

ONEXP 1.2

USER MANUAL

The official homepage of OnExp is: <http://onexp.textstrukturen.uni-goettingen.de>

Onexp is an experimental software that gives you the power and flexibility to create web-based experiments in a matter of minutes. But first, you need to understand, how onexp works. I will give you a simple example to start with.

1. GETTING THE IDEA!

1.1. WHAT YOU NEED FOR AN EXPERIMENT

You want a nice little experiment in which you have ten questions and for each of them, you want to give three answer options, with texts that differ from question to question, like this. So you want participants to see something like this:

Dear participant – please answer the following question:

Question 1

Choose one of these answers

Answer 1 specific to question 1

Answer 2 specific to question 1

Answer 3 specific to question 1

submit

And then you want to show Question 2, with the specific answers:

Dear participant – please answer the following question:

Question 2

Choose one of these answers

Answer 1 specific to question 2

Answer 2 specific to question 2

Answer 3 specific to question 2

submit

First thing you need, is the list of stimuli. Let us create a text file called input1.txt in which you just write up your questions and all the answers. Of course you don't write "question1" in your file, but instead your question 1 e.g. What do you think about OnExp?.

input1 . txt

```
Question1
answer1q1
answer2q1
answer3q1
Question2
answer1q2
answer2q2
answer3q2
...
```

Now, OnExp will want to know how your page should look like. There is no way, OnExp can guess that you want your answers one below the other, how many of the lines in the input1.txt should appear at once etc. While other programs will give you either a what you see is what you get editor or some funky tag-set to create your page design, OnExp relies on simple html tags. If

you have absolutely no idea about html, you can use our samples – but some of your colleagues will probably gladly help, after all someone must have written your homepage. PHP and Javascript is not allowed in this pages. So, here is an example .html page. We name it stage1.html.

stage1.html

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>
<body>
<div style="width:100%"><div align="left" style="width:600px;margin:auto">
<form action="next" method="post">
<br><br>
<p align="center"> <b> <font size="4"> Dear participant – please answer the following question: </font> </b> </p>
variablename
<center>
<table>
<tr><td>
<input type="radio" name="answer" value="7"> variablename<br>
<input type="radio" name="answer" value="6"> variablename<br>
<input type="radio" name="answer" value="5" required="required" > variablename<br>
</td></tr>
</table>
</center>
<p align="right">
<input type="submit" name="submit" value="continue">
</p>
</form></div></div></center> </body></html>
<!--
OnExp Variables:
NumberOfRepetitions=10;
variablename=input1.txt;
-->
```

You can see that at the bottom of the page, you inserted a comment that tells OnExp that you want the page to appear 10 times, and you want the variablename1 to be replaced by material from input1.txt. Now, OnExp will display stage1.html exactly ten times, and it will loop through your input1.txt. Every time it finds an occurrence of variablename in stage1.html, it will take the next element of input1.txt. This will produce the desired result.

NOTE THAT YOU ALWAYS NEED THE TURQUOISE MARKED LINES IN ORDER FOR ONEXP TO WORK. EVEN IF YOU KNOW HTML AND YOU KNOW THAT “NEXT” IS NO VALID ACTION FOR A FORM, DON’T WORRY. ONEXP WILL KNOW HOW TO HANDLE THAT.

1.2 RANDOMIZING

Of course, if you want an experiment, you probably want to have many participants and you want them to see your stimuli in a random order. So, while person1 sees question7 first, person2 could see question1 first and so on. But of course, you don’t want the specific answer to question 7 to appear below question 1. OnExp can do that for you, but you need to use two additional tags in your input1.txt. The <block> </block> tag and the <random> tag. Using <block> is simple: you just include a bunch of stimuli in a block, and OnExp will then keep those stimuli together whenever randomizing. OnExp will randomize a block if and only if the first tag that follows the <block> tag is <random>.

So, for our example we can do this:

input1 .txt

```
<block>
<random>
<block>
Question1
answer1q1
answer2q1
answer3q1
</block>
<block>
Question2
answer1q2
answer2q2
answer3q2
</block>
...
</block>
```

Now, OnExp will randomize the order of question-answer blocks, but will not change the order of questions and answers within each block. Assume for instance that you include the random tag into the first question-answer block, like this:

