Behavior Research Methods
2005, 37 (4), 703-710

Generic HTML Form Processor: A versatile PHP script to save Web-collected data into a MySQL database

ANJA S. GÖRITZ
University of Erlangen-Nuremberg, Nuremberg, Germany

and

MICHAEL H. BIRNBAUM
California State University, Fullerton, California

The customizable PHP script Generic HTML Form Processor is intended to assist researchers and students in quickly setting up surveys and experiments that can be administered via the Web. This script relieves researchers from the burdens of writing new CGI scripts and building databases for each Web study. Generic HTML Form Processor processes any syntactically correct HTML form input and saves it into a dynamically created open-source database. We describe five modes for usage of the script that allow increasing functionality but require increasing levels of knowledge of PHP and Web servers: The first two modes require no previous knowledge, and the fifth requires PHP programming expertise. Use of Generic HTML Form Processor is free for academic purposes, and its Web address is www.goeritz.net/brmic.

In the last decade, the execution of experiments and surveys via the World-Wide Web has become an established method (see Birnbaum, 2001, 2004a, 2004b; Kraut et al., 2004). Several reviews have concluded that the quality of data achieved in online studies can be comparable to, and sometimes better than, that obtained by more traditional methods involving a lab, paper questionnaires, or telephone interviews (Birnbaum, 2001; Krantz & Dalal, 2000; McGraw, Tew, & Williams, 2000).

There are other advantages of Web research. On the Web, people can be tested at any time and place, laboratory rooms or physically present experimenters are not necessary (so experimenter effects remain constant), and automated data handling reduces both the labor and error of data coding and entry (Birnbaum & Reips, 2005; Göritz & Schumacher, 2000). In addition, the Web method allows one to collect large samples inexpensively, which makes it possible to draw clear conclusions and to check their generality to different subsamples tested (Birnbaum, 1999; Reips, 2002).

An example of an HTML Web form is given in Birnbaum (2000). Such an HTML page can be placed on a server, where the participant can view it and fill in answers by typing in information and clicking on choices. When the participant is finished, he or she can then click on a button to send the data. Birnbaum's (2000) surveyWiz and factorWiz are freely available programs that make it easy to create HTML forms for simple surveys and within-subjects factorial experiments. Reips and Neuhaus (2002) have developed WEXTOR, which is useful for generating Web experiments that utilize (for instance) between-subjects factorial designs with multiple pages for different conditions.

Sending Data From an HTML Form

There are three methods of receiving data that have been collected through an HTML form. The first method is to use the "get" method and extract the form input from the server log files (see Birnbaum & Reips, 2005).[1] A second method is to have the form data e-mailed to the researcher. This can be done by means of the HTML form's action attribute—for example,

```
<form action="mailto:you@your.domain.net"
method="post" enctype="text/plain">.
```

However, some systems refuse such an action attribute if, for example, no e-mail client is set up (as might be the case for, e.g., computers in a public library). Moreover, some browsers issue a more or less draconian alert, which the participant has to confirm for the data to be e-mailed. Most problematic for large efforts, though, is the fact that each submission generates its own e-mail, so data need to be extracted from thousands of individual e-mails and merged into one data file. Thus, whereas the e-mail method might be useful during testing of a form

This work was supported by a University of Erlangen-Nuremberg postdoctoral scholarship (HWP) and by National Science Foundation Grants SES-9986436 and BCS-0129453. Correspondence concerning this article should be addressed to A. S. Göritz, Organizational and Social Psychology, University of Erlangen-Nuremberg, Lange Gasse 20, 90403 Nuremberg, Germany (e-mail: anja.goeritz@wiso.uni-erlangen.de).

or for small efforts like obtaining RSVPs to a party, the method is not practical for large research projects.

The third method is to use server-side CGI ("common gateway interface") scripts to process and save the form data. In this case, the action attribute of a Web form sends the form's data to a CGI script that is located on a Web server (Schmidt, 1997, 2000). An example of such an attribute is

<form action="http://your.domain.net/script.php" method="post">.

Thus, use of CGI requires the researcher to have access to a CGI-enabled directory on a Web server.

