

***phpPanelAdmin* Manual** by Anja S. Göritz

Copyright (C) 2009 Anja S. Göritz <<http://www.goeritz.net>>

This file is part of *phpPanelAdmin*.

**Contents**

Technological Environment and Set-up of <i>phpPanelAdmin</i> .....	2
Setting up a New Panel with <i>phpPanelAdmin</i> .....	3
Connecting <i>phpPanelAdmin</i> to an Existing Panel.....	6
Administer an Online Panel with <i>phpPanelAdmin</i> .....	7
Conducting a Study in the Online Panel .....	11
A Typical Example of Usage .....	12

*phpPanelAdmin* is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

*phpPanelAdmin* is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with *phpPanelAdmin*. If not, see <<http://www.gnu.org/licenses/>>.

URL of Program: <http://www.goeritz.net/panelware/>

### Technological Environment and Set-up of *phpPanelAdmin*

To use *phpPanelAdmin* a web server (e.g., *Apache*) with an interpreter for the scripting language PHP (version 4.3 or above) and a web-enabled *MySQL* database are needed. If one does not avail of suitable server space, there are two ways of setting up a new web server on a computer with a standing Internet connection: The average user is advised to install a pre-configured bundle that contains *Apache* web server, *MySQL* database, and a PHP interpreter. Many bundles of this kind can be downloaded from the web for free (e.g., [www.apachefriends.org](http://www.apachefriends.org)). Experienced users might prefer installing and configuring the three components separately. A web server, for example *Apache*, can be obtained from [www.apache.org](http://www.apache.org). PHP can be obtained from [www.php.net](http://www.php.net). The *MySQL* community server is available at [www.mysql.com](http://www.mysql.com). As an alternative, researchers might rent suitable web space from a provider, which now has become inexpensive (i.e., around 10-20 USD per month).

For full functionality of *phpPanelAdmin* it is required that JavaScript and Cookies are enabled in the browser and that PHP and *MySQL* are properly configured on the server. If e-mails are to be sent out of *phpPanelAdmin*, for example, for sending new panelists a welcome message or to invite panelists to a study, PHP needs to be configured to send e-mail via an e-mail server. On most systems, this setting is done in PHP's main configuration file *php.ini*. For testing purposes it is often sufficient to set the SMTP directive in *php.ini* to the same as what is set up as your outgoing e-mail server in your local e-mail client. In *MySQL*, the researcher needs ALTER, INSERT, SELECT, UPDATE, DELETE, and CREATE privileges.

If the researcher wishes to use a shared server, he or she might not own some of these privileges. There are three ways of dealing with this: (1) Ask the server administrator to create a minimal *MySQL* database and to allocate the privileges only for this database. (2) As the CREATE privileges are needed only once, namely to create all tables in the panel database, ask the server administrator to grant CREATE privileges only for a short time. In this interval, the file *unfold.php* should be called up from the server through any Web browser, upon which the panel database automatically unfolds. (3) Set up your own server and grant yourself the needed privileges. If desired, after some modifications in the code, advanced users might be able to connect *phpPanelAdmin* to databases other than *MySQL*. There is no limit other than the one posed by the hard drive's capacity as to how many variables or panellists the panel may harbor.

Once the technological environment is in place, that is, a web server, *MySQL*, and PHP have been installed and configured, the next step is to download *phpPanelAdmin* from [www.goeritz.net/panelware/](http://www.goeritz.net/panelware/). Unpack the zipped file into a web-enabled directory of the

server. There results the sign-up form *signup.php* and the folder "ppa\_admin", which contains PHP files that make up the panel's administrative interface. Besides the license information *gpl.txt* and the manual *documentation.pdf* the 17 PHP files depicted in Figure 1 should be present.

