Last saved 12 October 2004
C:\kailua\distributionCD\subDBmanual.doc
SubDB Manual

Overview

The subject database program (SubDB) is used to enter information about the experimental subjects and sessions whose psychophysiology data will be analyzed using puka. In the SubDB a *subject* is an individual from whom data will be collected and a *session* is a time that data is collected from that person for a particular experiment. Each subject should only be entered into the database once but may have multiple sessions. It may be convenient to create a separate database for each experiment, but not necessary. The database read by SubDB does not store the raw psychophysiological data files within the database but rather creates a hierarchy of zipped data files, whose location it stores in the database. The icon is ©2000 Saho, and is available at Saho's IconRoom, `http://saho.vis.ne.jp/`.

Installation

Both Java and MySQL must be installed before SubDB. These are both free programs. SubDB has been installed on Windows 2000 and Windows XP; its compatibility with other operating systems is unknown.

Java

SubDB requires the Java version 1.4 or higher runtime environment (J2SE JRE), which can be downloaded as a self-installing executable from `http://java.sun.com/` by following the download links. Java is available at no cost under Sun's Community Source License Program (SCSL, `http://wwws.sun.com/software/java2/license.html`). After installation the directory in which `java.exe` is located (such as `c:\j2sdk1.4.2_04\bin\`) must be added to the computer's system path.

MySQL

MySQL is an open-source database program available free of charge under the GNU General Public License (GPL). The most recent Generally Available release of the MySQL database server and standard client can be downloaded as a self-installing executable from `http://www.mysql.com/downloads/index.html`. Follow the instructions to start the MySQL server before running SubDB. On Windows XP the server is started using `winmysqladmin.exe`;

consult the MySQL documentation for more details. The database creation code and directions are included in the puka manual.

SubDB

The SubDB directory and all its contents should be copied to a convenient place on the computer, such as `c:\program files\subDB\`. Do not extract or otherwise modify the files in the SubDB directory.

Configuration

Preliminary Steps

Before SubDB can be used the database to store the information about the subjects and sessions must be created. Also, the subjectDataConfig.txt text file needs to be edited so that SubDB will use the correct database and directories.

1. Start the MySQL server, if it is not already running. Follow the MySQL installation directions for your operating system and configuration.
2. Open a dos command window by using the mouse to click on `Start`, then `All Programs`, then `Accessories`, then `Command Prompt`. The path to `Command Prompt` may vary on different versions of windows.
3. Access MySQL by typing `cd C:\mysql\bin` (assuming MySQL was installed to `c:\mysql`) at the command prompt. At the next prompt type `mysql`. This should cause the command prompt to change to `mysql>`, indicating that you can send commands to MySQL.
4. At the `mysql>` prompt type `create database NEWNAME;` (where `NEWNAME` is the name you will use for the new database) followed by enter.
5. The command window should print a line reading `Query OK, 1 row affected (0.00 sec)`. If different output appears something has gone wrong. You may need to log in to MySQL with a different username (such as `mysql -u root`).
6. Next, type `use NEWNAME;` and press enter, after which the words `Database changed` should appear.
7. Next, type `source C:/programming/makeANewDatabase/makeDBscript.sql;` replacing `c:/programming` with the path to the location at which you installed SubDB. MySQL should output many lines, ending with `Query OK, 1 row affected (0.00 sec)`. If these lines appeared the database was made properly and the command window can be closed by typing `exit`.
8. In the windows file explorer navigate to the `SubDB/makeANewDatabase/` folder. Make a copy of the file entitled subjectDataConfig.txt and move the copy to the SubDB installation directory (the directory that contains SubjectDatabase.jar).
9. Rename the copy as subjectDataConfig.txt and ensure that it is not read-only (right click the file and select `Properties`, then uncheck the `Read-only` box if it is checked).
10. Open subjectDataConfig.txt in a text editor. This file has several lines beginning with `dbname` and one with `savePlace`. Change the first `dbname` line to your database's name (`NEWNAME`) so it looks like `dbname=NEWNAME`. The other `dbname` lines may be left unaltered, deleted, or if you use more than one database, enter their names on separate `dbname` lines.
11. Change the path listed on the `savePlace` line to the directory in which you want SubDB to store the zip files it will create. DO NOT set this to an existing directory but use a new directory or path to an external hard drive.
12. Save the changes to subjectDataConfig.txt, leaving it in text format.
13. Move the `shortcut to SubDB` from the SubDB installation directory to the desktop or other convenient location.
14. If Windows has been previously configured to recognize self-executing jars double-click the shortcut to SubDB to start the program. If not, the "Executable Jar File" file type needs to be added to Windows by following Microsoft's instructions or the ones at `http://vietunicode.sourceforge.net/howto/file_association.html`. Alternatively, open a dos

window and navigate to the SubDB installation directory. SubDB can then be run by typing `java -jar subjectDatabase.jar` at the command prompt.

