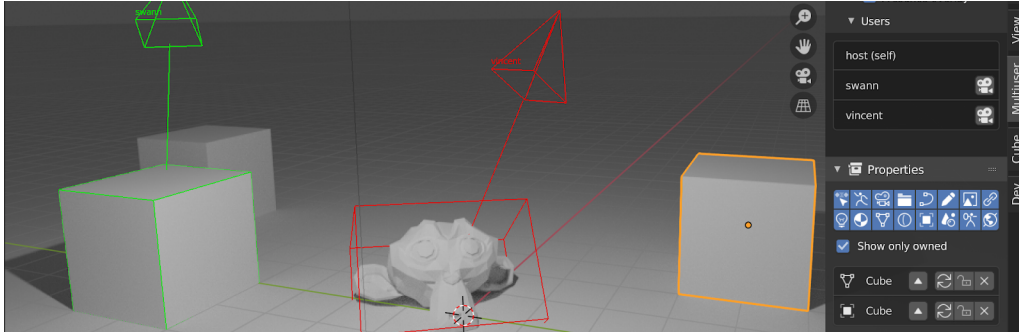

multi-user

Release 0.0.2

Jun 23, 2020

1	Main Features	3
2	Community	5
3	Status	7
	Index	25



The multi-user addon is a free and open source blender plugin. It tool aims to bring multiple users to work on the same .blend over the network.

Warning: Under development, use it at your own risks.

CHAPTER 1

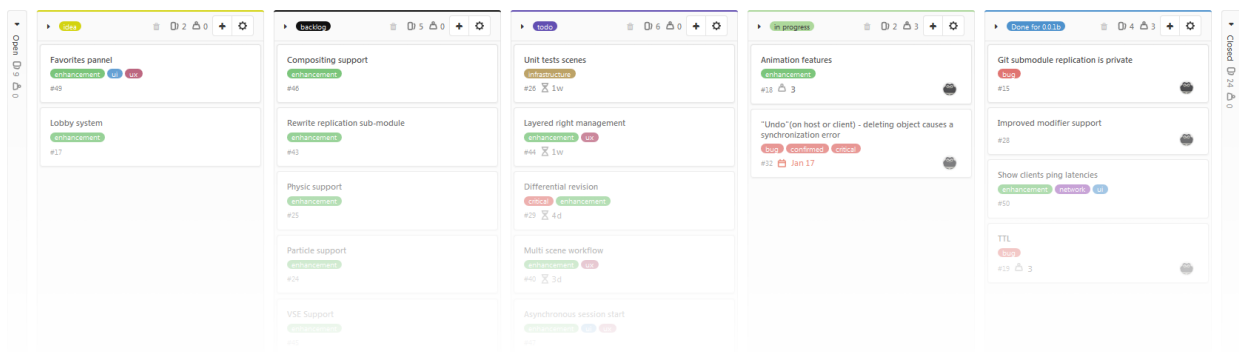
Main Features

- Collaborative workflow in blender
- Viewport users presence (active selection, POV)
- Datablocks right managment
- Tested under Windows

CHAPTER 2

Community

A [discord server](#) have been created to provide help for new users and organize collaborative creation sessions.



Follow the [roadmap](#) to be aware of last news.

Documentation is organized into the following sections:

3.1 Introduction

A film is an idea carved along the whole production process by many different peoples. A traditional animation pipeline involve a linear succession of tasks. From storyboard to compositing by passing upon different step, its fundamental work flow is similar to an industrial assembly line. Since each step is almost a department, its common that one person on department B doesn't know what another person did on a previous step in a department A. This lack of visibility/communication could be a source of problems which could produce a bad impact on the final production result.

Fig. 1: The linear workflow problems

Nowadays it's a known fact that real-time rendering technologies allows to speedup traditional linear production by reducing drastically the iteration time across different steps. All majors industrial CG solutions are moving toward real-time horizons to bring innovative interactive workflows. But this is a microscopic, per-task/solution vision of

real-time rendering benefits for the animation production. What if we step-back, get a macroscopic picture of an animation movie pipeline and ask ourself how real-time could change our global workflow ? Could-it bring better ways of working together by giving more visibility between departments during the whole production ?

The multi-user addon is an attempt to experiment real-time parallelism between different production stage. By replicating blender data blocks over the networks, it allows different artists to collaborate on a same scene in real-time.

3.2 Installation

Hint: The process is the same for linux, mac and windows.

1. Download latest [release](#) or [develop \(unstable !\)](#) build.
2. Run blender as administrator (to allow python dependencies auto-installation).
3. Install **multi-user.zip** from your addon preferences.

Once the addon is succesfully installed, I strongly recommend you to follow the [Quick start](#) tutorial.

3.3 Quick start

Hint: All session related settings are located under: 'View3D -> Sidebar -> Multiuser panel'

The multi-user is based on a session management system. In this this guide you will quickly learn how to use the collaborative session system in three part:

- [How to host a session](#)
- [How to join a session](#)
- [How to manage a session](#)

3.3.1 How to host a session

The multi-user add-on rely on a Client-Server architecture. The server is the heart of the collaborative session, it will allow each users to communicate with each others. In simple terms, *Hosting a session* means *run a local server and connect the local client to it*. When I said **local server** I mean accessible from the LAN (Local Area Network).

However sometime you will need to host a session over the internet, in this case I strongly recommand you to read the [Hosting on internet](#) tutorial.