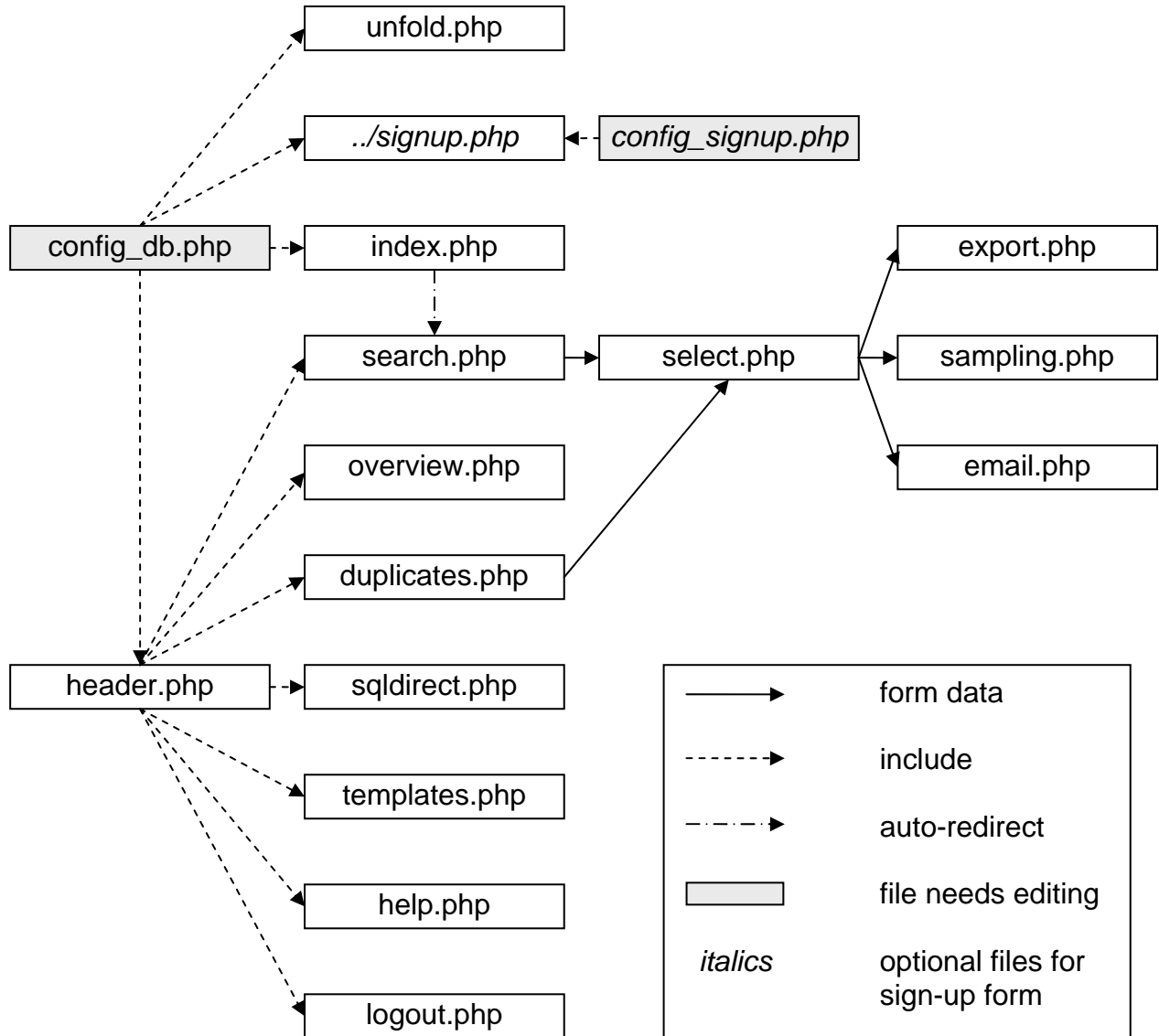


Figure 1. File structure of *phpPanelAdmin*

### Setting up a New Panel with *phpPanelAdmin*

First, open *config\_db.php* in a standard text editor (see Figure 2). Set your *MySQL* username and password in editing two lines in between the single quotes at the designated spot. You may need to adjust the host name (default: localhost) of the computer where *MySQL* is located. It is recommended that you set a default e-mail sender name and address for e-mail that is sent out of the platform. For more security, one can limit access to the panel

