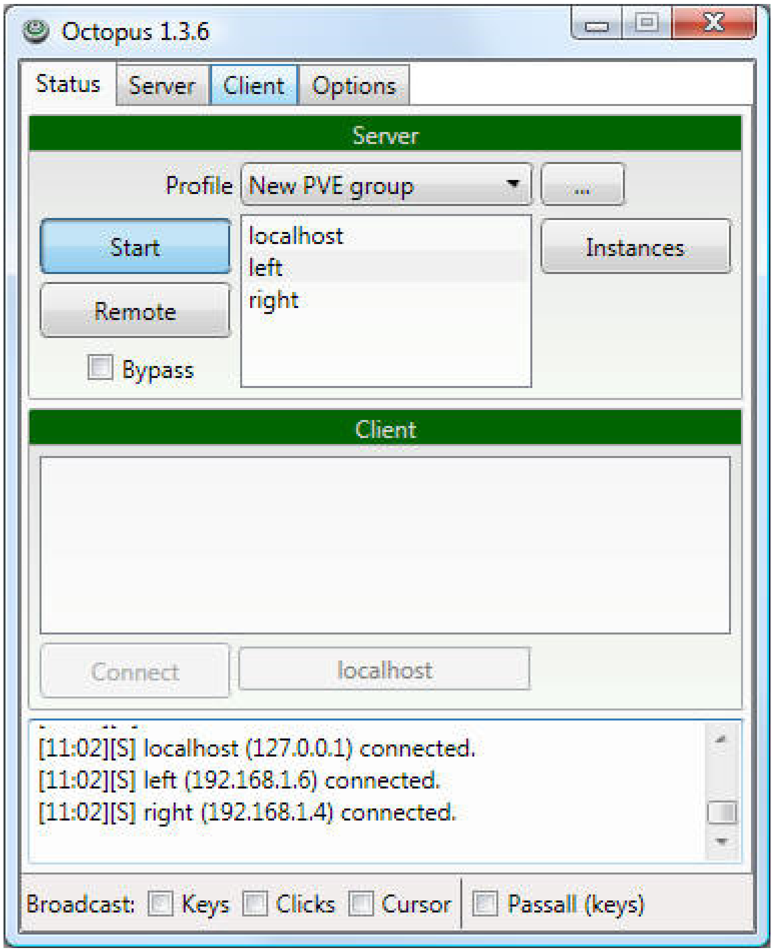
Note: Issues in Vista may arise if not run as an admin. See the FAQ .

This will take you through setup for single or multi PC rigs. Also many tooltips are on labels / controls in the program. Using it in game is not explained here. That requires assist macro's, /follow, etc.

1. Start the server, connect any clients. Read the FAQ if connection issues occur.

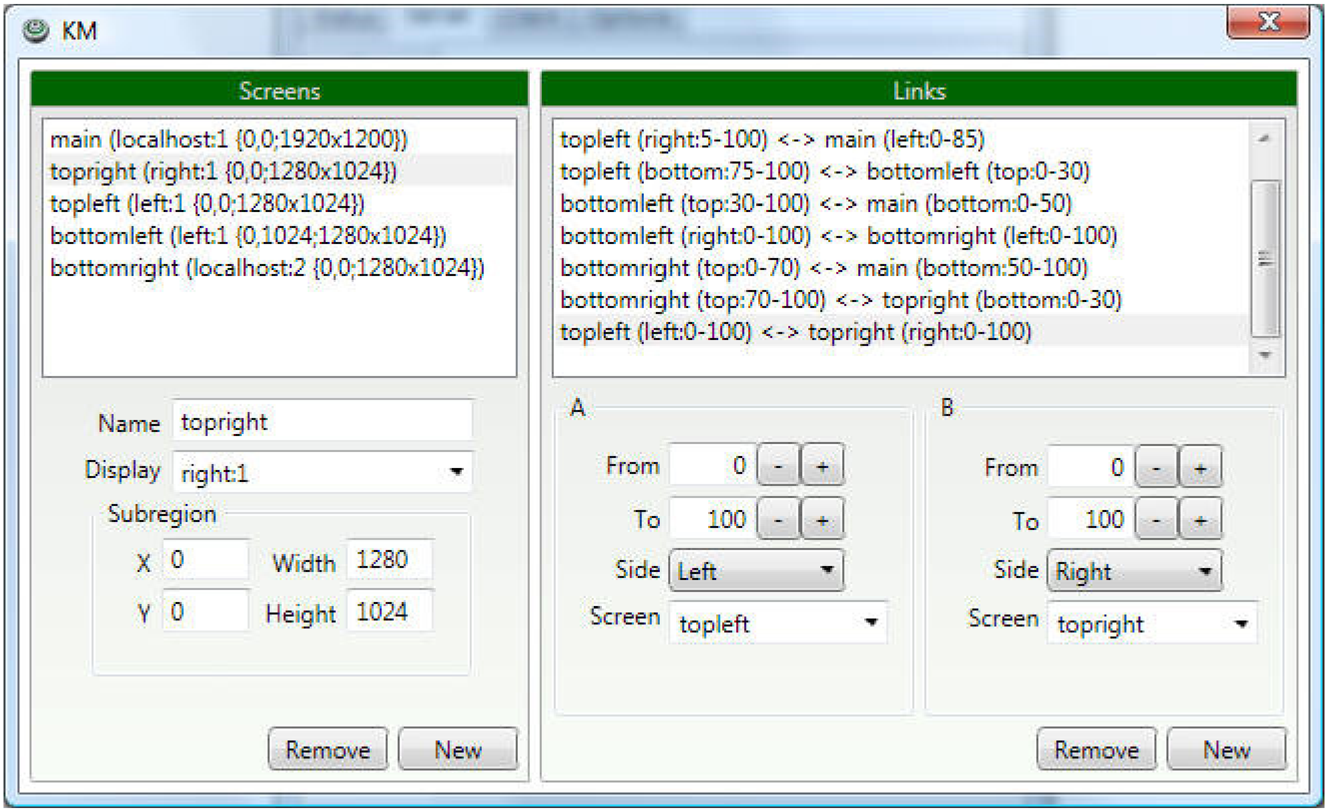
When connecting clients, an IP address or hostname is used in the textbox to the right of the Connect button.



```
[11:02][S] localhost (127.0.0.1) connected.
[11:02][S] left (192.168.1.6) connected.
[11:02][S] right (192.168.1.4) connected.
```

