

1. Hoops Net

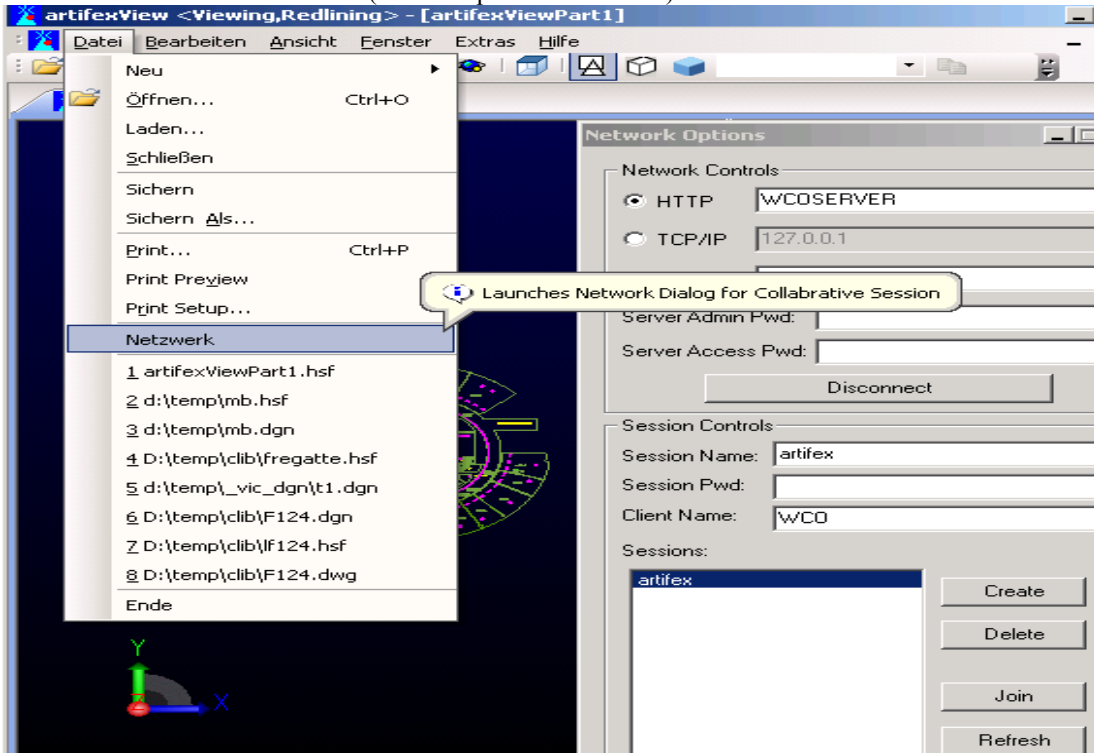
Hier die erste Kurzanweisung für den Hoops-Net-Server.

Das Prinzip ist ähnlich der Benutzerverwaltung.

Man starte den Server, hier HNET_SERVER1220.EXE am besten auf dem Server wie die Benutzerverwaltung.

Das .EXE ist genug, es braucht auf dem Server keine weiteren Hoops-Files.

Dann startet man artifexView.exe (oder HoopsPartViewer.exe) auf einem Client und drückt dort Datei->Netzwerk.

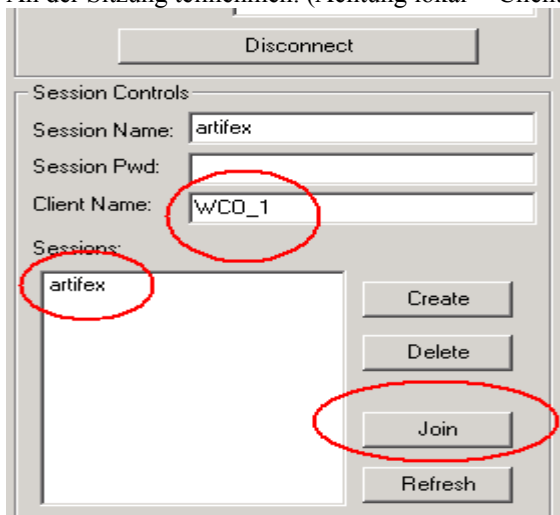


Oben unter Netzwerk Protokoll ist schon der Server aus wcoprj.ini eingetragen.

Eine Session ist eine gemeinsame Sitzung. In der Benutzerverwaltung ist die das aktuelle Projekt. Hier wird Als Vorschlag artifex genommen. Eine Session (Sitzung) könnte auch „Besprechung Mobiliar“ heißen. Der Client ist wie bei der Benutzerverwaltung der aktuelle Rechnername. Kann auch anders heißen. Achtung beim Testen auf einem Computer und zwei Clients. (zweimal artifexView.exe zum Testen) Ein Client kann nur einmal da sein, also zweiter Client anders nennen.