tool to one or more IP addresses. It is optional to adjust the names of the database and the tables into which *phpPanelAdmin* puts the panel data.

```
/* two lines you need to edit; alter only the words in single quotes */
$db_user='mysql_username'; //the username for the database (db) (if any)
$db_password='mysql_password'; //the password for the db (if any)

/* one line you may need to edit, e.g., mysql.myhost.net */
$db_host='localhost'; //host name or IP address where the db is located

/* it is recommended that you define a default e-mail sender name and
address for e-mail that is sent out of the platform */
$email_sender_name='XY-Panel'; //name of the sender of e-mails
$email_sender_address='yourpanel@www.yourpanel.net'; //e-mail address of panel

/* for more security, you can limit access to the tool to one or more IP
addresses; enter IP address(es) in format 'XXX.XXX.XXX.XXX' (if any);
comma-separate if more than one, e.g.,
$ips_allowed = array('100.100.200.101','100.100.200.102');
an empty array "array()" means there is not limitation*/
$ips_allowed = array();

/* in most instances you can leave the following lines as they are */
$db_database='ppa_paneltool'; //name of the db
$db_table='ppa_panelists'; //name of profile data table within the db
$db_table_admins='ppa_admins'; //name of table that lists admins of the panel
$db_table_deleted='ppa_deleted'; //name of table that lists deleted panelists
$db_table_templates='ppa_templates'; //name of table that lists e-mail templates
```

Figure 2. View of *config\_db.php* in text editor.

Second, load the file *unfold.php* from your server through any web browser, for example: [your.panel.net/ppa\\_admin/unfold.php](http://your.panel.net/ppa_admin/unfold.php). Make sure to load *unfold.php* from the server (i.e., type the Web address in the browser's location bar) but not to open it locally in the browser (i.e., do not use the "File->Open File" menu). This will cause the panel database to automatically spread out. If this step was successful the message "Databases were successfully created" is displayed on the screen. For security reasons, one should now remove the file *unfold* from the server. The following database entities should have been created: A database by the name specified in *config\_db.php* (default: *ppa\_paneltool*) with four tables in it (defaults: *ppa\_panelists*, *ppa\_admins*, *ppa\_deleted*, *ppa\_templates*). In the *admins* table there is one record defining the administrator of *phpPanelAdmin*. In the *templates* table there are four records with e-mail templates. Later on, one can delete these templates if not needed, and one can create one's own templates via the *phpPanelAdmin* web interface.

Most panels rely on a web form through which people can register to become panel members. Researchers can either keep using their existing sign-up form or create a new form based on a template that is part of *phpPanelAdmin*. If the panel is to work without a sign-up form simply delete the files *signup* and *config\_signup.php*.

To create a new sign-up form, open *config\_signup.php* to edit a few lines. One can choose whether to run a check on each submitted element in the sign-up form to make sure the panelist-to-be has not left any form element blank. Beyond this crude check, one may adjust the form elements' code to perform a selective completion check on individual form elements<sup>1</sup>. Furthermore, one can edit the thank-you text that is shown after people have submitted the sign-up form. Moreover, one can select if a confirmation e-mail is sent to new panelists, and edit the text of this e-mail. Sending e-mail presupposes that the sign-up form contains a field "email" through which a panelist's e-mail-address is collected and that the server is set up to send e-mail (see above). Next, one may switch on an automatic opt-out procedure that allows panelists to leave the panel at a mouse click. Moreover, *config\_signup.php* defines some variables that are saved along with each new registration (i.e., date, time, IP address and browser of the new panelist, and a random string called *reg\_urlcode* that is vital for the opt-out procedure and that may be included as an identifier in

