

5300
1999
no.99-021



University
of Michigan
Business
School

Faculty Research

WORKING PAPER SERIES

e-Experiment (v2.0) Documentation:
Software for Creating True Experiments on the Web

Eric DeRosia
University of Michigan Business School

Working Paper 99-021

KRESGE
BUSINESS
ADMINISTRATION
LIBRARY

e-Experiment (v2.0) Documentation:
Software for Creating True Experiments on the Web

Eric DeRosia

Marketing Department,
University of Michigan Business School
Ann Arbor, MI 48105

JUN 27 2000

The e-Experiment software is
Copyright © 1999 by Eric DeRosia. All commercial rights reserved.

Terms of Software Use

Disclaimer

The author makes and you receive no warranties or conditions, express, implied, statutory or in any communication with you, and specifically disclaims any implied warranty of merchantability or fitness for a particular purpose. The author does not warrant that the operation of the program will be uninterrupted or error free.

Limitation of Liability

In no event will the author be liable for any damages, including loss of data, lost profits, cost of cover or other special incidental, consequential or indirect damages arising from the use of the program or accompanying documentation, however caused and under any theory of liability. This limitation will apply even if the author has been advised of the possibility of such damage. You acknowledge that the download of the software package reflects this allocation of risk.

Contents

Chapter I)	Introduction	4
Chapter II)	Installation Instructions	8
Chapter III)	Creating a Project	13
Chapter IV)	Other Tasks	21
	- Adding (removing) a question to (from) an existing project	
	- Re-using a question from a previous project	
	- Renaming a question in an existing project	
	- Changing order questions are presented to subjects in an existing project	
	- Changing order questions are reported in data	
Chapter V)	Questions Files	24
	- Complete Questions	
	- Generic Response Formats	
Chapter VI)	Data Handling	35
Chapter VII)	Advanced e-Experiment	38
	- Advanced Conditions	
	- Advanced Questions	
	- Graphics	
	- Creating new question types	
Index	41

Chapter 1

Introduction

Description

This software is a tool for performing web-based experiments. It is a template that is copied by the researcher and modified to fit his or her new project. The template consists of 1) main files used in all projects, 2) previously created question files which ask common questions (e.g., demos), and 3) generic question files that are easily customized. The researcher modifies the files to create the desired web experiment. The procedures for modifying the files are fairly quick and easy. Most of the effort is typing the text for the questions into the files; even this step is quick if the text is pasted from a word-processing program.

Most importantly, no programming skills are required for using this software tool. If you're comfortable using a computer manual, you'll do fine. The files are written in HTML and JavaScript, and researchers with the ability and motivation to modify the files may certainly do so, but such programming is not required. In fact, the simple example available on the e-Experiment web site (www.umich.edu/~ederosia/e-exp) was created in less than one hour without the use of any special programming skills.

It should be emphasized, however, that e-Experiment was created to perform a number of functions not provided by current software. As a result, the program is less quick and easy to use than programs not having these features. The FAQ in the e-Experiment web site (www.umich.edu/~ederosia/e-exp) lists some programs which perform simple data collection and which require less researcher effort than e-Experiment.

In brief, e-Experiment allows a researcher to provide subjects with a single web address (i.e., URL) for participating in web-based data collection. Participants are randomly assigned to conditions defined by the researcher. The conditions may include text or graphics which differ based on the assigned condition.

One way of getting a sense of e-Experiment is by first imagining a paper-and-pencil questionnaire with only one stimulus element or question per page. Then imagine the questionnaire has a number of different versions, one of which is randomly assigned to each participant. e-Experiment is basically the web-based analog of such a questionnaire. Other features have been added, such as exposing subjects to a stimulus for a fixed amount of time, randomizing the order of questions and collecting response times. Each subject's responses (along with the response times) are reported to the researcher via e-mail. The responses can then be combined and imported into a statistics package.

It should be noted that skip patterns (i.e., determining which questions are asked based on the subject's previous responses) require advanced programming and are therefore not supported in this version of e-Experiment.

The following web site explains the features in somewhat more detail:

www.umich.edu/~ederosia/e-exp/features

Permission for Use

This software is free, but only for academic use. If your research is non-academic, please contact me for further details.

If you would like to use the software for academic research, I ask only two things in return:

1. Drop me a line

I would like to know who is using the software and also be able to notify people of bugs (not that there will necessarily be any) and future version releases (not that there will necessarily be any). So, if you want to use the software, please send me a quick e-mail. And, feedback is also appreciated.

2. Cite the software

If (when) you publish your results, please cite the software. Nothing elaborate – just something like: "The data were collected via the World-Wide-Web (DeRosia, 1999)."

The web site (www.umich.edu/~ederosia/e-exp) has the citation that is currently appropriate. This documentation is probably the thing to cite, but there may be something else (e.g., a paper). Please check the web site to see.

Ideally, I'd love it if you notify me of any citations.

Contact Me

I hope this documentation will answer most questions. Also the FAQ section of the web site (www.umich.edu/~ederosia/e-exp) may be helpful. If these sources are inadequate and you have questions, please contact me.

Also, very importantly, if you find this software does not perform a function you need for your research, let me know. There are a few things I did not include in the documentation, mostly because they are difficult to explain or require more advanced programming. Contact me to see if a feature you need is actually available but simply not described here.

My current contact information is as follows:

Eric DeRosia
Ph.D. student in marketing
University of Michigan Business School
ederosia@umich.edu

Like most people, I'm overly busy. So, I am not usually available for hire to do customized programming. If you are particularly motivated to hire me for small tasks, however, let me know.

This documentation...

Ideally, this documentation will be ...

... informal and non-technical (as much as possible). If you're a veteran of technical computer manuals, you may find this documentation a bit unusual. The goal is to make e-Experiment available to people of many skill levels.

... clear. If any issues are confusing, please let me know.

... brief. I suspect I failed on this account, but I wanted to provide as much info about the program's capabilities as possible.

This documentation assumes you (the researcher) ...

1) ... are not a programmer.

- if you are a programmer, there are a few notes scattered throughout the documentation for you.
- But, programming experience not required.

2) ... are familiar with files and directory (a.k.a. folder) structures.

3) ... have an account for hosting a web page.

4) ... understand how to use your account.

5) ... understand how to transfer files from your computer to the server (e.g., with FTP).

- You will need to transfer files in complex directory structures, so I recommend using an FTP package which transfers the directory structures automatically. Also, the FTP program should handle long filenames (i.e., longer than eight characters). I use WS_FTP LE because it fits these criteria. It can be downloaded for free by academic users at www.ipswitch.com.

If assumptions #3 and #4 above are incorrect, they can be made to be true fairly easily. Unfortunately, the information you need varies by institution, so it is difficult to give instructions here. If you are at the University of Michigan, the instructions for using ITD space are found at the following web sites:

<http://www.itd.umich.edu> (ITD's main site; a good launching point for many questions)

<http://www.umich.edu/~websvcs/umweb/how-to-homepage.html> (A tutorial on creating your own web page through your ITD account.)

<http://www.itd.umich.edu/bin/otfdoc?ID=S4283> (Directions on using 'Samba', which allows you to treat IFS space just like a local drive in Windows.)

<http://www.itd.umich.edu/bin/itddoc/otfdoc?ID=S4148> and also

<http://www.itd.umich.edu/bin/itddoc/otfdoc?ID=S4149> (Documentation of some commonly used UNIX commands; these will come in handy for working in your IFS space after you login to your account.)

If you are not at U-M, equivalent instructions can probably be found by contacting your Information Technology or Research Support representative.

Necessary programming skills

This software is billed as for non-programmers, and this is basically true. You will see some programming code but, for the most part, you can ignore it. In a few spots, you will need to make some minor edits to programmer-type code. These edits are relatively simple and are explained in detail in this documentation.

That said, you will be able to do much more with e-Experiment if you know how to program a simple web page with HTML. Fortunately, it's incredibly easy to learn and there are a number of tutorials on the web (e.g., <http://htmlprimer.com> ; with special emphasis on lessons 4 & 5).

Notation conventions for this documentation

IMPORTANT = Important notes that shouldn't be missed.

PROGRAMMERS = A note for programmers & advanced users. If don't get it, don't worry because it's not crucial.

