

Experiment Wizard

Overview

This section describes the problem that researchers might have while running experiments.

Problem

Multiple computers with multiple applications can be used to run experiments (see Illustration 1). It is up to the experimenter to go through a checklist to start and configure all the hardware/computers and all the applications on all the computers. The problem is that this process is error prone and time consuming.

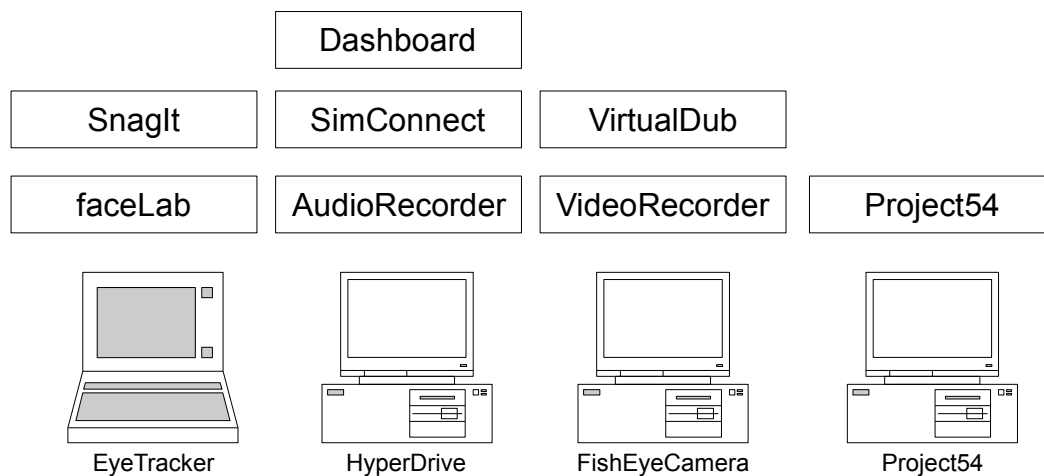


Illustration 1: Sample of computers and applications used in experiments in [Project54](#)

Goal

The goal is to streamline the process of running the experiments by allowing all the applications to be configured from a central locations, while minimizing manual intervention from the experimenter. This goal is motivated by the fact that automatic solution should reduce the chance of a human error in setting up experiments and hence result in a more consistent data collection.

Approach

All participating computers are connected into a single network. One computer is used as a controller. This computer sends configuration parameters for applications on all computers. Each computer has a configuration server running. This server applies received configurations to the applications specific to that computer.

An experiment often times consist of multiple stages, each of which might require different configuration of the participating computers and applications. These stages are described in a

configuration file used by ExpWiz application¹ described below.

Setup

Illustration 2 shows the flow of communication from a controlling computer to a controlled computer. **Experiment Configuration** contains the information about the settings required for the experiment. Experimenter uses **ExpWiz GUI** to run the experiment. **ExpWiz** communicates to the controlled computers using perl client/server. Every controlled computer has a set of **Application Configuration** scripts designed to configure **Experiment Applications**.

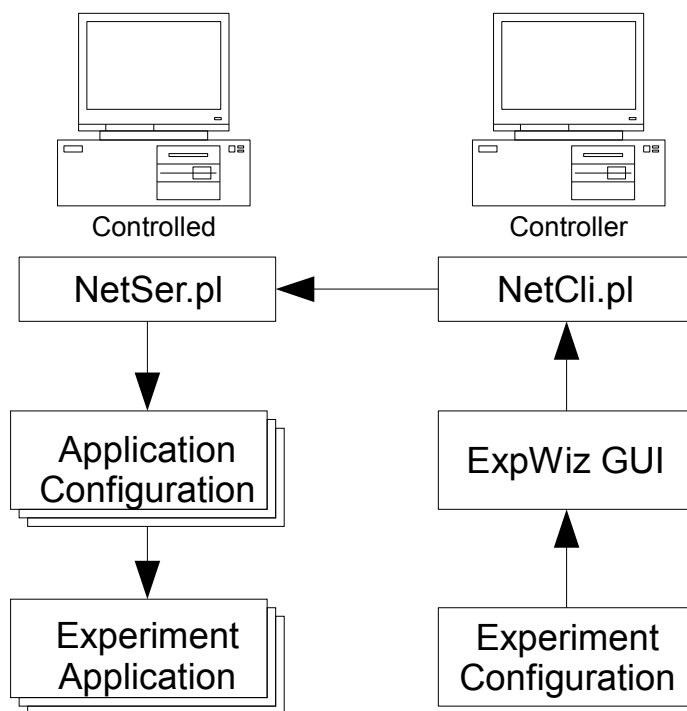


Illustration 2: Communication process

Tools

On Microsoft Windows AutoHotKey² automation application can be used to perform GUI manipulations for the applications that do not support command line configuration.