<sup>1</sup> With text fields, switch validation off by adding some characters to the value-attribute, for example: ... or enter your answer here. A sample text field without validation is `<input name="email" type="text" value="" size="40">`, whereas the same field with validation looks `<input name="email" type="text" value="..." size="40">`. With text areas, switch validation off by adding some characters between the opening and closing textarea-tag. A sample text area with validation is `<textarea name="comments" rows="2" cols="5"></textarea>`. The same field without validation looks `<textarea name="comments" rows="2" cols="5">...</textarea>`. With radiobuttons, switch validation on by adding a hidden radio button to a set of radio buttons and leave its value empty, for example `<input name="gender" type="hidden" value=""><input name="gender" type="radio" value="1">female <input name="gender" type="radio" value="2">male`. To skip the validation, do not insert a hidden radio button or set its value to zero. With dropdown menus, switch validation on by leaving the value of one option empty, for example, `<select name="country"><option value="" selected>Select here</option><option value="1">Germany</option> <option value="2">Not from Germany</option></select>`. To skip the validation, set the value of one option to zero, for example: `<option value="0" selected>Select here</option>`. Finally, with checkboxes it usually does not make sense to require an answer. If you still want to make checking a checkbox mandatory, add a hidden field with empty value by the same

invitations to studies. One can define more variables that are to be saved along with each new registration according to a provided template. Finally, *config\_signup.php* contains the HTML code for the input fields that are to be displayed within the actual sign-up form *signup.php*. Each type of HTML input field is included at least once as an example (i.e., radio buttons, text area, single-line text input, drop-down menu, and checkbox). If more form fields are needed in *signup.php* one can clone and adjust an example input field and assign a descriptive name<sup>2</sup> to it. After everything is in place, load *signup.php* from your server through a browser, for example: [your.panel.net/signup.php](http://your.panel.net/signup.php). It is recommended to submit the form a few times with test data to create some mock-up panelists. Optionally, one can adjust the layout and style of *signup.php*. We are now ready to recruit real panelists through *signup.php*.

### Connecting *phpPanelAdmin* to an Existing Panel

Connecting *phpPanelAdmin* to an existing panel presupposes that your panel database is a *MySQL* database with panelists' profile data in a single table. The presence of additional tables (e.g., for administrators or deleted panelists) is unproblematic as they will not be affected by *phpPanelAdmin*. There are no restrictions on the number, name, and format of fields in the profile data table as long as there is an "id" field that holds a unique auto-incrementing integer for each panelist and an "email" field that holds the e-mail address.

At the outset, edit a few lines in *config\_db.php* in a text editor: Insert your *MySQL* username and password and set the database name to the panel's existing database and set the name of the profile data table to the panel's existing profile data table<sup>3</sup>. In addition, you may set a default e-mail sender name and address and limit access to the tool to one or more IP addresses, as described above. Second, call up the file *unfold.php* from your server through a browser, for example: [your.panel.net/ppa\\_admin/unfold.php](http://your.panel.net/ppa_admin/unfold.php). This will cause all *phpPanelAdmin*'s database tables to be created with the exception of the already existing profile data table. Third, if you intend to continue using the panel's already extant sign-up

name as the checkbox, for example: `<input name="agree" type="hidden" value=""><input name="agree" type="checkbox" value="1">`.

<sup>2</sup>The variable name should be as descriptive as possible to give some orientation to registrants who skipped this item in the sign-up form. In the course of the input validation, these registrants will be asked to fill out the field called by this name.

<sup>3</sup>In the unlikely event that one or more tables by the same name do already exist in your panel database, you may want to alter the default names of the other three tables to be created (defaults: *ppa\_admins*, *ppa\_deleted*, *ppa\_templates*).

form you should delete the files *signup.php* and *config\_signup.php*. Otherwise, follow the steps described above to set-up a sign-up form from a template.

### Administer an Online Panel with *phpPanelAdmin*

First, call up the administrative platform's log-in page *index.php* from your server through a browser, for example: [your.panel.net/ppa\\_admin/index.php](http://your.panel.net/ppa_admin/index.php).<sup>4</sup> Log on to *phpPanelAdmin* with your *MySQL* username and password. Upon a successful login, you are automatically re-directed to *search*, which is the home page of *phpPanelAdmin*.