There are several benefits of using a CGI script: First, the CGI can process the data and save them in a file format ready for analysis. Order bias can be eliminated by presenting items and alternative answers in random order, and the CGI can reorganize the data. Also, skip patterns can be incorporated into questionnaires. In addition, participants' input can be validated in real time; for example, data errors can be detected and respondents pointed to omitted items (Göritz & Schumacher, 2000).

CGI scripts can be written in any language that a given server can execute, such as ASP ("active server pages"), Perl ("practical extraction and report language"), or PHP ("hypertext preprocessor"). PHP is an increasingly popular scripting language (see www.php.net/usage.php). As with Perl, PHP interpreters are open-source, free, and available for many different platforms. One can check the availability of PHP for one's own platform and download a suitable installation package from the Downloads section of the PHP home at www.php.net.

Run Your Own Server With Apache, PHP, and MySQL

There are many advantages to running your own server (Birnbaum & Reips, 2005; Schmidt, Hoffman, & MacDonald, 1997). You can configure the most common Web servers to work with PHP. You can install PHP on a server by following the installation instructions that come with the downloaded package. In most cases, PHP is installed on the same server where the HTML forms reside, but it can also be installed on any other server where the data are to be saved. Apache is a powerful, widespread, and flexible open-source Web server. Apache's Web server comes already installed in new Macintosh computers, as are Perl and PHP, and is freely available for PCs and almost any other platform from www.apache.org.

There are two options for how a PHP script can store the data from the HTML form. One simple method is to have the script save the form data into a text file that is located on the server. For example, this might be a comma separated values (CSV) file. After all data have been collected, the file can be read into a spreadsheet or statistical application.

However, if participants' input at some stage of the research needs to be used dynamically to determine the next question, or if the questionnaire consists of more than one HTML form, it is advisable to have one's PHP script save the form data into a database. The advantage of the database is that it can store information about the participant, make computations on those data, and dynamically respond to the participant's behavior. Using a database allows the server to keep track of a participant who may perform many tasks over a period of time.

Servers can use various database applications, including Oracle, MS Access, and MySQL. We recommend MySQL because it is an open-source, free, compact, fast, reliable, robust, and multiuser database server that compiles on many platforms. Its home page is www.mysql.com. MySQL databases can easily be administered with the free tool MySQL Control Center (now succeeded by MySQL Administrator), which can also be downloaded from www.mysql.com.

The installation and configuration details of the PHP interpreter, Apache, MySQL, and MySQL Control Center are beyond the scope of this article. However, plenty of relevant information can be found on the Web. For example, there is an introductory tutorial on PHP, Apache, MySQL, and MySQL Control Center by the first author, which can be downloaded from www.goeritz.net/ati/. Moreover, with precompiled binaries available, installation has nowadays become fairly easy. Less experienced users might want to install an automatically configuring all-in-one bundle containing Apache, MySQL, and PHP. Searching the Web will locate many sites from which such a bundle can be downloaded.

To sum up, a powerful way of collecting data from HTML forms is to have the data sent to a CGI script, which processes the input and writes the processed data into a database. The most cost-effective way of accomplishing this is to use free, open-source software. We recommend the combination of an Apache server to host your Web site, PHP for CGI scripting, and MySQL for the database.

A Generic HTML Form Processor in PHP

A versatile PHP script called Generic HTML Form Processor has been developed that processes any syntactically correct input from HTML forms. It is available, along with sample HTML forms, from www.goeritz.net/brmic. The script creates "on the fly" a MySQL database containing one data table. In the table, the script dynamically sets up columns for all submitted HTML input fields and saves the data in the previously created columns. Thus, this script relieves researchers of the burdens of writing a CGI script and building a database to store their data for each new project. To run a survey or experiment, researchers merely have to complete the relatively simple task of creating HTML forms that fit their needs. There are many commercial and noncommercial HTML editors available that can assist the researcher with this task. For example, one of the two free programs surveyWiz and factorWiz (Birnbaum, 2000) might be used. The following versions have been written to automatically include the proper link to the script described in this article:

psych.fullerton.edu/mbirnbaum/programs/surveyWiz4.htm

and

psych.fullerton.edu/mbirnbaum/programs/factorWizRB4.htm.

