Linking surveys

https://www.limesurvey.org/forum/can-i-do-this-with-limesurvey/108384-linking-surveys-start-of-documentation (this article in the forum got you here, I guess…)

Hi, reader of these pages.

Here I want to set up a documentation article about linking surveys. Since not everything is clear to me, I invite you to annotate this and so we together find the power of the community and we hopefully come to the best documentation.

Linking surveys turned out to be necessary, because I ran into the limitations, as described in:
https://manual.limesurvey.org/General_FAQ#...regarding_survey_size

One of the power users of LimeSurvey helped me and sent me two linked surveys. Let's call them survey1 (with sid 111111) and survey2 (with sid 222222).

1. all this effort is done just because LS has limitations on the number of database columns that can be used in one survey? Or are there other use cases where linking surveys has other advantages?

2. I guess the limitation of LS concerning database columns has to do with the basic architecture of surveys and their results? And this is hard to change? Or are there other reasons for not changing this architecture?

3. Passing forward variables can be done, but there are limitations. What are these limitations? Is it the number of characters that can make up a URL , is it the number of variables as such that can be passed or are there other limitations? Can limitation be overthrown by passing through in another way? Something like a file containing all needed variables? Or does that need an awful lot of programming?

4. How to combine multiple variables in the URL?

5. I heard that you can also work with tokenized surveys, as long as the token tables of linked survey are identical. On second thought: I guess you only need a token table for the first survey and just pass the token to the second survey as a variable that is stored somewhere. Later you have to combine the results tables based on the tokens. Have you ever used that?

Greetings from Haarlem, The Netherlands,
Tammo ter Hark

Start of documentation article

Linking surveys

Documented for LS 2.6.1 and 2.50 and higher
This may also work for older versions, but has not been tested

Sometimes a survey can grow too large. One survey can only hold as much as your database allows you to. This is not in the number of responses, but the number of questions, so pay attention to that. Think of it as a table: your survey design with all questions make up the columns and each response is a row in that table.

In many cases, for day to day surveys, you will not run into trouble, but there are some tricky things. For this, please read the documentation about the max survey size at:
https://manual.limesurvey.org/General_FAQ#...regarding_survey_size
[I think that this article is too technical for non-developers. The examples at the end do help, could be expanded]
For people that want to know more about the database structure the developers of LimeSurvey have made an overview of the database structure 261, which is the basis as of now (beginning 2017):
https://manual.limesurvey.org/LimeSurvey_1.8_database_layout_documentation

When can I run into trouble?

When you have very many complicated questions. Think of it as follows: each field that you can fill in in a survey (text field, number field, checkbox, collection of radio buttons) takes one column in your results table. The maximum number of columns in your database depends on the type of encoding you use for your database. Follow the rule of thumb as stated in the article in the documentation:

You can roughly calculate the size of your survey like this:

- Every multiple numerical question: 20 chars for each answer
- Every multiple choice & array question answers: 5 chars for each answer
- Every other question type: 5 chars
- Add 10% for the usual overhead

And for the maximum number of characters:

- MySQL ISAM: The sum of the lengths of the VARCHAR and CHAR columns in a table may be up to 64KB. [this is too difficult for non-developers, what is VARCHAR and CHAR?]
- MySQL InnoDB: Maximum number of 1000 columns
- MS SQL Server 2000: Maximum number of 1024 columns
- Postgres: Maximum number of 250-1600 columns depending on column types. The maximum number of columns can be quadrupled by increasing the default block size to 32k. See Installation FAQ and PostgreSQL FAQ.

Be aware!

Currently, there is no warning when you exceed the limits of survey size, due to the fact that LimeSurvey does not know what database engine is used. I have opened a feature

request for this, but feature requests are normally only dealt with when there are not many pressing bugs and between development rushes. So this may remain a request.
You can however help this request by supporting it. Please vote for it, so the developers are "nudged" once more.
You can find the request at:
https://bugs.limesurvey.org/view.php?id=11099

Disclaimer: workaround only

Please note that what is sketched here, is only a workaround with many disadvantages, but it may help if it fits your use case.

Rule 1: try to make your design fit into 1 survey. Limesurvey works best this way. Then you will not need this workaround.

Known disadvantages:

- once the respondent has left survey1 and has begun filling in survey2, stepping back to survey1 is not possible anymore (unless you do a lot of programming). So take care when planning your survey.
- the progress indicator of survey2 starts at 0%, so you may need to adjust that. Or leave the indicator away, but in a very large survey that also has disadvantages for the respondent who needs to know where (s)he is on the scale of the whole survey.
- the indicator for "there are x questions in this survey" starts again in survey2, so you may need to adjust that
- when your survey heavily makes use of conditions where questions and/or answer options are depending on previous answers, or where variables are needed for computing scores during the survey (often at the end) you need to know that passing many variables from one survey to another is a very secure job, errors are easily made and there is a maximum to where you can go. When you want to pass many variables the URL may become too long. The maximum length or a URL is 2083 characters (see: Stackoverflow).
  To save space: keep the names of the variables as short as possible. And test often, there is nothing as frustrating as building a large construct and in the end finding out you have taken a wrong path...

How to link surveys?

The easiest way to link from one survey to the next is the following:
define the two surveys, for this example we will call them **survey1** (with sid 111111) and **survey2** (with sid 222222).

At the end of survey1 you can link to survey2 using the following end-url:

```
[path]/index.php/222222/newtest/Y
```

This way you go on with survey2, after finishing survey1. Survey2 will be ignorant about anything that has happened in survey1, you will not be able to access any value from survey1! Later you will also have difficulties knitting the results tables together, because it is not clear which response from survey2 corresponds with results from survey1. In order to make this possible (letting survey2 know what happened in survey1 and who filled it in), you need to pass on variables.

Passing a value to a following survey

Suppose you have a variable that you want to pass from survey1 to survey2. Let's say variable is the answer to a question (with the question code: **qvarout**) in survey1.

For this to happen, you need to have a hidden question in survey2. This question must have as question code the name of the variable you want to pass, so in this case, you will need a hidden question. For the answer to be saved, this question in survey2 can be of the type:

- numerical question
- short text question

Let's assume that the receiving question (in survey2) has the question code **qvarin**

*Note: passing a value **to** an equation type question does not work. Passing a value **from** an equation does work.*

The way to construct the end-URL using the variable from **qvarout** is:

```
[path]/index.php/222222/newtest/Y?qvarin={qvarout}
```

If you want to know who answered what, you need to do some more work. In most cases it is not safe to think that the order of responses in survey1 is exactly the same as in survey2, since people may not go through your survey at the same pace.

[something about SAVEID]

Passing multiple values to a following survey

When you want to pass multiple variable to the following survey, this goes similar:

Suppose you want to pass two variables, which are stored in **qvarout1** and **qvarout2** respectively.

The receiving questions must exist in survey2, of course, and they must have a format for storing whatever value **qvarout1** and **qvarout2** have (text, number etc.). Let's call the receiving question codes **qvarin1** and **qvarin2**

The way to construct the end-URL using the variables is:

```
[path]/index.php/222222/newtest/Y?qvarin1={qvarout1}&qvarin2={qvarout2}
```

[Is this the right syntax/way to do this?]

Working with tokens in linked surveys

Actually a token is a variable like others, so when you work with tokens you can link this to survey2 (survey1 must be configured to work with tokens, survey2 without tokens, but with a question (question type: short text). For this example we created survey2 with a receiving question with the question code **token_in**.

[path]/index.php/222222/newtest/Y?token_in={TOKEN}

[Needed: Explanation about combining results tables using the tokens]