On *search* one can search for one or more or all panelists by one search criterion or a combination of several criteria. On the left hand side in *search* all fields that exist in the profile data table are displayed within drop-down menus, and thus can be chosen as search criteria (see Figure 3)<sup>5</sup>. This search interface is quite powerful. Further information and example search patterns can be found in the respective help section that is part of *phpPanelAdmin*. For a quick start, select "id" as a criterion, enter % in the accompanying text field and hit "Perform Search" (see Figure 3). This will produce a list of all existing panelists.

<sup>4</sup> Most web servers are set up to serve up an index file if nothing else is specified. So the URL to be entered in the browser bar would simplify to [your.panel.net/ppa\\_admin/](http://your.panel.net/ppa_admin/).

<sup>5</sup> By default, only the first five profile variables are displayed. Hence, if there are more than five variables in the panel's profile data table and you want to display them all click on "More Fields". Moreover, one can choose the order in which the search results are displayed and whether the search should be case-sensitive. Furthermore, one can select the type of search upon each search criterion from among =, <, and >. With the search type = one can use wildcards: The percentage sign % matches any number of characters, including zero characters; an underscore \_ matches exactly one character. Finally, one can determine the type of combination among the different search criteria to be either of the four options: or, and, or not, and not.

[ [search](#) | [panel overview](#) | [duplicates](#) | [sql query](#) | [e-mail templates](#) | [help](#) | [logout](#) ]

## Search Panelist(s)

Case-Sensitivity:	<input checked="" type="radio"/> no (P = p)	<input type="radio"/> yes (P ≠ p)
Order Results by:	<input checked="" type="radio"/> ascending	<input type="radio"/> descending
<input type="text" value="id"/>		

Search Criterion	Search Type	Search Pattern
<input type="text" value="id"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input type="text" value="%"/> <input checked="" type="radio"/> or <input type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not
<input type="text" value="email"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input type="text"/> <input checked="" type="radio"/> or <input type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not
<input type="text" value="sex"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input type="text"/> <input checked="" type="radio"/> or <input type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not
<input type="text" value="stamp"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input type="text"/> <input checked="" type="radio"/> or <input type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not
<input type="text" value="browser"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input type="text"/> <input checked="" type="radio"/> or <input type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not

<input type="button" value="More Fields"/>	<input type="button" value="Reset Form"/>	<input type="button" value="Perform Search"/>
--	---	---

Figure 3. Browser view of the search mask in *search.php*.

The search result (i.e., the panelists that match the search pattern) is displayed on the same page underneath the search mask, whereby all profile data are displayed. Next, check the panelists on whose profiles you would like to perform one of the following operations: View the profile data in detail and perhaps modify their values ("View & Alter"), draw a random sample ("Sampling"), send an e-mail ("E-Mail"), delete panelist ("Delete Person(s)"), or export the profile ("Export"). If desired, check the variable(s) on which would like to perform one of the following operations: Delete one or more profile variables in all panelists ("Delete Variable(s)") or clone one profile variable into another variable for all panelists ("Clone Variable(s)") (see Figure 4).

Checked panelist(s):	<input type="button" value="View &amp; Alter"/>	<input type="button" value="Sampling"/>	<input type="button" value="E-Mail"/>	<input type="button" value="Delete Person(s)"/>	<input type="button" value="Export"/>
Checked variable(s):	<input type="button" value="Delete Variable(s)"/>	<input type="button" value="Clone Variable"/>			

Figure 4. Browser view of available functions in *search.php*.



The functions "View & Alter", "Delete Person(s)", "Export", "Delete Variable(s)", and "Clone Variable(s)" are self-explanatory. Just follow the on-screen instructions. With the function "Sampling" there is the choice to put all of the checked panelists into a sample or to draw at random a desired number out of the checked panelists into a sample. In either case one needs to specify a name of the sampling variable and the value assigned to sampled panelists in this sampling variable. One can either sample into a newly created variable or into an existing sampling variable<sup>6</sup>. In the latter case, the affected values of the old variable are overwritten with the new values. To set this sampling variable apart from ordinary profile variables – which supports clarity and might be useful for future reference – the name of each sampling variable is automatically prefixed by the string "sample\_ *currentdate* \_" when it is created.