Besides the basic functionalities of creating a database and saving the form input into this database, Generic HTML Form Processor can point respondents to omitted questions in the HTML form. Also, the researcher can choose whether the HTML input will be written into the database in chronological or alphanumeric order. Moreover, the wording of feedback messages to participants (e.g., the text to be displayed upon omission of an item or the thank-you message that appears after submission of the questionnaire) can be customized. To enable the identification of multiple submissions (i.e., most likely submissions from the same IP address within a short period of time) and of nonserious participants (e.g., those who filled out the questionnaire too fast or too slowly), the date, IP address, and browser information are logged, along with the submission time stamps of each questionnaire page. The script supports both one-page and multipage questionnaires. Researchers with access to a PHP-enabled Web server with MySQL can use Generic HTML Form Processor on their own server at no cost. They are also free to further customize the script. Researchers without access to a server can use Generic HTML Form Processor and a MySQL database on a dedicated server at the first author's university.

As a result of this spectrum of possible usage, we will now describe five modes of use that require different levels of knowledge of PHP and server issues.

1. The easiest mode for using Generic HTML Form Processor is with one-page HTML questionnaires designed to save data on a server at the first author's university. For this purpose, no knowledge of PHP and Web servers is required. The only preparation necessary is to set the action attribute of your HTML form to the URL (Internet address) of Generic HTML Form Processor. The HTML file itself can reside on any server. Let us look at an example HTML form called sample.htm. It contains most of the existing HTML input field types (see Figure 1).

The HTML code of sample.htm can be found in Listing 1. Note the action attribute in the form tag

<form method="post" action="http://www.goeritz.net/brmic/generic.php">,

which is printed in bold in Listing 1.

To use the script on our server, leave this line (the action attribute) as it is. If you would like to have the order of items or alternative answers in your HTML form randomized or rotated, you can use Birnbaum's (2000) factorWiz or include a special JavaScript program (Birnbaum & Wakcher, 2002).[2]

To obtain your study data from our server, call up the display script display_generic.php by typing the URL www.goeritz.net/brmic/display_generic.php in your

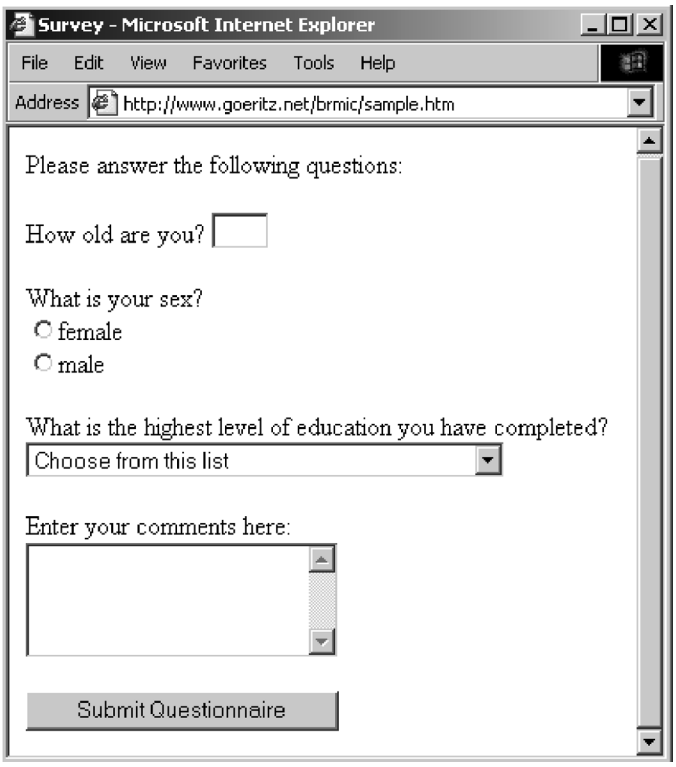


Figure 1. Browser view of example one-page HTML form sample.htm.

browser and hitting return. Next, read the instructions on this page and enter the URL of your HTML survey in the appropriate box. When you press the button, all data sets belonging to your study are output either as Excel files or in tab-delimited format. Once you have fetched your data, you can clean from them your own test runs as well as any multiple submissions. For more information on data analysis and filtering of multiple submissions, see Birnbaum (2001).

