

Geo-replication and parallel uploads

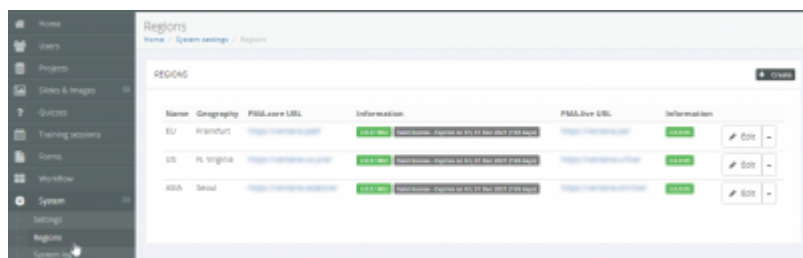
Sometimes you have slides that you want to upload to multiple destinations. Two examples of scenarios where this is appropriate may be:

- You're using [PMA.control](#) on a global scale. Your PMA.control setup connects to multiple [PMA.core](#) instances for geo-replication purposes. Each PMA.core instance therefore must have access to the same slides locally.
- You're setting up one or more instances of [PMA.slidebox](#). As you're explaining to the various instructors that are coming to your event how they can each manage their own slides, you wish to seed a number of root-directories with a pre-defined demonstration dataset.

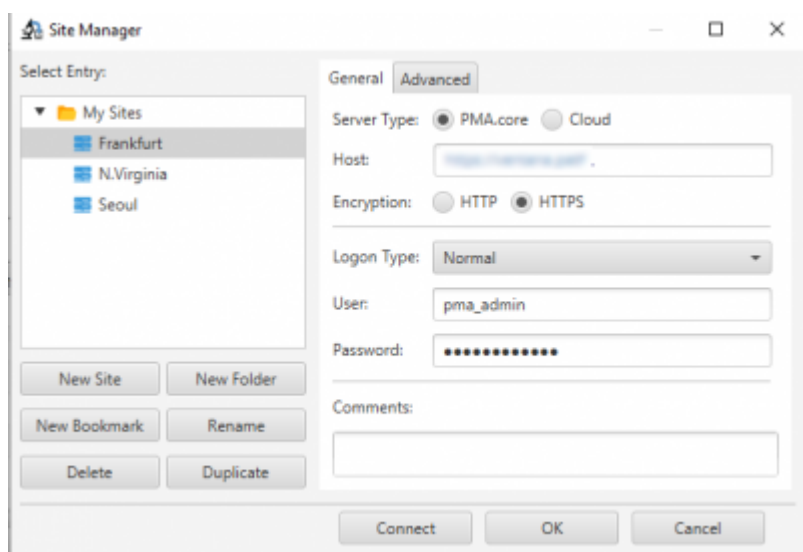
In both scenarios, the solution is work via the [site manager](#) to register your target locations, and to subsequently run multiple PMA.transfer instances side by side.

Geo-replication (multiple PMA.core instances)

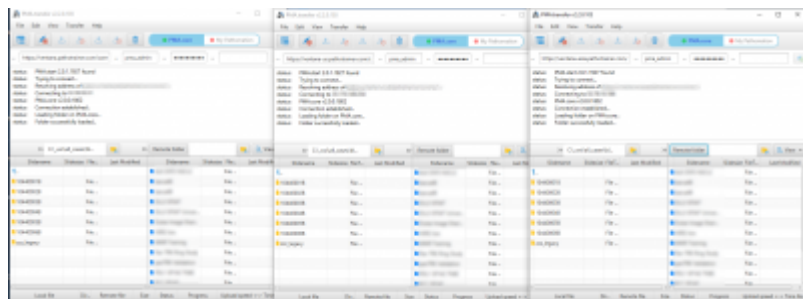
Within your PMA.control installation, login with an administrative account and determine what the PMA.core instances are that accomplish respective geo-replication.



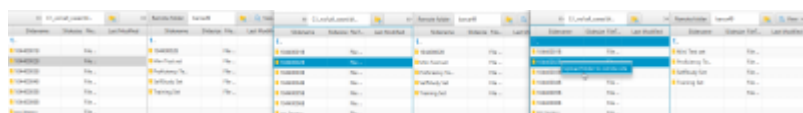
Copy the PMA.core instance information (with appropriate credentials for each) to the site manager in PMA.transfer. At this point, you only need a single instance of PMA.transfer open.



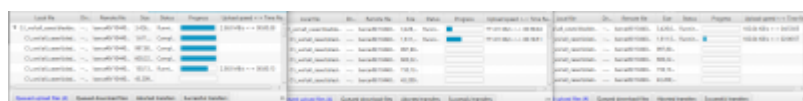
Make sure the site manager has the correct information stored, and start additional instances of PMA.transfer as needed. Arrange them on your screen so you can conveniently manage them simultaneously side by side (a second monitor may come in handy).



Once you're connected to each PMA.core, set your source folder to the same location in each PMA.transfer instance. Then select the right target folder for each respective PMA.core instance, and initiate the transfers.



You'll notice quickly that the transfer doesn't proceed at the same pace for each PMA.transfer instance. Indeed, that is the purpose of geo-replication in the first place: to make sure that all the slides are available at each geographic location and can be accessed by end-users with the same performance, whether somebody resides in Tokyo, Japan, or Sao Paulo, Brazil.



Parallel uploads (one PMA.core instance, multiple root-directory targets)

Parallel uploads can be set up in similar fashion as geo-replicated transfers. The following aspects differ:

- You can still use the [site manager](#) to register different target locations. This time, register the same instance of PMA.core multiple times under a different name. Then use the [Advanced](#) tab for each instance to configure different target paths on the same PMA.core.
- Your actual transfer should run significantly smoother than under a geo-replication scenario. The reason being that you're connecting to the same PMA.core in each instance of PMA.transfer, and therefore the connection speed in each instance should roughly be the same, too.

What about automation?

It's very tempting to imagine a mechanism that would allow you to automatically queue a transfer to multiple PMA.core instances, or multiple root-directories on the same PMA.core). We thought about it, too.

Our conclusion is that, realistically speaking, it is not possible to facilitate this in a general fashion.

The architecture of PMA.transfer is such that it allows multiple instances to be run side by side, simultaneously, in parallel. Why mess up a good thing?

But, ok, you want to know about automation...

For highly controlled environments, we recommend that you look into automation through our Python or PHP SDK. The `PMA::Core` module has `Upload()` and `Download()` methods that can be scheduled to operate on what new data you have, and can be combined to integrate in pipelines and complex slide distribution scenarios.

More information about these can be found through our [developer portal](#) or [GitHub landing page](#).

From:

<https://docs.pathomation.com/PMA.transfer/> - **PMA.transfer**

Permanent link:

<https://docs.pathomation.com/PMA.transfer/doku.php?id=geo-replication>

Last update: **2021/06/22 15:10**

