

This method of skipping transactions is not suitable when you have enabled GTID assignment on a replication channel using the `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS` option of the `CHANGE REPLICATION SOURCE TO` statement.

If the failing transaction generated an error in a worker thread, you can obtain its GTID directly from the `LAST_SEEN_TRANSACTION` field in the Performance Schema table `replication_applier_status_by_worker`. To see what the transaction is, issue `SHOW RELAYLOG EVENTS` on the replica or `SHOW BINLOG EVENTS` on the source, and search the output for a transaction preceded by that GTID.

When you have assessed the failing transaction for any other appropriate actions as described previously (such as security considerations), to skip it, commit an empty transaction on the replica that has the same GTID as the failing transaction. For example:

```
SET GTID_NEXT='aaa-bbb-ccc-ddd:N';
BEGIN;
COMMIT;
SET GTID_NEXT='AUTOMATIC';
```

The presence of this empty transaction on the replica means that when you issue a `START REPLICATION | SLAVE` statement to restart replication, the replica uses the auto-skip function to ignore the failing transaction, because it sees a transaction with that GTID has already been applied. If the replica is a multi-source replica, you do not need to specify the channel name when you commit the empty transaction, but you do need to specify the channel name when you issue `START REPLICATION | SLAVE`.

Note that if binary logging is in use on this replica, the empty transaction enters the replication stream if the replica becomes a source or primary in the future. If you need to avoid this possibility, consider flushing and purging the replica's binary logs, as in this example:

```
FLUSH LOGS;
PURGE BINARY LOGS TO 'binlog.000146';
```

The GTID of the empty transaction is persisted, but the transaction itself is removed by purging the binary log files.

## Skipping Transactions Without GTIDs

To skip failing transactions when GTIDs are not in use or are being phased in (`gtid_mode` is `OFF`, `OFF_PERMISSIVE`, or `ON_PERMISSIVE`), you can skip a specified number of events by issuing a `SET GLOBAL sql_slave_skip_counter` statement. Alternatively, you can skip past an event or events by issuing a `CHANGE MASTER TO` statement to move the source binary log position forward.

These methods are also suitable when you have enabled GTID assignment on a replication channel using the `ASSIGN_GTIDS_TO_ANONYMOUS_TRANSACTIONS` option of the `CHANGE REPLICATION SOURCE TO` statement.

When you use these methods, it is important to understand that you are not necessarily skipping a complete transaction, as is always the case with the GTID-based method described previously. These non-GTID-based methods are not aware of transactions as such, but instead operate on events. The binary log is organized as a sequence of groups known as event groups, and each event group consists of a sequence of events.

- For transactional tables, an event group corresponds to a transaction.
- For nontransactional tables, an event group corresponds to a single SQL statement.

A single transaction can contain changes to both transactional and nontransactional tables.