1. Fill your user information

The **User Info** panel (See image below) allow you to constomize your online identity.

Let's fill those tow field:

- **name:** your online name.
- **color:** a color used to represent you into other user workspace(see image below).

During online sessions, other users will see your selected object and camera hilghlited in your profile color.

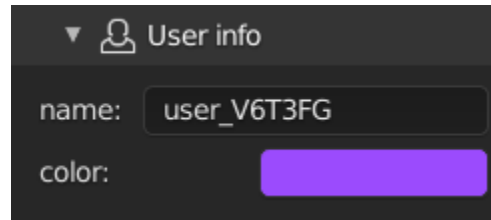


Fig. 2: User info panel

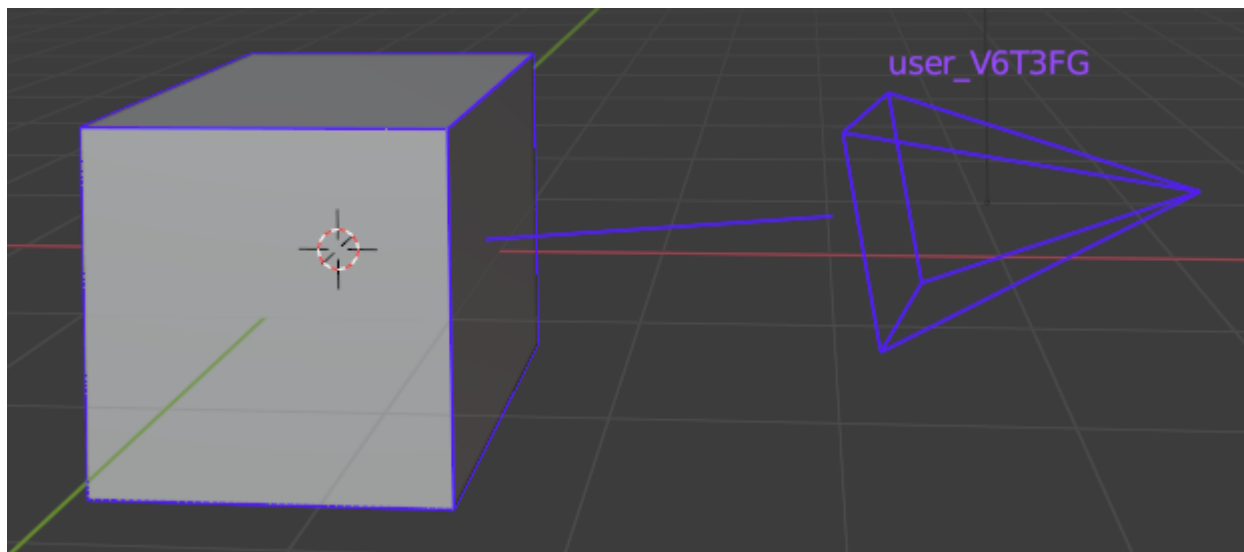


Fig. 3: User viewport representation

2. Setup the network

When the hosting process will start, the multi-user addon will launch a local server instance. In the network panel select **HOST**. The **Host sub-panel** (see image below) allow you to configure the server according to:

- **Port:** Port on which the server is listening.
- **Start from:** The session initialisation method.
 - **current scenes:** Start with the current blendfile datas.
 - **an empty scene:** Clear a data and start over.

Danger: By starting from an empty, all of the blend data will be removed ! Ensure to save your existing work before launching the session.

- **Admin password:** The session administration password.

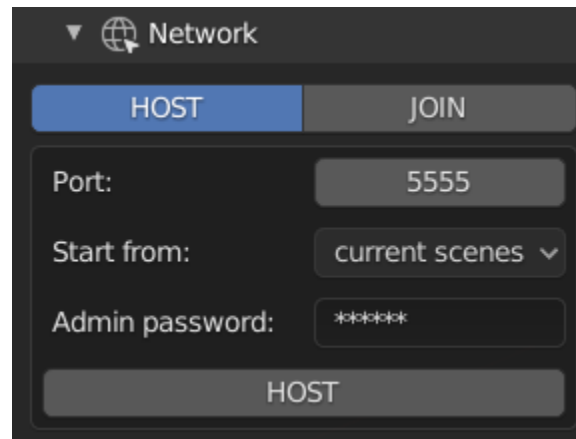


Fig. 4: Host network panel

Note: Additionnal configuration setting can be found in the [Advanced configuration](#) section.

Once everything is setup you can hit the **HOST** button to launch the session !

It will do two things:

- Start a local server
- Connect you to it as an admin

During online session, various actions are available to you, go to [How to manage a session](#) section to learn more about them.

3.3.2 How to join a session

This section describe how join a launched session. Before starting make sure that you have access to the session ip and port.

1. Fill your user information

Follow the *user-info* section for this step.

2. Network setup

In the network panel select **JOIN**. The **join sub-panel** (see image below) allow you configure the client to join a collaborative session.

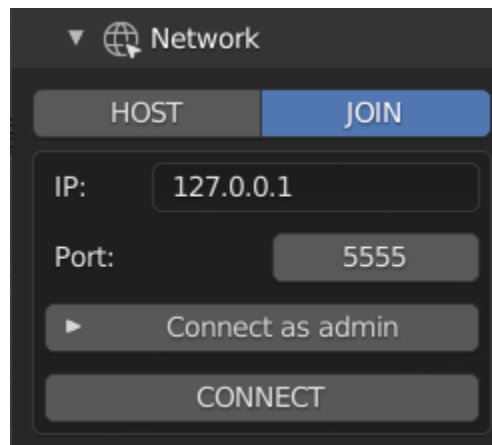


Fig. 5: Connection panel

Fill those field with your information:

- **IP:** the host ip.
- **Port:** the host port.
- **Connect as admin:** connect you with **admin rights** (see admin) to the session.