15. If the installation and configuration succeeded the SubDB program will start with the `Connect to the Database` dialog loaded, as shown in Figure 1. From the list of database names in the `connect to this database` box select the database you wish to use (these names are read from the `dbname` entries in subjectDataConfig.txt). Enter the username and password if required by the database and click `OK`.

16. If no errors were output the installation and configuration of SubDB is complete and the program can be used.
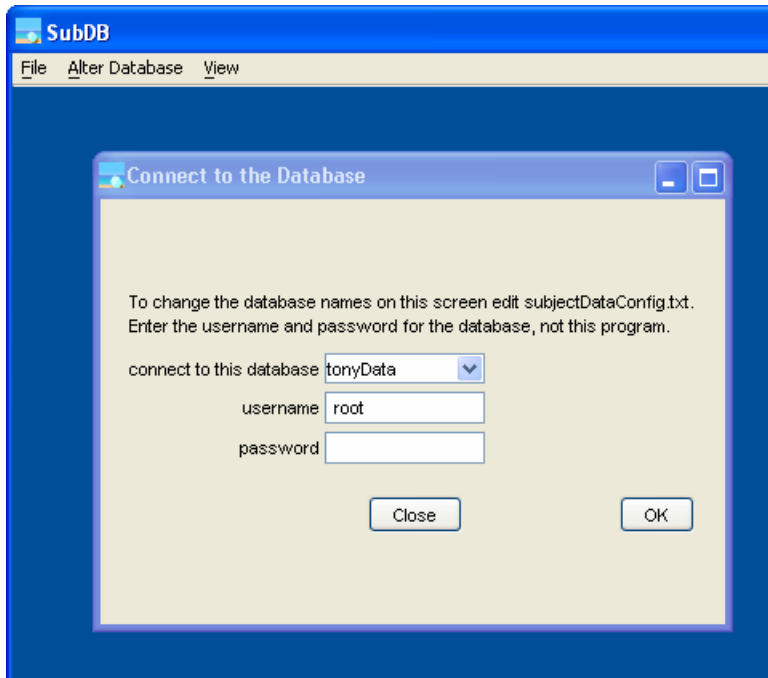

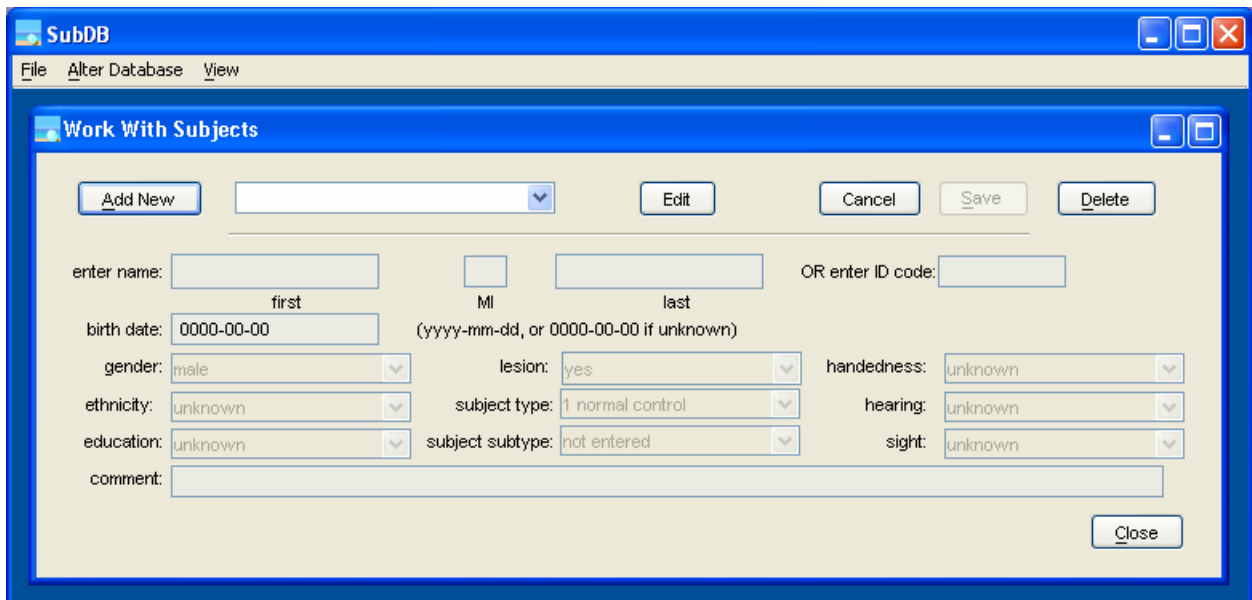**Figure 1. Appearance of SubDB at startup.**