STEP = A step in a process which must be taken. This documentation describes a number of procedures as lists of basic steps, with commentary strewn about in between.

Chapter II

Installation Instructions

STEP 1 -- Installation Instructions

Operating System requirements

- When using e-Experiment to setup an online experiment, it would be best if you use an operating system which supports long filenames, such as Win 95, 98 or NT.
- If you Windows, I recommend you do not 'hide extensions for known filetypes' because this can cause confusion when using e-Experiment.
- This is changed in Explorer under View / Options. Uncheck 'hide file extensions for known filetypes'.
- When using this software, you will be creating a number of files and directories. The names of all these directories and filenames should be kept lowercase.
- Note: WinNT (and I believe 9x also) has a habit of changing the first char to UpperCase, which can cause trouble. Double-check to ensure your filenames stay in lower case.

STEP 1a Create "e-exp" directory

- Create a directory called "e-exp" and put it in a location where you feel comfortable editing files. Perhaps ...
 - ... on your local hard drive or a network drive. If you do this, you will FTP a copy of these file to the web server when it is time to put them onto the web.
 - ... on the server. You would then edit the files directly on the server.

This documentation assumes you will do the editing on your local hard drive or network drive, mostly since doing it on the server requires you to use a UNIX editor. If you know how to use a UNIX editor, you're the type of person who can work around the assumptions.

STEP 1b Unzip the file

- The installation file (`e-exp.zip`) is available for downloading from the e-Experiment web site at <http://www.umich.edu/~ederosia/e-exp>.
- Download “`e-exp.zip`” and move it to the directory you made in the previous step.
- Unzip the file, maintaining the directory structure in the original file (e.g., “use folder names” with WinZip). You will need some type of ‘unzipping’ program, such as WinZip (which can be downloaded for free at <http://www.winzip.com>).
- After unzipping, you may delete “`e-exp.zip`”

Comments on directory structure

The e-experiment directory contains two main subdirectories. There will be much more detail in later sections of this documentation, but here is a brief overview of the e-Experiment subdirectories:

- `factory_template`
 - These are the default or 'factory preset' files. There are four necessary files for an e-Experiment project (the three files with names starting with the letter “m” and “`index.html`”). There are also four temporary question files in this directory.
 - In a procedure described below, you will modify this directory to make your own 'project_template'
 - After creating your own 'project_template', you will use it as the basis for each of your new projects.
 - In this documentation, a project refers to one specific online experiment you create.
- `question_files`
 - You will create new questions for your projects from these files.
 - Basically, these files are question templates. As such, they will replace the four temporary question files in the 'project_template'.

IMPORTANT: Editing Instructions

- Many of the steps will involve “editing” a file.
- When editing the files, use a text editor.
 - Do not use a word processor (e.g., Word) to edit the files.
 - You could use Windows' notepad (if you can stand it). I use UltraEdit (see www.ultraedit.com), and I think it's the best editor around.
- **IMPORTANT:** When editing the files, edit only the left-justified lines.
 - The programming stuff is indented with two (or more) tabs to help differentiate it from the things which you should edit. The things which need to be edited in the files are left-justified (i.e., against the left margin).
 - I strongly recommend you turn off the word-wrap feature in your text editor, allowing the text to go as 'far' as necessary to the right without wrapping to the next line. By turning off word-wrap, the lines which are left-justified in the file (i.e., the ones you will edit) will become obvious.

- PROGRAMMERS: If you have the motivation and ability, go ahead and edit the indented programming stuff.
- Sometimes the line number is referred to in this documentation to help you locate the line to edit. The line numbers are labeled “approximate” because after you edit the file, the number of the line in question may change. However, it should always be obvious which lines to edit.

STEP 2 Setup for your own use

STEP 2a Copy the directory

- Within the main “e-exp” directory, copy the entire subdirectory “factory_template” as “project_template”.
- This step outlines how to modify the “project_template” so it works for you. Then it will serve as the template for your future projects.
- Leave “factory_template” unmodified (in case you want to re-start Step 2).

STEP 2b Update e-mail address

- e-Experiment displays your e-mail address to subjects so they know whom to contact with questions. Your e-mail address is displayed on the screen and a ‘mailto’ allows participants to click on your e-mail and start their browser’s e-mail software.
- To set this up, perform the tasks below for two files:
 - project_template/index.html (about line 77)
 - project_template/mthanks.htm (about line 20)
- The line to be edited is reproduced below:


```
<A HREF="mailto:researcher@univ.edu">researcher@univ.edu</A></center>
```
- Simply replace `researcher@univ.edu` with your e-mail address. Note that it should be replaced two times in this line.

STEP 2c Update CGI stuff

- The following edits must be made to the file “project_template/mfinish.htm”:
- In line 137, replace “absolute_location” with the web address of the project_template. The web address depends on your particular setup with your account. Typically, the URL has a boiler-plate part which you must always use (that part for me is “www.umich.edu/~ederosia”). After that, you should add “/project_template”. Thus, for me, I would replace “absolute_location” with “www.umich.edu/~ederosia/project_template”.
- In line 140, replace “researcher@univ.edu” with your e-mail address. This is the e-mail address to which the data will be sent.

STEP 2d Test

- It is time to see if your project template works correctly.
- First, transfer the “project_template” directory to the server, leaving the directory structure intact.
- Visit your new site on the web with your browser . The URL will be the same one you entered in Step 2c above (e.g., www.umich.edu/~ederosia/project_template). By visiting this URL, the file “index.html” file will automatically be referenced and the template questionnaire will begin.
- You will need to use Netscape 3+ or Internet Explorer 4+ to view the project template. If you made any mistakes, an error will tell you where (i.e., the filename and line number) the problem can be found. For testing, I recommend Netscape 3 because its error messages are the clearest to understand. If you use Netscape 4, the errors can be seen by typing “javascript:” as a new web address.
- At the opening screen, you should see your e-mail address. Clicking on it should open a 'mailto' screen (if your browser is equipped to do so).
 - If not, carefully do step 2b again.
- After starting, you should see four 'temporary' questions.
 - Write down your answers to these questions for checking the resulting data.
 - **IMPORTANT:** The question order is randomized in this template, so you will need to pay attention to the question number. More will be said about this in Chapter III.
- After answering, you should see a 'send data' screen.
- After sending the data, you should see a 'thank you' screen with your e-mail address; clicking on it should open a 'mailto' screen (if your browser is equipped to do so).
 - If you see the thank-you screen but your e-mail is not correct, carefully do step 2b again.
 - If you do not see the thank-you screen or if you see an error, carefully do steps 2c and 2d again.
- After a few minutes, you should receive the data via e-mail.
 - The time required depends on a lot of things, such as your server. In general, though, the e-mail should arrive within a few minutes.
- **IMPORTANT:** When you receive the data e-mail, you will see that the data seems to be in an ugly format, but format is more useful than it seems. See Chapter VI.
- In the data portion of the e-mail, it should say “temp01##” and immediately after that the data you entered (e.g., “1”) for the first question.
 - For now, ignore the other numbers & stuff.
 - Also compare the data for temp02 - temp04 with how you answered.

- If you received the data correctly, your template works. Go to Step 2f.
- If you do not receive an e-mail containing the data, carefully do steps 2d and 2e again.

STEP 2e Backup

- Your project template works!
- Create a backup of the whole `e-exp` directory.
- You may delete 'factory_template' if you like.
- You're ready to use this project template as the basis for a new project (as described in the next chapter).

Chapter III

Creating a Project

Start here with every new project you want to do.

STEP 3 Plan the questionnaire

STEP 3a Plan the questions

- Plan the questions you will ask and the stimuli you will present, including the order you want them presented.
- Also plan the information screens (i.e., screens with only information and a “continue” button) you want to present to subjects. These are often used for instructions.
- The first time you go through this process, I recommend you do a small ‘test’ project rather than a full-sized questionnaire.

STEP 3b Plan the conditions

- Plan and define the Conditions for the project. These will probably be based on design of the experiment. For example, for a 2x2x2 experiment, you will have eight Conditions. Chapter VII has information about other ways to use Conditions within e-Experiment
- Assign a number to each Condition (i.e., 1, 2, 3, etc.).