2. Single PC, go to step 5 (the KM is only for multi pc's).

3. Open the Server/KM tab and hit 'Configure...'.



Screens

main (localhost:1 {0,0;1920x1200})
topright (right:1 {0,0;1280x1024})
topleft (left:1 {0,0;1280x1024})
bottomleft (left:1 {0,1024;1280x1024})
bottomright (localhost:2 {0,0;1280x1024})

Name: topright
Display: right:1
Subregion
X: 0 Width: 1280
Y: 0 Height: 1024

Links

topleft (right:5-100) <-> main (left:0-85)
topleft (bottom:75-100) <-> bottomleft (top:0-30)
bottomleft (top:30-100) <-> main (bottom:0-50)
bottomleft (right:0-100) <-> bottomright (left:0-100)
bottomright (top:0-70) <-> main (bottom:50-100)
bottomright (top:70-100) <-> topright (bottom:0-30)
topleft (left:0-100) <-> topright (right:0-100)

A
From: 0
To: 100
Side: Left
Screen: topleft

B
From: 0
To: 100
Side: Right
Screen: topright

A screen is a subregion of a display. Usually a display is a monitor, but may be 2 or more in the case of span modes or other multi screen devices.

A link is a portion of a screen edge that transfers the mouse to another screen edge.

Internally, the program has an active screen for the active client and only checks for links from the active screen. This means you may not need to create a screen entry for every single physical monitor, but there must be a flow between screens. It will not activate a new screen unless linked to. So in my case here, which is just for show, I have 4 physical screens on 'localhost' and 1 on 'right', but 2 screen entries will suffice because it does not bind the mouse inside 'main'.

Tip: If a particular game doesn't allow the mouse to exit over the right or bottom of the screen, try lowering the height/width of the subregion by 1 pixel.

4. Close the dialog and confirm the mouse will cross from one PC to the next.

Note: If the cursor switches PC but then stops moving on the networked PC, it's because UDP traffic (only used for mouse movement) is firewalled or not routing correctly.

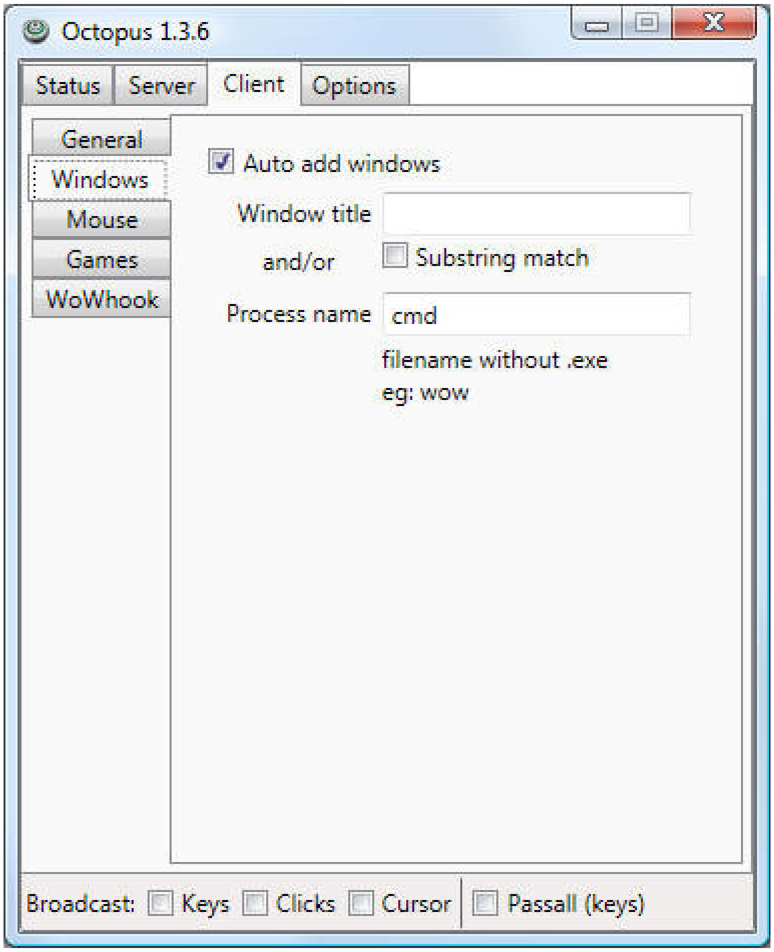
Note: If the KM doesn't work at this point, try opening the KM config again and closing it. You could also restart the server or restart the application to force an update.

5. The simplest way to get started (with key broadcasting) is run a few cmd.exe (notepad.exe will not work like this).

Broadcast keys on the status bar is checked.

From what you see below, pressing any keyboard key should input to windows in the client marked [k] to receive keys.

Incidently [c] is for clicks and [m] is for cursor, the default behaviour for clicks is unbroadcast (ie an empty whitelist), where as the default for keys is to pass them all.