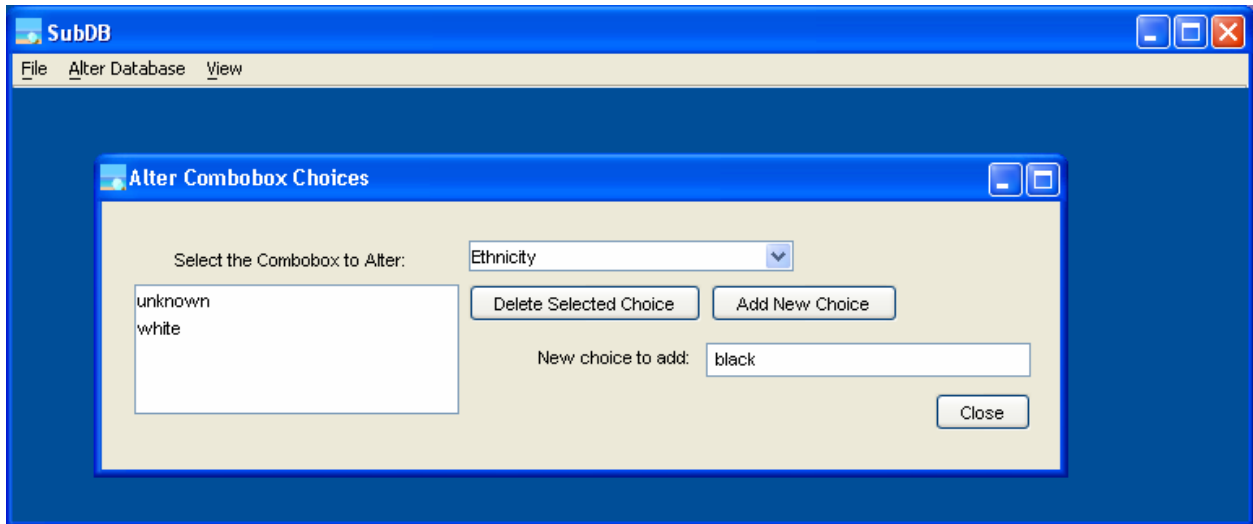**Figure 2. SubDB `Work With Subjects` dialog, used to modify subject information.**