STEP 4 Setup the project

STEP 4a Copy the directory

- Copy the directory "project_template" (including its subdirectory structure) to a new directory called "new_project".
- You may replace "new_project" with your project name in all the steps below.
- **IMPORTANT:** This step only modifies files in the “new_project” directory; be sure not to edit the “project_template” in this step.

STEP 4b Absolute references

- Edit the file “new_project/mfinish.htm”.
- In line 137, replace “project_template” with “new_project”. This line reflects the absolute location of your web experiment. This is similar to Step 2c above.
- After you become familiar with these procedures and can create a new project quickly, be sure to remember to take this step and update the absolute reference. Let’s just say I happen to know it’s an easy thing to forget.

STEP 5 Modify the project created in Step 4

STEP 5a Create one page for each question

- **STEP 5a.1** Copy the appropriate file from `question_files`.
 - These are the question templates.
 - See Chapter V for a description of each question file.
 - See Chapter V for details on modifying question files.
 - See Chapter VII for advanced techniques for modifying question files.
 - Questions can also be copied directly from previous e-Experiment projects (see Chapter IV).
 - Copy the question file you choose into the "new_project" directory (i.e., the same directory as `index.html`).
 - Rename the copied file. The new filename is **important**:

IMPORTANT: Save the files with a ".htm" extension in the filename. It must not be ".html" but rather ".htm".

The filename must be in all lower-case letters.

The filename you choose (sans the ".htm") is the question name. For example, if the filename is "q01.htm", the question name is "q01". The question name is used in a number of ways by e-Experiment, including reporting it as an identifier in the data.

There are some restrictions on the filename. The operating system has the usual restrictions (e.g., the character "*" cannot be used). Also, do not include spaces, single quotes or double quotes.

IMPORTANT: Do not name the question a word or something a subject would reasonably use in an open-ended response. This will cause problems in the data reporting stage. For example, "like" would be a poor choice for a question name. However, "like001" is probably just fine, since subjects are not likely to write "like001" in their open-ended responses.

IMPORTANT: There another restriction on the question names based on the way the computer searches through the list of question names. The question name cannot be a subset of other question names used in the project. In other words, if the names of question A and question B are such that question A's name is the same as question B's name except that question B's name has additional letters at the end, there will be a problem. If, for example, one question were named "q1" and another were named "q11", the computer's search through the list for "q1" would succeed when it hits "q11". That's a bad thing.

All these restrictions may be confusing. Fortunately, a system of letter identifiers followed by sequential numbers fits all the restrictions and works quite well. For example, you could mix the following question names in a project with no problem:

q001, q002, q003, scale001, scale002, scale003, demo001, demo002

- **STEP 5a.2** Edit the question text.
 - If necessary, edit the question file to customize the question text.
 - Recall that text for editing is left-justified, while programming stuff is indented two tabs.
 - The “Adding question text to generic question files” section of Chapter V (p. 29) explains this step in detail.
 - After editing the question text, the question file is complete.
 - Repeat Step5a for each planned question.
 - In the end, you will have one file for each question or screen you want to display.

- **STEP 5b** Edit `mstart2.htm`
 - Recall that text for editing is left-justified, while programming stuff is indented two tabs (which is particularly important in this file).
 - Admittedly, this step seems ugly. This is probably the most complex part of setting up a new project.

This step is much easier to do if you are sure about the project's design you created in Step 3.

- **IMPORTANT:** The following steps are all edits to the following file:
 - `new_project/mstart2.htm`
- **STEP 5b.1** Edit the line which says:
 - `NumConditions=2;` (about line 67)
 - Replace the 2 with the number of conditions you have planned.
 - You must have at least 1 condition.
 - There is no upper bound on number of conditions.
 - Each condition you define in this step and in the steps to come will have an equal chance of being assigned to a participant.
- **IMPORTANT:** Definition of “code section”
 - We need to define something here.
 - A "code section" is defined here as four lines ...
 - starting with a line that begins `if (Condition=="`
 - and ending with a line that is simply `}"`
 - In the rest of Step 5, a "code section" refers to this four-line block.
 - All four lines -- including the character `}"` on its own line -- must be included when we consider a "code section".
 - We will refer to a code section by it's Condition number, which is in column 16 of the first line in the code section.

For example, the code section which starts with ...

```
if (Condition==2) [blah blah]
```

... is defined here to be the code section for Condition 2.

- **STEP 5b.2** Delete the code section for Condition 2.
- **IMPORTANT:** Do not remove the code section for Condition 1; remove only Condition 2.
- **STEP 5b.3** Within the code section for Condition 1, edit the line which says:


```
CommentCookie= [blah blah] Study=template##Condition=1 [blah blah]
```
- Replace 'template' with the name of your new project.
- Name the project as you want it reported to you in the e-mails you receive; the questionnaire name is not displayed to the subject.
- Do not use punctuation (other than periods) in the name. Spaces are okay.
- **STEP 5b.4** Copy the code section for Condition 1 and paste it a number of times so that you have one code section for each planned condition.
- **STEP 5b.5** Modify the pasted code sections.
- For each of the pasted code sections, change the condition number from 1 to the new number.

Change once at almost the beginning of the code section.

- After 'Condition==' (i.e., line 1 of the code section, column 16)
- Note: there are supposed to be two equal signs here

Change again a little later in the code section.

- After 'Condition=' (i.e., line 2 of the code section, column 58)
- Note: only one equal sign here

- **STEP 5b.6** Verify code sections for Conditions.
- Review the lines created in the two previous steps:
 - Each planned condition number should have one and only one code section.
 - Each line should have the same number at column 16 and column 78.

An example for 3 conditions:

```

if (Condition==1){
CommentCookie=" [blah blah] Condition=1 [blah blah]
SequenceCookie=" [blah blah]
}

if (Condition==2){
CommentCookie=" [blah blah] Condition=2 [blah blah]
SequenceCookie=" [blah blah]
}

if (Condition==3){
CommentCookie=" [blah blah] Condition=3 [blah blah]
SequenceCookie=" [blah blah]
}

```

- In the steps outlined below, you will use the code sections you've created to define the conditions you planned in Step 3b.
- **STEP 5b.7 Define the CarryText for each Condition.**
- See Chapter VII for more information about using CarryText variables.
- This step is optional. If you do not plan to use any CarryText fields, simply skip this step.
- For each condition, text can be defined which can be brought into the questions.
- Four such text fields can be defined for each condition, called CarryText.
- For example,
 - Condition 1 might define...
 - CarryText1=Coca Cola
 - CarryText2=Diet Coke
 - CarryText3=2-liter
 - CarryText4=It's just as unique and refreshing as you are.
 - Condition 2 might define these four variables differently.
- When a question designed to use these CarryText fields is brought up, the CarryText for the condition will be put onto the screen (like a form letter).
- The CarryText fields are created by editing the text in the code section for the appropriate condition.
 - CarryText 1 is defined after "CarryText1=" in the second line of each code section. For example, in the template file, CarryText 1 is defined as "text for carrytext1" for each Condition.
- In this step, modify the code section for each condition to reflect the CarryText fields you desire.
 - Replace "text for carrytext1" in line 2 of the code section with the text you desire for that Condition.
 - Repeat for CarryText2 through CarryText4.
- CarryText fields defined within a given condition's code section will only be defined that way if that condition is randomly assigned to the subject.
- **STEP 5b.8 Define graphic files for each Condition.**
- This advanced topic is described in Chapter VII.
- This step is optional. If you do not plan to use graphic files, simply skip this step.
- **STEP 5b.9 Define the sequence for each Condition.**
- The questions which are to be asked -- and the sequence of those questions -- are defined for each Condition in this step.
- In the code section for Condition 1, edit the line which starts ...
 - SequenceCookie=' [blah blah] (about line 72)
- Replace "temp01" and the other temporary question files with the names of the questions you created in Step 5a.
- Recall that the question name is the filename you used to save the question.
- Recall also that only lower-case letters should be used.

- For this step, the ".htm" part of the filename is not used.
For example, if you saved the question as "q01.htm", use "q01" here.
- Each question name in the sequence is separated (i.e., delimited) by two number symbols (i.e., "##").

