

Permission - No errors reported, but data not synced

Last updated by | Keith Elmore | Apr 5, 2021 at 7:57 AM PDT

Contents

- [Symptom](#)
- [Cause 1](#)
 - [Resolution](#)
- [Cause-2](#)
 - [Mitigation](#)
- [Cause-3](#)
- [Classification](#)

Symptom

Sync agent reported no error, but no data propagated to the destination

Sync agent log showed normal processing of sync

But no data was synced.

Cause 1

all triggers on hub/member databases got dropped somehow

Resolution

PG changed hub state & member state to ReProvFailed, which triggered a sync starting with a re-provision, which added all the sync related objects to the database. After the re-provision, the sync works fine,

If the tracking table already exists in the member database, we won't provision the tracking table even when triggers are missing.

Here is the action plan:

1. Run command below to drop the tracking table

```
DROP TABLE DataSync.OIM_USR_dss_tracking
```

2. Go back to your sync metadata database, and change memberstate to 0 for this member in dss.syncgroupmember. Provide your member name in below query.

```
UPDATE dss.[syncgroupmember] SET memberstate = 0 WHERE name = 'xxx'
```

3. Trigger sync again so it will do provisioning now which should add all the sync related objects to the database.

You may also change it to 17.

Here is the difference between 0 and 17.

0 means **Unprovisioned**, so we will create everything from scratch (skip creating triggers if the tracking tables already exist).

17 means **UnReprovisioned**, which indicate a reprovision operation is pending. So it will also create everything that is missing.

Both of them almost have the same effect, except that 0 will assume that we didn't sync before. In this case, setting state to 0 will also clean up the knowledge, while setting to 17 will still keep the existing knowledge so we are not start from the beginning.

Cause-2

If they use bulk insert DSS will not work

This is by design because Data Sync tracks changes using insert, update, and delete triggers. The changes are recorded in a side table in the user database.


Mitigation

They need to add the option FIRE_TRIGGERS to the bulk insert

<https://docs.microsoft.com/en-us/sql/t-sql/statements/bulk-insert-transact-sql?view=sql-server-2017> 

<https://github.com/MicrosoftDocs/azure-docs/pull/19877> 

Cause-3

DSS will not work if they use Truncate table. TRUNCATE TABLE cannot activate a trigger because the operation does not log individual row deletions. <https://docs.microsoft.com/en-us/sql/t-sql/statements/truncate-table-transact-sql?view=sql-server-ver15> 

Classification

Root cause Tree - DataSync/Service issue/SyncFail

How good have you found this content?



-