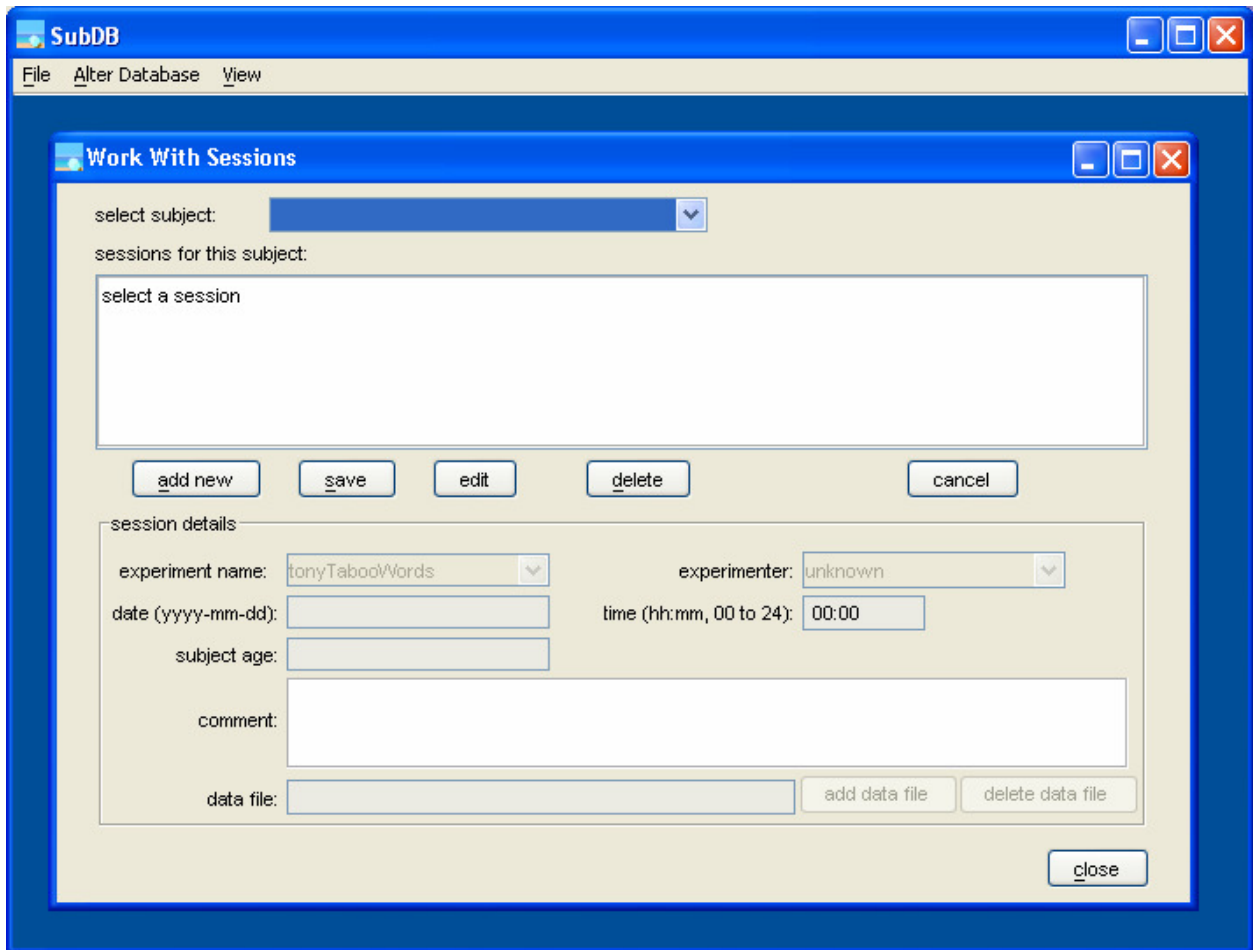**Figure 3. SubDB dialog to change combobox items.**

Figure 4. SubDB `Work With Sessions` dialog, used to modify session information.

Combobox Choices

Many types of subject and session information (such as the subject's gender and the experiment name) can be set by choosing from the options listed in comboboxes on the `Work With Subjects` and `Work With Sessions` dialogs. When the subject database is created in MySQL an initial set of choices are written in to the database. These choices are changed using the `Alter Combobox Choices` dialog and saved in the database. Since the combobox choices are saved in the database every database can have its own set of options.

The `Alter Combobox Choices` dialog is opened by selecting `File` then `Change Combobox Choices` from the top SubDB menu. This dialog, shown in Figure 3, lists the comboboxes that can be changed in the `Select the Combobox to Alter` box. When a box's name is selected the list at the left side of the dialog shows the items in the box. An existing item can be removed by highlighting it and clicking `Delete Selected Choice`. A new item can be added by typing it in the `New choice to add` box and clicking the `Add New Choice` button.

For example, in Figure 3 the *Ethnicity* combobox is selected. This box contains two choices: *unknown* and *white*. *black* is typed in the `New choice to add` box and so will be added beneath *white* when the `Add New Choice` button is clicked.

Working with Subject Information

To add subjects to the database select `Alter Database` then `Work With Subjects` from the top SubDB menu, which opens the `Work With Subjects` dialog. The initial appearance of this dialog is shown in Figure 2. This dialog is also used to view and change subject information and to delete subjects. At least one subject must be added to the database before working with sessions.

