

MySQL 8.0 Réplication : Réconcilier Source/Replica avec Master/Slave

Aurélien LEQUOY · 8 avril 2026

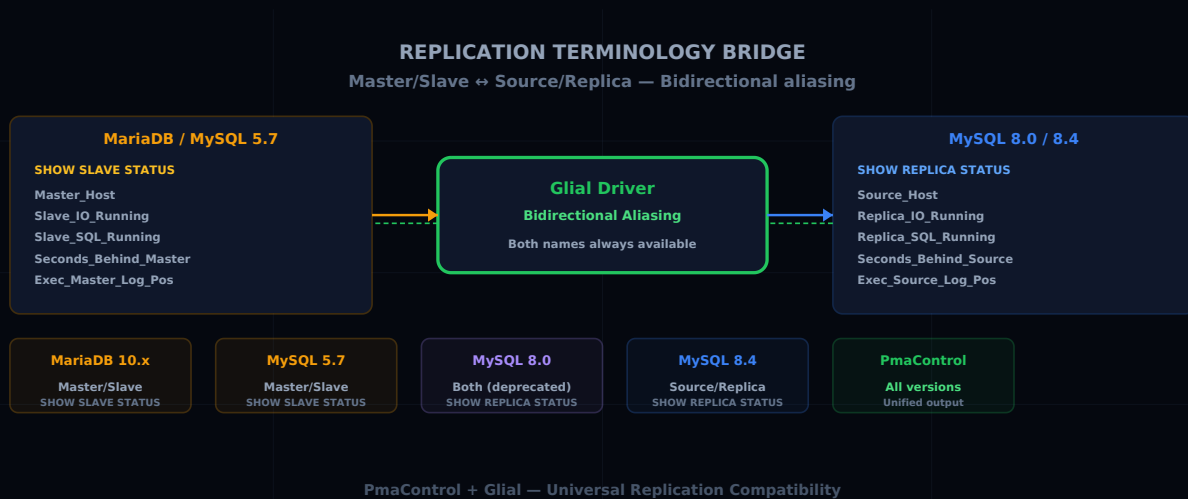
MYSQL

REPLICATION

COMPATIBILITY

MYSQL-8.0

MIGRATION



Le renommage

MySQL 8.0.22 a introduit un changement terminologique majeur : les termes **Master** et **Slave** dans les commandes et les colonnes de sortie de réplication ont été remplacés par **Source** et **Replica**.

Ce n'est pas qu'un changement cosmétique. Concrètement :

- `SHOW SLAVE STATUS` devient `SHOW REPLICATION STATUS`
- `CHANGE MASTER TO` devient `CHANGE REPLICATION SOURCE TO`
- La colonne `Master_Host` dans la sortie devient `Source_Host`
- `Slave_IO_Running` devient `Replica_IO_Running`
- `Slave_SQL_Running` devient `Replica_SQL_Running`
- `Seconds_Behind_Master` devient `Seconds_Behind_Source`

Et ainsi de suite pour une trentaine de colonnes.

MySQL 8.0 maintient la compatibilité : les anciennes commandes fonctionnent encore (avec un deprecation warning). Mais MySQL 8.4 commence à supprimer les anciens alias. Et MariaDB, de son côté, conserve la terminologie historique.

Le problème pour les outils de monitoring

Tout outil qui parse la sortie de `SHOW SLAVE STATUS` / `SHOW REPLICA STATUS` doit maintenant gérer **deux jeux de noms de colonnes** selon la version du serveur :

Ancien (MariaDB, MySQL 5.7)	Nouveau (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

Le problème est double :

1. **Le code existant** utilise les anciens noms de colonnes partout. Des centaines de références à `$row['Master_Host']` ou `$row['Slave_IO_Running']`.
2. **L'infrastructure est mixte** : MariaDB 10.6 et 10.11, MySQL 5.7 en fin de vie, MySQL 8.0, MySQL 8.4 fraîchement déployé. Tous dans le même PmaControl.

La solution naïve (et pourquoi elle ne marche pas)

La première idée : détecter la version et utiliser le bon nom de colonne.

```
// Ne faites pas ça
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'];
}
```

Ce pattern est catastrophique à maintenir. Chaque endroit du code qui lit un champ de réplication doit être dupliqué. Avec 30 colonnes et des dizaines d'endroits dans le code, on arrive à des centaines de conditions.

Et quand MySQL 9.0 ajoutera de nouveaux alias ? On triple le code ?

L'approche Glial : aliasing bidirectionnel au niveau du driver

La solution implémentée dans le framework Glial (qui fournit la couche d'accès MySQL de PmaControl) est plus élégante : **l'aliasing bidirectionnel au niveau du driver**.