When using the function "E-Mail" one may choose one of the existing e-mail templates. E-mail templates can be created, edited, and deleted under the link "e-mail templates" in the header menu that can be reached from all pages within *phpPanelAdmin* (see top of Figure 3). In the subject and body of the e-mail message one can include personalized text that is fetched from the database prior to sending the message. To include a piece of personalized text one needs to enter a placeholder. The placeholder needs to comply with the format ##name\_of\_variable##. For example, to address the receiver of the e-mail by his or her forename one could write "Hello ##forename##". This presupposes that there actually is a variable "forename" in the profile data table. To keep track of people who are invited to a study it is recommended to append their id or other unique identifiers to the study's URL and log this identifier when respondents hit on the first page of the study. Finally, in using the function "E-Mail" one may alter the default size of the e-mail batch that is sent at one go from 50 to any other positive integer.<sup>7</sup> When pressing "Send E-Mail" the messages in the next batch are sent at once. At the same time the variable "status\_email\_delivery" is created in the profile data table provided it does not yet exist from earlier mailings. This variable is filled on-the-fly as the e-mailing proceeds. It informs about the e-mail delivery status. This variable can assume the values "email sent", "selected for emailing", and "not selected". If for some

<sup>6</sup> Sampling into a newly created variable is useful when drawing a new sample. By contrast, sampling into an existing sampling variable is useful when modifying an existing sample, for example, for dividing an existing sample into several experimental conditions (see section "A Typical Example of Usage" below).

<sup>7</sup> It is recommended to sound out the capacity of your mail server successively by starting small and prudently increasing the batch size, provided sending out the messages has worked fine in the past.

unexpected reason the e-mailing is interrupted one knows which panelists have been sent the e-mail and which not. Thus one can resume the e-mailing where left off by sending an e-mail to those with value "selected for emailing".

In addition to the function "search" described above, the header menu that can be reached from all pages within *phpPanelAdmin* (see top of Figure 3) encompasses the functions "panel overview", "duplicates", "sql query", "e-mail templates", "help", and "logout". The functions "e-mail templates", "help", and "logout" are self-explanatory. Through the function "panel overview" one can obtain real-time panel statistics at a glance<sup>8</sup>. The function "duplicates" helps in identifying panelists who have registered multiple times. One can search for duplicates after checking one criterion or several criteria at the same time. Upon hitting "Identify Duplicates" all panelists will be displayed that have identical values in the checked criteria. One can now check one or several of the affected panelists and either view and alter their profile data or delete them. Through the interface "sql query" the advanced user can send queries written in SQL (i.e., Structured Query Language) to the database tables that are part of *phpPanelAdmin*<sup>9</sup>. It is recommended to read the respective help section that is part of *phpPanelAdmin*; among other information it contains example queries.

<sup>8</sup> The leftmost column lists all profile variables. Next to each profile variable, a drop-down menu contains a summary of all existing values of that variable. The summary's format is: (absolute frequency) (relative frequency) : value

For example, assuming there exists the panel variable "sex" and there are four panelists in the panel. Unfolding the respective drop-down menu reveals three menu items:

( 1 ) ( 25% ) : 0

( 1 ) ( 25% ) : 1

( 2 ) ( 50% ) : 2

Depending on the values you have set in the sign-up form, this might read: One panelist (i.e., 25% of all panelists) has a value of "0" in the sex variable. In this example, value "0" indicates that he or she did not give his or her sex. Next, one panelist (i.e., 25% of all panelists), has a value of "1" in the sex variable. That is, she is a woman. Finally, there are two panelists, who make up 50% of the panel population. They have a value of "2" in the sex variable, and hence are men.