Perl³ is used to create automation scripts.

1 <http://expwiz.sf.net>

2 <http://autohotkey.com>

3 <http://perl.org>

ExpWiz

Experiments can be organized in stages. With each stage requiring possibly different configuration of the hardware and software. Stages are a set of commands that can be performed as a unit. For example an experiment can have the following stages:

1. Hardware power up
2. Experiment
3. Data collection
4. Hardware power down

Most hardware will require you to power it up manually, so ExpWiz can perform a role of a checklist. Execution of **Hardware power up** stage can prompt an experimenter to power up all participating pieces of hardware one by one. Once everything is ready for an actual experiment the execution of the **Experiment** stage will start data acquisition (by starting appropriate applications and starting recording). Once the experiment is over, execution of **Data collection** stage will stop and data recording and copies the collected data to a predefined location. Execution of **Hardware power down** stage could send messages to the hardware to power down or prompt the experimenter to power down the hardware manually.

All these operations are specified in the configuration file that is required to run ExpWiz.

Configuration

ExpWiz uses an xml file with Experiment Configuration, with the schema defined by bin/ExpWiz.xsd. Template for new configuration files can be generated from the schema file⁴.

Graphical User Interface

This section describes ExpWiz application. When application is executed without command line parameters, the user is shown help about possible command line options. The only required command line option is **--ExperimentConfig** or **-c** with the configuration file for the experiment.

```
ExpWiz.exe --ExperimentConfig YourFile.xml
```

By default the GUI shows a welcome message with the description of the experiment (see Illustration 3). It's possible to disable this by using a command switch **--NoIntroduction**.

⁴ Free community edition of Liquid XML Studio - <http://www.liquid-technologies.com/> can generate xml from the schema.

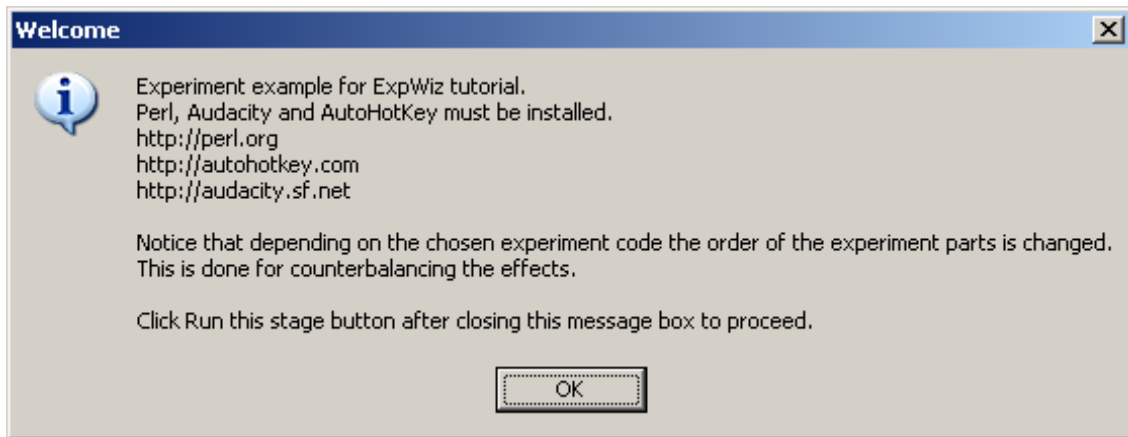


Illustration 3: Example of a welcome message

The next step is to choose one of the experiment codes declared in the configuration file (see Illustration 4). It is possible to add a suffix to the experiment code. This is required to cover a situation when the same experiment must be repeated, but previous data files must be preserved. This is an emergency tool.

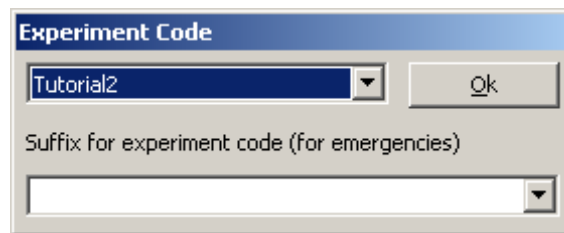


Illustration 4: Subject code dialog

Illustration 5 shows the screen shot of the main window of the application. Log pane shows the log of the experiment. The same information is duplicated in the log file (can be specified with --LogFile option.). Notice that log pane uses different colors to differentiate between the log entries. Errors are displayed in red, while output of the executed commands is displayed in gray; black is used for ExpWiz messages. Commands pane shows the stages and commands inside the stages. When selected more information about them is displayed in Controls pane. Controls pane also has buttons to control the experiment. Use **Run this stage** button to execute all the commands in the current stage. It's possible to select individual commands and execute them with **Run Selection** button. Status bar will display the information about the currently executed command. **Abort** button can be used to stop execution of the commands. **Back** and **Skip** buttons are used to change the current stage.