☑ Auto add windows

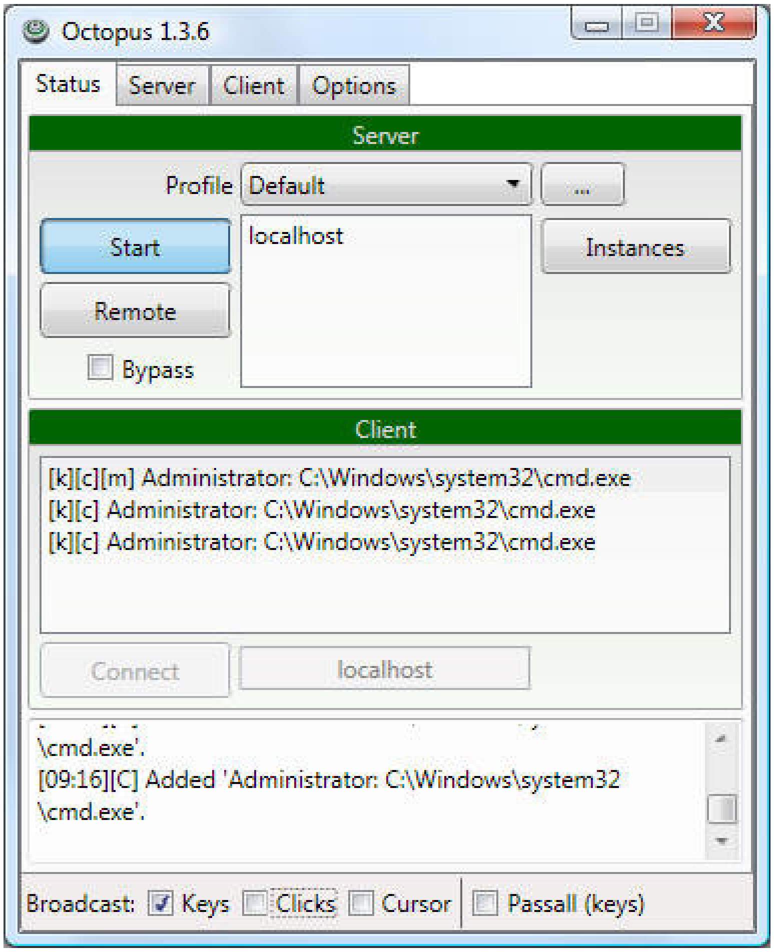
Window title

and/or □ Substring match

Process name: cmd

filename without .exe
eg: wow

Broadcast: □ Keys □ Clicks □ Cursor □ Passall (keys)



Profile: Default

localhost

□ Bypass

```
[k][c][m] Administrator: C:\Windows\system32\cmd.exe
[k][c] Administrator: C:\Windows\system32\cmd.exe
[k][c] Administrator: C:\Windows\system32\cmd.exe
```

```
\cmd.exe'.
[09:16][C] Added 'Administrator: C:\Windows\system32
\cmd.exe'.
```

Broadcast: ☑ Keys □ Clicks □ Cursor □ Passall (keys)

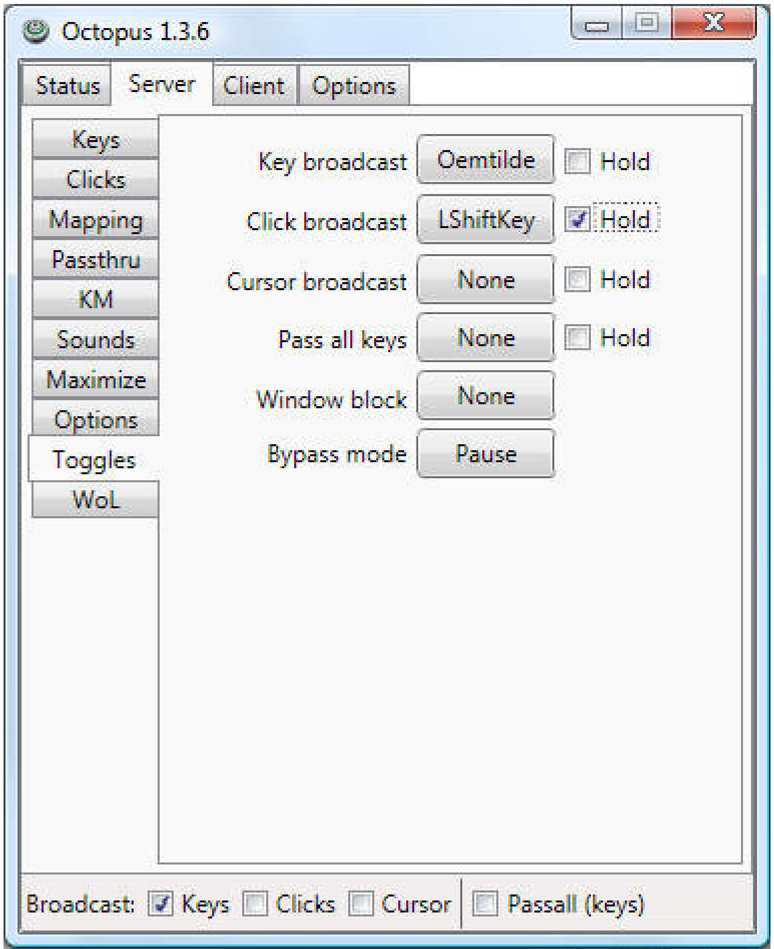
Note: This does not broadcast to network clients with an empty window list. At least one must be on the list.

6. Now I've set ~ to toggle key broadcasting and LSHIFT for clicks. Pressing one of these keys will toggle the checkbox on the status bar.

Toggle keys do not pass through (or broadcast) unless the key is also self mapped on the mapping list (like LSHIFT->LSHIFT). Using LSHIFT like this without the mapping trick (see below) will mean LSHIFT combos on network clients won't function correctly.

'Hold' means down is on and up is off. Broadcast/mapping pausing prevents broadcasting when in any non game window.

With networked Keyclone's, check the anti double-key before you start the server (requires Run as administrator when using UAC) because the order of windows hooks is last in goes to the start of the queue, which is not what you want with a KM too, or Keyclone may see the keystrokes on the server and the client.