```
<block>
<random>
Question1
answer1q1
answer2q1
answer3q1
</block>
```

This would make OnExp shuffle the order of Questions and answers, so you could end up having the question appear where an answer should.

Of course, you might want to keep the order of the questions fixed, but you want to randomize only the order of the answers for each question. This can be achieved like this:

input1 .txt

```
<block>
Question1
<block>
<random>
answer1q1
answer2 q1
answer3 q1
</block>
Question2
<block>
<random>
answer1q2
answer2 q2
answer3 q2
</block>
...
</block>
```

And of course, you could want to randomize the order of the questions AND the order of the answers for each question. This is done as follows: (note that it may take some thinking to realize why we need nested blocks).

There is no limit for the embedding of blocks. So you can embed a block in a block in a block in a block...

input1 . txt

```
<block>
<random>
<block>
Question1
<block>
<random>
answer1q1
answer2q1
answer3q1
</block>
</block>
<block>
Question2
<block>
<random>
answer1q2
answer2q2
answer3q2
</block>
</block>
...
</block>
```

1.3 STIMULUS IDS

You don't need to use stimulus IDs in OnExp. But there are good reasons to do so and that is related to the way OnExp will save your material: OnExp will create one "answers" file for each experiment. And it will append a new line with every new answer that is chosen by any participant. Of course, you can keep the participants apart, since OnExp will start every new line with the unique participant ID. Also it will save the relative time the answer has been given with an exactness going into microseconds. But what you will not know is which stimulus/question your participant has seen. Of course, if the data is not randomized, you could reconstruct that, but it will take a long time. If you choose to use IDs, however, OnExp will save the ID of the stimulus the participant has seen when selecting the particular answer.

You can see an example here:

input1 . txt

```
<id:q1>Question1</id>
<id:q1-a1>answer1q1</id>
<id:q1-a2>answer2q1</id>
<id:q1-a3>answer3q1</id>
<id:q2>Question2</id>
<id:q2-a1>answer1q2</id>
<id:q2-a2>answer2q2</id>
<id:q2-a3>answer3q2</id>
...
```

There is one more reason you may want to use id-s. Per default, OnExp will assume that every new line is a stimulus, if you don't use IDs. In fact, if you only use the "opening" tags for ids, and you don't close them, OnExp will assume that every end of a line is a closing ID. But if you have long stimuli containing several lines, it may be useful to actually close the ids. This way OnExp will know that several lines belong to only one stimulus.

It is good practice to give your stimuli IDs according to your experimental factors and conditions, such that you can import them easily into your statistics program.

Also note that if you want to have new lines actually appear in the html file, you should use html tags for this, such as `
` or `<p></p>` etc.

DON'T EVER USE THE " " OR ' ' SIGNS IN YOUR INPUT FILES. THAT WILL MOST LIKELY CAUSE SEVERE ERRORS. IN SOME LATER VERSION OF ONEXP THIS WILL BE TAKEN CARE OF.

1.4 SEVERAL INPUT FILES

It is not particularly likely that you will ever need several variables for one stage in OnExp. But it's good to know that you CAN have that, if you need it. Assume you want an experiment, in which you want to randomly pair two sets of stimuli, say: stimulus 1-n should be randomly paired with stimulus a-z. It is easy to see that you cannot do this with only one variable. In such a case, OnExp allows you to have two distinct variables in one stage. Each of them will come with their own input file, which you can specify, and every input file will be structured as you please. The stagefile could look like this:

stage1.html

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>
<body>
<div style="width:100%"><div align="left" style="width:600px;margin:auto">
<form action="next" method="post">
<br><br>
<p align="center"> <b> <font size="4"> Dear participant – please answer the following question: </font> </b> </p>
variablename1
<br>
variablename2
<center>
<table>
<tr><td>
<input type="radio" name="answer" value="7"> yes<br>
<input type="radio" name="answer" value="6"> no <br>
<input type="radio" name="answer" value="5" required="required" > I don't know <br>
</td></tr>
</table>
</center>
<p align="right">
<input type="submit" name="submit" value="Weiter">
</p>
</form></div></div></center> </body></html>
<!--
OnExp Variables:
NumberOfRepetitions=10;
variablename1=input1.txt;
variablename2=input2.txt;
-->
```

In this case, variablename1 will be seeded from input1.txt and variablename2 will be seeded from input2.txt

