

Labcontrol Manual

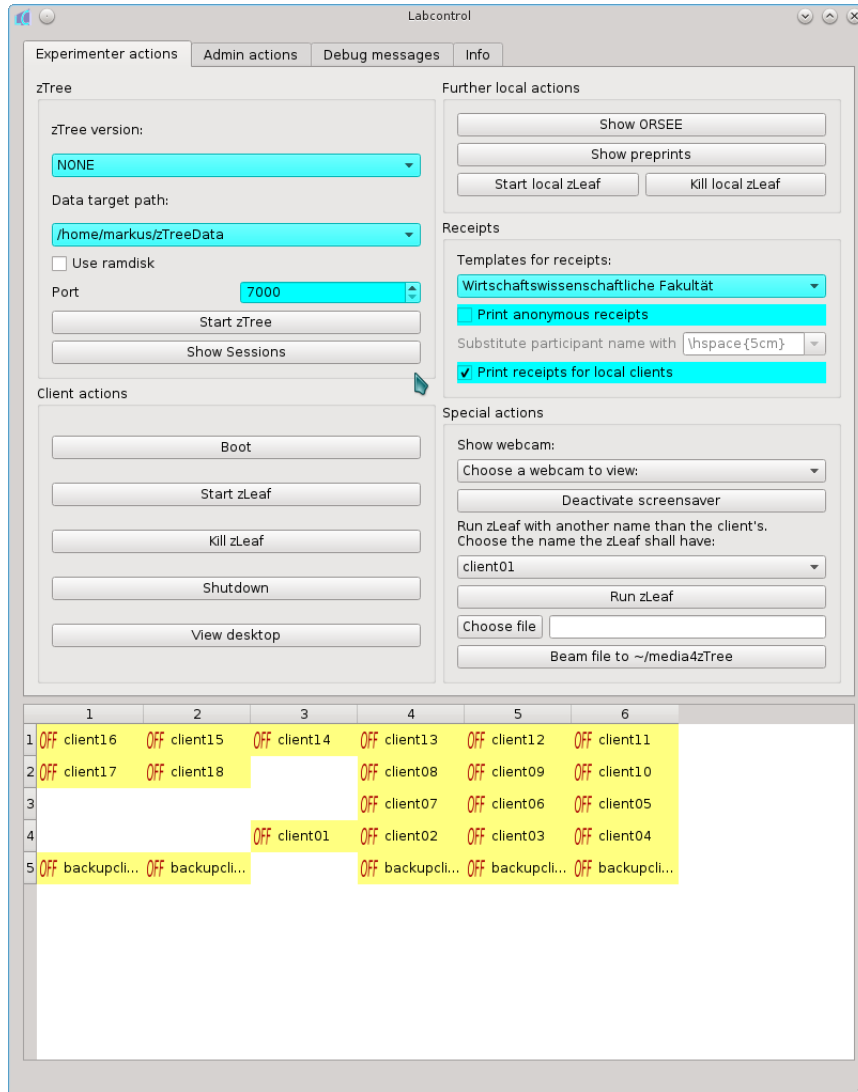
Markus Prasser

June 2, 2020

Contents

1 Introduction

Labcontrol is a tool to control a laboratory used for economic experiments with the software *z-Tree*. This laboratory consists of a *Linux* server and 18 *Linux* clients used to conduct those experiments. The primary functions of *Labcontrol* are the startup and management of experiment sessions, automated receipts printing and easy administration of the clients.



1.1 Working Features

- A *Qt5* GUI for experiments and administration showing the clients' current states
- boot and shutdown the clients, start and kill *z-Leaves* on them using *ssh*
- detect all installed *z-Tree* versions
- automatic receipts printing after creation of the payment file by *z-Tree*
- beam media files to the clients using *rcp*
- watch clients' desktops using *VNC*

- open external programs showing the laboratories' webcams
- execute admin functions on the clients
- automatic and regular querying of the clients' statuses using *ping* and *netstat*
- simultaneous running of multiple *z-Tree* instances on different ports
- administrator and experiment users with different privileges

1.2 Wish-list

- logging support for a better overview over experiments and error discovery in the laboratory
- integration into *ORSEE* for automated experiment finishing
- improve handling of multiple concurrent sessions, currently just viewing them is supported

2 Requirements

Labcontrol was designed and programmed to be as error insensitive as possible. Even with all data and settings (except the executable of course) missing it should still be able to start up and run. Only the functionality will be very limited. This was made to make installing and testing as easy as possible by allowing a step by step approach.

2.1 Files And Basic Environment

Labcontrol was tried to be programmed as versatile as possible, e.g. all important paths are configurable with *Qt*'s *QSettings* configuration system. Also we tried to include no compilation time dependencies except *Qt5*.