Please note that 3-month-old records are automatically deleted from this public database. Thus, make sure to retrieve your data in time. Furthermore, researchers intending to collect sensitive data (which should by no means be accessible to strangers) should use a script on their own server (see Modes 3–5 described below). Moreover, because of the public nature of this database, only up to 1,000 records and 70 unique HTML input fields per study are saved. If you plan a study larger than that, you are requested to use the script on your own server.

2. The second mode consists of using Generic HTML Form Processor with a multipage HTML questionnaire on the server at the first author's university. This also requires no knowledge of PHP or server issues. However, you would have to make a few more changes in your HTML forms, as follows: First, as in the previous mode, you need to set the action attribute of each of the HTML forms to the URL of Generic HTML Form Processor. Second, to tell the script which HTML page it needs to call up after processing the previous page, one extra line of HTML code must be inserted within the form tags. This line defines the hidden variable *next_page* and its value, which is the location of the next HTML page. An example is shown below in which sample2.htm is in the same Web folder as Generic HTML Form Processor:

> <input type="hidden" name="next_page" value="sample2.htm">.

Alternatively, if the next page were called "page_two.html," this line would read

> <input type="hidden" name="next_page" value="page_two.html">.

If the next survey page were located on another server or not within the same directory as the first page, the value should be set to the absolute URL of the second page—for example,

> <input type="hidden" name="next_page" value="http://full_path_on_server.net/pagename.htm">.

Let us inspect an example survey that consists of three HTML pages. The first page is sample1.htm (see the source code in Listing 2 and the browser view in Figure 2), the second is sample2.htm, and the third is sample3.htm.

To track individual participants across the pages of a multipage survey, a unique identifier needs to be passed from page to page. This is automatically accomplished by extracting an identifier for each participant (called *op*56) from the database and appending it to the query string of the next survey page—for example, www.goeritz.net/brmic/generic2.htm?op56=16&nr93=1.

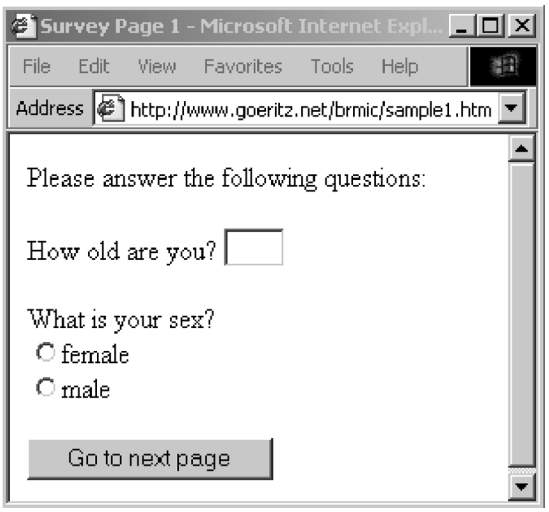


Figure 2. Browser view of example HTML form sample1.htm.

The other transmitted variable, *nr*93, tells Generic HTML Form Processor which page of a multipage survey it is dealing with. With the help of JavaScript, these two variables are extracted from the query string and inserted into the HTML form as hidden variables. Therefore, for each page of a multipage survey except the first, a JavaScript snippet (see the bold print in the sample2.htm source code in Listing 3) needs to be pasted within the form tags.[3]

Figure 3 depicts the browser view of the HTML file sample2.htm.

On the third page of our multipage study, sample3.htm, the researcher again needs to paste the JavaScript snippet between the form tags, but inclusion of the hidden variable *next_page* is no longer necessary, because sample3.htm is the last page of the survey (see the source code in Listing 4).

Figure 4 depicts the browser view of the HTML file sample3.htm.