1.5 THE _NOADVANCE FEATURE.

Assume, you have an input file and a stage-file. For some reason you want a stimulus to appear twice on the same page. There are two ways to achieve this: either you do some hard thinking and organize your input txt such a way that you every stimulus appears twice and randomization effects are successfully cancelled. Or you use the _noadvance feature. If you write "variablename_noadvance" into your stage-file, OnExp will not take the NEXT element of your input file, but the same it used before.

stage1.html

```
<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head>
<body>
<div style="width:100%"><div align="left" style="width:600px;margin:auto">
<form action="next" method="post">
<br><br>
<p align="center"> <b> <font size="4"> Dear participant – please answer the following question: </font> </b> </p>
variablename1
<br/>here it appears again <br/>
variablename1_noadvance
<center>
<table>
<tr><td>
<input type="radio" name="answer" value="7"> yes<br>
<input type="radio" name="answer" value="6"> no <br>
<input type="radio" name="answer" value="5" required="required" > I don't know <br>
</td></tr>
</table>
</center>
<p align="right">
<input type="submit" name="submit" value="Weiter">
</p>
</form></div></div></center> </body></html>
<!--
OnExp Variables:
NumberOfRepetitions=10;
variablename1=input1.txt;
-->
```

In this example, the stimulus from you input1.txt will appear twice on the page.

NOTE, HOWEVER, THAT IF VARIABLENAME_NOADVANCE APPEARS BEFORE OR WITHOUT VARIABLENAME YOU GET TOTALLY UNEXPECTED RESULTS OR AN ERROR MESSAGE.

YOU CAN USE THE _NOADVANCE FEATURE FOR INSTANCE TO SAVE THE TEXT OF A STIMULUS IN THE RESULTS FILE AS IN:

```
<FORM ACTION="NEXT" METHOD="POST">
  <P>
    VARIABLENAME
  </P>
  <INPUT TYPE="HIDDEN" NAME="TEXT" VALUE="VARIABLENAME_NOADVANCE"/>
  <P> IHRE WEITERFÜHRUNG: <INPUT TYPE="TEXT" NAME="ANSWER" SIZE="70"
AUOFOCUS="AUTOFOCUS" REQUIRED="REQUIRED"></P>
  <BR>
  <P ALIGN="RIGHT">
    <INPUT TYPE="SUBMIT" NAME="SUBMIT" VALUE="WEITER">
  </P>
</FORM>
```

1.6 RESOURCE FILES

You can use resource files, such as audio, video or picture files. Basically the way to use them is very simple: you include the filenames (with respective path or url) in your input file, and loop through the filenames with variablename. So for instance you can display a sound with

variablename in the stage-file

and the input file could contain something like:

<audio src="http://myhomepage.com/mystimulus1.mp3".. >

Note that you can:

- a) either upload your resource files onto the onexp server. In this case, you can use relative path to display them. So you could simply have "resources/mystimulus1.mp3" in your input file
- b) or you can store them on some independent website, say, your own homepage, and let them be displayed from the external location using full url-s in the input file.

I STRONGLY RECOMMEND THE LATTER ALTERNATIVE: PROBABLY YOU WILL BE USING AN ONEXP SERVER THAT SOMEONE ELSE HAS SET UP AND HAS ALLOWED YOU TO USE – WHY UPLOAD MB AFTER MB OF RESOURCE FILES TO HIS SERVER? AFTER ALL, HE MAY HAVE LIMITED BANDWIDTH THAT YOU ARE UNNECESSARILY EATING UP THIS WAY.

Of course, you are free to choose your own way of displaying videos, images and audios. You can use flash or whatever method you want. OnExp will only replace the variable name with the resource file name you specify, and will not interfere any further.

1.6 PROGRESS BAR

Maybe you have one hundred or more stimuli, you want your subject to see. But you want him to know, how much is still to come, such that they don't quit on the penultimate slide, being annoyed that this experiment will never end. For this OnExp provides two progress bars:

- The first one is displayed on the title bar of the browser. It looks like "YOUR-TITLE - experiment running. Progress i/n".
You can decide for each stage file, whether you want such a progress bar to be shown or not. If you want it, please include <title></title> tags into the header of your html file. If you don't want it, just delete the <title></title> tags. It is as simple as that.
Note that some browsers do not support this feature.
- The second one is a real html5 progress bar, also only supported by some browsers. To achieve this you simply include the <progress> tag inside your stage file at the location you wish. You can also have several progress bars.