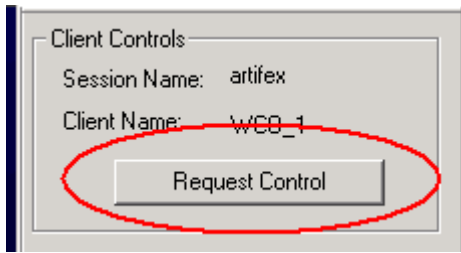
Mit CREATE wird eine Sitzung neu erzeugt.

Jetzt noch mal artifex View laden. Auf einem anderen Computer, am LAN , übers Web oder Lokal zum Testen und dort auch Datei-Netzwerk und mit Connect einloggen.. Die offenen Sitzungen werden angezeigt, hier artifex., anwählen und mit JOIN An der Sitzung teilnehmen. (Achtung lokal-> Clientnamen ändern, hier WCO_1)

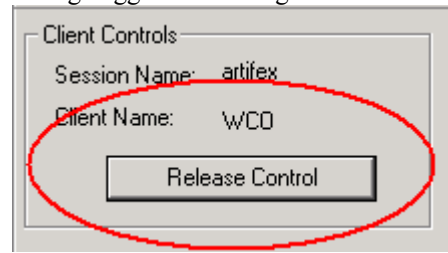


Jetzt auf einem der Clients die Kontrolle übernehmen.

Eine Kontrolle übernimmt man mit



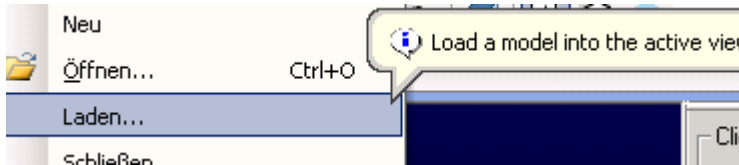
Eine geloggte Kontrolle gibt man mit



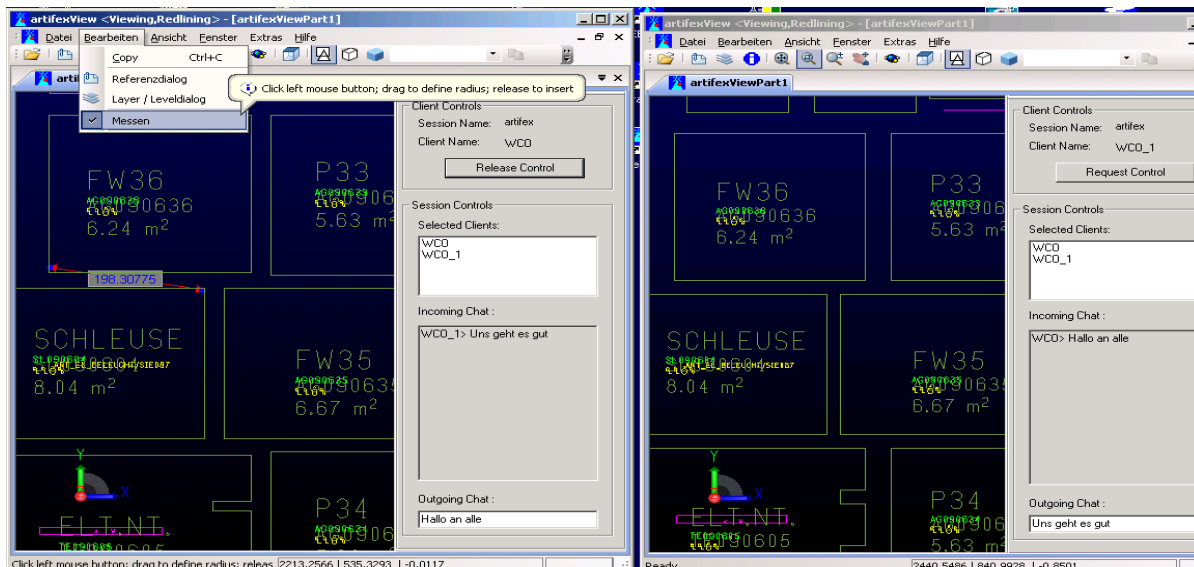
wieder frei.

Unten im Chat kann man ja allen sagen, dass Sie die Kontrolle freigeben sollen.

Derjenige der die Kontrolle hat kann nun eine Zeichnung Laden (NICHT Öffnen, sonder Laden)



Alles was nun derjenige mit der Kontrolle macht. Laden, zoomen sehen jetzt auch die anderen.



So, das ist jetzt die einfachste Variante.

