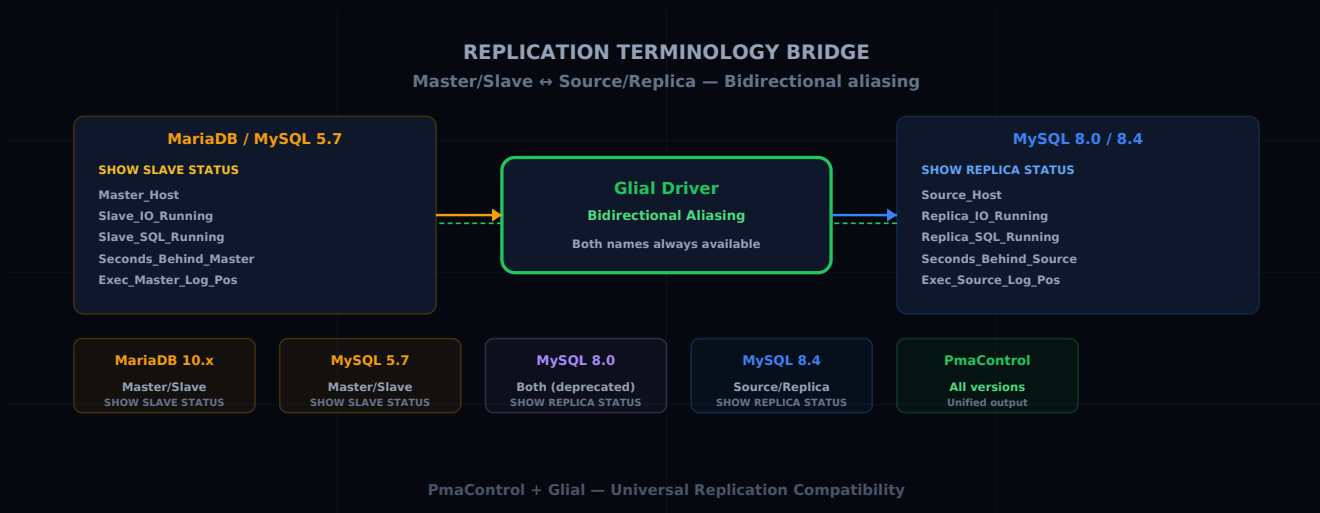


# MySQL 8.0 Replication: Bridging Source/Replica and Master/Slave

Aurélien LEQUOY · April 8, 2026

MYSQL REPLICATION COMPATIBILITY MYSQL-8.0 MIGRATION



## The Rename

MySQL 8.0.22 introduced a major terminology change: the terms **Master** and **Slave** in replication commands and output columns were replaced by **Source** and **Replica**.

This is not just a cosmetic change. Concretely:

- SHOW SLAVE STATUS becomes SHOW REPLICATION STATUS
- CHANGE MASTER TO becomes CHANGE REPLICATION SOURCE TO
- The Master\_Host column in the output becomes Source\_Host
- Slave\_IO\_Running becomes Replica\_IO\_Running
- Slave\_SQL\_Running becomes Replica\_SQL\_Running
- Seconds\_Behind\_Master becomes Seconds\_Behind\_Source

And so on for about thirty columns.

MySQL 8.0 maintains backward compatibility: the old commands still work (with a deprecation warning). But MySQL 8.4 starts removing old aliases. And MariaDB, on its side, keeps the

historical terminology.

## The Problem for Monitoring Tools

Any tool that parses the output of `SHOW SLAVE STATUS` / `SHOW REPLICHA STATUS` must now handle **two sets of column names** depending on the server version:

Old (MariaDB, MySQL 5.7)	New (MySQL 8.0+)
Master_Host	Source_Host
Master_User	Source_User
Master_Port	Source_Port
Master_Log_File	Source_Log_File
Read_Master_Log_Pos	Read_Source_Log_Pos
Slave_IO_Running	Replica_IO_Running
Slave_SQL_Running	Replica_SQL_Running
Slave_IO_State	Replica_IO_State
Seconds_Behind_Master	Seconds_Behind_Source
Last_IO_Error	Last_IO_Error
Last_SQL_Error	Last_SQL_Error
Exec_Master_Log_Pos	Exec_Source_Log_Pos

The problem is twofold:

1. **Existing code** uses the old column names everywhere. Hundreds of references to `$row['Master_Host']` or `$row['Slave_IO_Running']`.
2. **The infrastructure is mixed:** MariaDB 10.6 and 10.11, MySQL 5.7 at end-of-life, MySQL 8.0, freshly deployed MySQL 8.4. All in the same PmaControl.

## The Naive Solution (and Why It Fails)

The first idea: detect the version and use the right column name.

```

// Don't do this
if ($version >= '8.0.22') {
    $host = $row['Source_Host'];
    $io_running = $row['Replica_IO_Running'];
} else {
    $host = $row['Master_Host'];
    $io_running = $row['Slave_IO_Running'];
}

```

This pattern is catastrophic to maintain. Every place in the code that reads a replication field must be duplicated. With 30 columns and dozens of code locations, you end up with hundreds of conditions.

And when MySQL 9.0 adds new aliases? Triple the code?