Key broadcast Oemtilde □ Hold

Click broadcast LShiftKey ☑ Hold

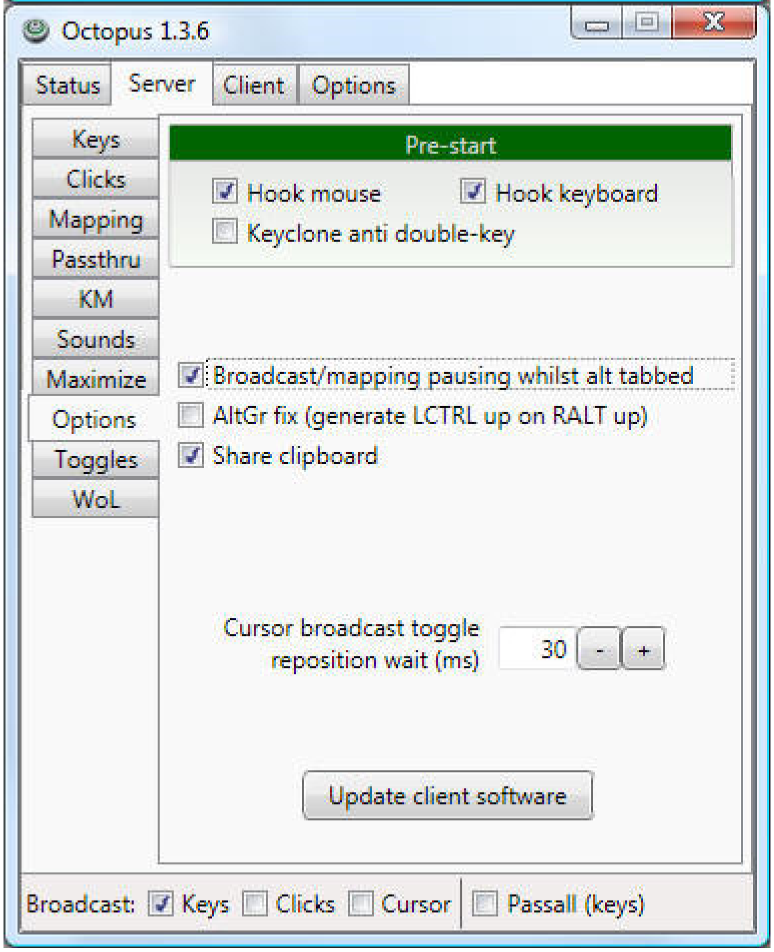
Cursor broadcast None □ Hold

Pass all keys None □ Hold

Window block None

Bypass mode Pause

Broadcast: ☑ Keys □ Clicks □ Cursor □ Passall (keys)



Pre-start

☑ Hook mouse ☑ Hook keyboard

□ Keyclone anti double-key

☑ Broadcast/mapping pausing whilst alt tabbed

□ AltGr fix (generate LCTRL up on RALT up)

☑ Share clipboard

Cursor broadcast toggle reposition wait (ms) 30 - +

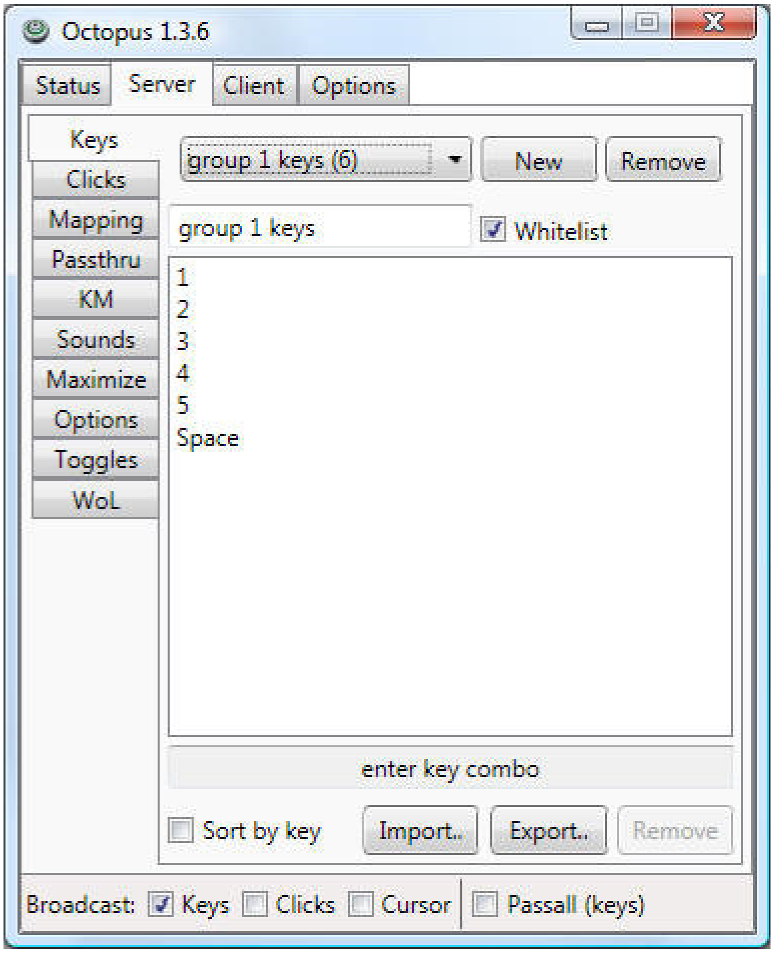
Broadcast: ☑ Keys □ Clicks □ Cursor □ Passall (keys)

7. Now I set up a basic key list, this will send 1-5 and SPACE to all [k] marked windows. Whitelist is checked to only allow certain keys, otherwise it's a blacklist.

I'll also set up a click list with LSHIFT+Btn1.

These lists switch with the profile.

Note: With key broadcasting, modifiers keys (ALT,CTRL,SHIFT) are always sent to each window which enables action bar addons to switch pages. Key-up's are always sent as well (so keys don't get locked down by releasing modifiers before the defining key).



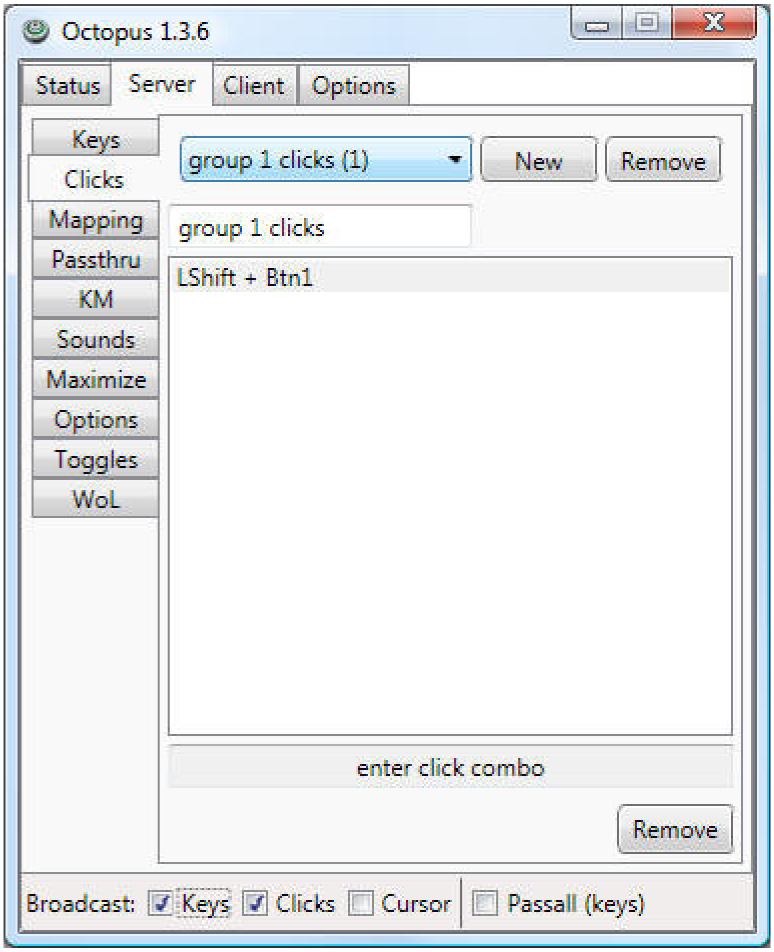
group 1 keys (6)