Again, the script creates a MySQL database called "generic" with results table "generic." Because now there are three HTML pages, the names and submission time stamps of pages 2 and 3 are recorded as well (see Figure 5). Retrieval of the survey data follows the same procedure described for the one-page case.

For implementing skip patterns in your survey, there are three options. The first method requires only HTML and is illustrated at the following URL: www.goeritz.net/brmic/gender.htm. Both other methods use JavaScript. First, the skip pattern might be included within the same page of a survey (e.g., by having JavaScript display an additional item or response option if a particular answer has been selected prior to that). For example, if a respondent indicated having had a particular experience, an additional item could come up that would ask the respondent to de-

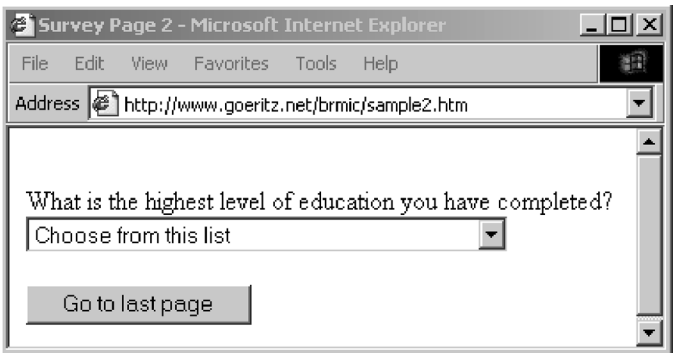


Figure 3. Browser view of example HTML form sample2.htm.

scribe this experience in more detail. The third option (and the second requiring JavaScript) would be to display any contingent material on a consecutive page. As an example, look at the source code of the HTML file sample1a.htm in Listing 5. This is a modified version of sample1.htm, with the bold text added. If in sample1a.htm a participant indicated that she was female, she would be directed to the interposed page sampleextra.htm instead of directly to the page sample2.htm (not shown here, but you can see this example as well as others in action at www.goeritz .net/brmic).

The same method can also be used for randomization of survey pages. For this purpose, a JavaScript snippet could draw an element of an array at random. The next page displayed would be contingent on the element selected.

3. The third mode involves using Generic HTML Form Processor with a one-page HTML questionnaire on an accessible server with PHP and MySQL. First, download Generic HTML Form Processor from www.goeritz.net/brmic and extract it onto your server in a suitable directory. To customize the script's behavior, follow the guidelines in the script. Alter the action attribute in your HTML form to the location of the script on your server—for example,

<form method="post" action="http://My.domain .net/MyFolder/generic.php">.

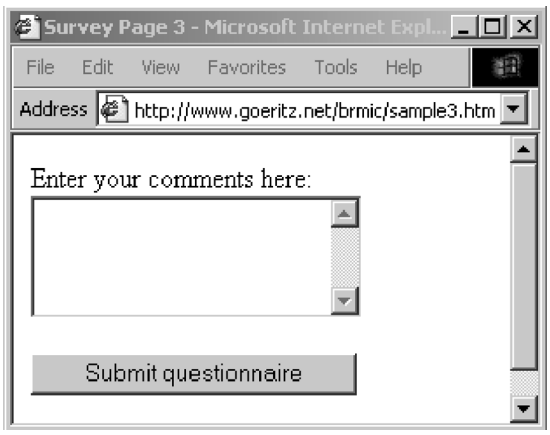


Figure 4. Browser view of example HTML form sample3.htm.

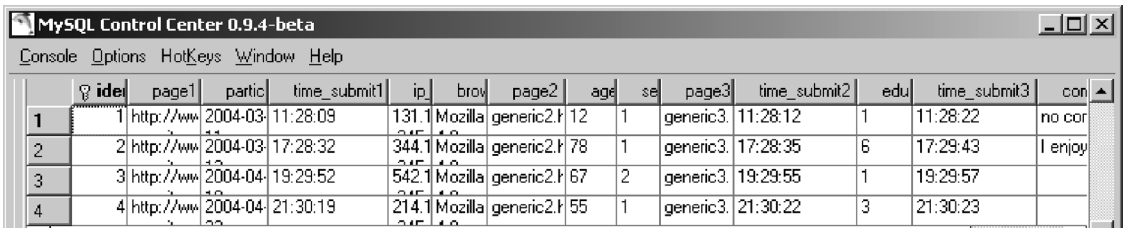
The first time the HTML form is submitted, the script creates a MySQL database called "generic" with a results table also called "generic." Data may be retrieved from your MySQL database either by running a SELECT syntax command or with the help of a graphical administration tool such as MySQL Control Center.

