SSMA for Oracle common errors and how to fix it

Prepared by

Paula Berenguel

Date: Mar-18

This document was written in a context of an Oracle 11g to Azure SQL DB migration project.

1. O2SS0001: SQL Server Migration Assistant for Oracle Error message: Unsupported SQL clause(1)

Oracle allows you to assign a non-scalar condition in WHERE clause. The problem is that SQL Server doesn't support conditions of this type. Thus, the SQL Server Migration Assistant (SSMA) for Oracle doesn't convert queries with a a non-scalar condition in WHERE clause. So, when you try to convert the source code that includes a non-scalar condition in a WHERE clause, SSMA will generate an error O2SS0001.

```
SELECT *
FROM test_functions.test_func_emp_constr
WHERE (first_name, last_name) IN
(
SELECT 'SASHA','PIT'
FROM dual
UNION
SELECT 'BRED', 'PIT'
FROM dual
);

SELECT *
FROM test_functions.test_func_emp_constr
WHERE (first_name, last_name) NOT IN
(
SELECT 'SASHA','PIT'
FROM dual
UNION
SELECT 'BRED', 'PIT'
FROM dual
);
```

Possible Remedies:

In order to resolve this issue, you should rewrite the converted query using EXISTS condition. This condition tests for the existence of rows in a subquery. Then you should substitute the AND condition with IN condition. Also, replace the OR condition with NOT IN. Thus, the updated SQL Server code will look as follows:

```
select *
from dbo.test_func_emp_constr e
where EXISTS (
select 1
```

```
where e.first_name IN ('BRED','SASHA')
AND e.last_name IN ('PIT'))
select *
from dbo.test_func_emp_constr e
where EXISTS (
select 1
where e.first_name NOT IN ('BRED','SASHA')
OR e.last_name NOT IN ('PIT'))
```

2. O2SS0004: Unparsed SQL

Possible Remedies:

- Replace Pivot with exists, aliases or case;
- Create Linked server for the Oracle Database Links

https://www.dbbest.com/blog/oracle-sql-server-migration-ssma-convert-pivot-operator/

3. O2SS0005: Source datatype not recognized

Possible Fixes:

Oracle has an object data type called *anydata*. This data type supports wide range of data types. For example when creating a table with a column defined as *anydata* type, the column can store many types of data from string to numeric . SSMA does not support migration of *anydata* and when migrating Oracle database containing the type, SSMA raise a migration error **O2SS0005: Source datatype not recognized.**

MSSQL has a similar data type called *sql_variant*

MSSQL Server what can come close to TABLE OF TableName%RowType is Declare a table type variable (in case it is collection). Another way may be create a #temp table. But temp table method may involve IO which need to be aware of and also in there are limitation where you can create #temp table. For a single row, it is better to declare separate scalar variable for each columns.

Table variable Method
Declare @Dept Table (Department_id Int,Name Varchar(100),Description Varchar(1000))
The below statement will insert all the rows from Department to @Dept variable.
Select *INTO @Dept
FROM
Department

Temp table method
Select *into #Temp_Dept
FROM Department
The above statement will create a temp table on the fly. This can be used in Procedure but not in Function.

4. O2SS0006: Type ‘INTERVAL DAY(5) TO SECOND(3)’

Possible Remedies:

https://www.dbbest.com/blog/ssma-convert-interval-expressions/

5. O2SS0013: SQL Server Migration Assistant for Oracle Error message

Possible Remedies:

Execute Immediate, needs to be replaced manually for the SQL equivalent. Guidance on how to do that available on:

https://channel9.msdn.com/Events/Ignite/Microsoft-Ignite-Orlando-2017/BRK3356

6. O2SS0029 Cannot convert EXIT statement

Possible Remedies:
http://migration572.rssing.com/chan-6919524/all_p1.html#item2

7. O2SS0050: SQL Server Migration Assistant for Oracle Error message: Conversion of identifier is not supported(1)

Possible Remedies:
In order to resolve this issue, you should convert all aggregate expressions manually. Guidance on:
http://migration572.rssing.com/chan-6919524/all_p1.html#item2

8. O2SS0083: SQL Server Migration Assistant for Oracle Error message: Unresolved identifier(1165)

Possible Remedies:
Error is commonly related to the existence of database links, create a linked server or use openrowset. More guidance on:
https://www.dbbest.com/blog/ssma-convert-database-links/

9. O2SS0086: The literal 'INTERVAL YEAR TO MONTH'

Possible Remedies:
https://www.dbbest.com/blog/ssma-convert-interval-literals/

10. O2SS0217 Call to identity Sequence CURRVAL not supported

Possible Remedies:
https://blogs.msdn.microsoft.com/ssma/2011/06/29/migrating-oracle-to-sql-server-using-ssma-error-o2ss0217-call-to-identity-sequence-currval-not-supported/