Thus, "##" is found on either side of each question name.

- The first question must be "mstart2"
- The last question must be "mfinish"
- The 'in-between' questions are the ones you created in Step 5a. For example, if you have only two questions (called q01 and q02), your line would look like this:
`SequenceCookie="##STARTOFSEQUENCE##mstart2##q01##q02##mfinish##ENDOFSEQUENCE";`
- The Sequence is defined separately in each condition's code section.
- If you desire the sequence to be the same for all conditions, simply copy the same sequence to each Condition's code section, replacing the template's sequence with your own sequence.
- **PROGRAMMERS:** By moving the code which defines the variable SequenceCookie out of the IF statement, the sequence will be defined the same for all conditions using only the one line of code.
- Different sequences can be used for different Conditions. This allows large differences between the Conditions and is, therefore, probably the most important feature of e-Experiment. See Chapter VII for more information and a few restrictions.
- Question files which are actually information screens (e.g., rinfo.htm) are treated the same way as other questions in this sequencing step.
- Question files which have multiple questions per screen (e.g., qdemo.htm) are referred to in the sequence only by their question name (i.e., the filename).
- The sequence can often become quite long, so check closely for typing errors.

- **STEP 5b.10 Randomize the sequences.**

- The sequences defined in the previous step are fixed (i.e., the same for each subject).
- e-Experiment has the option of presenting a subset of questions in random order.

A different random order is chosen for each subject.

This feature is usually used to compensate for order effects.

- The code necessary for doing this is best presented as an example:

The line below will present q01, q02, and then randomize q03 and q04:

```
SequenceCookie="##STARTOFSEQUENCE##mstart2
##q01##q02##"+randomize("q03##q04")+ "##mfinish
##ENDOFSEQUENCE";
```

(all on the same line in the file)

- Thus, the sequence is altered somewhat to include the extra randomize syntax.
- The ## delimiter is used between the question names within the randomization (e.g., between q03 and q04).
- **IMPORTANT:** When randomizing questions, the quote marks, plus signs and other syntax must be exactly as shown above; the correct syntax is also found in the template version of mstart2.htm

- In the current version of e-Experiment, one subset of questions cannot be randomized as a subset of another.

For example, the following cannot currently be done:

```
Randomize (randomize (q01, q02) , q03 , randomize (q04, q05) )
```

(note: abbreviated code is used above for illustration)

- By the way, it should be noted that the outcome described above could be accomplished by considering careful Sequence definitions within Conditions with the Randomize feature. Chapter VII covers this topic in more depth.

More than one randomization can be done in each sequence, as long as one is not a subset of another.

For example, the following is acceptable:

```
randomize (q01, q02) , randomize (q03, q04) , randomize (q05, q06)
```

(note: abbreviated code is used above for illustration)

- Just as each Condition has its own sequence (from Step 5b.9), each Condition can have its own randomizations.

STEP 5c Edit mfinish.htm

- The following step involves editing the following file:
new_project/mfinish.htm
- This file defines which questions will be reported in the data. This is an important contrast: mstart2.htm defines what questions will be asked, while mfinish.htm defines what questions will be reported in the data.
- The lines edited in this step are almost at the end of the file.
- Each question name should be defined in the following format:
GetDataPiece("temp01")+
... replacing the question name temp01 with the desired question name.
- **IMPORTANT:** The syntax should be copied exactly as shown above (or as in the original "mfinish.htm").
- Thus, there will be one line for each question you are reporting in the data.
For example, the original file reports the data for questions temp01, temp02, temp03 and temp04.
Note: the code in the original file created the data e-mail for the test you undertook in step 2e.
- The question names should be defined in the order they should be reported.
No matter what order the questions were asked, the data will be reported in the order defined in this step.
- You need not request all questions be reported in the data.
- For example, information screens (e.g., rinfo.htm) are treated as questions in the sequencing, but since a 1 is written as data when each subject presses "continue", reporting the data may not be useful. By omitting the name of the information screen in this step, the data will not be reported.

- **IMPORTANT:** Since inadvertently omitting a question name in this step will cause the data to not be reported, this step should be performed carefully.
- **IMPORTANT:** A few question files have multiple fields (i.e., more than one question on the screen at once). The file `qdemo.htm` is one such file. These files are referred to in this step **NOT** by their question name but rather by the name of their individual fields.
- See the documentation for the individual multiple-field question for its field names and more specific instructions.

STEP 5d Delete unnecessary files

- The following files are left over from the project template:
 - `temp01.htm`
 - `temp02.htm`
 - `temp03.htm`
 - `temp04.htm`
- These files can be deleted.

STEP 6 Test, test, test

- The project is finished! Transfer the whole project directory to the web server (maintaining the subdirectory structure) and view the online questionnaire with your browser (just as you did in step 2e).
- A number of steps are required to setup a new project – this is no surprise to you if you have worked through the steps! Since mistakes can be made anywhere, the on-line questionnaire must be thoroughly tested.
- Prior to the customary pre-testing with subjects, I suggest you make multiple trips through the questionnaire looking for errors. The on-line questionnaire can be compared to your planned questionnaire, and the data you enter can be compared to the data you receive via e-mail. Be sure to try enough times to be assigned at least once to each condition.
- One hint for effective testing is to temporarily include in a CarryText the number of the assigned Condition; then, if the first question displays this CarryText, the tester can know at the onset which condition has been assigned.

Chapter IV Other Tasks

The previous chapter has details on creating your first project.

Assuming you already have an e-Experiment project, here are some abbreviated step-by-step procedures:

Adding a question to (removing a question from) an existing project

STEP 1 Create (delete) question file

- If removing a question, deleting the file is not absolutely necessary.
- See Step 5a in Chapter III for filename restrictions and further instructions.

STEP 2 Add (remove) question name in `mstart2.htm`

- Add (remove) question name to (from) the sequence in EACH condition's code section in `mstart2.htm`.
- See Step 5b.9 in Chapter III.

STEP 3 Consider randomizations

- Consider whether randomized sequences are (or should be) affected by this change.
- See Step 5b.10 in Chapter III.

STEP 4 Add (remove) question name in `mfinish.htm`

- Add (remove) question name to (from) the data reporting code in `mfinish.htm`.
- See Step 5c in Chapter III.

STEP 5 Test the resulting questionnaire

Re-using a question from a previous project

STEP 1 Copy the question file

- Copy the question file from the donor project's directory into the recipient project's directory.
- The file may be renamed if desired.
- See Step 5a in Chapter III for filename restrictions and further instructions.

Steps 2 through 5 are the same as in the previous section (i.e., adding a question to a project).

If you often re-use one type of question, you might consider putting a copy of the file into the "my_questions" subdirectory of "question_files", which is provided for that purpose.

Renaming a question in an existing project

STEP 1 Rename the question file

- See Step 5a in Chapter III for filename restrictions and further instructions.

STEP 2 Edit "mstart2.htm"

- Rename the question references in the sequence for EACH condition's code section in the file mstart2.htm .
- See Step 5b.9 in Chapter III.

STEP 3 Edit "mfinish.htm"

- Rename the question reference in the data reporting code in mfinish.htm .
- See Step 5c in Chapter III.

STEP 4 Test the resulting questionnaire

Changing order questions are presented to subjects

STEP 1 Edit "mstart2.htm"

- Change order of question names in sequence in EACH condition's code section in mstart2.htm (see Step 5b.9 in Chapter III).

STEP 2 Consider randomizations

- Consider whether randomized sequences are (or should be) affected.
- See Step 5b.10 in Chapter III.

STEP 3 Test the resulting questionnaire

Changing order questions are reported in data

STEP 1 Edit "mfinish.htm"

- Change order of question names in data reporting portion of mfinish.htm .
- See Step 5c in Chapter III.

STEP 2 Test the resulting questionnaire

Chapter V

Question Files

The software comes with 75 question files, stored in the "question_files" directory. These question files are designed to be placed within a project (See step 5a of Chapter III). Some of the files are commonly used, ready-made questions, and other files are templates which can be easily modified for your particular purposes.

All the files in the directory "question_files" may be displayed in your browser to see what they will display to a subject. However, they will not accept data input correctly until added to a project, as discussed in Step 5a of Chapter III.