group 1 keys ☑ Whitelist

1
2
3
4
5
Space

□ Sort by key

Broadcast: ☑ Keys □ Clicks □ Cursor □ Passall (keys)



group 1 clicks

LShift + Btn1

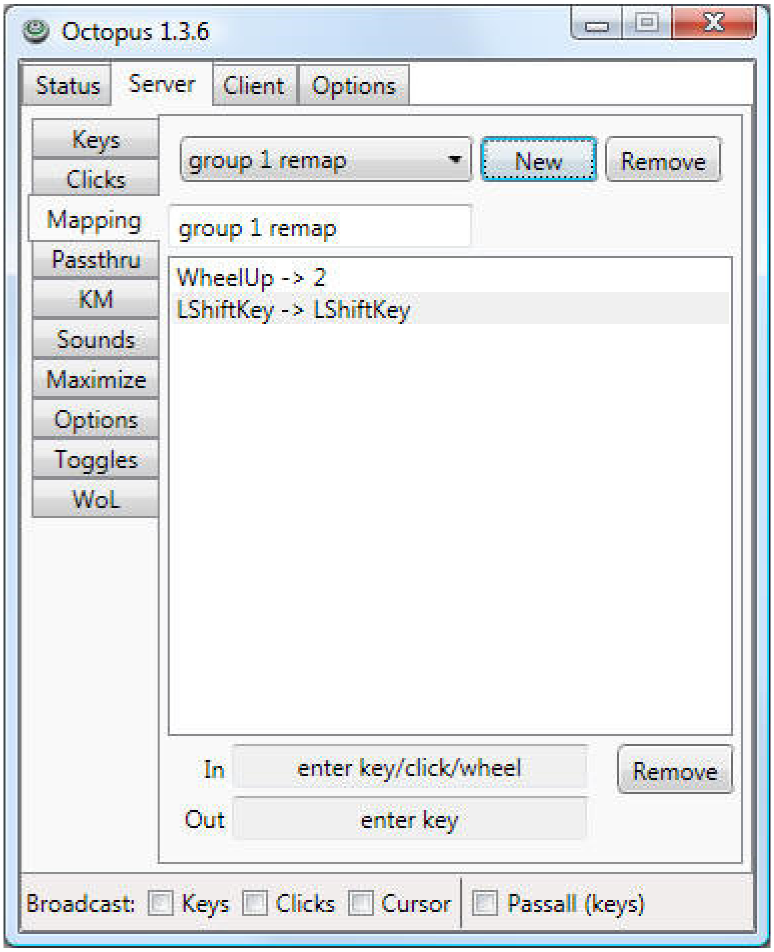
Broadcast: ☑ Keys ☑ Clicks □ Cursor □ Passall (keys)

8. Now only the specific keys above should broadcast, and also LSHIFT + Btn1 should cycle through all cmd.exe windows and click in the same relative position. (actual x,y converted to 0-1 scale along x/y axis and scaled onto target window - i.e, resolution doesn't matter with click broadcasting).

With click broadcasting it is done sequentially on each client (after you release the button).

9. This will map wheel up to 2.

Note: if you have a toggle bound to LSHIFT and it won't enter into the mapping, the bypass option (on the status panel) will allow it to enter.



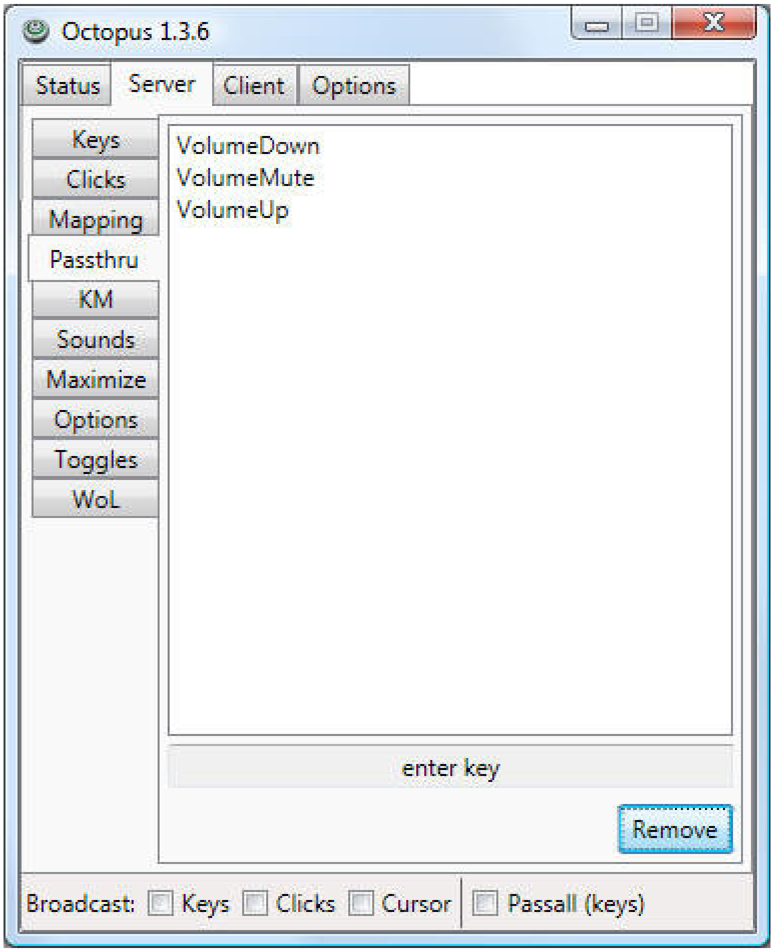
group 1 remap

group 1 remap

WheelUp -> 2
LShiftKey -> LShiftKey

Broadcast: □ Keys □ Clicks □ Cursor □ Passall (keys)

10. The Passthru list means when the mouse is on a client, the keys still stay on the server, useful for keys like volume or Ventrillo.