<sup>9</sup> To query the profile data table use "\$db\_table", to query the table that contains panel administrators use "\$db\_table\_admins", to query the table that contains deleted panelists use "\$db\_table\_deleted". For security reasons, some potentially risky queries are blocked. Despite this safeguard, this interface is geared towards the expert to administer the panel even more efficiently.

## Conducting a Study in the Online Panel

*phpPanelAdmin* is not wired or limited to any survey tool or provider, that is, *phpPanelAdmin* and the study to be conducted are independent. This means one is free to use whatever questionnaire generating software or field partner one likes – if one relies on those at all. One is also free where one's study is hosted.

In terms of open-source programs, you might try *Generic HTML Form Processor* (Göritz & Birnbaum, 2005<sup>10</sup>) to create your web-based study. *Generic HTML Form Processor* is a free program for facilitating data collection with HTML forms. It automatically creates a *MySQL* database with one table in it containing columns that are named according to the variables that were submitted through the HTML form. Thus, *Generic HTML Form Processor* relieves researchers from writing a script that parses form input and writes the form input into a database as well as from setting up a database in the first place. *Generic HTML Form Processor* can be used for one-page surveys, multiple-page surveys, surveys with input validation, random assignment of participants to experimental conditions, skip patterns, and password-protection. One can obtain this program along with a tutorial and examples from [www.goeritz.net/brmic/](http://www.goeritz.net/brmic/).

Whatever the set-up of the study, it is usually desirable to know who of the invited panelists have heeded the request for participation. This information is important for the panel's quality management. Besides, it is the precondition for merging panelists' profile data with data from a particular study. To keep track of respondents from the panel into the study one can append the id and/or other unique identifiers to the URL of the survey (for details see description of the function "E-Mail" above). On the first survey page, the identifier should be extracted from the query string. If one uses *phpPanelAdmin* in conjunction with *Generic HTML Form Processor* one can achieve this by following these steps:

(1) When sending the e-mail invitation to a study out of *phpPanelAdmin*, after a ? append a unique identifier "id" for each panelist to the URL of the study, for example:  
[www.yourstudy.net/page1.php?id=uniqueidentifier](http://www.yourstudy.net/page1.php?id=uniqueidentifier).

(2) The first page of the study (for example: page1 . php) should have the extension .php instead of .htm or .html.

<sup>10</sup> Göritz, A. S. & Birnbaum, M. H. (2005). Generic HTML Form Processor: A versatile PHP Script to save Web-collected data into a *MySQL* database. *Behavior Research Methods*, 37, 703-710.

(3) Insert this one line of code into the first page of the study in between the form tags:  
`<input name="id" type="hidden" value="<?php echo $_GET[id]?>">.`

### A Typical Example of Usage

This section describes the typical steps to undertake in *phpPanelAdmin* when conducting a study in an online panel that is administered through *phpPanelAdmin*. As a sample scenario, suppose our panel contains 8,000 German panelists from all walks of life. Among other profile data such as the panelists' "id" and their "e-mail", we have information about their "surname", "sex" with values 1=female and 2=male, and a five-digit "postcode". The aim is to conduct an experiment among 700 panelists who live within postal areas starting with "8" or "9". There are two experimental groups to which respondents shall be randomly assigned. The conditions differ in the instruction on the first study page; thus the two groups get invited to different study URLs. In each condition, 350 panelists shall be invited.

First, on *search* we identify all panelists who have a postal code starting with "8" or "9" using the wildcard symbol % (see Figure 5). This results in a list of 1,498 panelists. Next, we draw a random sample of 700 panelists from the 1,498 eligible panelists. To achieve this, we leave the 1,498 panelists checked and press "Sampling" (see Figure 6). On the subsequent screen entitled "Sampling" we enter "700" in the text box and hit "Draw Sample" (see Figure 7). On the next screen as a name and value for the new sampling variable we enter "experiment\_postcode" and "1" in the two text boxes and hit "Proceed" (see Figure 8).