The files have been segmented in the "question_files" directory in the following way:

- complete questions
- generic response formats
- my questions

Each of these categories of question files will be explained in turn below, along with a commentary on the function of each question file.

Complete Questions

These questions are designed for special purposes and are ready to be used in any project. They can, of course, be modified; saving such modified files under a different name and leaving the original unmodified may be best.

The question descriptions refer to footnotes, which are located at the end of this section. Some of the footnotes are marked as IMPORTANT, and they are indeed so.

Each question file in this category is described below.

`qagree.htm` -- A simple example of an agree/disagree question as it could be used in a project.

See footnotes (1) and (3).

`qbench1.htm` -- Provides a benchmark of the speed of the subject's internet connection. The file downloads a file of known size, then reports the time required as the response time (see Chapter VI). This is designed to be a potential covariate in analysis of subjects' response times.

PROGRAMMERS: If multiple measures are desired, this file may be included more than once, perhaps at different places throughout the q're. To do this, a few steps must be taken: 1) The different instances of this file must have different question names, 2) The graphic file (`bench1.jpg`) must be copied and renamed within the same gif directory, and 3) the question file (line 22) must reference the renamed file instead of `bench1.jpg`. Steps 2 and 3 above prevent the cached copy of the file from being used instead of downloading the file again.

IMPORTANT: If you use this file, be sure to refer to the question name in "`mfinish.htm`" to receive the elapsed time in the data e-mail.

See footnotes (1) and (3).

`qbench2.htm` -- Provides a benchmark of the processing speed of the subject's computer. The file performs a loop (counting from 0 to 50,000) and then reports the time required as the response time (see Chapter VI). This is designed to be a potential covariate in analysis of subjects' response times.

IMPORTANT: This file may be included more than once, perhaps at different places throughout the q're; it must, however, have a different filename/question-name for each instance.

IMPORTANT: If you use this file, be sure to refer to the question name in "mfinish.htm" to receive the elapsed time in the data e-mail.

See footnotes (1) and (3).

`qcenter.htm` -- Simply asks the subject to press a box in the middle of the screen to continue. Designed to force the subject to center the cursor before another stimulus is presented in the two-option (good vs. bad) response format. Without such centering, the cursor would be 'over' the previous response, possibly affecting responses and response times.

See footnotes (1) and (4).

`qdemo.htm` -- This is a multiple-field file which has a few commonly used demographic questions. No data verification (e.g., ensuring subjects who ask for credit via "qcredit.htm" also complete the 'for student' questions) is attempted. The exception is that either a two-digit number or no response (which indicates a refusal) is required for the year of birth.

This file creates the following field names:

"gender" (1=male, 2=female)

"year" (i.e. 2-digit year of birth. Sure, it's not 'Y2K ready', but the way I figure it, there won't be trouble here until the people born in the year 2000 start answering questionnaires. The year 2010?)

"city"

"state" (2-character postal abbreviation for the state; also includes Wash. D.C. (DC), Puerto Rico (PR) & US Virgin Islands (VI))

"level" (i.e., current grade level; 1=Freshman, 2=Sophomore , 3=Junior, 4=Senior, 5=Masters, 6=Ph.D.)

"university"

"major"

See footnotes (1) and **IMPORTANT:** (6)

`qpanel.htm` -- Asks the subject to provide an e-mail address if he or she would be willing to participate in future studies.

See footnotes (1) and (3).

qpin.htm -- Asks the subject to validate their permission to participate by entering a seven digit PIN (Personal Identification Number) which they were previously assigned (e.g., during recruiting). Their response to the question can be reported in the data normally (see footnote 3), which could be used to identify the subject or simply to prevent duplicate participations (see Chapter VI).

IMPORTANT: The PIN must be valid for the subject to participate. Valid PINs have the following properties: 1) they are all numeric digits (0 to 9 inclusive), 2) the first four digits can be any combination of digits, and the last three digits are the sum of the first four digits. (i.e., the fifth digit is always zero.) For example, 1234010 is a valid PIN, but 1234011 is not.

Also, the PIN 0000000 is invalid.

The web site (www.umich.edu/~ederosia/e-exp) has an Excel file which generates valid PINs.

This is not a completely secure measure, since someone who knows JavaScript can simply view the code and trace through it to determine what checks are taking place (although, I did try to make that portion of the code difficult to interpret). Also, a valid PIN can be guessed (the probability of a random seven-digit number being a valid PIN is .01) Security can be made even tighter if you not only use a PIN but collect the subject's e-mail address and compare it to a recruitment list.

See footnotes (1) and (3).

qconsent.htm -- A simple informed consent screen, of the type required by the University of Michigan if no deception is involved in the study (i.e., what they would call a survey). The screen text must be edited to reflect the time required for the study.

See footnotes (1) and (3).

qintro1.htm -- This is a typical "welcome to the study" screen, and it does nothing special. Note: you should not use this file along with "qintro2.htm" because doing so will duplicate the messages displayed for the subject.

See footnotes (1) and (3).

qintro2.htm -- This is an alternative to "qintro1.htm". The same messages as "qintro1.htm" are displayed to the subject, but this file is designed to do much more. It sends an e-mail notifying the researcher that a study has been initialized. (Actually, the file "qloss1.htm" sends the e-mail; see below.) The e-mail has the subject line "ANNOUNCEMENT for [studyname]" and includes the Condition number to which the subject is assigned. By comparing the number of subjects who began the experiment (i.e., the number of announcement e-mails) to the number of subjects who completed the experiment (i.e., the number of data e-mails) the researcher can determine the overall dropout rate for the study and whether that rate is different across Conditions. This may be especially important, for example, if some conditions are less pleasant than others.

IMPORTANT: If you use this file, you must also use the files "qloss1.htm" and "qloss2.htm"; see below.

Also, see footnotes **IMPORTANT:** (1) and **IMPORTANT:** (5).

qloss1.htm -- This file is only used in conjunction with "qintro2.htm".

IMPORTANT: This is the file which actually sends the participation announcement to the researcher, so if it is to be used, it must be edited in the same way "mfinal.htm" was edited in Step 2c of Chapter III (except it is the line numbers 29 and 32 which are to be edited). I recommend that you not bother editing this file unless you decide to use "qintro2.htm" (described above).

See footnotes **IMPORTANT:** (2) and **IMPORTANT:** (5).

qloss2.htm -- This file is only used in conjunction with "qintro2.htm".

See footnotes **IMPORTANT:** (2) and **IMPORTANT:** (5).

qcredit.htm -- Designed for studies where you need to collect the names and other related information from subjects so they can receive some type of course credit. If the subject says he or she desires credit and enters his or her full name, full e-mail address and the instructor's name, two things will happen:

1) An e-mail (separate from the data e-mail) will be sent to the researcher containing data from the three fields. This function is designed for researchers who want to avoid attaching identifying information to the data (i.e., researchers who need human-subject review board approval). Also, the recipients receive a "cc" of the e-mail to confirm that their responses have been received and that they will receive credit for participating. The e-mail generated by this file has the subject line "PARTICIPATION CONFIRMATION" and has a message designed to help subjects understand its purpose.

2) The information will be available to researchers and can be reported in the data e-mail (using mfinish.htm) if the researcher desires. The information is available under the field names "RespName", "Respemail", and "Instructor".

IMPORTANT: If you use this file, you must also use the files "qconfirm.htm" and "qconfirm2.htm"; see below.

Also, see footnotes **IMPORTANT:** (1) and **IMPORTANT:** (6).

qconfirm.htm -- This file is only used in conjunction with "qcredit.htm".

IMPORTANT: This is the file which actually sends the credit information to the researcher, so if it is to be used, it must be edited in the same way "mfinal.htm" was edited in Step 2c of Chapter III (except it is line numbers 23, 24 and 25 which are to be edited).

I recommend that you not bother editing this file unless you decide to use "qcredit.htm" (described above).

See footnotes **IMPORTANT:** (2) and **IMPORTANT:** (5).

qconfirm2.htm -- This file is only used in conjunction with "qcredit.htm".