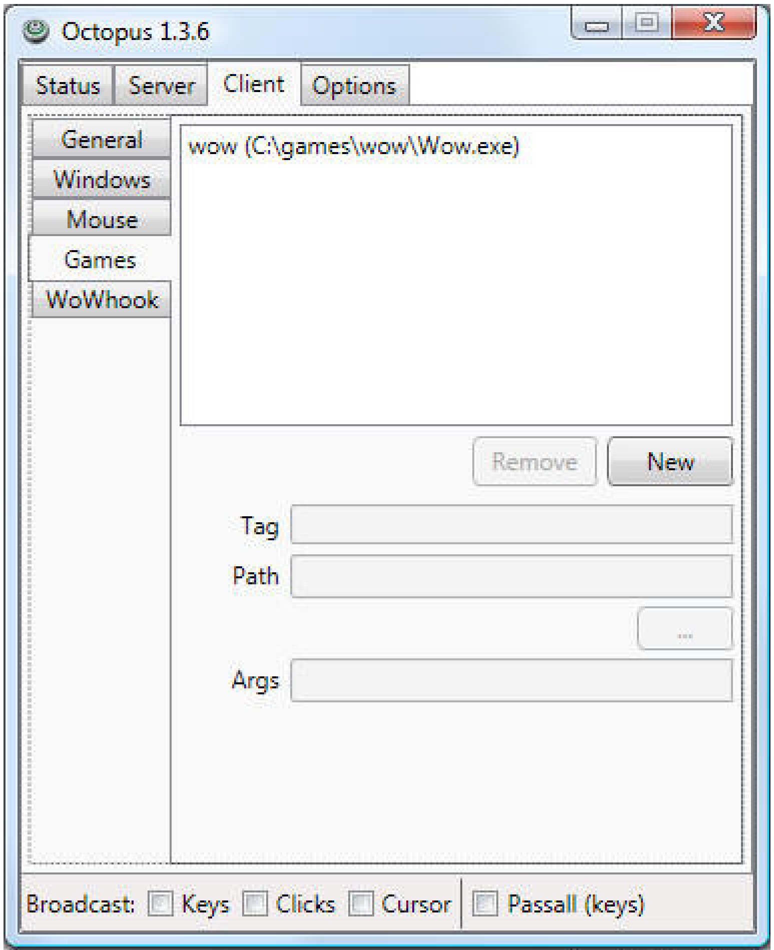


VolumeDown
VolumeMute
VolumeUp

11. Setting up the maximizer and profiles.

Before I create a profile I setup what game EXE's to run (on each client), and where I want my windows to maximize to. (this is for a 1920x1200 monitor).

The maximizer is projected to whatever display is set in the profile.



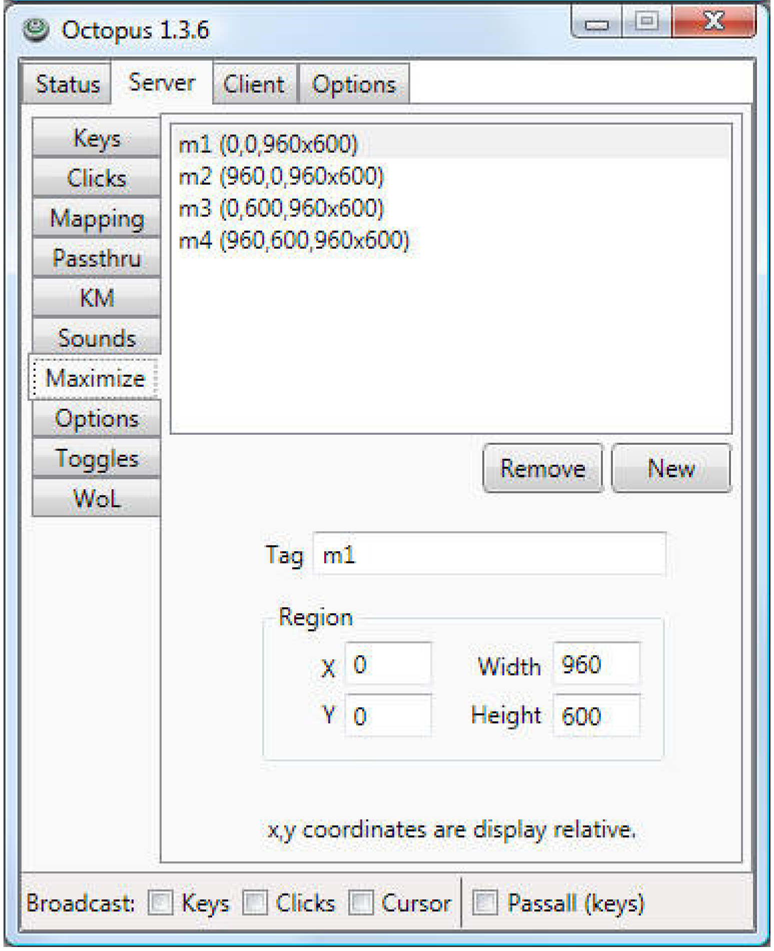
wow (C:\games\wow\Wow.exe)

Tag

Path

Args

Broadcast: □ Keys □ Clicks □ Cursor □ Passall (keys)



m1 (0,0,960x600)
m2 (960,0,960x600)
m3 (0,600,960x600)
m4 (960,600,960x600)

Tag m1

Region

X	0	Width	960
Y	0	Height	600

x,y coordinates are display relative.