Note: Additionnal configuration setting can be found in the *Advanced configuration* section.

Once you've set every field, hit the button **CONNECT** to join the session ! When the session-status is **ONLINE** you are online and ready to start to collaborate.

During online session, various actions are available to you, go to *How to manage a session* section to learn more about them.

3.3.3 How to manage a session

The collaboration quality directly depend on the communication quality. This section describes various tools made in an effort to ease the communication between the different session users. Feel free to suggest any idea for communication tools [here](#) .

Monitor online users

One of the most vital tool is the **Online user panel**. It list all connected users information's including yours such as :

- **Role** : if user is an admin or a regular user.

- **Location:** Where the user is actually working.
- **Frame:** When (in frame) the user working.
- **Ping:** user connection delay in milliseconds

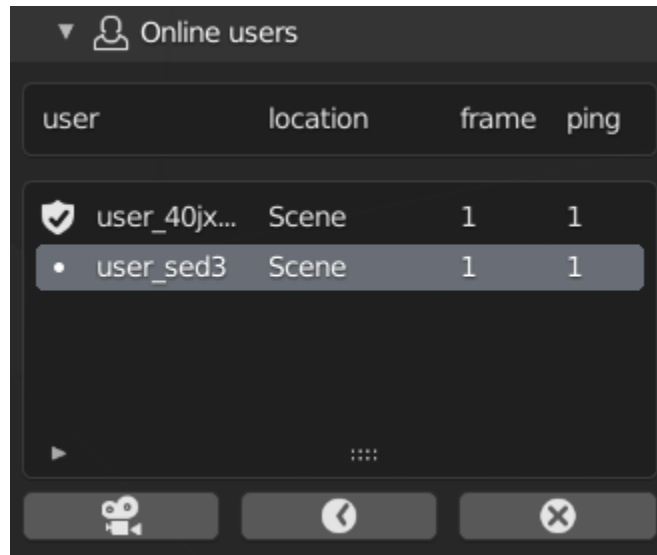


Fig. 6: Online user panel

By selecting a user in the list you'll have access to different user related **actions**. Those operators allow you reach the selected user state in tow different dimensions: **SPACE** and **TIME**.

Snapping in space

The **CAMERA button** (Also called **snap view** operator) allow you to snap on the user viewpoint. To disable the snap, click back on the button. This action served different purposes such as easing the review process, working together on wide world.

Hint: If the target user is localized on another scene, the **snap view** operator will send you to his scene.

Fig. 7: Snap view in action

Snapping in time

The **CLOCK button** (Also called **snap time** operator) allow you to snap on the user time (current frame). To disable the snap, click back on the button. This action is built to help various actors to work on the same temporality (for instance multiple animators).

Fig. 8: Snap time in action

Kick a user

Warning: Only available for admin !

The **CROSS button** (Also called **kick** operator) allow the admin to kick the selected user. On the target user side, the session will properly disconnect.

Change users display

Presence is the multi-user module responsible for users display. During the session, it draw users related information in your viewport such as:

- Username
- User point of view
- User selection

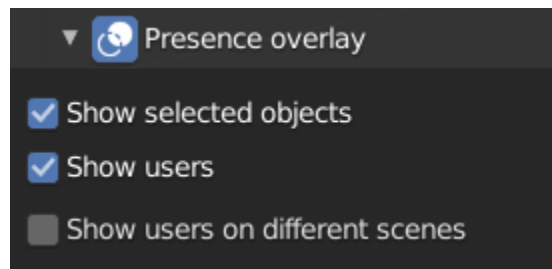


Fig. 9: Presence show flags

The presence overlay panel (see image above) allow you to enable/disable various drawn parts via the following flags:

- **Show selected objects:** display other users current selection
- **Show users:** display users current viewpoint
- **Show different scenes:** display users working on other scenes

Manage data

In order to understand replication data management, a quick introduction to the multi-user data workflow is required. First thing to know: until now, the addon rely on a data-based replication. In simple words, it means that it replicate user's action results. To replicate datablocks between clients the multi-user rely on what tends to be a distributed architecture:

- The server store the “master” version of the work.
- Each client have a local version of the work.

When an artist modified something in the scene, here is what is happening in the background:

1. Modified data are **COMMITTED** to the local repository.
2. Once committed locally, they are **PUSHED** to the server
3. As soon as the server is getting updates, they are stored locally and pushed to every other clients

At the top of this data management system, a right management system prevent multiple users from modifying same data at same time. A datablock may belong to a connected user or be under common-right rights.

Note: In a near future, the right management system will support roles to allow multiple users to work on different aspect of the same datablock.

The Repository panel (see image below) allow you to monitor, change datablock states and right manually.