Note that for the script to function properly, the researcher needs ALTER, INSERT, SELECT, UPDATE, and CREATE privileges for MySQL. If the researcher wishes to use a public server, he or she might not have some of these privileges, because it is a security hazard to allocate global ALTER privileges to any user (in this case the researcher). There are three ways of dealing with this: (1) The researcher may ask the server administrator to create a minimal MySQL database with one table in it (this should take less than 1 min) and to allocate the relevant privileges only for the single table within this database. There is no security risk involved in that. (2) Since the ALTER privileges are needed only once—namely, to define all possible columns in the MySQL database—the researcher may ask the server administrator to grant him or her ALTER privileges for 10 min only. In this interval, the researcher can fill out all input fields on all pages of the questionnaire with practice data and submit it.[4] (3) The researcher can set up his or her own server and grant him- or herself the needed privileges for this step.

4. The fourth mode involves using Generic HTML Form Processor with a multipage HTML questionnaire on a server to which you have access. First, put the script on your server and, if desired, customize it. You will need to alter your HTML forms as described above for Mode 2.

5. Finally, you are free to further customize Generic HTML Form Processor to your particular needs—for example, by implementing a more sophisticated input validation. Learning PHP will allow you to completely customize your study.

In summary, our versatile PHP script Generic HTML Form Processor can assist researchers and students in quickly setting up surveys and experiments that can be administered via the Web. The script accomplishes the CGI



	ider	page1	partic	time_submit1	ip_	brov	page2	age	se	page3	time_submit2	edu	time_submit3	con
1	1	http://ww	2004-03-	11:28:09	131.1	Mozilla	generic2.h	12	1	generic3.	11:28:12	1	11:28:22	no cor
2	2	http://ww	2004-03-	17:28:32	344.1	Mozilla	generic2.h	78	1	generic3.	17:28:35	6	17:29:43	I enjoy
3	3	http://ww	2004-04-	19:29:52	542.1	Mozilla	generic2.h	67	2	generic3.	19:29:55	1	19:29:57	
4	4	http://ww	2004-04-	21:30:19	214.1	Mozilla	generic2.h	55	1	generic3.	21:30:22	3	21:30:23	

Figure 5. Sample table with results from a three-page survey.

scripting for coding and storing Web-collected data in a database. The strengths of Generic HTML Form Processor are its universality and simplicity of use. It processes any data input from an HTML form and saves it into a dynamically created open-source database. The script also allows the processing and recording of the data to be further customized. Researchers with access to a PHP-enabled Web server with MySQL can use Generic HTML Form Processor on their own server, and researchers without access to a server can use the script and a database on our server. Use of the script is free for noncommercial academic purposes.

REFERENCES

Birnbaum, M. H. (1999). Testing critical properties of decision making on the Internet. *Psychological Science*, **10**, 399-407.

Birnbaum, M. H. (2000). SurveyWiz and factorWiz: JavaScript Web pages that make HTML forms for research on the Internet. *Behavior Research Methods, Instruments, & Computers*, **32**, 339-346.

Birnbaum, M. H. (2001). *Introduction to behavioral research on the Internet*. Upper Saddle River, NJ: Prentice Hall.

Birnbaum, M. H. (2004a). Human research and data collection via the Internet. *Annual Review of Psychology*, **55**, 803-832.

Birnbaum, M. H. (2004b). Methodological and ethical issues in conducting social psychology research via the Internet. In C. Sansone, C. C. Morf, & A. T. Panter (Eds.), *The Sage handbook of methods in social psychology* (pp. 359-382). Thousand Oaks, CA: Sage.