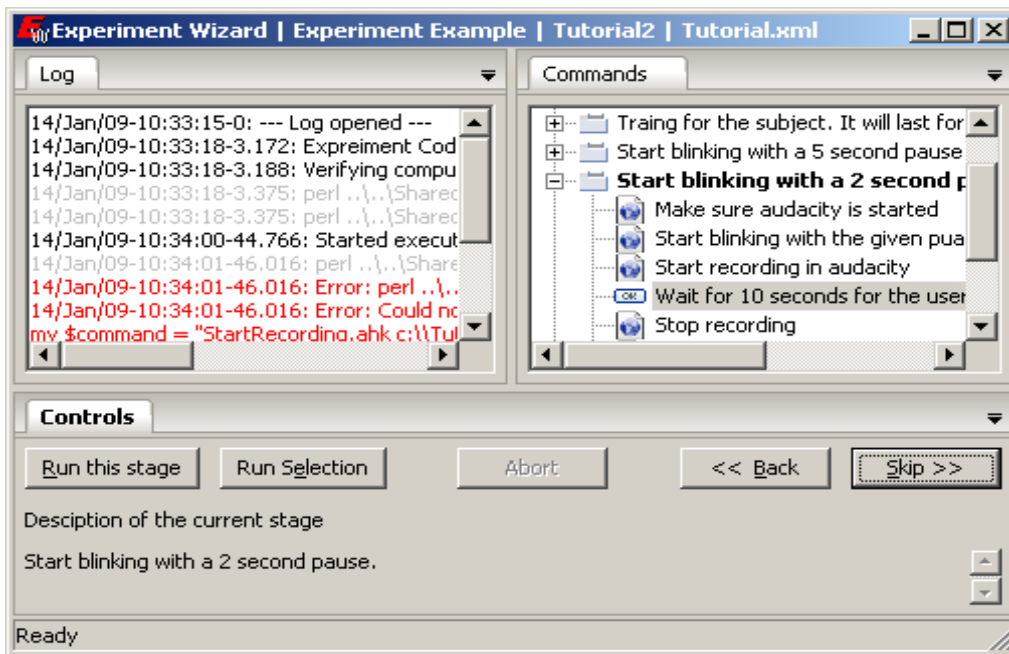


Illustration 5: ExpWiz screen shot

Execution of the last stage, will result in the termination of the ExpWiz application. It will also update the settings file for the experiment that stores the information about last successful experiment. When the same experiment is launched again the next experiment code is chosen by default.

Most messages are presented using Notification dialog (see Illustration 6). This dialog can be timed. If **Auto Close** is checked, the dialog will close automatically. Otherwise the experimenter is expected to click **Proceed** button. **Abort** button can be used to abort the execution of the current stage.

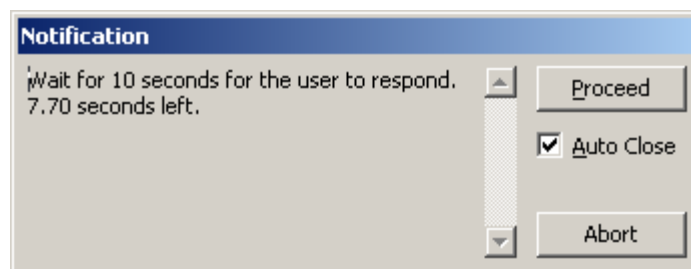


Illustration 6: Example of the notification dialog

Server/Client programs

ExpWiz uses NetSer.pl and NetCli.pl⁵ perl scripts to organize communication between the computers. Every computer that will be used by ExpWiz must run NetSer.pl.

Use IP of the computer as the first parameter when running NetSer.pl, otherwise the server may not work as expected under all operating systems.

The working directory from which NetSer.pl is executed is important. Relative file names given in the attribute **RemoteFile** are relative to this directory. This directory may be changed through a perl command if it is necessary, but usually all commands are given relative to that path without a need to

⁵ <http://pc20m229.unh.edu/svnrepos/hyperdrive/Automation/Shared/>

change the working directory.