Local Installations - Peering Setup

Peering Setup - Local Installations:

ProVision versions 5.3.0 to 6.1.2

ProVision uses a locally-hosted mirror of the PeeringDB database in order to perform non-edit Peering functions. There are a few steps to take in order to set up your locally hosted instance to coordinate with PeeringDB information.

> As of PeeringDB 2.0, SQL dump files are no longer provided. If you are using ProVision 5.3.0 or higher, you must follow this new install process. If you are using a lower version of ProVision, then please follow the instructions in the previous version of this page.

1) Create a new database to store the PeeringDB data. This **must** be on the same server as the database which is used by ProVision.

2) Download, install, and use the PeeringDB Python Client to populate the database.

The PeeringDB documentation is available here: http://peeringdb.github.io/peeringdb-py/

3) Once this has been done, edit the ProVision global configuration file located here:

```
[ProVision Root]/data/globals.php
```

It must be updated with the following variables to inform ProVision of the location of this new install. The username and password fields correspond to the username and password of the MySQL account which has access to the database (Not the username and password to your PeeringDB account).

> This can, but does not have to be, the same MySQL user which is used for the ProVision database. However, the ProVision MySQL user **must** have at least READ access to the PeeringDB database.

```
$peeringdb_host = 'localhost';          // Database host must be the same for provision and peering
$peeringdb_username = 'username';       // username for the MySQL user
$peeringdb_password = 'password';       // password for the MySQL user
$peeringdb_db_name = 'peeringdb';       // name of the database in MySQL
```

4) Periodically sync with the PeeringDB server to get the latest updates. This can be done manually, or there are instructions in the PeeringDB documentation on how to automatically schedule syncs using cron (http://peeringdb.github.io/peeringdb-py/cli/#sync).

ProVision versions 7.x and newer

For ProVision versions 7.x and later, ProVision directly interfaces with PeeringDB's API to update exchange and peering data, caching the data for a default time of 12 hours.

This requires a PeeringDB account, and for the account credentials to be set in ProVision. The credentials may either be hard coded into globals.php, or set into the database via the Admin/Peering GUI. See Admin Preferences and Peering for detailed information.

Additional Peering constants may be added into globals.php to change the PeeringDB URL between the main and beta site (some users may find the beta site to have faster response times), and to adjust the PeeringDB cache TTL.

For real-time updates, TTL may be set to 0. However, some users may experience severe lag with a TTL = 0; we recommend using a 10 to 15 minute or greater TTL if this occurs.

Peering Constants

In globals.php, the following constants can be defined to tweak the Peering internals:

PEERINGDB_USERNAME

```
define('PEERINGDB_USERNAME', 'username');
```

Default value: none

The username for the account used to connect to the PeeringDB API

Instead of saving the username and password in the database, the values can be hard coded into globals.php

PEERINGDB_PASSWORD

```
define('PEERINGDB_PASSWORD', 'mypass');
```

Default value: none

The password for the account used to connect to the PeeringDB API

PEERINGDB_URL

```
define('PEERINGDB_URL', 'https://peeringdb.com/api/');
```

Default value: https://peeringdb.com/api/

The URL of the PeeringDB API. Alternate value: https://beta.peeringdb.com/api/

PEERINGDB_CACHE_TTL

```
define('PEERINGDB_CACHE_TTL', 43200);
```

Default value: 43200 (12 hours)

How often (in seconds) to purge the cached PeeringDB API calls. If a customer wants real time access, this can be set to 0.

If experiencing major lag issues with real time access, it is recommended to increase to increase the cache TTL from 0 to 5, 10, or 15 minutes.

Additional Information:

For additional information on working with Peering, see the following documentation sections:

Peering

Import Peering Sessions