Caught exception","SP2-", "PLS-","IMP-","Warning:"," Caught exception", "SQL* Loader-"
," SQLException", "SQLRecoverableException" or "<Error>"
```

Run the following commands and then review the log_check.txt file for possible errors.

```
grep "ORA-" *.log > log_check.txt
grep "Caught exception" *.log  >> log_check.txt
grep "SP2-" *.log >> log_check.txt
grep "PLS-" *.log >> log_check.txt
grep "IMP-" *.log >> log_check.txt
grep "Warning:" *.log >> log_check.txt
grep "Caught exception" *.log >> log_check.txt
grep "SQL* Loader-" *.log >> log_check.txt
grep "SQLException" *.log >> log_check.txt
grep "SQLRecoverableException" *.log >> log_check.txt
grep "<Error>" *.log >> log_check.txt
```

Review the log_check.txt file that is generated by the commands for any errors generated during the migration. If no errors are found you are ready to start the new OTM 6.4.3 instance.

dbpatch.sh failures

The migration to version 6.4.3 is a very complex migration that cannot be restarted after a migration failure. In the event of a failure, clients should investigate the cause, restore the database to the restore point, correct any data issues that may have caused the failure and restart dbpatch.sh script again.

Checking for Customizations After a Migration Failure

One of the most common reasons the migration to 6.4.3 fails is due to customizations that were not removed before running dbpatch.sh.

The following queries may be useful in finding customizations on your instance after a migration failure. The following queries should be run before the database is restored to the restore point you created earlier.

glogowner

```
select 'Non-editioned '||TYPE||' '||OWNER||'.'||NAME ||', References '||REFERENCED_TYPE||'
'||REFERENCED_OWNER||'.'||REFERENCED_NAME|| ' (To Be Editioned) with Reference Type
'||DEPENDENCY_TYPE Message from dba_dependencies , dba_users

where REFERENCED_OWNER='GLOGOWNER'

and REFERENCED_TYPE in ('FUNCTION','LIBRARY','PACKAGE','PACKAGE
BODY','PROCEDURE','TRIGGER','TYPE','TYPE BODY','SYNONYM','VIEW','NON-EXISTENT')

and not (OWNER=REFERENCED_OWNER and TYPE='PACKAGE BODY' and REFERENCED_TYPE='PACKAGE' )

and not (REFERENCED_OWNER='SYS' and REFERENCED_NAME='STANDARD' and REFERENCED_TYPE like
'PACKAGE%' )

and OWNER=USERNAME

and EDITIONS_ENABLED='N' order by 1;
```

reportowner

```
select 'Non-editioned '||TYPE||' '||OWNER||'.'||NAME ||', References '||REFERENCED_TYPE||'
'||REFERENCED_OWNER||'.'||REFERENCED_NAME|| ' (To Be Editioned) with Reference Type
'||DEPENDENCY_TYPE Message from dba_dependencies , dba_users

where REFERENCED_OWNER='REPORTOWNER'

and REFERENCED_TYPE in ('FUNCTION','LIBRARY','PACKAGE','PACKAGE
BODY','PROCEDURE','TRIGGER','TYPE','TYPE BODY','SYNONYM','VIEW','NON-EXISTENT')

and not (OWNER=REFERENCED_OWNER and TYPE='PACKAGE BODY' and REFERENCED_TYPE='PACKAGE' )

and not (REFERENCED_OWNER='SYS' and REFERENCED_NAME='STANDARD' and REFERENCED_TYPE like
'PACKAGE%' )

and OWNER=USERNAME

and EDITIONS_ENABLED='N' order by 1;
```

Restoring the Database to a Restore Point

Use the following commands to flashback to the guaranteed restore point if needed.

```
sqlplus / as sysdba;
select current_scn from v$database;
shutdown immediate;
startup mount;
select * from v$restore_point;
flashback database to restore point BEFORE_643;
alter database open resetlogs;
select current_scn from v$database;
```

Starting OTM 6.4.3

Go to the $GLOG_HOME/install/webserver and run the following command.

```
./glogweb-wl start
```

When the command completes run the following command to view the console.log.0 file for possible problems during startup.

```
tail -f ../../logs/console/console.log.0
```

Common Problems

Many clients moving from older versions of OTM to 6.4.3 report the following problem when trying to start the instance after completing the migration.

```
INFO | 2024/01/12 13:28:27 | [GC (Allocation Failure)  1707534K->744370K(3988992K), 0.1750618
secs]
INFO | 2024/01/12 13:30:49 | <Jan 12, 2024 1:30:49,499 PM GMT> <Notice> <Security> <BEA-090078>
<User guest in security realm myrealm has had  5 invalid login attempts, locking account for 30
minutes.>
INFO | 2024/01/12 13:30:49 | <Jan 12, 2024 1:30:49,504 PM GMT> <Notice> <Security> <BEA-090078>
<User guest in security realm myrealm has had  5 invalid login attempts, locking account for 30
minutes.>
```

This is a common problem since the password requirements for the guest and other users changed to require the password be 8 characters and have at least 1 number. Many clients need to run the update_password.sh command to set the password for this user in the gl_user table to same value used during installation of 6.4.3.

Review the installer.properties used for the 6.4.3 installation to verify the password you have for the guest and otmSystem password in the wallet.shared file. You will need the otmSystem password to run the update_password.sh script.

```
OTM_SYSTEM_PASSWORD=<password>
GUEST_USER_PASSWORD=<password>
```

Shut the OTM 6.4.3 instance down

Go to the $GLOG_HOME/glog/oracle/script8 directory and run the update password script.

```
./update_password.sh
```

Below is the script output and expected response.

```
No user should run this script except Application Administrator.
Enter Application Administrator user name (Press Enter for default of otmSystem): Press Enter
Using default administrator user otmSystem
Enter Application Administrator password:  <OTM_SYSTEM_PASSWORD value>
Enter guest user password if application is running, otherwise Press Enter: Press Enter
Enter the glog properties path (Press Enter for default of ../../config): Press Enter
Using default property path ../../config
Enter user name for which password need to be updated: guest
Enter new password: <GUEST_USER_PASSWORD value>
Enter Database TNS connection id: <DB SID>
```

You should see something similar to this being displayed.

```
TNSSTR: <DB Server Name> <DB Server Port> <DB SID>
Attempting to change Password for user guest on database <DB SID>
Attempting to change Password for user guest on database <DB SID>
guest: 1 password changed
```

The log file that is generated, update_password_<DB SID>_<YYYYMMDD>_<HHMMSS>.log, should show content similar to this.

```
Attempting to change Password for user guest on database <DB SID>
guest: 1 password changed
```

Restart the instance again after making this change.

Moving to 19C Database

Review the information on the following note if you are moving from 12c to 19c database.

- Reported Issues and Resolution for OTM 6.4.3 Using Oracle 19C Database (Doc ID 2695259.1)

Exporting and Importing Data in 6.4.3 and above

Review the information in the following note to export and import data from 6.4.3 and above.

- Cloning Oracle Transportation Management Database Schemas for 6.4.3 and 6.5.x (Doc ID 2593800.1)