In HoopsPartViewer sind noch Schalter

Z.B. dass nur das geladen wird wo man gerade Zoomt.

Das heißt auch bei riesigen Dateien ist ein gemeinsames Gehen in einem kompletten Gebäude möglich, oder man kann Kollisionen einschalten. Wenn man jetzt was schiebt. Z.B. einen Stuhl kann man dies nicht durch die Wand...

Oder man kann auf dem Server eine Textdatei legen welche Files in welcher Session zugelassen sind und und und...

2. Server

Hier ein Auszug aus der möglichen Server Konfiguration.

Das einfache arbeiten übers Netz wie oben beschrieben geht ohne eine Einstellung.

2.1 Launching the Server

The server is distributed in the form of an executable. To launch the server you should simply run the `hnet_server` executable located in the `/dev_tools/hoops_net/bin` directory of your HOOPS/3dAF installation. On startup the server reads that `hnet_server.conf` file for initial server settings. Within this file you can set initial settings such as the administrator's password and the port on which the server should listen for initial client connections.

2.2 Server Configuration File

The `hnet_server.conf` file is an ASCII text file within which the administrator sets up the initial server configurations. The different options are:

`server_name = <string>`: Since multiple servers can be run simultaneously on the same machine, you can name each server differently.

`server_admin_password = <string>`: This is the password which must be provided for administrator access to that particular server.

`server_access_password = <string>`: This is the password which must be provided for client to connect/access to that particular server.

`server_debug_log_file_name = <stderr/stdout/null/string>`: This is the output device/filename for the server log

`server_file_sessions_enabled = <yes/no>`: When this flag is set the file sessions hosting capabilities of server be enabled.

`server_file_sessions_directory = <string>`: Directory path where all the file sessions reside. Each subdirectory under this directory indicates a file session.

`server_file_sessions = <string>`: A space separated list of all the file sessions. For each entry in this list, a corresponding directory and configuration file must exist under `server_file_sessions_directory`.

`server_port = <string>`: The port on which the server listens for initial client connections. Note, after a connection is made the server opens and maintains a separate connection for that client over a separate port.

`control_token_policy = <none/one/fair>`: This flag governs the control token queue. Once a user requests control they cannot then remove their name from the control queue unless they quit the session. The meaning of the different control policy flags are:

none: Server does not keep track of requests for control. Clients can only get control when nobody else has control

one: Server keeps track of the first person to request control. All other request for control are ignored.

fair: Server keeps an ordered list of the people as they request control. Control is passed to clients in the order in which they are in the list. No one client can have more than one entry in the list.

`log_standard_messages_enabled = <yes/no>`: When this flag is set the server will log all messages into a message log file.

`log_standard_message_file_location = <string>`: Specifies the location the message log file should be written to.

`log_unique_standard_messages = <string>`: A space separated list of all the messages which should only have their last entry stored in the message log. Camera position entries would be a typical example of a unique standard message entry.

`log_never_standard_messages message names = <string>`: A space separated list of all the messages which should not be put into the message log

`log_reset_standard_messages = <string>`: The message identifier which forces all entries in the message log to be cleared out. This might be needed when there is a new file load.

`session_inactivity_timeout = <string>`: Time, in seconds, of inactivity that causes the session to shutdown.

`client_inactivity_timeout = <string>`: Time, in seconds, of inactivity of a certain user before they are expelled from a session.

Below is an example of a hnet_server.conf file:

```
# strings must be quoted if longer than one word
server_name = "MCAD Server"
server_admin_password = "ROMULUS9"
server_port = "11111"
control_token_queue_policy = "fair"
log_standard_messages_enabled = "yes"
log_standard_message_file_location = "./logs"
log_unique_standard_messages = "H_SET_CAMERA H_SET_RENDERMODE
H_SET_MODELLING_MATRIX H_SET_WINDOW_COLOR"
log_never_standard_messages = "H_DEV_NULL_DATA"
server_file_sessions_enabled = "yes"
server_file_sessions_directory = "file_sessions"
server_file_sessions = "tsa_files"
session_inactivity_timeout = "3600"
client_inactivity_timeout = "0"
```

2.3 File Session Configuration File

The *<file_session_name>.conf* file is an ASCII text file within which the administrator sets up the initial configurations for the given file session. The different options are:

name = <string>: The name of the file session. This must match with the one listed in *hnet_server.conf* under *server_file_sessions*.

password = <string>: This is the password which must be provided for client to access to that particular file session.

file_list = <string>: A space separated list of all the files in this file session. For each entry in this list, a corresponding file must exist.

Below is an example of a *tsa_files.conf* file:

```
# strings must be quoted if longer than one word
name = "tsa_files"
password = "SimSim8"
file_list = "foo.hsf bar.hsf"
```