Quand le driver Glial exécute `SHOW REPLICA STATUS` (ou `SHOW SLAVE STATUS`), il enrichit le résultat en ajoutant **les deux noms** pour chaque colonne :

```
// Dans le driver MySQL de Glial
$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',
    // ... toutes les paires
```

```
];

// Après fetch du résultat
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];
    }
}
}
```

Le résultat : **le code consommateur peut utiliser indifféremment l'ancien ou le nouveau nom**. Que le serveur soit MariaDB 10.6, MySQL 5.7, MySQL 8.0 ou MySQL 8.4, les deux noms existent toujours dans le résultat.

L'avantage de cette approche

1. Zéro modification du code existant

Les centaines de références à `$row['Master_Host']` dans PmaControl continuent de fonctionner. Aucune migration de code nécessaire.

2. Le nouveau code peut utiliser la nouvelle terminologie

Les développeurs qui écrivent du nouveau code peuvent utiliser `$row['Source_Host']` dès maintenant. Quand l'ancien code sera progressivement refactorisé, la transition sera transparente.

3. Un seul point de maintenance

L'aliasing est défini une seule fois, dans le driver. Si MySQL 9.0 ajoute de nouveaux champs ou renomme d'autres colonnes, il suffit d'étendre le tableau d'aliases.

4. Compatibilité avec SHOW SLAVE STATUS et SHOW REPLICA STATUS

Le driver tente d'abord `SHOW REPLICA STATUS`. Si la commande échoue (MySQL 5.7, MariaDB), il fait un fallback sur `SHOW SLAVE STATUS`. Dans les deux cas, le résultat contient les deux jeux de noms.

```
try {
    $result = $db->query('SHOW REPLICA STATUS');
} catch (Exception $e) {
    $result = $db->query('SHOW SLAVE STATUS');
}
// Dans tous les cas, $result contient Master_Host ET Source_Host
```

Le cas multi-source

MySQL 8.0 supporte la réplication multi-source (plusieurs channels). La commande `SHOW REPLICA STATUS` retourne alors **plusieurs lignes**, une par channel. L'aliasing s'applique à chaque ligne individuellement.

MariaDB supporte également la réplication multi-source mais avec une syntaxe différente (`SHOW SLAVE 'channel_name' STATUS` ou `SHOW ALL SLAVES STATUS`). Le driver Glial normalise ces différences pour que PmaControl reçoive un format uniforme.

Impact sur PmaControl

Grâce à cet aliasing dans le driver Glial, PmaControl gère de manière transparente :

- **MariaDB 10.6 / 10.11** : terminologie Master/Slave, `SHOW SLAVE STATUS`
- **MySQL 5.7** : terminologie Master/Slave, `SHOW SLAVE STATUS`
- **MySQL 8.0** : terminologie Source/Replica, `SHOW REPLICA STATUS` (avec ancien fallback)
- **MySQL 8.4** : terminologie Source/Replica uniquement, `SHOW REPLICA STATUS`

L'interface PmaControl affiche les informations de réplication de manière uniforme, quelle que soit la version du serveur. Le dashboard de réplication montre `Source_Host` dans les colonnes, mais les requêtes internes fonctionnent avec les deux noms.

Les commandes aussi

Le renommage ne concerne pas seulement les colonnes de sortie. Les commandes SQL elles-mêmes ont été renommées :

Ancien	Nouveau
--------	---------

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

PmaControl utilise la même approche : tenter la nouvelle commande, fallback sur l'ancienne.

Recommandations pour vos propres outils

Si vous maintenez des scripts ou outils qui parsent la sortie de réplication :

1. **N'utilisez pas de conditions sur la version** — c'est fragile et ne scale pas
2. **Implémentez l'aliasing bidirectionnel** au plus bas niveau possible (driver, couche d'abstraction)
3. **Écrivez du nouveau code avec la nouvelle terminologie** — Source/Replica
4. **Testez sur les 4 versions majeures** : MariaDB 10.x, MySQL 5.7, MySQL 8.0, MySQL 8.4
5. **Préparez-vous à MySQL 9.x** — les anciens alias pourraient disparaître complètement

Conclusion

Le renommage Master/Slave vers Source/Replica dans MySQL 8.0 est un changement simple en apparence mais qui casse potentiellement tout outil de monitoring, de backup ou d'orchestration qui parse la sortie de réplication.

La solution propre est l'aliasing bidirectionnel au niveau du driver : une seule implémentation, zéro changement dans le code consommateur, compatibilité totale de MariaDB 10.x à MySQL 8.4.

C'est ce que fait le framework Glial, et c'est ce qui permet à PmaControl de superviser des infrastructures mixtes MariaDB / MySQL sans configuration spéciale par version.