Broadcast: □ Keys □ Clicks □ Cursor □ Passall (keys)

12. So I hit the profile "..." button (on the status panel) and enter 4 instances.

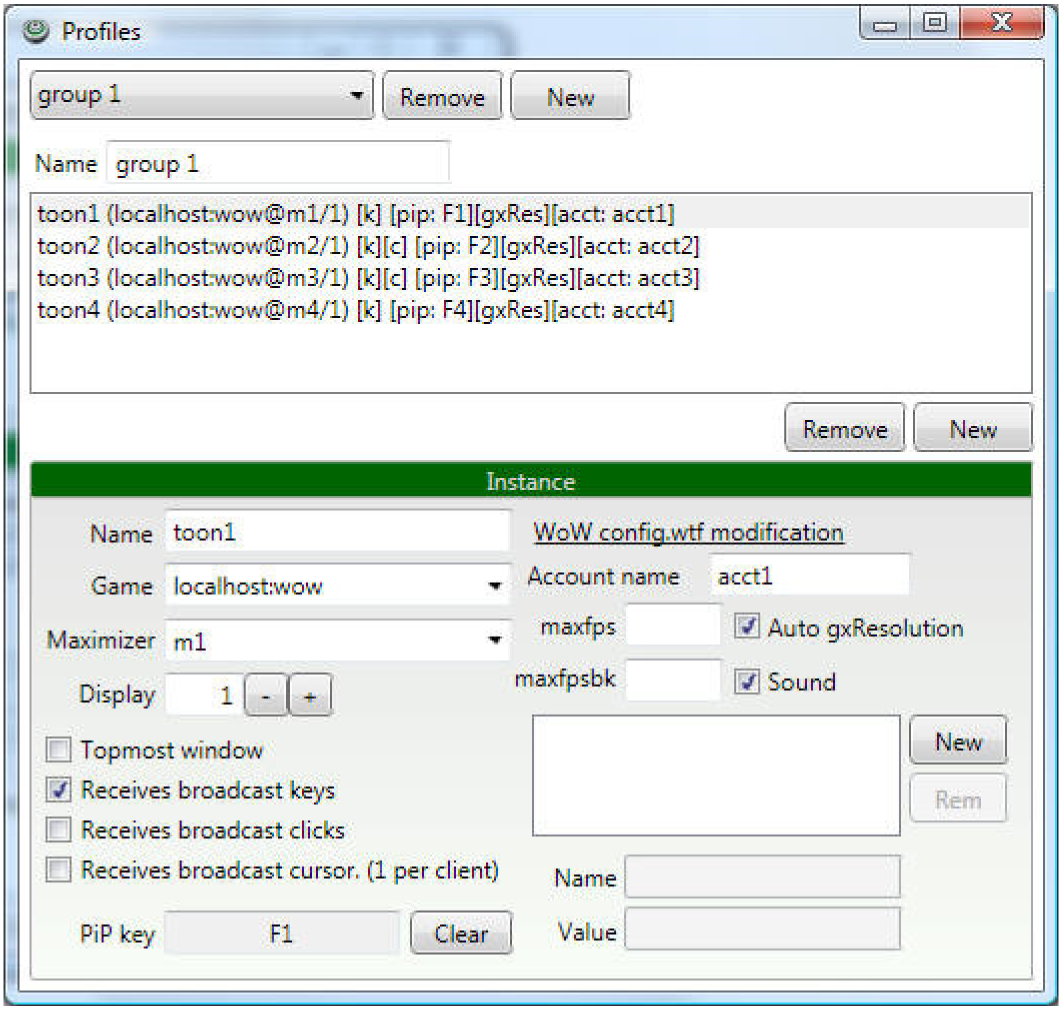
Auto gxResolution means the resolution of the maximizer will be applied to WoW (whether or not WoW supports this resolution is not considered).

maxfps and maxfpsbk limit the FPS (foreground/background).

The PiP key will switch the instance window with the foreground window.

Custom variables can be added also.

When the instance is launched WoW is set to window mode.



Profiles

group 1

Name: group 1

toon1 (localhost:wow@m1/1) [k] [pip: F1][gxRes][acct: acct1]
toon2 (localhost:wow@m2/1) [k][c] [pip: F2][gxRes][acct: acct2]
toon3 (localhost:wow@m3/1) [k][c] [pip: F3][gxRes][acct: acct3]
toon4 (localhost:wow@m4/1) [k] [pip: F4][gxRes][acct: acct4]

Instance

Name: toon1
Game: localhost:wow
Maximizer: m1
Display: 1

□ Topmost window
☑ Receives broadcast keys
□ Receives broadcast clicks
□ Receives broadcast cursor. (1 per client)

PiP key: F1

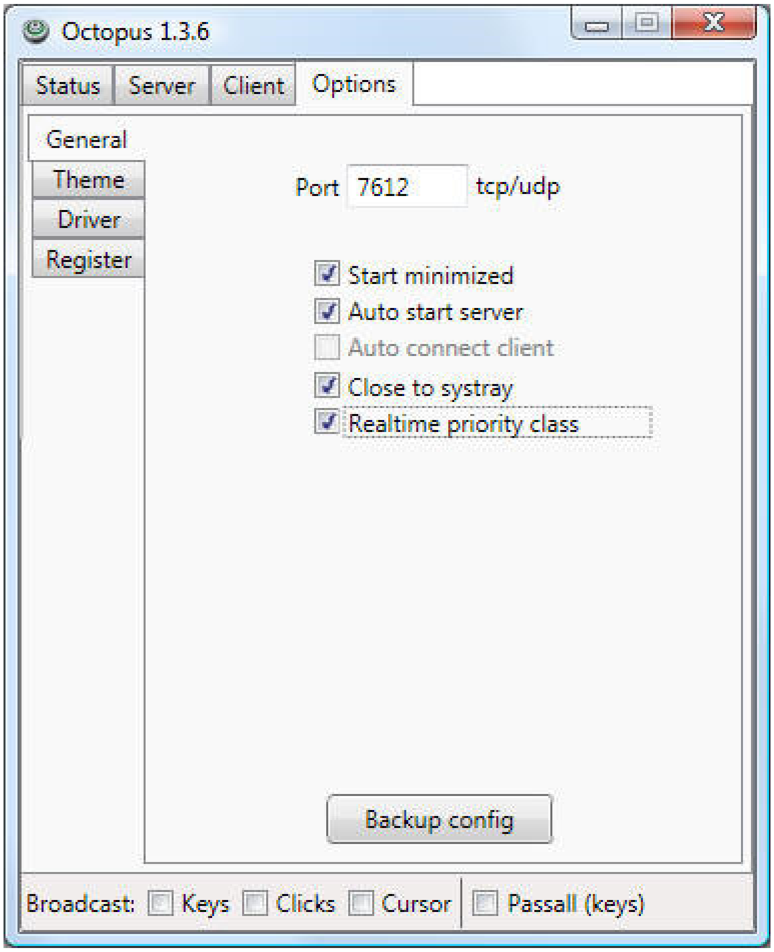
WoW config.wtf modification

Account name: acct1
maxfps:
☑ Auto gxResolution
maxfpsbk:
☑ Sound

Name:
Value:

13. Instances can be run now via the sys-tray icon or the instance button on the status tab.

14. Lastly I change a few options, and set all clients to real time (may help). :-)



Port 7612 tcp/udp

☑ Start minimized
☑ Auto start server
□ Auto connect client
☑ Close to systray
☑ Realtime priority class

Broadcast: □ Keys □ Clicks □ Cursor □ Passall (keys)

Cursor broadcasting

Instances have an option to "receive broadcast cursor" [m]. This is where the position of the mouse relative to the [m] window on the server is projected onto the [m] window on each client.

Using cursor broadcasting will allow listed click downs to process immediately (providing the correct modifiers reach the client via key-broadcasting).

Resolution and aspect ratio are ignored (as with clicks) - it scales to the target window on each client.

My setup was to use LSHIFT for a toggle for clicks and cursor, both in 'Hold' mode, and then target from my main to mages on my clients. To aid this my AoE spells were bound to a LSHIFT+<key> combo, so it would be like LSHIFT, aoe key, and then click, for example. To support this mapping LSHIFT->LSHIFT would also be required, otherwise LSHIFT will not register.

With the wowhook DLL it will broadcast to multi windows on each client, providing the clients have a '[m] window. To see multiple cursors in WoW, you can switch it to software cursor otherwise it's an invisible cursor.

Click timings

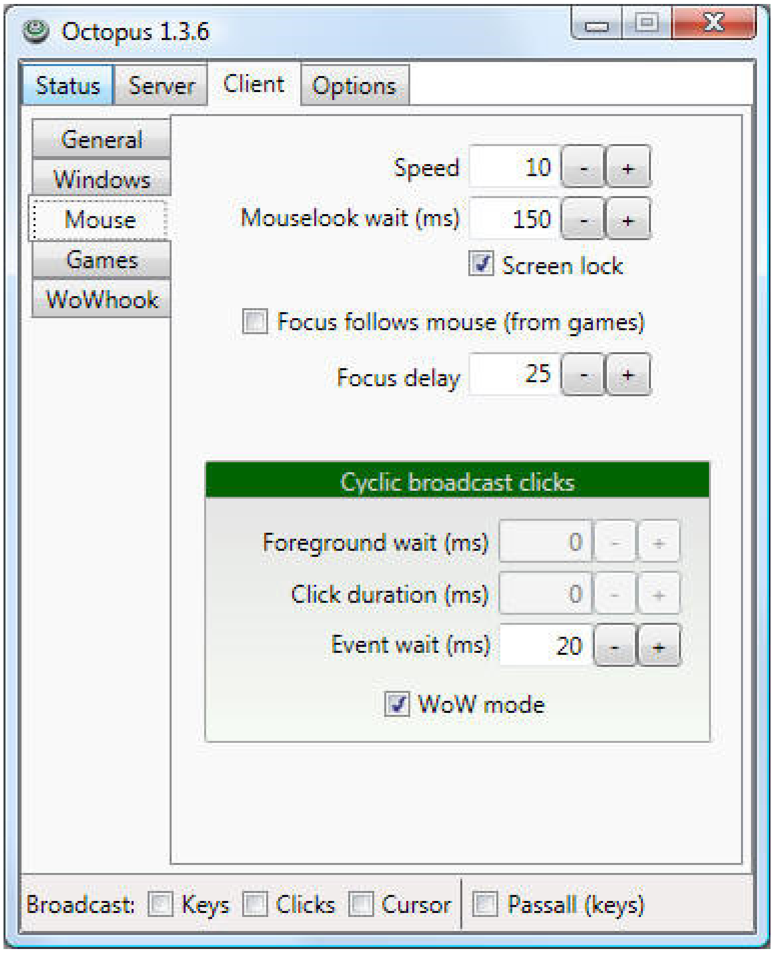
The foreground wait time is in-case a window takes a hit when switching to foreground (like a sound stutter or something). Increase it if the window misses the down click all together.

The click duration is time between down and up, which is not for WoW.

The event wait is the time for the game to respond to the event. How games are coded in regards to click/movement events is an unknown, thus some time is necessary between events (movement and click) before moving onto the next so the camera doesn't jerk and so forth. Increase it if there are issues.

The (non wow mode) sequence for passing clicks to multiple windows on a client is
a) make the window foreground if not already
b) move the mouse to the new click position
c) wait **foreground wait** time if it wasn't foreground
d) wait **event wait** time
e) make the click
f) wait **click duration**
g) release the click
h) wait **event wait** time

In 1.3.3 - 'wow mode' was added which will leave the window in the background and move the cursor around, and try to reduce the time by forcing wow to process messages on the spot.



Speed 10
Mouselook wait (ms) 150
☑ Screen lock
☐ Focus follows mouse (from games)
Focus delay 25

Cyclic broadcast clicks
Foreground wait (ms) 0
Click duration (ms) 0
Event wait (ms) 20
☑ WoW mode

Broadcast: ☐ Keys ☐ Clicks ☐ Cursor ☐ Passall (keys)

Mouse options

Mouse speed is just adjusting the slider in control panel. The original speed will be reset when you exit the program.

Mouselook wait is used firstly with **Screen lock**.

WoW moves the mouse cursor invisibly to the centre of the desktop (or something unhelpful), which potentially triggers links. If Screen lock is not on, it may trigger links.

Secondly it is used when you hold a mouse button with cursor broadcasting and click broadcasting together (as in broadcast mouselooking). Normally it will broadcast absolute positions to clients but it switches to relative movement when a button is down so that you can mouse look on clients. This is a safety margin to stop erratic camera movements. 150 ms is a value I found to work.

Screen lock is only active when in a listed game window.