Adding a New Subject

A new subject is added by clicking the `Add New` button on the `Work With Subjects` dialog. When this button is clicked the lower part of the dialog becomes active and can be changed. Every subject needs to be given either a name (first, middle initial, last) or an ID code (letters, numbers, or a combination) on this screen. SubDB will assign an identification number to every subject, as well. The name or ID code together with SubDB's number will be used to identify the subject in the SubDB program and the database.

Only the subject's name or ID code, not both, should be entered. The default values will be saved for the other information (birth date, handedness, etc.) unless these values are changed by selecting the desired options from the comboboxes. These options are changed using the `Alter Combobox Choices` dialog, as previously described. Other information can be entered in the `Comment` field, if desired.

After all subject information has been entered click the `Save` button on the top of the dialog. The lower part of the dialog will grey, indicating that it is inactive, and the new subject's name and number will be added to the top combobox.

Editing a Subject

Subject information can be changed using the `Work With Subjects` dialog. Select the subject to modify from the list in the top part of the dialog and click the `Edit` button. The subject's information will be displayed and the bottom part of the dialog enabled. Make the needed changes and click the `Save` button. If the `Edit` button was clicked erroneously, click the `Cancel` button to undo any changes.

Deleting a Subject

A subject can be deleted from the database using the `Work With Subjects` dialog. Only subjects without any sessions can be deleted. If there are sessions in the database delete the sessions first, using the `Work With Sessions` dialog. Select the subject to be deleted from the combobox at the top of the screen and click the `Delete` button. A confirmation dialog will appear asking if you are sure you wish to delete the subject, and if you click `yes` the subject as well as all information about the subject will be removed from the database.

Working with Session Information

The `Work With Sessions` dialog is used to add, modify, or delete session information from the database. This dialog, shown in Figure 4, is opened by selecting `Alter Database` then `Work With Sessions` from the top SubDB menu. A *session* is an episode of data collection from a subject, so sessions are added after the subjects have been entered into the database.

Adding a Session

A session is added to the database by first selecting the subject that completed the session. When a subject is selected in the `select subject` box at the top of the `Work With Sessions` dialog the `sessions for this subject` list is filled from the database, if any are present. To add a new session to the subject click the `add new` button, which enables the bottom portion of the dialog. The relevant information about the session can then be set by typing in the boxes or selecting the desired choices from the comboboxes. As with subject information, the combobox options are changed using the `Alter Combobox Choices` dialog. When the session information has been entered click the `save` button, which will add the session to the database and list of sessions. If the `add new` button was clicked accidentally, click the `cancel` button and no changes will be made.

Once the session is in the database data files can be added. Select the newly added session from the list and click the `edit` button. The session information will be displayed, and the `add data file` button enabled. Clicking the `add data file` button brings up a file open dialog box. In this box select the data file(s) corresponding to this session and click `Select Text Files`. The selected files will be copied into a zipped archive in a subdirectory of the directory listed on the `savePlace` line of subjectDataConfig.txt. The path and name of this archive is shown on the bottom of the `Work With Sessions` dialog. Click the `save` button again to confirm the changes to the session.

Editing a Session

Session information is changed using the `Work With Sessions` dialog. First select the subject from the top combobox, then the session from the list. Next, click the `edit` button, which displays the session information on the bottom of the dialog. Change the needed information on the screen. To make a change to the data file the entire data file must be deleted (using the `delete data file` button) and recreated (using the `add data file` button). The changes are saved by clicking the `save` button, or canceled by clicking `cancel`.

Deleting a Session

Sessions can also be deleted using the `Work With Sessions` dialog. First select the subject and session to delete and click the `edit` button. Then, click the `delete data file` button, which will bring up a confirmation dialog. Click `yes` in the confirmation dialog, then click the `save` button. Now that the data file has been removed the session can be deleted. Select the session in the list again and click the `delete` button. Another confirmation dialog will appear, after which the session will be deleted from the database.

View Menu Items

The two choices under the `View` menu, `Query Database` and `Database Key`, are used to display basic subject information from the database. These screens can be used to verify that the data is being written into the database correctly. The `Database Key` screen lists the database ID code, subject name, session number, and session date for all subjects and sessions in the database. This is useful to create a key tying the subject names to their database codes. The `Query Database` screen lists all subjects in the database by experiment or subject type. This screen also indicates whether data has been written into the database by puka, which is a convenient way to verify which subjects' data have been analyzed.