

The first part of the form is the conditional logic where we collect user info for each registrant. As you suggest, we can add a ‘add friend’ button pretty easily, which means that at any point during the form we can ‘break out’ and go to the billing page.

To do so, we first need to realize that the Payment items are hard-coded during form creation to prevent tampering by the end user. For example, we don’t want to make it easy to change the number of registrants, price, and so on. Thus, we will create 4 different e-commerce pages, one for each number of possible registrants.

At this point then, we start to build the form. I have decided the easiest way to route users is, as you suggest, an **add friend** button.

Each additional user field will then simply get the number of the user it is appended to the name of the field. Thus, email1, email2, email3, and so on.

Your Name

Add Friend

Submit Form

F

^

v

Add Second User

Second User Name

The Submit Route property is the key here, with the value typed in 'user2.php', being the name of the page we want to visit should the button be

Add Friend

Name/Value

submit4

id

submit4

Submit Route

user2.php

Wrapper Class

Extra Attribute(s) JS Style Etc

However—and again, this is the way I’d do it—I will not use the Conditional processing built into FormBoss, simply because what we want is a bit more complex, and more importantly, should be flexible enough such that should anything change in the basic form, we only need to change a few elements, not the whole form. I will warn you though, I’m a huge nerd, so this may look complex at first, but I promise, is probably the best route to take.

Basically, what I do is create a separate page for routing between the number of users and the ecommerce items. This page will never actually show itself to the user, it just acts as a bridge.

Submit Form

F

^

v

CUSTOM ROUTER

This page is not really seen by the end user. it is simply used by us to route to the correct payment page. Wd get to this page those 'Submit Form' buttons.

F

^

v

Add 1 User

Add 1 User To This Event

Single User - \$1.00

Submit

C

^

v

Confirm 1 User

This page uses the PHP Top Code block to inspect the name fields, and if they are filled out, to route to the payment item page that matches the number of users entered.

Page PHP Top Code

PHP Top Code

```
<?php
session_start();

// Security Code - Not Needed For First Page Of A Form.

if(!isset($_SESSION['entry_key'])) { die(); }

// Your Code Below...

if(isset/#{user4}) && #{user4} != ''){
header("Location: proc4.php");
exit;
}

if(isset/#{user3}) && #{user3} != ''){
header("Location: proc3.php");
exit;
}

if(isset/#{user2}) && #{user2} != ''){
header("Location: proc2.php");
}

Toggle Large Editor  Insert PHP Tag

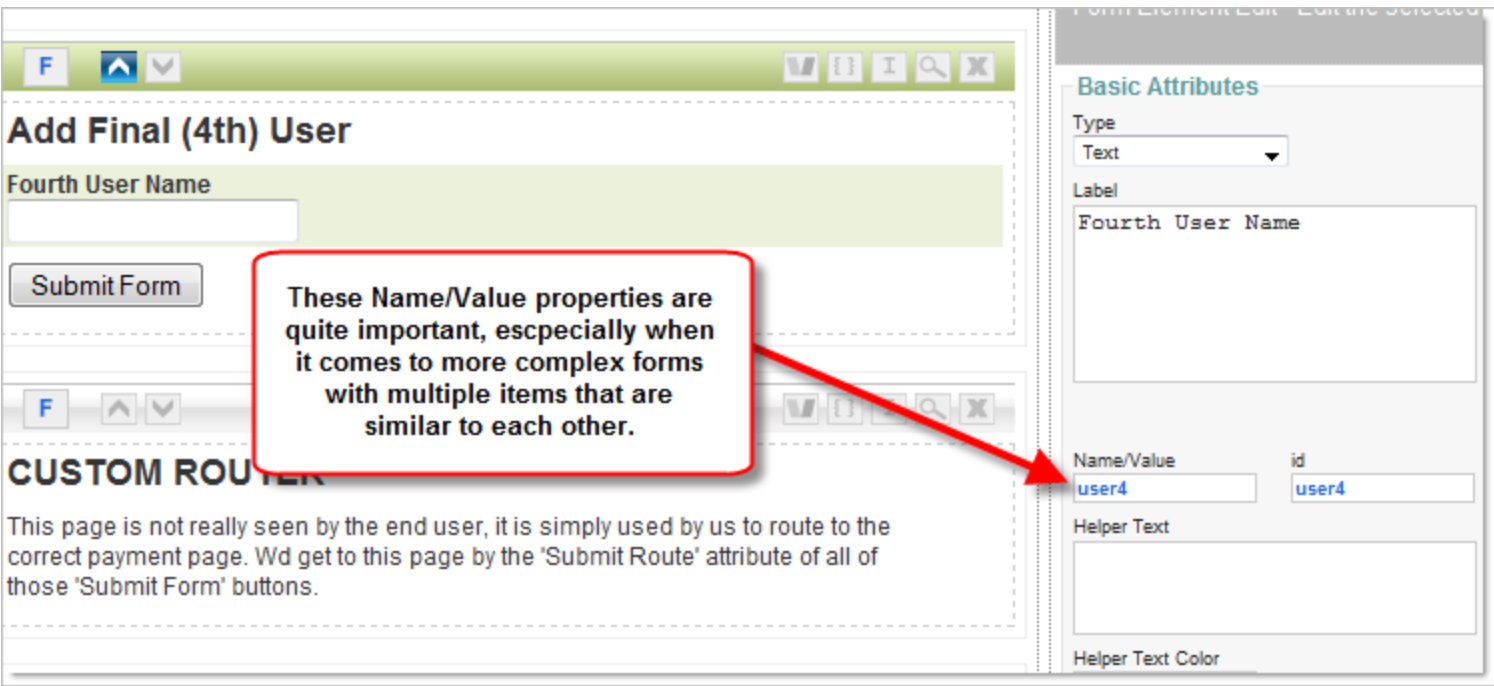
Retrieve PHP Field Elements*:
```