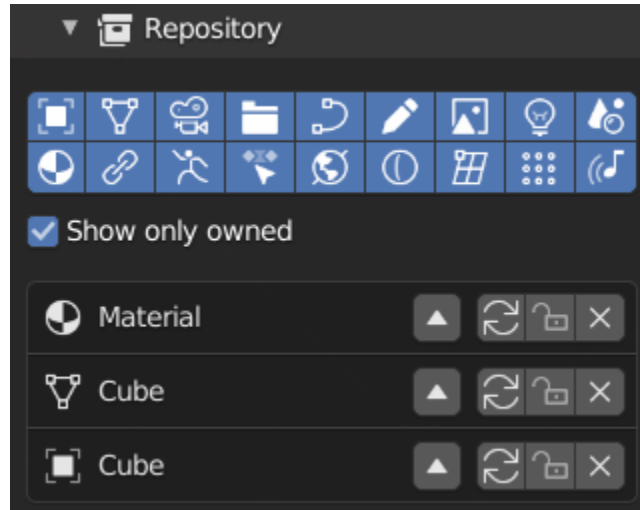


Fig. 10: Repository panel

The **show only owned** flag allow you to see which datablocks you are currently modifying.

Warning: If you are editing a datablock not listed with this flag enabled, it means that you do not have right granted to modify it. So it won't be updated to other client !

Here is a quick list of available actions:

icon	Action	Description
▲	Push	push data-block to other clients
▼	Pull	pull last version into blender
↺	Reset	Reset local change to the server version
🔒	Lock/Unlock	If locked, does nothing. If unlocked, grant modification rights to another user.
✕	Delete	Remove the data-block from network replication

3.3.4 Advanced configuration

This section contains optional settings to configure the session behavior.

Network

IPC Port is the port used for Inter Process Communication. This port is used by the multi-users subprocesses to communicate with each others. If different instances of the multi-user are using the same IPC port it will create

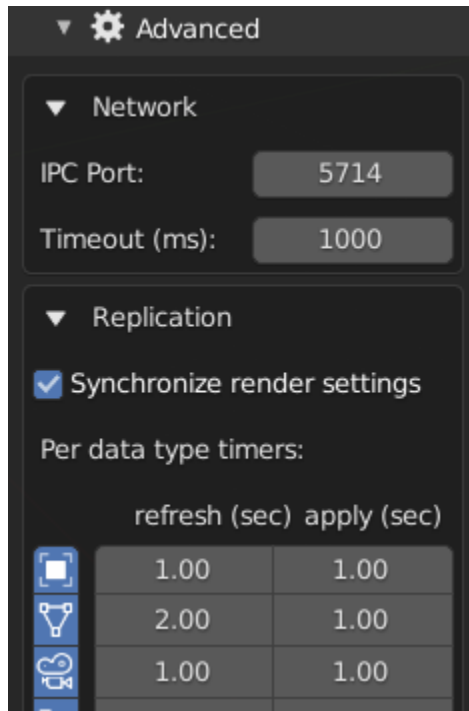


Fig. 11: Repository panel

conflict !

You only need to modify it if you need to launch multiple clients from the same computer(or if you try to host and join on the same computer). You should just enter a different **IPC port** for each blender instance.

Timeout (in milliseconds) is the maximum ping authorized before auto-disconnecting. You should only increase it if you have a bad connection.

Replication

Synchronize render settings (only host) enable replication of EEVEE and CYCLES render settings to match render between clients.

Properties frequency gird allow to set a custom replication frequency for each type of data-block:

- **Refresh:** pushed data update rate (in second)
- **Apply:** pulled data update rate (in second)

Note: Per-data type settings will soon be revamped for simplification purposes

3.4 Glossary

administrator *A session administrator can manage users (kick) and have a write access on each datablock. He could also init a dedicated server repository.*

session status *Located in the title of the multi-user panel, the session status show you the connection state.*

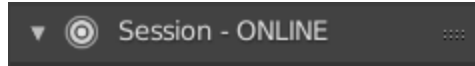


Fig. 12: Session status in panel title bar

All possible state are listed here with their meaning:*

State	Description
WARMING UP DATA	Commiting local data
FETCHING	Dowloading snapshot from the server
AUTHEN- TIFICATION	Initial server authentication
ONLINE	Connected to the session
PUSHING	Init the server repository by pushing ours
INIT	Initial state
QUITTING	Exiting the session
LAUNCH- ING SER- VICES	Launching local services. Services are spetialized daemons running in the background.)
LOBBY	The lobby is a waiting state triggered when the server repository hasn't been initiated yet Once initialized, the server will automatically launch all client in the LOBBY .

common right When a data block is under common right, it is available for everyone to modification. The rights will be given to the user that select it first.

3.5 Hosting on internet

Warning: Until now, those communications are not encrypted but are planned to be in a mid-term future (*Status*).

This tutorial aims to guide you to host a collaborative Session on internet. Hosting a session can be done is several ways:

- *From blender*: hosting a session directly from the blender add-on panel.
- *From the dedicated server*: hosting a session directly from the command line interface on a computer without blender.

3.5.1 From blender

By default your router doesn't allow anyone to share you connection. In order grant server access to people from internet you have tow main option:

- The *Using a connection sharing solution*: the easiest way.
- The *Using port-forwarding*: this one is the most unsecure, if you have no networking knowledge, you should definitively go to *Using a connection sharing solution*.

Using a connection sharing solution

Many third party software like [ZEROTIER](#) (Free) or [HAMACHI](#) (Free until 5 users) allow you to share your private network with other people. For the example I'm gonna use ZeroTier because its free and open source.

1. Installation

Let's start by downloading and installing ZeroTier: <https://www.zerotier.com/download/>

Once installed, launch it.