Search Criterion	Search Type	Search Pattern
<input type="text" value="postcode"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input type="text" value="8%"/>
<input type="text" value="postcode"/>	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	<input checked="" type="radio"/> or <input type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not
		<input type="text" value="9%"/>

Figure 5. Browser view of example search in *search.php*.

<input checked="" type="checkbox"/>	10418	bach@testemail.de	1	86423	Bach
<input checked="" type="checkbox"/>	10433	beethoven@testemail.de	1	89890	Beethoven
<input checked="" type="checkbox"/>	10441	heine@testemail.de	1	85317	Heine
<input checked="" type="checkbox"/>	10442	goethe@testemail.de	1	84578	Goethe

☒ uncheck all 1498 panelists

Checked panelist(s):

Figure 6. Browser view of example search result in *search.php*.

## Sampling

Sample all panelists who were checked:

Sample this number from checked panelists:

Figure 7. Browser view of sampling example in *select.php*.

## Sampling

Create or choose a variable and set the value for this variable:

Name of new variable:

Use existing variable:  ▼

Set value of new or existing variable:

Figure 8. Browser view of sampling example in *sampling.php*.

Next, we randomly divide the 700 panelists into two experimental conditions: Back on *search* we select the newly created variable "sample\_currentdate\_experiment\_postcode" in

the dropdown menu and enter value "1" in the search pattern text box before hitting "Perform Search" (see Figure 9).

Figure 9. Browser view of example search mask in *search.php*.

At the bottom of *search* we leave all 700 panelists checked and press "Sampling". On the next screen, we enter "350" in the text box and hit "Draw Sample". On the following screen in the dropdown menu next to "Use existing variable:" we select the name of our existing sampling variable "sample\_currentdate\_experiment\_postcode", and as the variable's value we enter "2" before hitting "Proceed" (see Figure 10).

## Sampling

Create or choose a variable and set the value for this variable:

Name of new variable:

Use existing variable:

Set value of new or existing variable:

Figure 10. Browser view of sampling example in *sampling.php*.

We are now ready to invite the participants to the study. There are two experimental groups with different URLs. Moreover, since in Germany women and men are addressed differently (i.e., as *Frau* and *Herr*) the salutation in the invitation e-mail needs to be varied accordingly. In sum, there are four groups of invitation e-mails to send out: (1) to women in Group 1, (2) to men in Group 1, (3) to women in Group 2, and (4) to men in Group 2. One can prepare an e-mail template beforehand through the screen "e-mail templates" or have the four

versions handy in a text file. To invite women in Group 1, on *search.php* we identify all panelists who have value "1" in the sampling variable "sample\_currentdate\_experiment\_postcode" AND value "1" in the profile variable "sex" (see Figure 11).

Search Criterion	Search Type	Search Pattern
sex	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	1
sample_20081010_experimer	<input checked="" type="radio"/> = <input type="radio"/> < <input type="radio"/> >	1
		<input type="radio"/> or <input checked="" type="radio"/> and <input type="radio"/> or not <input type="radio"/> and not

Figure 11. Browser view of example search mask in *search.php*.

On the bottom of *search.php* we leave the (in our example: 178) panelists checked and hit "E-Mail". Since we have the invitation handy in a text file, without selecting a template on the next screen we press "Compose E-Mail". On the next screen, from the dropdown menu we select the sender name and e-mail address we defined when setting up the panel. In addition, we enter a subject of the e-mail, for example, "Invitation to a Study". Next, we paste the text of the invitation in the text area captioned "E-Mail Body". For example, the text might read:

```
Dear Frau ##surname##,

We invite you to take part in a study about XXtopicXX. It
takes approximately XXnumber_of_minutesXX min to fill out the
questionnaire. The study is open until XXdeadlineXX.

To access the study please click on:
http://www.XXyour.panel.net/studyXX.htm?code=##id##

Kind regards,

The Panel Team
```

Replace the placeholders XX...XX with your own text. On the next screen, one may alter the size of the e-mail batch that is sent at one go. Upon pressing "Send E-Mail" the messages are sent. We proceed accordingly with the three remaining groups.