11. O2SS0231: SQL Server Migration Assistant for Oracle Error message: Foreign key cannot be converted(1)

Possible Remedies:
https://www.dbbest.com/blog/oracle-sql-server-convert-foreign-keys/

12. O2SS0245: Cursor conversion in return statements not supported

Possible Remedies:
https://blogs.msdn.microsoft.com/ssma/2011/05/19/migrating-oracle-to-sql-server-using-ssma-error-o2ss0245-cursor-conversion-in-return-statements-not-supported/

13. O2SS0264: Unable to convert cursor or cursor variable as a function or procedure call parameter

Possible Remedies:
https://blogs.msdn.microsoft.com/ssma/2011/06/20/migrating-oracle-to-sql-server-using-ssma-o2ss0264-unable-to-convert-cursor-or-cursor-variable-as-a-function-or-procedure-call-parameter/

https://www.dbbest.com/blog/oracle-sql-server-convert-cursor/

14. O2SS0302: Unable to convert recursive call in inline call conversion

Possible Fix:
It is caused by the fact that SQL doesn't support calling a function from the function itself.
In pseudo code your issue looks like:

```
Function couper_texte(string){
        PerformOperationSubstring
        Do While (NOTOK){
Couper_texte(string)   }
Return string
}
```

A solution here should be implemented from the Oracle side to be a quick win:

```
Function couper_texte(string){
Do While (NOTOK){
        Couper_texte_2(string) }
Return string }
 Function couper_texte_2(string){        PerformOperationSubstring}
```

By creating a new function with the original name we make sure that not a single call to it in other functions/procedures has to be adjusted.

15. O2SS0339: Cannot convert usage of standalone user-defined type

Possible Remedies:
https://www.dbbest.com/blog/oracle-sql-server-migration-ssma-convert-unsupported-table-expressions/

16. O2SS0343 FORALL statement with SAVE EXCEPTION clause is not supported

Possible Remedies:
http://migration572.rssing.com/chan-6919524/all_p1.html#item3

17. O2SS0359 cannot get description for return type of function call expression

Possible Remedies:
https://blogs.msdn.microsoft.com/ssma/2011/06/20/migrating-oracle-to-sql-server-using-ssma-o2ss0359-cannot-get-description-for-return-type-of-function-call-expression/

18. O2SS0408 Collection or Record Type is not supported

Possible Remedies:
http://migration572.rssing.com/chan-6919524/all_p1.html

19. O2SS0456: SQL Server Migration Assistant for Oracle Error message: User defined types conversion is not supported(1)

Possible Remedies:
https://channel9.msdn.com/Events/Ignite/Microsoft-Ignite-Orlando-2017/BRK3356

20. O2SS0474: SQL Server Migration Assistant for Oracle Error message: User defined type variable not converted(1)

Possible Remedies:
https://www.dbbest.com/blog/oracle-sql-server-migration-ssma-convert-unsupported-table-expressions/

21. O2SS0482: Conversion of following TABLE expression is not supported: TABLE()

Possible Remedies:
https://www.dbbest.com/blog/oracle-sql-server-migration-ssma-convert-unsupported-table-expressions/

22. O2SS0516: SQL Server Migration Assistant for Oracle Error message: Init block can't be used inside function

Possible Fix:

In Oracle we can create a special block that will initialise the package before the first usage by user session. In Azure SQL DB, the initialization procedure can't be run from the context of a function.

As a workaround the call for ssma_oracle.db_check_init_package procedure is placed before the calling package function or package variables getters inside the code that use them but not inside the function.

SSMA replaces the initialization function call to ssma_oracle.db_check_init_package procedure call when is possible, but in some cases this should be done manually depending on the code that uses the converted package and its routines.

23. O2SS0518: SQL Server Migration Assistant for Oracle Error message: Wrapper functions are not supported by SQL Azure platform(578) Estimated manual conversion time: 1156 hr(s)

Possible Fix:
Package State Cannot Be Changed Inside Functions In Oracle, package variables values can be changed inside user defined functions.
For SQL Server, such functions are emulated using xp_ora2ms_exec2_ex extended stored procedure and implementation procedures. Azure SQL DB does not support extended stored procedures. That is why the above emulation is not applicable for this database.
SSMA marks the converted function that contains variables changed inside it with the following error message: "Wrapper functions are not supported by Azure SQL DB platform. Use $impl procedures instead."

24. O2SS0522: Materialized view with float type can't be converted (restriction)

Possible Remedies:
https://www.dbbest.com/blog/ssma-convert-materialized-view-float-type/

Feedback and suggestions

If you have feedback or suggestions for improving this data migration asset, please contact the Data Migration Jumpstart Team (askdmjfordmtools@microsoft.com). Thanks for your support!

Note: For additional information about migrating various source databases to Azure, see the Azure Database Migration Guide.