2. Network creation

To create a ZeroTier private network you need to register a ZeroTier account on my.zerotier.com (click on **login** then register on the bottom)

Once your account is activated, you can connect to my.zerotier.com. Head up to the **Network** section (highlighted in red in the image below).

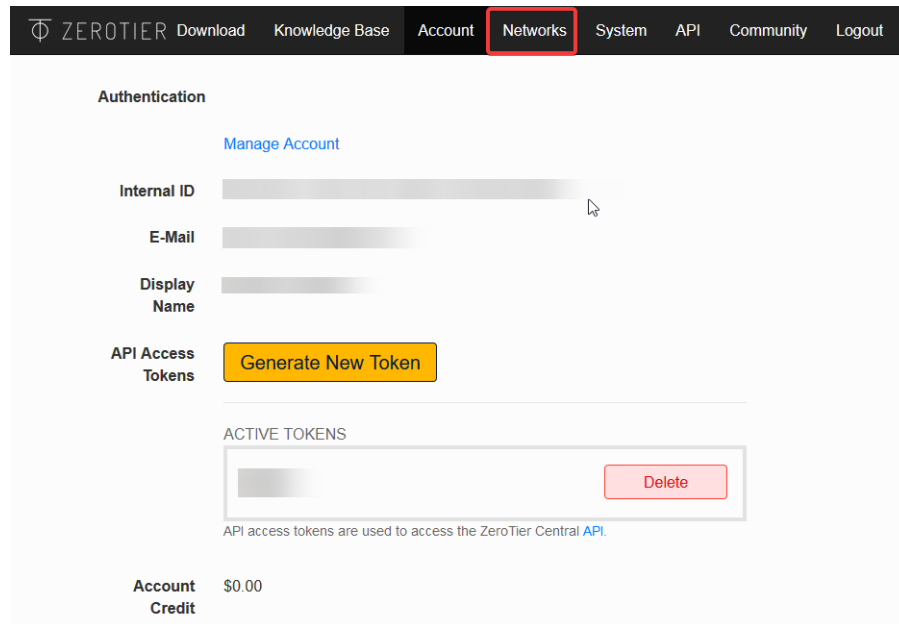


Fig. 13: ZeroTier user homepage

Hit 'Create a network' (see image below) and go to the network settings.

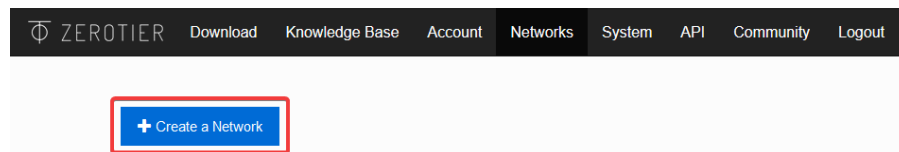


Fig. 14: Network page

Now that the network is created, let's configure it.

In the Settings section(see image below), you can change the network name to what you want. Make sure that the field **Access Control** is set to **PRIVATE**.

Hint: If you set the Access Control to PUBLIC, anyone will be able to join without your confirmation. It is easier to set up but less secure.

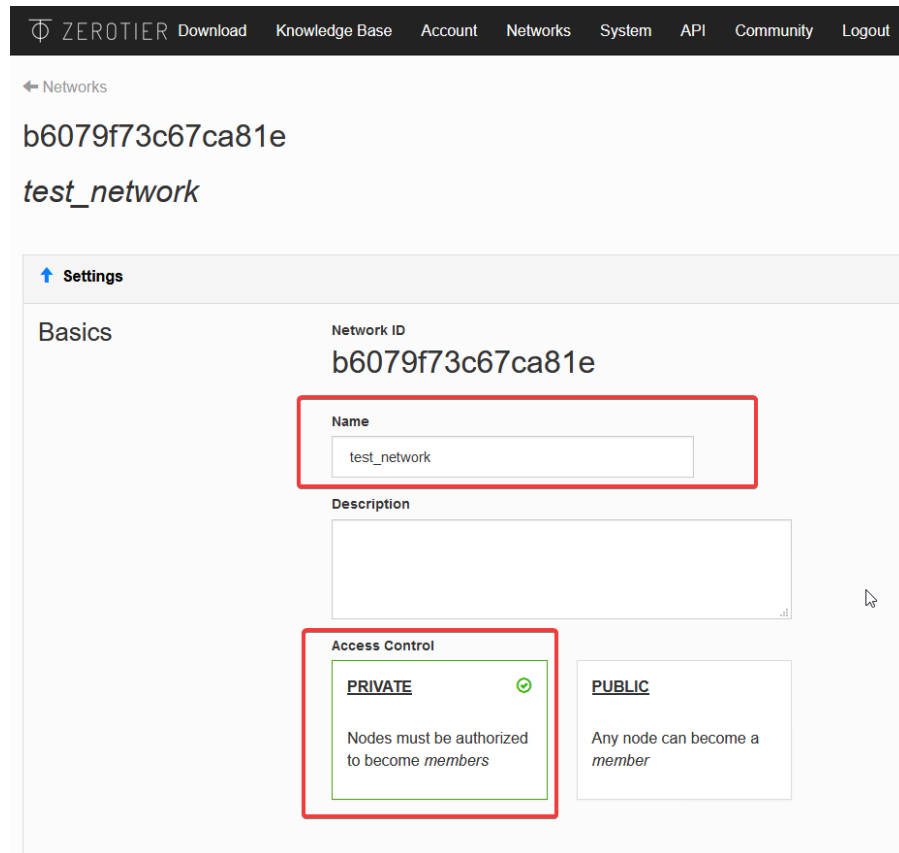


Fig. 15: Network settings

That's all for the network setup ! Now let's connect everyone.