1.8 SEVERAL STAGES

In an experiment you may need to have many stages: stage 1 gives some general information, stage 2 collects personal data, stage 3 is for practice, stage 4 is for warm-up, stage 5 is for the actual experiment and stage 6 asks for comments and observations. This can easily be done in OnExp. All you need is to create several stage html files. You will name them stage1.html ... stage2.html etc. OnExp will show every stage as often as specified in the "NumberOfRepetitions" variable one after the other. So, first it will completely finish stage1 and only then proceed to stage2.

NOTE:

Every stage has to specify the OnExp variables. AT LEAST the Number of Repetitions. If you don't have variables to replace, you don't need an input file either.

Every stage has to advance via a "form". If you don't have a form containing the "action="next"", tag, the next stage and even the next repetition will not be linked correctly. So I advise to create the "instructions" page independently of OnExp or at least include a "proceed" button with "action ="next" on it.

1.7 METADATA

Assume you need 20 participants for your experiment. It may take a day or a couple of weeks to get that number of participants. How do you know when you reached it? Simple: OnExp will send you a mail. And even more: if you do this experiment in cooperation with some other people. OnExp will send an E-Mail to everyone involved.

What you need to do is this: create a file called "metadata.txt". You include the number of needed participants into this file and all the involved e-mail addresses. As soon as the number of participants is reached, OnExp will close down the experiment and send out a notification to all e-mail addresses. The metadata file could look like this:

metadata.txt

```
<!--  
OnExp Variables:  
NumberOfParticipants=100;  
NumberOfMails=3;  
first@domain.com;  
second@otherdomain.com;  
third@something.something;  
-->
```

NOTE THAT THE METADATA.TXT IS OBLIGATORY.

1.8 LATIN SQUARE AND THE LIKE

Now, assume that for some reason you have one big experiment in which you have to split your stimuli across participants. In other words, you want some group of participants to see one set of stimuli, and another group of participants to see another group of stimuli. With other software, you could simply create two independent experiments, and send one link to one group of people, and another link to the other group of people. OnExp fortunately helps here.

You simply place your stage files, your stimuli files and your metadata files for each sub-experiment into a directory. And you zip all of these directories together. When you upload the zip.file, OnExp will recognize that it has to distribute people across these subexperiments and will simply do it. You only have to distribute ONE link to your participants.

This is the correct way to zip your experiment: (the blue ones are directories!)

experimentname.zip

```
experiment1  
    metadata.txt  
    stage1.html  
    stage2.html  
    stage3.html  
    input1.txt
```

```
input2.txt
input3.txt
experiment2
  metadata.txt
  stage1.html
  stage2.html
  stage3.html
input1.txt
input2.txt
input3.txt
```

These are wrong ways to zip your experiment:

Here you have embedded your experiment-data too much!

experimentname.zip

```
experimentname
  experiment1
    metadata.txt
    stage1.html
    stage2.html
    stage3.html
    input1.txt
    input2.txt
    input3.txt
  experiment2
    metadata.txt
    stage1.html
    stage2.html
    stage3.html
    input1.txt
    input2.txt
    input3.txt
```

Here you have not embedded the files enough.

experimentname.zip

```
metadata.txt
stage1.html
stage2.html
stage3.html
input1.txt
input2.txt
input3.txt
```

NOTE THAT YOU WILL HAVE TO EMBED YOUR STIMULI AND STAGE FILES UNDER A DIRECTORY EVEN IF YOU HAVE ONLY ONE EXPERIMENT. ONEXP ALWAYS ASSUMES MULTIPLE EXPERIMENTS AND TREATS ONE SINGLE EXPERIMENT JUST AS A SPECIAL CASE.

2. TEN GOOD RULES FOR ONEXP

1. Test your stage files. They should be displayed properly in your browser on your local machine. If they are not, OnExp will not fix them.