See footnotes **IMPORTANT: (2)** and **IMPORTANT: (5)**.

qall.htm -- This is not a question designed to be given to subjects. Rather, this file gives the researcher an example of the code necessary for a number of advanced functions. Some of the advanced functions displayed in this file will not display properly unless the file is referenced from within an e-Experiment project (i.e., not if the file is simply viewed from a browser). See Chapter VII for more information about this file and about advanced question creation.

Footnotes for file descriptions:

- (1) This file is added to a project in the normal manner, including renaming the file and referring to it in "mstart2.htm" (see Step 5a in Chapter III for more details).
- (2) This file is added to a project in a special manner: 1) copy the file to the project directory (i.e., with the rest of the question files), 2) maintain the filename (i.e., do not rename the file), and 3) do not refer to the file in "mstart2.htm". The file will be used automatically.
- (3) The subject's response to a question created from this file can be reported in the data e-mail in the normal manner, including referring to it in "mfinish.htm" (see Step 5c in Chapter III for more details).
- (4) The data created by this file is not likely to be useful, so it may be omitted from "mfinish.htm" if desired (see Step 5c in Chapter III for more details). If desired, however, the data can be reported via "mfinish.htm".
- (5) This file does not create data which can be reported in the data e-mail, so it should not be referenced in "mfinish.htm". Simply omit any reference to the file in "mfinish.htm".
- (6) This file presents multiple fields on the same screen. When reporting the resulting data in the data e-mail, the filename should not be referenced in "mfinish.htm" (as is usually the case). Instead, each field name should be referenced separately in the place of a question name. For example, instead of referring to "qdemo", refer to "gender", "year" and all the other field names created by this multiple-question file. See Step 5c in Chapter III for more details on how to refer to a file in "mfinish.htm".

Generic Response Formats

Of course, the complete questions detailed in the previous section do not begin to address the many types of questions which must be asked in a project. This section documents generic files which can serve as the basis for creating such diverse questions.

A question file itself can be considered to have two parts:

- 1) the question portion (including the question text and the screen formatting of that text)
- 2) the response portion (including the number of response options, the text defining each option, the screen formatting of the options, and the method of recording the subject's response)

The question portion of a question file is much easier for researchers to edit than the response portion. Also, researchers often use a few standard response formats. Therefore, e-Experiment includes a number of question files with generic question portions and pre-defined response portions based on standard response formats. These generic question files are described in this section. Thus, each question file in this section serves as a template for its response type. Before describing the many different response formats included in these question files, this section describes how to edit the generic question portion.

As first discussed in Step 5a in Chapter III, creating a new question requires two steps:

Step 1: Copy the appropriate question file to the project's directory and rename the file as the new question name.

Step 2: Edit the new question file to reflect the desired question text.

The section below describes Step 2 above in some detail.

Adding question text to generic question files

- Recall that the left-justified text should be edited, but all the programming stuff is indented with two tabs. The generic question portion of these files (i.e., the text to be edited) is as follows:

```
<center><h2>Topic</h2></center>
<h3>Primary question text</h3>
<h4>Secondary question text</h4>
```
- The question text is everything but the stuff between the "<" and ">" symbols. The generic code would display something similar to the following:

Topic

Primary question text

Secondary question text

- To change the text, simply replace the generic text (e.g., "Topic") with your text (e.g., "Some info about you ..."). If you don't want a given line (e.g., "Secondary question text"), simply delete the whole line from the file.
- The formatting of the question text is created by the other stuff (e.g., "<center>"). This portion of the question file is simply HTML, so if you are familiar with HTML and if you want to change the formatting of the question text, you can certainly do so. By the way, if you are

unfamiliar with HTML but want to learn how to use it; it is not difficult in the least. Try one of the many tutorials on the web (e.g., <http://htmlprimer.com> ; with special emphasis on lessons 4 & 5) **PROGRAMMERS:** Any HTML tags (e.g., tables) can be used to define the question text. For greatest browser compatibility, however, you may want to avoid using 1) specific fonts and 2) HTML which cannot be read by earlier browsers such as Netscape 3 and I.E. 4 (e.g., styles).

Closed-Ended Questions

As mentioned above, editing the response portion of the question file is more difficult than editing the question portion. Therefore, a number of response types have been created, including different numbers of options, different text defining the options and different formatting of the options. Some of the templates are segmented in the "generic_response_format" subdirectory in the following way:

1. Closed-ended with a "don't know" option

a. Horizontal format

i. 4 options

Type D

Type E

Type Generic

ii. 5 options

Type A1

Type A2

Type B

Type Generic

iii. 7 options

Type A

Type B

Type Generic

iv. 9 options

Type A

Type B

Type Generic

b. Vertical format

[same sub-categories as 'a' above]

2. Closed-ended without a "don't know" option

[same sub-categories as '1' above]

As the segment names imply, the question files in (1) include a "don't know" option, and the files in (2) do not. Both #1 and #2 are further segmented by whether the responses are presented (a) horizontally or (b) vertically. Each sub-type is further segmented by how many options are included. So, for example, a 9-option vertical-format sans-DK-option response type is represented.

Within each of these categories, a few traditional option texts are available. They are saved as /pes, and the types are outlined below. Note that the last letter of each question filename denotes the response type. For example, the file "rhy4d.htm" is type D.

Four options	Five options	Seven options	Nine options
<u>Type D</u> 1 = Never 2 = Once in a while 3 = Sometimes 4 = Often	<u>Type A1</u> -2 = disagree completely -1 = 0 = neutral +1 = +2 = agree completely	<u>Type A</u> -3 = disagree completely -2 = -1 = 0 = neutral +1 = +2 = +3 = agree completely	<u>Type A</u> -4 = disagree completely -3 = -2 = -1 = 0 = neutral +1 = +2 = +3 = +4 = agree completely
<u>Type E</u> 1 = not at all likely 2 = not very likely 3 = somewhat likely 4 = very likely	<u>Type A2</u> -2 = disagree strongly -1 = disagree somewhat 0 = neither agree nor disagree +1 = agree somewhat +2 = agree strongly	<u>Type B *</u> -3 = anchor 1 -2 = -1 = 0 = +1 = +2 = +3 = anchor 2	<u>Type B *</u> -4 = anchor 1 -3 = -2 = -1 = 0 = +1 = +2 = +3 = +4 = anchor 2
<u>Type Generic *</u> 1 = option 1 2 = option 2 3 = option 3 4 = option 4	<u>Type B *</u> -2 = anchor 1 -1 = 0 = +1 = +2 = anchor 2	<u>Type Generic *</u> -3 = option 1 -2 = option 2 -1 = option 3 0 = option 4 +1 = option 5 +2 = option 6 +3 = option 7	<u>Type Generic *</u> -4 = option 1 -3 = option 2 -2 = option 3 -1 = option 4 0 = option 5 +1 = option 6 +2 = option 7 +3 = option 8 +4 = option 9
	<u>Type Generic *</u> -2 = option 1 -1 = option 2 0 = option 3 +1 = option 4 +2 = option 5		

Types marked with “*” are designed to be modified by the researcher. The method for doing so is similar to the method for editing the question text, described above. As always, to aid in the process, the option text which should be modified is left-justified while the programming stuff is indented with two tabs.

Miscellaneous

"Miscellaneous" is another category of question templates segmented in the "generic_response_format" subdirectory. Four files are found here:

- `rhn2.htm` -- used for evaluating a stimulus with only two options, possibly in connection with response time measurements. Two large buttons with symbolic colors are used. This file was designed to work with "qcenter.htm" described earlier.
- `rinfo.htm` -- a frequently used file which simply presents information (e.g., instructions or stimuli) to the subject and a simple "continue" button. The elapsed time between the displaying of the screen and the subject's "continue" click is reported as the response time.
- `rmulti.htm` -- an example of a generic multiple-field question file, provided for researchers with programming experience. Since advanced programming is required to use such files, they are not specifically supported or described in this documentation.
- `rtimed.htm` -- a file which presents the question text for a fixed amount of time before moving to the next question automatically. This may be useful, for example, for researchers who desire to present a stimulus for exactly three minutes. The amount of time the question text is displayed can be varied by editing line 18 of the file:

```
NumOfSeconds=3;
```

The number determines how many seconds the question text will be shown. The file has not been tested for small values (e.g., less than one), as it was not designed for subliminal presentations. There is no maximum value. Since no response is collected from the subject by this file, zero is reported if the question name is referred to in "mfinish.htm".

