Data exists in destination prior to sync setup

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

Contents

Issue

If both the source and destination databases have the same data before setting up the data sync, such as a scenarios that delete and re-create a sync group, or destination is a copy of the source, customer can avoid expensive provisioning by using noinitsync feature.

The noinitsync feature is tested, and allow listed (ETA is end of Dec or Jan).

Option 1

For now, we can use below steps to accomplish:

- Create sync group with hub database.
- Add sync member
- Add the tables and save, DO NOT SYNC now.
- Connect to the metadata database and reset the noinitsync state for the member:

```
update [dss].[syncgroupmember] set [noinitsync]=1 where name = '<sync member name>'
```

- Click sync.

The initial provisioning will still be performed, all sync objects will be added to the destination, but data processing will be skipped.

Please make sure the source and the destination indeed have the same data, and no changes made while setting up the sync.

Option 2

In case customer already had a sync group:

1. Drop the old sync group, and wait for deprovision to complete.
2. **Make sure the data is consistent between hub and member.**

3. Create sync group with hub database.
4. Add sync member
5. Refresh schema (either hub or member).
6. Add the tables you want to sync and save, **DO NOT SYNC** now.
7. Connect to sync database

```
update [dss].[syncgroupmember] set [noinitsync]=1 where name = '<sync member name>'
```

8. Click sync. The initial sync will not happen.

In case you already have two heads (new data exists on both sides) please refer to this blog article to recommended way to setup the sync:
https://blogs.msdn.microsoft.com/azuresqldbsupport/2018/03/08/azure-data-sync-resolve-slow-initial-sync-on-two-way-sync/

Classification

Root cause Tree - DataSync/User Issue/error/Uncoded

How good have you found this content?