Birnbaum, M. H., & Reips, U.-D. (2005). Behavioral research and data collection via the Internet. In R. W. Proctor & K.-P. L. Vu (Eds.), *The handbook of human factors in Web design* (pp. 471-492). Mahwah, NJ: Erlbaum.

Birnbaum, M. H., & Wakcher, S. V. (2002). Web-based experiments controlled by JavaScript: An example from probability learning. *Behavior Research Methods, Instruments, & Computers*, **34**, 189-199.

Göritz, A. S., & Schumacher, J. (2000). The WWW as a research medium: An illustrative survey on paranormal belief. *Perceptual & Motor Skills*, **90**, 1195-1206.

Krantz, J. H., & Dalal, R. S. (2000). Validity of Web-based psychological research. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 35-60). San Diego: Academic Press.

Kraut, R., Olson, J., Banaji, M., Bruckman, A., Cohen, J., & Couper, M. (2004). Psychological research online: Report of Board of Scientific Affairs' Advisory Group on the Conduct of Research on the Internet. *American Psychologist*, **59**, 105-117.

McGraw, K. O., Tew, M. D., & Williams, J. E. (2000). The integrity of Web-delivered experiments: Can you trust the data? *Psychological Science*, **11**, 502-506.

Reips, U.-D. (2002). Internet-based psychological experimenting: Five dos and five don'ts. *Social Science Computer Review*, **20**, 241-249.

Reips, U.-D., & Neuhaus, C. (2002). WEXTOR: A Web-based tool for generating and visualizing experimental designs and procedures. *Behavior Research Methods, Instruments, & Computers*, **34**, 234-240.

Schmidt, W. C. (1997). World-Wide Web survey research: Benefits, potential problems, and solutions. *Behavior Research Methods, Instruments, & Computers*, **29**, 274-279.

Schmidt, W. C. (2000). The server-side of psychology Web experiments. In M. H. Birnbaum (Ed.), *Psychological experiments on the Internet* (pp. 285-310). San Diego: Academic Press.

Schmidt, W. C., Hoffman, R., & MacDonald, J. (1997). Operate your own World-Wide Web server. *Behavior Research Methods, Instruments, & Computers*, **29**, 189-193.

NOTES

1. For example, the form tag might appear as follows:

   <form method="get" action="http://psych.fullerton.edu/mbirnbaum/thanks.htm">,

where the action directs the user to a "thank you" page for the study. For more information on this method, see Birnbaum and Reips (2005).

2. JavaScript is a client-side scripting language. That is, it runs in the participant's browser. Details on JavaScript programming are beyond the scope of this article, but more information can be found at psych.fullerton.edu/mbirnbaum/brmic. In addition, there are plenty of free JavaScript snippets for this purpose available for download on the World-Wide Web.

3. If a participant altered bits of the query string in "opt56=XY&nr93=XY" by hand, he or she could not access other participants' data and the functioning of the script would in no way be impaired, because the JavaScript snippet in the called-up page extracts information from the query string before a participant could interfere with it.

If a respondent in a multipage study had JavaScript disabled, a new record would be added to the database for each page instead of the respondent's entire session being written in one line. With the help of the data that supplement each record (i.e., URL of the survey page, date, time, IP address, and browser), a researcher could then reconstruct these sessions after data collection was over. That is, he or she could manually merge different lines for the same person into one line. Most of the (somewhat varying) statistics on JavaScript support indicate that 1%–10% of Web users have JavaScript disabled.

4. This procedure would have to be repeated for each new survey project, because the HTML input fields used will likely differ from one project to another.