3. Network authorization

Since your ZeroTier network is Private, you will need to authorize each new users to connect to it. For each user you want to add, do the following step:

1. Get the client **ZeroTier id** by right clicking on the ZeroTier tray icon and click on the *Node ID*, it will copy it.
2. Go to the network settings in the Member section and paste the Node ID into the Manually Add Member field.

4. Network connection

To connect to the ZeroTier network, get the network id from the network settings (see image).

Now we are ready to join the network ! Right click on the ZeroTier tray icon and select **Join Network** !

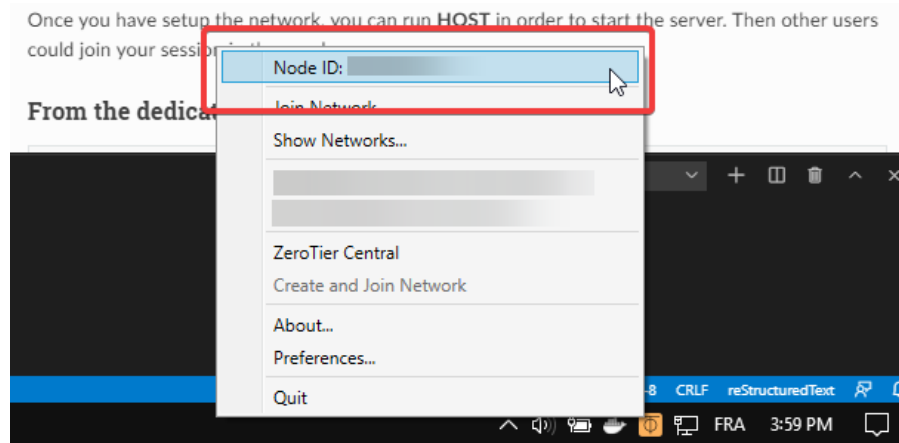


Fig. 16: Get the ZeroTier client id

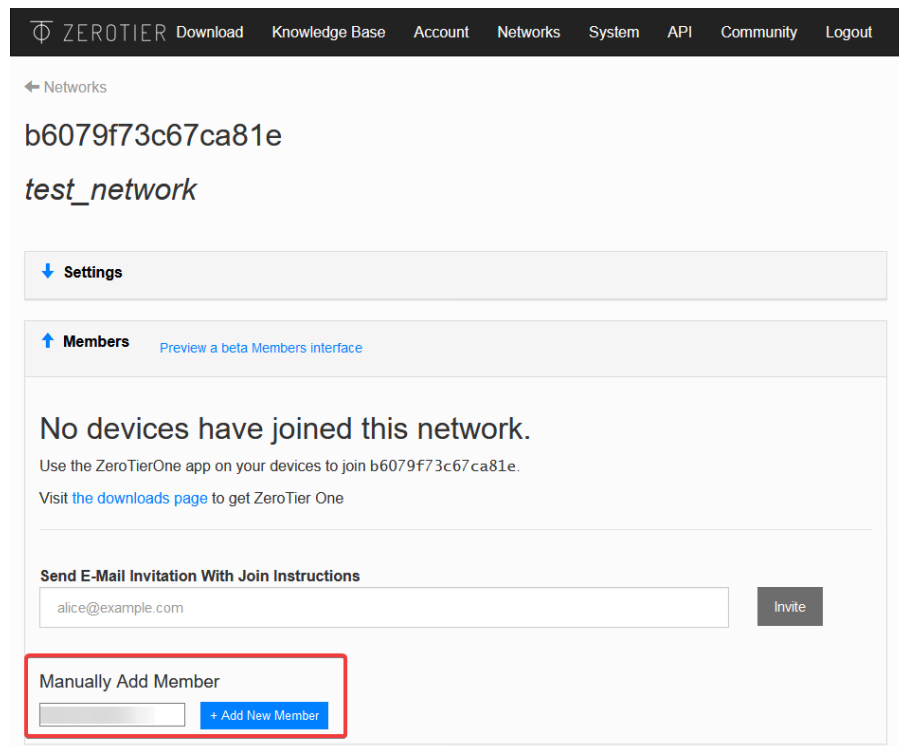


Fig. 17: Add the client to network authorized users

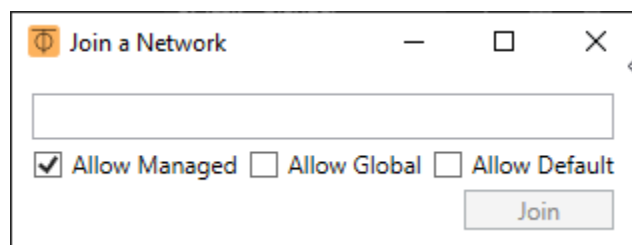
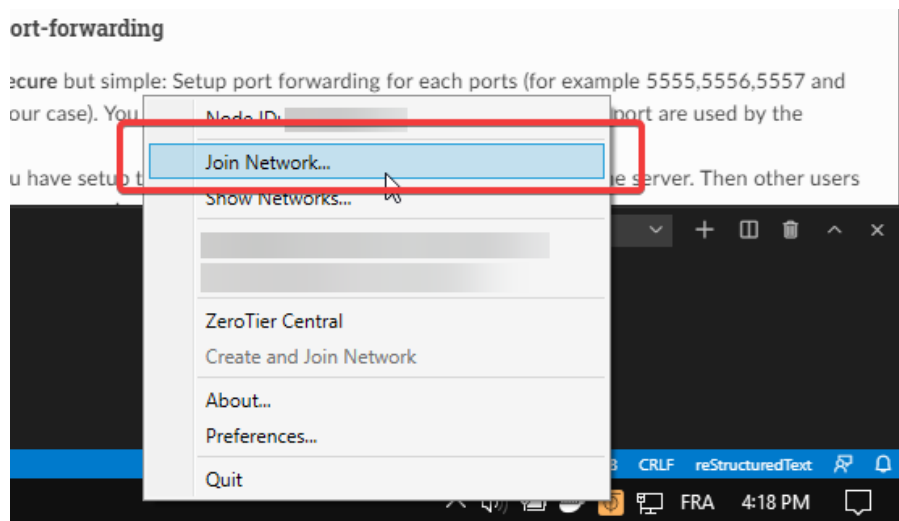
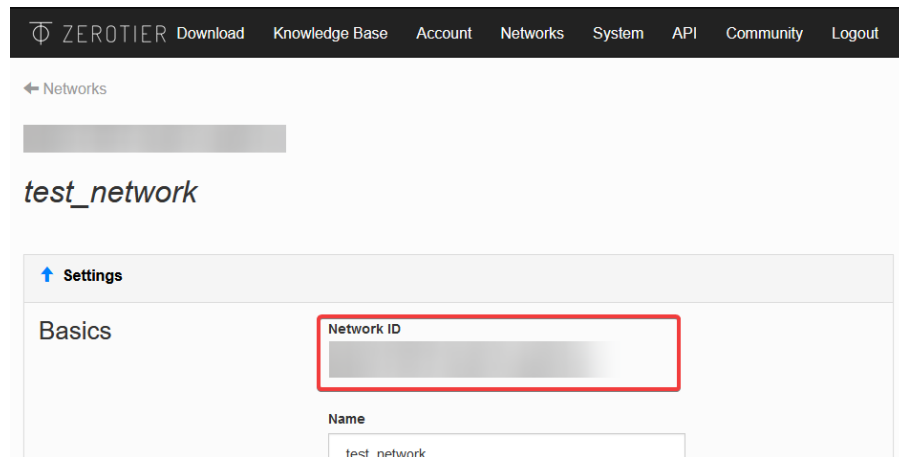


Fig. 18: Joining the network

Past the network id and check **Allow Managed** then click on join ! You should be connected to the network.

Let's check the connection status. Right click on the tray icon and click on **Show Networks...**

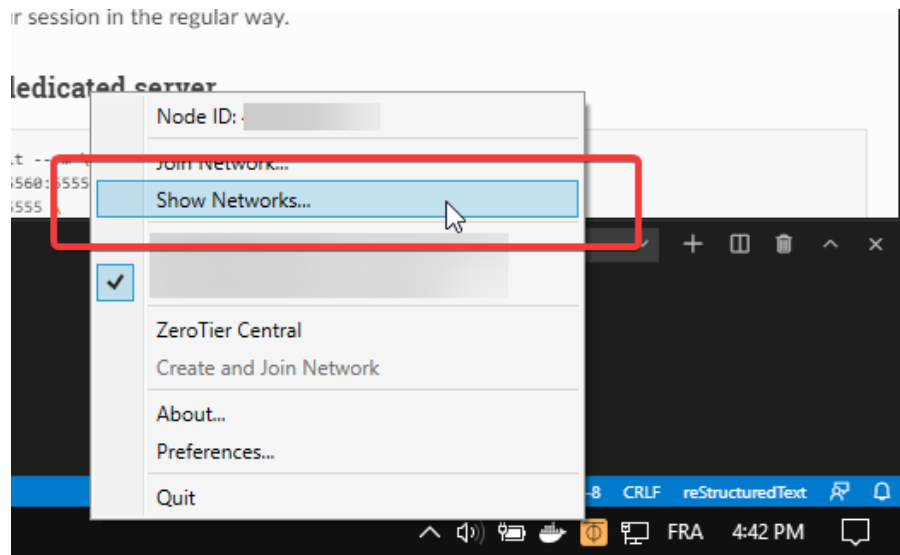


Fig. 19: Show network status

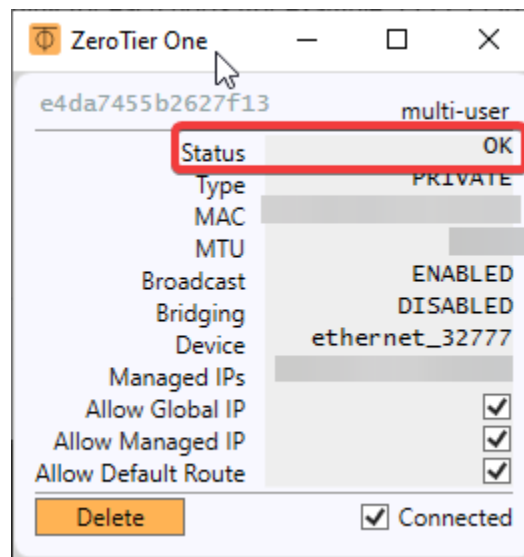


Fig. 20: Network status.

