Distribution Agent errors 2601 2627 20598

Last updated by | Vitor Tomaz | Jun 8, 2022 at 5:37 AM PDT

Contents

Issue

The Distribution Agent is failing with either of the following error messages:

- Error 20598: The row was not found at the Subscriber when applying the replicated command
- Error 2601: Cannot insert duplicate key row in object 'xxxx' with unique index 'yyyy'. The duplicate key value is (zzzz).
- Error 2627: Violation of PRIMARY KEY constraint 'xxxx'. Cannot insert duplicate key in object 'yyyy'. The duplicate key value is (zzzz).

These errors are closely related, they share the same potential root cause and the same approach to troubleshooting:

- Error 20598 usually occurs if you update or delete a row at the Publisher which no longer exists at the Subscriber.
- Errors 2601 and 2627 usually occur if you insert a row at the Publisher which already or still exists at the Subscriber.

It is a rather nasty error that is not easy to fix or mitigate. The customer will have to make some tough decisions for resolving the issue (see troubleshooting section below).

Troubleshooting:

The troubleshooting challenge is that the root cause for this error is usually in the past, but we only see the error symptom now. In addition to mitigating the issue, you need to learn as much as possible from the current

symptoms in order to deduce what has led to the issue (maybe several weeks back). Otherwise the same event might happen again any time. Please see the More Information section below for further details.

1. Disable the Distribution cleanup job

Disabling the SQL Agent job "Distribution clean up: Distribution" will avoid the expiration of the failing subscription. The subscription will be marked as expired 72 hours after the distribution agent started to fail. The disadvantage is that the size of the distribution database will grow while you continue troubleshooting.

This step could buy you valuable time for troubleshooting and/or deciding on an action plan with the customer.

2. Turn off Distribution History Cleanup

Disabling the SQL Agent job "Agent history clean up: Distribution" keeps the history message in place for further review.

3. Get the last replicated transaction ID from the subscriber database:

```
SELECT [transaction_timestamp], * FROM [MSreplication_subscriptions]
```

4. Record the complete error message

You will find the errors 2601/2627/20598 in Replication Monitor, but you can also see them by querying MSdistribution_history. The key data to take from Replication Monitor is the Transaction Sequence Number and Command ID.
The message in Replication Monitor will look something like this:

```
Command attempted:
if @@trancount > 0 rollback tran
(Transaction sequence number: 0x0000002500000167000400000000, Command ID: 1)
```

5. Identify the failing data change

Take the Transaction Sequence Number from the error message and pass it in as a parameter to the system stored procedure `sp_browsereplcmds`.
Please drop any trailing zeros after the last non-zero number:

```
exec distribution..sp_browsereplcmds '0x00000025000001670004', '0x00000025000001670004'
```

The output may include several rows. Try to match up the `command_id` column with the ID given in the error message. When you find the correct command_id, you will notice that there is a `command` column that includes the actual command being replicated. If you can't isolate the actual command that is failing, you may need to look at the other rows in the output until you find one that references a row that doesn't exist at the subscriber.

Here is a sample command retrieved from sp_browsereplcmds:

```
{CALL [sp_MSupd_dboTable_1] (,N'firstdat  ',,1,0x02)}
```

It tells us that we are running the update procedure for a table "Table_1". The primary key value is listed second to last and for this row it is "1". For sp_Msupd_ stored procedures, the primary key will always appear at the end of the column list.
This means that our missing row is where Primary Key = 1. The string N'firstdat ' is the data being modified. Ignore the hex value 0x02.

6. Discuss with customer to get an idea where this change originated

This could render valuable hints about what might have caused the issue:

- Was this data change part of a batch job?
- Was there anything else running in parallel, e.g. some maintenance task?
- Has this occurred while the publication was changed by the customer, e.g. adding articles or subscriptions?
- Is there anything specific about the data change itself? For example, is an update performed as Delete/Insert by the application? Or is an Insert immediately followed by an Update?

Mitigation

Either of the following steps will help to overcome the immediate error:

- Reinitialize the subscription from a new snapshot - **THIS IS THE PREFERRED SOLUTION**
  Reinitializing a subscription will make the Distribution Agent to bulk copy all data from the publication table(s) to the Subscriber table(s) for the next synchronization. This is the clean solution for this issue, but might be very painful for the customer, especially if the replicated database is very large. It might take several hours or even days to complete.

- Insert the row(s) missing at the Subscriber for a failing Update/Delete.
  This might help if only one or a few rows are affected; often enough a lot more rows are affected, but it is still worth a try.

- Use the -SkipErrors parameter on the Distribution Agent to skip errors 2601:2627:20598. Or use the corresponding agent profile "Continue on data consistency errors"
  The distribution agent will overcome the immediate error, but there is a high probability that data remains different on Publisher and Subscriber, causing data inconsistencies between the Publisher and the Subscriber tables. This may or may not be acceptable for the customer.

- Drop the affected article from the publication and re-add it later.
  Applies if you are sure that only this one article is affected, and if the article is not referencing any other articles in the publication. It may or may not resolve the issue.

Do not forget to revert any changes you have made in the troubleshooting steps - especially regarding the cleanup agents.

More Information

Although you have a definite datetime for the first occurrence of the error, it is very likely not the time of the event that had caused the issue. The cause might have happened any time between the last (re-)initialization of the subscription and the occurrence of the first error.

If the customer is asking for an RCA, you will need to establish if the root cause is inside or outside of replication:

- If it is inside of replication, either the Log Reader or the Distribution agent might have skipped an earlier change.
- If it is outside of replication, the customer might have changed the data at the subscriber, or made a mistake while initializing the subscription.

List of possible causes, ordered descending by their likelihood

1. The customer has inserted/updated/deleted data in the subscriber database (which is supposed to be handled as read-only data), thus causing a mismatch between Publisher and Subscriber data.
2. The subscription had been created with the "replication support only" sync_type option, and the Subscriber database was not a 1:1 copy of the Publisher database when the subscription had been set up.
3. If a filtered article is affected, there might be a problem with the filter configuration
4. The Insert/Update/Delete procedures at the Subscriber were customized by the customer; e.g. a preceding Insert command might have been skipped which now fails the current Update or Delete.
5. The customer has enabled the Delayed Durability feature on the Publisher or Subscriber database.
6. A defect in the Log Reader or Distribution agent.

Discussion of the possible causes

1. This is the most likely reason for the issue. A user or application might have connected to the Subscriber database and changed the data by mistake or unknowingly.
2. This might have occurred after a previous replication issue, e.g. if the subscription was expired and the customer recreated it without applying a new snapshot. Pending changes between the start of the earlier issue and the re-create of the subscription were lost, thus causing a data mismatch between Publisher and Subscriber.
3. This is a rather obscure issue; to detect it, you need the customer to provide you with a script of the affected publication.
4. Another rather obscure issue; the customer might know if they are using custom stored procedures at the Subscriber.
5. Delayed Durability opens the door for data loss if the SQL service is restarted before writing the transaction log to disk. In this case, the replication might have replicated changes to the Subscriber that are then rolled back at the Publisher. The customer can check if this option has been enabled in their topology.
6. This is unlikely but not impossible, and extremely hard to narrow down. It requires transaction log backups of the Publisher database and backups of the distribution database to confirm it as cause, thus always needs an IcM if such a defect is suspected.

How good have you found this content?