Open-Ended Questions

Open-ended question templates are also available:

- `roeln.htm` -- has room for a long open-ended response. The response is not checked in comparison to any criteria and is simply reported in the data as-is (with a few minor exceptions; see Chapter VI)
- `roesn.htm` -- a short (i.e., one line) open-ended response. As with the above file, no data checking is performed.
- `roesy.htm` -- a short open-ended response which performs some checking of the response. The check performed by this file is that the response is a number between 0 and 100 (or, left blank for "Don't Know"). Other checks can be performed, but this requires advanced programming and is therefore not detailed here. **PROGRAMMERS:** Hopefully the checks performed in this file will be a decent springboard for you to create any customized checks you need to program.

My Questions

The “my_questions” subdirectory is empty. It is provided for you to have a convenient location to store commonly used questions. Since re-using questions from previous projects is particularly quick and easy (see Chapter IV), saving copies of often-used question files may be very useful.

Chapter VI

Data Handling

The data e-mail

The data e-mail created by "mfinish.htm" will be sent with the subject line "DATA for [study name]". The study name is defined in Step 5b.3 in Chapter III.

The e-mail identifies the browser (e.g., Netscape 3.0) used by the subject. Also, the e-mail identifies the sequence in which the questions were asked. This sequence will show all questions as defined in "mstart2.htm", regardless of whether they are referenced for data reporting in "mfinish.htm". The main purpose of this function is to show how any randomized ordered questions were presented (see Step 5b.10 in Chapter III). It may be important in some circumstances to know what order was randomly assigned. Since the data are reported in a fixed order each time, as described below, the exact sequence of questions as presented cannot be determined from the data without such a sequence report.

The data itself is identified by one (potentially very long) string of characters. The string is delimited to identify question names, response data and response time. The delimiting uses the characters "##" to delimit between questions and the characters "##" to delimit between elements of that question, as shown below:

```
##question_name##response_data##response_time
```

For example, the following portion of the data string...

```
##q01##1##10500
```

... shows the subject responded with a "1" for question "q01", and 10500 milliseconds (i.e., 10.5 seconds) elapsed between the time the question was displayed and the response button was clicked by the subject. The response time may not be useful for all questions, but it is reported for all questions in case it is useful.

Using the data

Admittedly, this long data string looks messy in the e-mail. But, it's in a useful format. The data can be imported into a statistics package by taking the following steps. This assumes you're familiar with importing into your stats package. For this purpose, I recommend using the syntax code instead of any pull-down menus because it's a copy & paste type of thing. Plus, if there are any errors, you'll need to modify a bit and repeat the process.

Prepare the data file for importing:

STEP 1 Open the data e-mail message.

STEP 2 Copy the "D A T A" portion of the e-mail (i.e., the long data string)

- Start copying immediately after the "D A T A" line.

STEP 3 Paste this string into a new text editor

STEP 4 Accumulate all the data lines (i.e., the responses) before continuing.

STEP 5 Backup the data file

Import the data file into the statistics package:

STEP 6 Edit the data file with a text editor and replace all commas (i.e., ",") with "***"

STEP 7 Replace all "##" with ","

STEP 8 Replace all "##" with ","

- Admittedly, the previous two steps equate the "##" and "##" delimiters. They are included for more advanced data handling.

STEP 9 Import into the stats package using the following format for each question:

```
[QuestionName]N, [QuestionName]D, [QuestionName]T
```

- For example, for the portion of the data line which reads a question named "q01", define the variables as ...

```
q01N, q01D, q01T
```

- Thus, for each question there will be three variables.

STEP 10 After importing, verify the [QuestionName]N variables:

- The value of the variable should be the same for all records.
- The value should be the expected value (i.e., the name of the question)

If either of these is not true, there is a problem with either the data file (e.g., one of the pasted records is incomplete) or with the import specifications (e.g., a variable was forgotten). After fixing the error, return to Step 9.

STEP 11 Delete all the [QuestionName]N variables and any [QuestionName]T variables for which the response time is not needed.

STEP 12 From within the stats package, replace all "***" in the dataset with ",".

STEP 13 From within the stats package, replace all "999" values with your favorite missing value holder (e.g., ".").

Missing Values

Missing values are handled in a consistent way in the reported data. For all closed-ended questions, "Don't Know" responses are reported as "999". Of course, this can be replaced in your dataset with your preferred value for missing data as described in the previous section. For open-ended questions, the response of the subject is reported exactly as it is typed; if the subject simply presses "continue" without entering any text, the reported string has no characters. In this case, the data might look something like the following:

```
##q001####10500
```

The above indicates that for q001, the subject did not enter any text (and spent 10.5 seconds thinking about it).

After performing the steps described immediately above, this portion of the data would be the following:

```
,q001,,10500
```

The response for this open-end will be read as a missing value by most statistics packages.

Open Ended Data

For technical reasons, some of the characters entered by subjects in open-ended questions must be changed in reporting. The following table shows characters which could be entered by subjects and then shows the character e-Experiment reports instead of that character:

entered		reported
'		~
;		@
"		*
#		}
\		{
[space]		+
newline -or- carriage returns		^

Although the open-ended responses are usually quite readable even with the above character substitutions, the substitutions can be reversed by editing the final data file and "replacing all" the "reported" characters with the "entered" characters.

Multiple Submissions

Finally, there is a potential problem with multiple submissions of the data e-mail, but the problem is minor and there is a workaround. The problem is based on the possibility of a subject immediately and repeatedly pressing the "submit" button displayed by "mfinish.htm". Doing so is difficult, since an 'okay' button is programmed to appear immediately after the "submit" button is pressed, but if the subject purposely clicks the button multiple times the same data e-mail will be sent multiple times.

There are two potential workarounds for this problem. First, a PIN can be used, and duplicate PINs can be checked for in the whole dataset during data processing. Any duplicate PINs are submissions from one subject. Examining the time stamps on the data e-mails will then show whether it was a problem of multiple clicking on the 'submit' button or whether it was a subject participating more than once. If the timestamps are not very close together in time, it is a case of the latter. Since having a subject participate more than once (e.g., a few days later) is usually not a savory prospect, this procedure may be a good practice anyway.

A second workaround for the multiple submission problem is simply to examine the timestamps of the data e-mails before you paste them into your data file. If the times of submission for any two data e-mails are within a few seconds, they are potentially 'multiple-click' submissions. If the responses for these submissions are equivalent, they are definitely 'multiple-click' submissions. When the e-mails are sorted according to send date/time, this is a relatively quick test to perform.

It should be noted that the "participation confirmation" e-mail generated through "qcredit.htm" and the "announcement" e-mail generated through "qintro2.htm" are also susceptible to multiple submissions.

Personally, I have not experienced any and do not expect many such multiple submissions, but they are possible and can be tested for as described above if you are particularly concerned.

Chapter VII

Advanced e-Experiment

This chapter addresses some of the advanced features of e-Experiment, so the explanations are not as detailed as some of the explanations in earlier chapters. Please contact me if you have any questions.

Advanced Conditions

Although the concept of Conditions in e-Experiment was meant to coincide with the designed conditions of the experiment, this definition may be somewhat limiting, and thinking of Conditions as potentially more powerful may be helpful. If, for example, the same stimulus will be presented to all subjects but a number of different question orderings should be assigned to subjects, the different orderings could be considered Conditions and the Sequence could reflect each Condition's special question ordering.

Another example of the flexible use of Conditions is using CarryText to vary the stimuli referred to in the questionnaire. For example, if brand personality measures are to be administered for a number of different stimuli brands, this can be accomplished by assigning the brand names to the CarryText variables and referring to them in the question files. In this way, a number of brands can be tested using only one set of question files (as opposed to having one set of question files for "Diet Coke", another for "Pepsi", etc.)

One limitation of the current version of e-Experiment is that the data reported in "mfinish.htm" must be the same across conditions. This forces each condition to have the same data-generating questions. This last point is important and deserves clarification. The question files defined in "mstart2.htm" do not have to be the same across conditions, but the question names reported in "mfinish.htm" do need to be the same across conditions. Therefore, although one condition can present information and stimuli which the other conditions do not, one condition cannot gather responses which are not gathered by all the others. If this limitation is problematic, contact me; I have an idea for a workaround.