- Your setup should consist of a server running *z-Tree* and clients to run the *z-Leaves*.
- Our laboratory works fine using *Debian Jessie* with a self compiled *wine* in version 1.4.1.
- All clients should be listed in */etc/hosts* on the server and the server on the clients. For most functions the IP addresses are used directly, but the status querying relies on hostnames.
- All clients should have a special user dedicated for experiments. Since *Labcontrol* uses *ssh*, password-less public-key authentication must be available at least for the user. For administrative tasks on the client also *root* must be available for public-key authentication.
- All *z-Tree* and *z-Leaf* executables should be stored in a central folder on the server and the clients with each version in a separate folder following the naming scheme **zTree_X.Y.Z**, otherwise the automatic detection will not work.
- The directories in the **data** directory should be copied to the so called *labcontrol installation directory*, both on the server and the clients.
- The **Labcontrol.conf** file should be copied to the place where *Qt* stores the *QSettings* files (on *Linux* this should be */etc/xdg/Economic Laboratory*). Afterwards it should be adjusted to match the laboratory's specifications.
- To automatically deploy media files on the clients in the client's experiment user's home directory there should be a read-write accessible directory **media4ztree**

2.2 External Runtime Dependencies

- on *Linux* *wine* to run the *z-Tree* and *z-Leaf* binaries
- a wake-on-lan program for remote client startup
- public-key enabled *ssh* access to the clients
- optionally a *VNC* client to show the clients' desktops (and a *VNC* server on them)
- a web browser to show *ORSEE*
- *latex*, *dvips*, *ps2pdf* and *lpr* for automatic receipts creation and printing
- a postscript viewer to display the generated receipts

3 Compilation And Installation

To compile *Labcontrol* it should be sufficient to open the `Labcontrol.pro` file with *Qt Creator* and compile after choosing the desktop profile.

In the compilation process a special build directory will be created in which the compiled executable will be found after successful compilation. You can place this executable anywhere you want. On *Linux* we suggest to place it in `/usr/local/bin`. *Linux* users also can use the supplied `labcontrol.desktop` file and place it in `/usr/share/applications` to have an accessible desktop icon.

Like already mentioned all *z-Tree* and *z-Leaf* versions should be copied to a central directory, the *zTree installation directory*. This directory will be automatically scanned for installed *z-Tree* versions by searching for folders matching the pattern `zTree_X.Y.Z`.

The directories in the `data` directory should be copied to the *zTree installation directory*.

Afterwards place the `Labcontrol.conf` in a subfolder `Labcontrol` in the place where *QSettings* stores its definitions on your platform (on *Linux* this would be `/etc/xdg/Labcontrol/Labcontrol.conf`) and adjust it to your needs and prerequisites.

To allow *Labcontrol* to control the clients password-less public-key authentication must be available at least for the experiment user on the clients. For administration this is also required for `root`. Also all clients should be known by `/etc/ssh/ssh_known_hosts`. This file must be readable by the users! If this is missing *Labcontrol* will query the user on every connection attempt, which is not user-friendly.

4 Using Labcontrol

After startup *Labcontrol* initializes all available components and loads its settings stored in *QSettings*. On any possible error *Labcontrol* will display an error message what type of error occurred and what consequences it has for the functionality of *Labcontrol*.

Afterwards *Labcontrol* will greet the user with its main window. This consists of a table view at the bottom showing all known clients and their current states. This view may take some seconds to initialize depending on the time it takes to query the initial client statuses. In the upper area there is a tabbed area consisting of three tabs. Most users will only use the *Experimenter actions*

tab which hosts all options needed to conduct experiments. Only admins are allowed to use the *Admin actions* tab, which will be deactivated for all other users. The *Debug messages* tab does exactly what its title says: display debug messages. It's useful for checking configuration and other errors but is of no interest for normal users. At last license and building information will be shown in the *Info* tab.

4.1 The Experimenter actions tab

This tab consists of five group boxes grouping tasks and options of one kind. The first one is the *z-Tree* one. Before you start a session you need to choose a particular *z-Tree* version, the port you want to use and where the data shall be stored. If this path does not exist it will be created by *Labcontrol*. Since this path will be passed to *z-Tree* please do not use any special characters of spaces in it. Otherwise there will be the possibility of data loss. After all settings where done *z-Tree* can be started by clicking the respective button. Because of the possibility to run multiple simultaneous sessions (which likewise means *z-Tree* instances) a function to display all active sessions was also implemented. It allows no modifications but just the display of currently running sessions.

To the start of a session is also bound the start of all receipts printing facilities. So users are strongly advised not to kill *Labcontrol* while conducting a session because then the automated receipts printing will not work. *z-Tree* is always started as a stand-alone program so killing *Labcontrol* or a crash of it will mean no harm to *z-Tree*.

The *Client actions* and *Further local options* group boxes are self explanatory.

Another important group box for the sessions is the *Receipts* group box. All settings made there apply to the next started session and cannot be altered after starting a session. The user is able to choose between several installed receipt templates (if there are any installed) and can optionally make the receipts anonymous. If the user chooses to do so he is also able to choose the string which shall replace the user name. Printing receipts for local clients (*z-Leaf* instances on the server) can be deactivated. This only works if the name given to the local client contains the string `local`. Also the user should take care, that none of the participants has `local` in his name, because then this participant would not get a receipt generated.

The last one is the *Special actions* group box. This allows you to choose a webcam to show. After choosing a webcam an external program will open showing a window with the cameras view. Second there is a button to manually deactivate the screen savers of the clients. In some cases (e.g. replacing a client with a maintenance one) *z-Leafs* can be started with others names than their own client's host name. To do this the user can choose the name of any client and afterwards click the *Run z-Leaf* button. Last there is the option to beam files to an already existing directory `media4ztree` in the client user's home directory. All files which shall be beamed should be placed in one folder. The folder name and the in it contained files' names should also include no special characters of spaces.

4.2 The Admin actions tab

This tab allows to do some administrative tasks. The user can open terminals or file managers connected to the clients. Also specific commands can be run for the entire laboratory.

4.3 The Debug messages tab

This tab is very useful for installing and debugging. All configuration options get listed there and if you run a command or click a button (e.g. the "Start z-Leaf" button) the command behind this function is displayed there, so you can copy and paste it into a terminal and check there, why it is not working.