LISTING 1
sample.htm

```
<html><head><title>Survey</title></head><body>
Please answer the following questions:<br><br>
<form method="post" action="http://www.goeritz.net/brmic/generic.php">
How old are you?<input type="text" name="age" size="2" maxlength="3">
<br><br>What is your sex?<br>
<input type="radio" name="sex" value="1">female<br>
<input type="radio" name="sex" value="2">male
<br><br>What is the highest level of education you have completed?<br>
<select name="education">
<option value="0" selected>Choose from this list
<option value="1">Less than 12 years
<option value="2">Graduated High School (12 years education)
<option value="3">Some College (13-15 years education)
<option value="4">Graduated from College (Bachelor's degree)
<option value="5">Master's degree
<option value="6">Doctoral Degree (Ph.D., M.D., J.D., etc.)
</select><br><br>Enter your comments here:<br>
<textarea name="comment" cols="21" rows="4"></textarea><br><br>
<input type="submit" value="Submit Questionnaire">
</form></body></html>
```

LISTING 2
sample1.htm

```
<html><head><title>Survey Page 1</title></head><body>
Please answer the following questions:<br><br>
<form method="post" action="http://www.goeritz.net/brmic/generic.php">
<input type="hidden" name="next_page" value="sample2.htm">
How old are you?<input type="text" name="age" size="2" maxlength="3">
<br><br>What is your sex?<br>
<input type="radio" name="sex" value="1">female<br>
<input type="radio" name="sex" value="2">male<br><br>
<input type="submit" value="Go to next page">
</form></body></html>
```

LISTING 3
sample2.htm

```
<html><head><title>Survey Page 2</title></head><body><br>
<form method="post" action="http://www.goeritz.net/brmic/generic.php">
<input type="hidden" name="next_page" value="sample3.htm">
<script language="JavaScript" type="text/javascript">
var qs=location.search.substring(1);
var nv=qs.split('&');
var url=new Object();
for(i=0; i<nv.length; i++)
{eq=nv[i].indexOf('=');
url[nv[i].substring(0,eq).toLowerCase()]=unescape(nv[i].substring(eq+1)); }
document.write('<input type="hidden" name="identification" value="'+url["op56"]+'">
   <input type="hidden" name="counter" value="'+url["nr93"]+'">');
</script>
What is the highest level of education you have completed?<br>
<select name="education">
<option value="0" selected>Choose from this list
<option value="1">Less than 12 years
<option value="2">Graduated High School (12 years education)
<option value="3">Some College (13-15 years education)
<option value="4">Graduated from College (Bachelor's degree)
<option value="5">Master's degree
<option value="6">Doctoral Degree (Ph.D., M.D., J.D., etc.)
</select><br><br>
<input type="submit" value="Go to last page">
</form></body></html>
```

LISTING 4
sample3.htm

```
<html><head><title>Survey Page 3</title></head><body>
<form method="post" action="http://www.goeritz.net/brmic/generic.php">
<script language="JavaScript" type="text/javascript">
var qs=location.search.substring(1);
var nv=qs.split('&');
var url=new Object();
for(i=0; i<nv.length; i++)
{eq=nv[i].indexOf('=');
url[nv[i].substring(0,eq).toLowerCase()]=unescape(nv[i].substring(eq+1)); }
document.write('<input type="hidden" name="identification" value="'+url["op56"]+'">
   <input type="hidden" name="counter" value="'+url["nr93"]+'">');
</script>
Enter your comments here:<br>
<textarea name="comment" cols="21" rows="4"></textarea><br><br>
<input type="submit" value="Submit questionnaire"><br><br>
</form></body></html>
```

LISTING 5
sample1a.htm

```
<html><head><title>Survey Page 1</title>
<script language="JavaScript" type="text/javascript">
function branch1()
{document.form1.next_page.value='sampleextra.htm';}
function branch2()
{document.form1.next_page.value='sample2.htm';}
</script>
</head>
<body>Please answer the following questions:<br><br>
<form name="form1" method="post" action="http://www.goeritz.net/brmic/generic.php">
<input type="hidden" name="next_page" value="sample2.htm">
How old are you?<input type="text" name="age" size="2" maxlength="3">
<br><br>What is your sex?<br>
<input type="radio" name="sex" value="1" onClick="branch1()">female<br>
<input type="radio" name="sex" value="2" onClick="branch2()">male
<br><br><input type="submit" value="Go to next page">
</form></body></html>
```

(Manuscript received May 28, 2004;
revision accepted for publication January 5, 2005.)