**WYSIWYG Interface**

After running an experiment with WYSIWYG, the data needs to be further processed:

**Background:**

The way turns.txt creates the transcript is the following: All characters typed by the participants are sent to the server, in order to be relayed to the other client(s). The server analyzes this stream of incoming characters.

1. Suppose at the start of the experiment participant 1 types a character. This is received by the server and treated by the server as the first character of Participant 1’s turn.
2. If participant 1 types more characters, all subsequent characters are treated as being part of the same turn.
3. If another participant (e.g. participant 2) sends a character, the server records a turn-change – now participant 2 “has the floor”. All subsequent characters produced by participant 2 are treated as being part of participant 2’s turn.

The problem with this approach is that it ignores network delay and doesn’t necessarily reflect what participants themselves see on their screen. This is less of a problem with turn-by-turn interfaces, but with character-by-character interfaces it can be problematic. Consider if both participants type exactly the same text at the same time – due to inevitable network delay they will both perceive themselves typing the letters before their partners…So, to avoid this, the chattool records on the clients what the participants saw, and then sends that to the server. (N.B. If you are programming the chattool – this information is stored in clientinterfaceevents.obj” in the experiment directory).

This means that there is now no longer one single objective view of the interaction. Each client records what the participant on that client saw! (in practice, if network latency is low this shouldn’t make much of a difference). When generating the transcripts, for each client, it generates 5 files. The data from each pair is stored in four separate files which store the data associated with each keypress in a CSV format (with the “¦” character as separator):

**To generate the new files:**

In the chattool, select the menu option “utils”, then select “WYSIWYGInterface: generate transcripts”



Run it and select the folder(s) in /data/saved/experimental data/**FOLDER NAME(S)** where your data is. This should process the recorded data and create a set of new files:

**The data captured on the clients:**

Consider a hypothetical experiment between two participants

Participant 1: Participant ID: LLLL0 and Username: Michael  
Participant 2: Participant ID: RRRR0 and Username: Jenny

The following files will be generated:

**NEWwysiwyg\_cie\_LLLL0\_Michael\_s\_.txt**This file contains all the characters typed by Participant 1 (ID=LLLL0 with Username=Michael)  
The “cie” in the filename stands for “clientinterfaceevents” and the “s” in the filename stands for “self”  
The first few characters might be:  
h¦e¦¦¦¦¦¦h¦e¦l¦l¦o¦w¦a¦t¦ ¦v¦o¦o¦r¦

**NEWwysiwyg\_cie\_LLLL0\_Michael\_o\_RRRR0.txt**The file contains all the keypresses from the other participant in the pair, in this case the person with participant ID RRRR0 (The “o” in the filename stands for “other”)  
The first few characters in this file might be:  
¦¦h¦e¦l¦l¦o¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦

**NEWwysiwyg\_cie\_LLLL0\_Michael\_t\_.txt**The file contains the time gap between each character and the preceding character   
(captured on the client interface – not captured on the server!)  
(The “t” in the filename stands for “time”). You should ignore the first two values in this file.  
 The first few characters in this file are  
¦-1¦-1¦238¦11¦182¦7¦468¦447¦1851¦145¦131¦117¦183¦9586¦137¦94¦121¦152¦104¦152¦

**NEWwysiwyg\_cie\_LLLL0\_Michael\_w\_.txt**This file contains information which interface was being used.   
If the single line interface was being used, all values are “-“  
If the double line interface was being used, all values are “-“ or “\_”  
- means that the text appeared on the top row.  
\_means that the text appeared on the bottom row.  
The first few characters in this file are:  
-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-

**transcriptNEWwysiwyg\_cie\_LLLL0\_Michael\_o\_RRRR0.txt**

This is a readable transcript which is calculated from the other files in the following way:

Putting the files together (i.e. one file per row of text) gives this:

¦¦h¦e¦l¦l¦o¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦   
h¦e¦¦¦¦¦¦h¦e¦l¦l¦o¦w¦h¦y¦ ¦w¦e¦r¦e¦  
¦-1¦-1¦238¦11¦182¦7¦468¦447¦1851¦145¦131¦117¦183¦9586¦137¦94¦121¦152¦104¦152¦  
-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-¦-

If padding is added so that the timestamps can be displayed, it gives:   
 ¦ ¦h ¦e ¦l ¦l ¦o ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦   
h ¦e ¦ ¦ ¦ ¦ ¦ ¦h ¦e ¦l ¦l ¦o ¦w ¦h ¦y ¦ ¦w ¦e ¦r ¦e ¦   
- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦- ¦  
 ¦238 ¦11 ¦182 ¦7 ¦468 ¦447 ¦1851¦145 ¦131 ¦117 ¦183 ¦9586¦137 ¦94 ¦121 ¦152 ¦104 ¦152 ¦112 ¦

This is then used to generate the transcript, e.g.

Hello were  
he hellowhy  
-------------------

**N.B.** Because of the fade-out of the interface, participants often don’t use spaces if there is a longer pause (e.g. there is no space character between “hello” and “why” in the example above. This is because “hello” would have faded out before the participant typed “why”)

**Also, the chattool will generate CSVs and a transcript for the other participant Jenny**