## The Glial Approach: Bidirectional Aliasing at the Driver Level

The solution implemented in the Glial framework (which provides PmaControl's MySQL access layer) is more elegant: **bidirectional aliasing at the driver level**.

When the Glial driver executes `SHOW REPLICA STATUS` (or `SHOW SLAVE STATUS`), it enriches the result by adding **both names** for each column:

```

// In Glial's MySQL driver
$replication_aliases = [
    'Master_Host'          => 'Source_Host',
    'Master_User'         => 'Source_User',
    'Master_Port'         => 'Source_Port',
    'Master_Log_File'     => 'Source_Log_File',
    'Read_Master_Log_Pos' => 'Read_Source_Log_Pos',
    'Slave_IO_Running'    => 'Replica_IO_Running',
    'Slave_SQL_Running'   => 'Replica_SQL_Running',
    'Slave_IO_State'      => 'Replica_IO_State',
    'Seconds_Behind_Master' => 'Seconds_Behind_Source',
    'Exec_Master_Log_Pos' => 'Exec_Source_Log_Pos',
    // ... all pairs
];

// After fetching the result
foreach ($replication_aliases as $old => $new) {

```

```
if (isset($row[$old]) && !isset($row[$new])) {
    $row[$new] = $row[$old];
}
if (isset($row[$new]) && !isset($row[$old])) {
    $row[$old] = $row[$new];
}
}
```

The result: **consuming code can use either the old or the new name interchangeably.** Whether the server is MariaDB 10.6, MySQL 5.7, MySQL 8.0 or MySQL 8.4, both names always exist in the result.

## The Advantages of This Approach

---

### 1. Zero Changes to Existing Code

The hundreds of references to `$row['Master_Host']` in PmaControl continue to work. No code migration required.

### 2. New Code Can Use the New Terminology

Developers writing new code can use `$row['Source_Host']` right now. When old code is gradually refactored, the transition will be transparent.

### 3. Single Maintenance Point

The aliasing is defined once, in the driver. If MySQL 9.0 adds new fields or renames other columns, you only need to extend the aliases array.

### 4. Compatibility with Both SHOW Commands

The driver first attempts `SHOW REPLICA STATUS`. If the command fails (MySQL 5.7, MariaDB), it falls back to `SHOW SLAVE STATUS`. In both cases, the result contains both sets of names.

```
try {
    $result = $db->query('SHOW REPLICA STATUS');
} catch (Exception $e) {
    $result = $db->query('SHOW SLAVE STATUS');
}
```

```
// In all cases, $result contains Master_Host AND Source_Host
```

## The Multi-Source Case

MySQL 8.0 supports multi-source replication (multiple channels). The `SHOW REPLICA STATUS` command then returns **multiple rows**, one per channel. The aliasing applies to each row individually.

MariaDB also supports multi-source replication but with a different syntax ( `SHOW SLAVE 'channel_name' STATUS` or `SHOW ALL SLAVES STATUS` ). The Glial driver normalizes these differences so that PmaControl receives a uniform format.

## Impact on PmaControl

Thanks to this aliasing in the Glial driver, PmaControl transparently handles:

- **MariaDB 10.6 / 10.11**: Master/Slave terminology, `SHOW SLAVE STATUS`
- **MySQL 5.7**: Master/Slave terminology, `SHOW SLAVE STATUS`
- **MySQL 8.0**: Source/Replica terminology, `SHOW REPLICA STATUS` (with old fallback)
- **MySQL 8.4**: Source/Replica terminology only, `SHOW REPLICA STATUS`

The PmaControl interface displays replication information uniformly, regardless of the server version. The replication dashboard shows `Source_Host` in columns, but internal queries work with both names.

## Commands Too

The rename does not only affect output columns. The SQL commands themselves were renamed:

Old	New
<code>CHANGE MASTER TO</code>	<code>CHANGE REPLICATION SOURCE TO</code>
<code>START SLAVE</code>	<code>START REPLICA</code>
<code>STOP SLAVE</code>	<code>STOP REPLICA</code>
<code>RESET SLAVE</code>	<code>RESET REPLICA</code>

Old	New
<code>SHOW SLAVE HOSTS</code>	<code>SHOW REPLICAS</code>

PmaControl uses the same approach: attempt the new command, fall back to the old one.

## Recommendations for Your Own Tools

---

If you maintain scripts or tools that parse replication output:

1. **Do not use version-based conditions** — they are fragile and do not scale
2. **Implement bidirectional aliasing** at the lowest possible level (driver, abstraction layer)
3. **Write new code with the new terminology** — Source/Replica
4. **Test on all 4 major versions:** MariaDB 10.x, MySQL 5.7, MySQL 8.0, MySQL 8.4
5. **Prepare for MySQL 9.x** — old aliases may disappear completely

## Conclusion

---

The Master/Slave to Source/Replica rename in MySQL 8.0 is a seemingly simple change that can potentially break any monitoring, backup, or orchestration tool that parses replication output.

The clean solution is bidirectional aliasing at the driver level: a single implementation, zero changes to consuming code, full compatibility from MariaDB 10.x through MySQL 8.4.

This is what the Glial framework does, and this is what allows PmaControl to monitor mixed MariaDB / MySQL infrastructures without version-specific configuration.