

# Re: Multi phase processing

---

*Source:* <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2007-12/msg00213.html>

---

- *From:* Christian Winter <[thepoet\\_nospam@xxxxxxxx](mailto:thepoet_nospam@xxxxxxxx)>
  - *Date:* Thu, 06 Dec 2007 19:34:22 +0100
- 

Dave schrieb:

Hi folks,

I writing a perl/mysql backend to a website, part of which takes a number of pages of input, then, at the end, stores all the info on the database. Since Apache is stateless, all the data input from previous pages has to be put back into the next page in the form of hidden input tags, and then read back in each time. This is a touch long winded, but for most of the data works fine. I have however hit a snag with 1 set of data that is made up of a repeating set of which any number (or none) may be selected.

This data is a set of dynamically generated tick boxes of related data, all of which is taken from some data tables on the database. For example, if the page was asking which languages a person programmes in, it would then offer a list of choices from the database which the end user selects from. Part of the problem is that it also lists 3 separate options against each, for example used on VME, Windows, Linux.

The end result would look something like this, where each '@' is a separate input type="checkbox" html tag:

```
Language VME Windows Linux
Application Master @ @ @
C @ @ @
Cobol @ @ @
Perl @ @ @
SCL @ @ @
```

The idea being that if someone has programmed in Cobol on VM, and in Perl on Windows and Linux the appropriate boxes are ticked. The validation is fun because in the above, cut down, example, Application master and SCL are only available on VME, and Perl is the only one not available on VME. The other two being available on all platforms.

Where this is the only page, so that all the info is input in one go, I have no problems. Where I have a problem, is holding this in

## Re: Multi phase processing

temporary input type = "hidden" tags within the html page. Does anyone have any suggestions on the best way to hold this data?

As you're asking for the "best way", I'd suggest you get rid of the archaic "haul-all-params-ever-entered-with-me-around-hidden" approach completely and use a decent session module that lets you store intermediate results with a simple assignment, storing everything in one go in the database and fetching it on session initialization. This way, you only need one value (session id) on the client side to