2. Test your stage-files independently manually replacing the varnames with exactly what appears between the id-s in your stimuli file. This way you can check whether what you get as a result will look and function the way you want.
3. Never omit “;” or closing tags.
4. Never use uppercase filenames (windows and linux treat those differently, and it will most likely cause problems).
5. Don’t use special characters in your stimuli-files. Use html codes instead.
6. Don’t use “ and ‘ in stimuli files.
7. Check to set the number of repetitions for each stage correctly. It maybe lower than the number of available stimuli but not higher.
8. If your randomization fails and OnExp displays the wrong stimuli, you probably got the blocks wrong.
9. You can test OnExp on your local machine using “WAMPserver”.
10. You should always test your experiment online before sending out the link.

3. USING THE ONEXP SERVER

3.1 UPLOADING AN EXPERIMENT

Go to <http://onexp.textstrukturen.uni-goettingen.de> and click to “Register”. Fill out the Form and wait for activation by administrator. After activation you have to go to the upload page. Enter your user name and your password. Select the name of the zip file containing directories with the subexperiment: each directory must contain stage-files and input files. A metadata file is optional.

Submit.

If you get any errors try to fix the file where the error occurred. If you think the File should be OK send a copy to crcedv@zentr.uni-goettingen.de with the error message that occurs. If you think you found a bug please report it at <http://bugs.onexp-dev.textstrukturen.uni-goettingen.de>.

If you get no errors save the link that you get. This link is **essential**, you should

- a) Either include this link into your instruction page.
- b) If your instruction page is already a stage in the experiment, you will have to send out the whole link to your participants.

a) is the better and safer option for most cases.

Note that after testing the experiment, you should upload exactly the same zip again, if you are satisfied. This way, you make sure, you don’t see your test-answers in the results files in the end.

3.2 VIEWING RESULTS

Go the results page. Log in, and you will automatically see a number of csv files (if results already exist). Click on them and save them wherever you please.

4. SETTING UP YOUR OWN ONEXP SERVER

It is possible to create an own onexp server, but an installer is not yet provided. Please contact us for the download links and support.

DO NOT CREATE AN OWN SERVER IF YOU DON'T UNDERSTAND WHAT THE SCRIPTS DO. PLEASE LET YOUR WEBMASTER DO THAT FOR YOU. PUTTING UP AN ONEXP SERVER WITHOUT PROPER PHP AND SERVER-ADMIN KNOWLEDGE COULD RESULT IN CATASTROPHIC SECURITY ISSUES!

<A step to step howto is under construction>

5. HOW-TOS?

5.1 HOW TO DISPLAY AUDIO STIMULI:

```
<audio src="varname" controls autoplay>
  Your browser does not support the audio element.
</audio>
```

5.2 HOW TO DISPLAY RADIO BUTTONS

```
<input type="radio" name="answer" value="7"> yes<br>
<input type="radio" name="answer" value="6"> no <br>
<input type="radio" name="answer" value="5" required="required" > I don't know <br>
```

Note that you need to have the SAME “name” for each button. Only the value will be saved.

5.3 HOW TO INCLUDE THE INSTRUCTION FILE AS A STAGE:

The trick is to have a simple form and no variables:

Your instructions

```
<form action="next" method="post">
<input type="submit" name="submit" value="start">
</form>
<!--
OnExp Variables:
NumberOfRepetitions=1;
-->
```

5.4 HOW TO REPORT BUGS, REQUESTS AND TO HELP DEVELOPING.

If you want to help us developing this software you can help us with bug reports, wish lists or your developing skills.

We created a developing server under <http://onexp-dev.textstrukturen.uni-goettingen.de> here you can find the actual revision of the software. Here it is possible that the software is **not** stable.

For documentation we use <https://trac.onexp-dev.textstrukturen.uni-goettingen.de>. The revisions are stored in subversion <https://svn.onexp-dev.textstrukturen.uni-goettingen.de>. If you need an account for these please send an email to crcedv@zentr.uni-goettingen.de.

Yes, you are allowed to change and redistribute OnExp, but always include a link to the official homepage.

If you are a developer and you created some more features, please let me know. We might include it in the official distribution.

6. OTHER NOTES

1. Be aware of potential network lags if you want to do experiments with OnExp in which time plays a crucial role. This is especially problematic, if you have large picture files or audio stimuli, as their load time may vary from repetition to repetition. If you have simple text stimuli, the lag may not be relevant after all.

2. If anyone wants to do some serious testing on the useage of OnExp for timed experiments, I would like to know the results.

Copyright © Edgar Onea, 2011.

Thanks to Alexander Syring for lots of help in the development of OnExp.