The network status must be **OK** for each user (like in the picture above) otherwise it means that you are not connected to the network. If you see something like **ACCESS_DENIED**, it means that you were not authorized to join the network. Please check the [3. Network authorization](#) section.

This is it for the ZeroTier network setup. Now everything should be setup to use the multi-user add-on over internet ! You can now follow the [Quick start](#) guide to start using the multi-user add-on !

Using port-forwarding

The port forwarding method consists to configure your Network route to allow internet traffic through specific ports.

In order to know which port are used by the add-on, check the [Port setup](#) section. To set up port forwarding for each port you can follow this [guide](#) for example.

Once you have set up the network you can follow the [Quick start](#) guide to start using the multi-user add-on !

3.5.2 From the dedicated server

Warning: The dedicated server is developed to run directly on internet server (like VPS). You can also run it at home for LAN but for internet hosting you need to follow the [Using port-forwarding](#) setup first.

The dedicated server allow you to host a session with simplicity from any location. It was developed to improve internet hosting performance.

The dedicated server can be run in tow ways:

- [Using a regular command line](#)
- [Using a pre-configured image on docker engine](#)

Using a regular command line

You can run the dedicated server on any platform by following those steps:

1. Firstly, download and intall python 3 (3.6 or above).
2. Download and extract the dedicated server from [here](#)
3. Open a terminal in the extracted folder and install python dependencies by running:

```
python -m pip install -r requirements.txt
```

4. Launch the server from the same terminal with:

```
python scripts/server.py
```

Hint: You can also specify a custom **port** (-p), **timeout** (-t) and **admin password** (-pwd) with the following optionnal argument

```
python scripts/server.py -p 5555 -pwd toto -t 1000
```

As soon as the dedicated server is running, you can connect to it from blender (follow [How to join a session](#)).

Hint: Some commands are available to manage the session. Check [Dedicated server management](#) to learn more.

Using a pre-configured image on docker engine

Launching the dedicated server from a docker server is simple as:

```
docker run -d \
  -p 5555-5560:5555-5560 \
  -e port=5555 \
  -e password=admin \
  -e timeout=1000 \
  registry.gitlab.com/slumber/multi-user/multi-user-server:0.0.3
```

As soon as the dedicated server is running, you can connect to it from blender. You can check the [How to join a session](#) section.

Hint: Some commands are available to manage the session. Check [Dedicated server management](#) to learn more.

Dedicated server management

Here is the list of available commands from the dedicated server:

- `help`: Show all commands.
- `exit` or `Ctrl+C`: Stop the server.
- `kick username`: kick the provided user.
- `users`: list all online users.

3.5.3 Port setup

The multi-user network architecture is based on a clients-server model. The communication protocol use four ports to communicate with client:

- `Commands`: command transmission (such as **snapshots**, **change_rights**, etc.) [given port]
- `Subscriber` : pull data [Commands port + 1]
- `Publisher` : push data [Commands port + 2]
- `TTL (time to leave)` : used to ping each client [Commands port + 3]

To know which ports will be used, you just have to read the port in your preference.

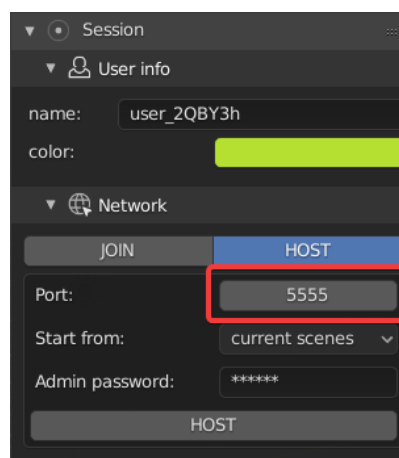


Fig. 21: Port in host settings

In the picture below we have setup our port to **5555** so it will be:

- Commands: 5555 (**5555**)
- Subscriber: 5556 (**5555** +1)
- Publisher: 5557 (**5555** +2)
- TTL: 5558 (**5555** +3)

Those four ports need to be accessible from the client otherwise it won't work at all !

3.6 Ways to contribute

Note: Work in progress

3.6.1 Testing and reporting issues

A great way of contributing to the multi-user addon is to test development branch and to report issues. It is also helpful to report issues discovered in releases, so that they can be fixed in the development branch and in future releases.

Testing development versions

In order to help with the testing, you have several possibilities:

- Test `latest` release
- Test `development` branch

Filling an issue on Gitlab

The `gitlab` `issue` `tracker` is used for bug report and enhancement suggestion. You will need a Gitlab account to be able to open a new issue there and click on “New issue” button.

Here are some useful information you should provide in a bug report:

- **Multi-user version** such as *latest*, *commit-hash*, *branch*. This is a must have. Some issues might be relevant in the current stable release, but fixed in the development branch.
- **How to reproduce the bug**. In the majority of cases, bugs are reproducible, i.e. it is possible to trigger them reliably by following some steps. Please always describe those steps as clearly as possible, so that everyone can try to reproduce the issue and confirm it. It could also take the form of a screen capture.

3.6.2 Contributing code

1. Fork it (<https://gitlab.com/yourname/yourproject/fork>)
2. Create your feature branch (`git checkout -b feature/fooBar`)
3. Commit your changes (`git commit -am 'Add some fooBar'`)
4. Push to the branch (`git push origin feature/fooBar`)
5. Create a new Pull Request

A

administrator, [15](#)

C

common right, [16](#)

S

session status, [15](#)