If, for some reason, you want assignment to one condition to be more probable than the others, this can be accomplished, albeit with some effort. For example, if you have two conditions (A and B) but you want A to be assigned 2/3 of the time and B to be assigned 1/3 of the time, simply create three Conditions and make Conditions #1 and #2 perform Condition A and Condition #3 perform Condition B. This same idea can be extended to cover many situations.

Advanced Questions

Almost all the question files provided with e-Experiment have question portions which are based on static text -- that is, question text which is the same for each subject. However, a number of other features are possible:

- Displaying the same graphic file across all conditions.
- Displaying text which varies based on the assigned treatment. This is accomplished through the CarryText variables as described in Step 5b.7 in Chapter III.
- Displaying different graphic files, depending on the assigned condition.
- Displaying a subject's response to a previous question as part of the current question text.

The file "complete_questions/qall.htm" shows the syntax which yields these features. Simply copy the appropriate portion to your new question file and make any necessary edits. For example, "qall.htm" shows the code which displays the value of a CarryText variable as part of the question:

```
<SCRIPT LANGUAGE="JavaScript1.1">
document.write(ReadComment('CarryText1'));</SCRIPT>
```

(note: this code is all on one line in the file)

When copying code from "qall.htm" into your own question files, be sure to copy all the appropriate code (i.e., do not omit any characters).

Graphics

Graphics files require special consideration. First, the graphic formats supported by web browsers are the "gif" and "jpg" format, so only these types of files should be used. If you have a file of another format, it must be converted to one of these two formats.

In the portion of "qall.htm" which displays one graphic file across all conditions, the following code is used:

```

```

This code refers to the file "sample.gif" in the subdirectory "gif" of the project directory. For organizational purposes, graphic files should be kept in the "gif" subdirectory. Files in either "gif" or "jpg" format can be used in this subdirectory.

In the portion of "qall.htm" which displays different graphic files depending on the assigned condition, the following code is used:

```
[blah blah] ReadComment('Graphic') [blah blah]
```

This code reads the name of the file which is assigned to the variable "Graphic". This variable is defined in "mstart2.htm" as part of the Condition, just as the CarryText variables are defined. In fact, the "Graphic" variable is defined immediately after the CarryText variables in "mstart2.htm". In the 'factory preset' version of "mstart2.htm", the variable is defined for Condition 1 with the following code:

```
[blah blah] Graphic=gif/graphic1.gif [blah blah]
```

This will cause the variable "Graphic" to refer to the file "graphic1.gif" located in the "gif" subdirectory of the project directory. By defining the file differently for different Conditions, the graphic file displayed by the question file will depend on the assigned Condition.

Creating new question types

If you are interested in creating new, more advanced question types which can be directly incorporated into e-Experiment, let me know and I will give you an outline of what e-Experiment's main files require from such files. Basically, if the question file provides the response and writes a Cookie in a certain format, the question file can be dropped right into e-Experiment. Responses can be collected from the subject in a virtually unlimited number of new and creative ways.

Similarly, if you have a question file you would like to share with others, let me know and I will post it on the web site and notify the list of users of its presence (giving proper credit to you, of course).

Index

<p>999 36</p> <p>absolute 13</p> <p>absolute location..... 10, 13</p> <p>academic..... 5</p> <p>adding a question..... 21</p> <p>adding question text to generic question files 29</p> <p>advanced features 38</p> <p>advanced questions..... 38</p> <p>automatic advancement 33</p> <p>benchmark 24, 25</p> <p>browser 11, 35</p> <p>CarryText 17, 20, 38, 39</p> <p>CGI..... 10</p> <p>changing question order 22, 23</p> <p>citation of the software..... 5</p> <p>city 25</p> <p>closed-ended questions..... 31</p> <p>code section, definition of 15</p> <p>complete questions 24</p> <p>conditions 15, 19, 38</p> <p>conditions, number of..... 15</p> <p>consent..... 26</p> <p>contact the author 5</p> <p>credit for participating 27</p> <p>data handling 35</p> <p>data reporting 19</p> <p>deleting a question..... 21</p> <p>demographics 25</p> <p>directory structure 9</p> <p>documentation, attributes of..... 5</p> <p>don't know option 31, 36</p> <p>dropout rate 26</p> <p>editing..... 8, 9</p> <p>e-mail address of researcher 10</p> <p>e-mails, ANNOUNCEMENT 26, 37</p> <p>e-mails, DATA 35, 37</p> <p>e-mails, PARTICIPATION CONFIRMATION 27, 37</p> <p>factory 9, 12</p> <p>field names 20</p> <p>filename..... 14, 18</p> <p>filename extensions 8, 14</p> <p>FTP..... 6, 8</p> <p>gender 25</p> <p>generic response formats 29</p> <p>graphics 4, 17, 39</p> <p>HTML 4, 6, 30</p>	<p>installation 8</p> <p>internet connection 24</p> <p>JavaScript 4</p> <p>level 25</p> <p>line number..... 10</p> <p>long filenames 6, 8</p> <p>major 25</p> <p>missing values 36</p> <p>multiple fields..... 20, 25, 27, 28, 33</p> <p>multiple submissions 37</p> <p>my_questions..... 33</p> <p>new question types 39</p> <p>notation conventions 7</p> <p>open ended data..... 37</p> <p>open-ended questions 33</p> <p>operating system..... 8</p> <p>permission for using software 5</p> <p>PIN 26, 37</p> <p>plan 13</p> <p>project..... 9</p> <p>project name 16</p> <p>qagree.htm 24</p> <p>qall.htm..... 28, 39</p> <p>qbench1.htm 24</p> <p>qbench2.htm 25</p> <p>qcenter.htm..... 25</p> <p>qconfirm.htm 27</p> <p>qconfirm2.htm 28</p> <p>qconsent.htm 26</p> <p>qcredit.htm..... 27</p> <p>qdemo.htm..... 25</p> <p>qintro1.htm 26</p> <p>qintro2.htm 26</p> <p>qloss1.htm 27</p> <p>qloss2.htm 27</p> <p>qpanel.htm 25</p> <p>qpin.htm..... 26</p> <p>question formatting 29</p> <p>question name..... 35</p> <p>question order 22, 23</p> <p>question text 29</p> <p>question_file 14</p> <p>question_files 24</p> <p>random..... 4, 11, 15, 18, 38</p> <p>random assignment..... 4</p> <p>randomize 18, 21, 35</p> <p>refusal 25</p>
---	--

removing a question	21	STEP 4b	13
renaming a question	22	STEP 5	14
reported vs. entered characters	37	STEP 5a.....	14
response times	24, 25, 33, 35	STEP 5a.1.....	14
response to a previous question.....	39	STEP 5a.2.....	15
response types	31	STEP 5b	15
re-using a question	21	STEP 5b.1.....	15
rhn2.htm	33	STEP 5b.10.....	18
rinfo.htm.....	33	STEP 5b.2.....	16
rmulti.htm.....	33	STEP 5b.3.....	16
roeln.htm	33	STEP 5b.4.....	16
roesn.htm.....	33	STEP 5b.5.....	16
roesy.htm.....	33	STEP 5b.6.....	16
rtimed.htm	33	STEP 5b.7.....	17
sequence	17, 38	STEP 5b.8.....	17
skip patterns	4	STEP 5b.9.....	17
state	25	STEP 5c.....	19
statistics package.....	35	STEP 5d	20
STEP 1a.....	8	STEP 6	20
STEP 1b	9	stimuli.....	13, 33
STEP 2	10	subliminal	33
STEP 2a.....	10	testing	11, 20
STEP 2b	10	text formatting.....	29
STEP 2c.....	10	timed display of stimulus	33
STEP 2d	11	university.....	25
STEP 2e.....	12	UNIX.....	6, 8
STEP 3	13	unzip.....	9
STEP 3a.....	13	using the data.....	35
STEP 3b	13	welcome	26
STEP 4	13	year	25
STEP 4a.....	13	zip file.....	9