The PHP code uses FormBoss tokens as shorthand references to the various name fields in the form. Thus, the first if statement asks:

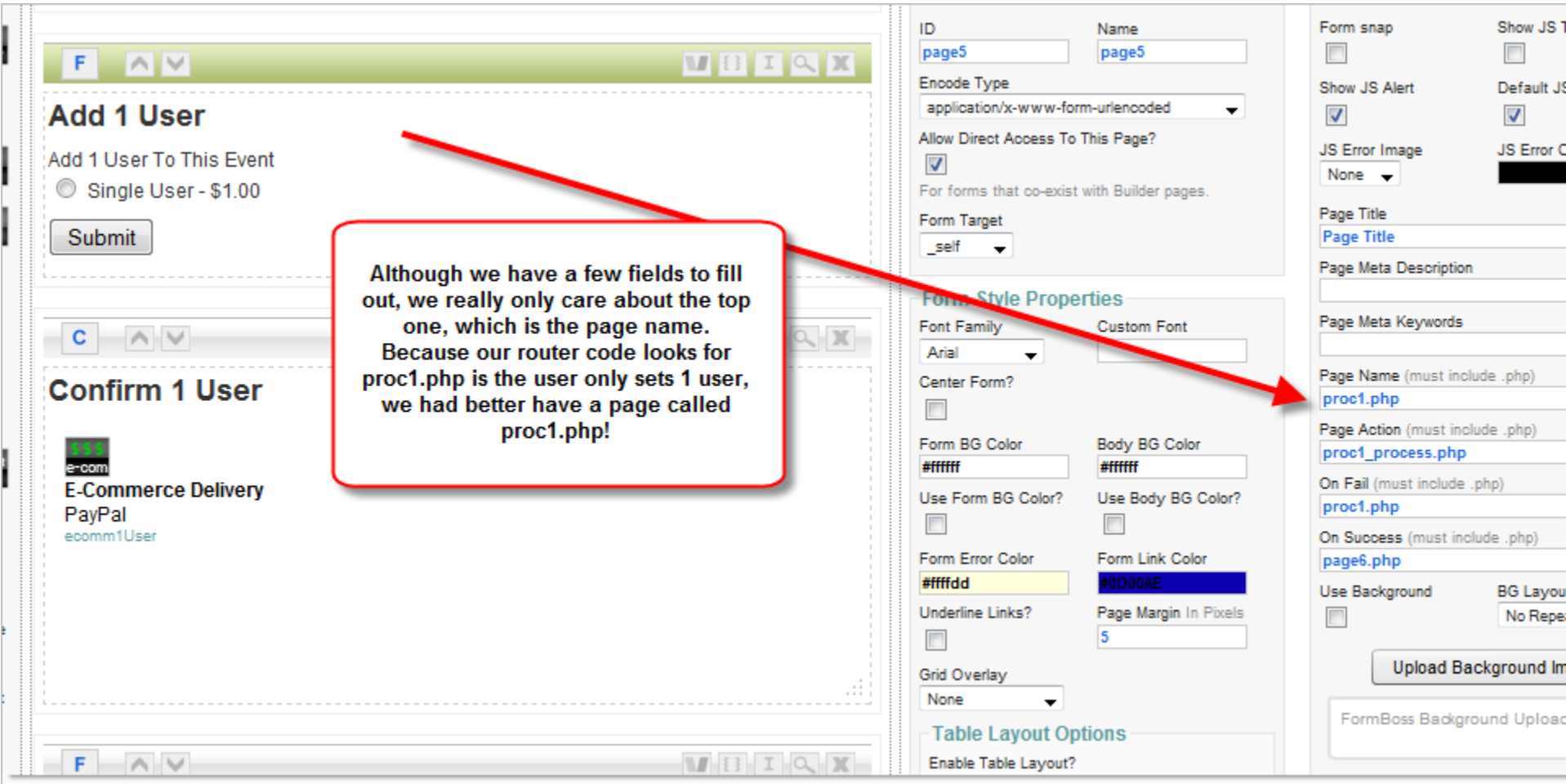
```
if(isset/#{user4}) && #{user4} != ''){
header("Location: proc4.php");
exit;
}
```

Which in English means: if the user4 field is set, *and* it has a non-blank value, we go to proc4.php, which is simply the page that has the 4-entry e-commerce item.

Just to be clear, the *user4* field designation comes from the Name/Value of the text box:



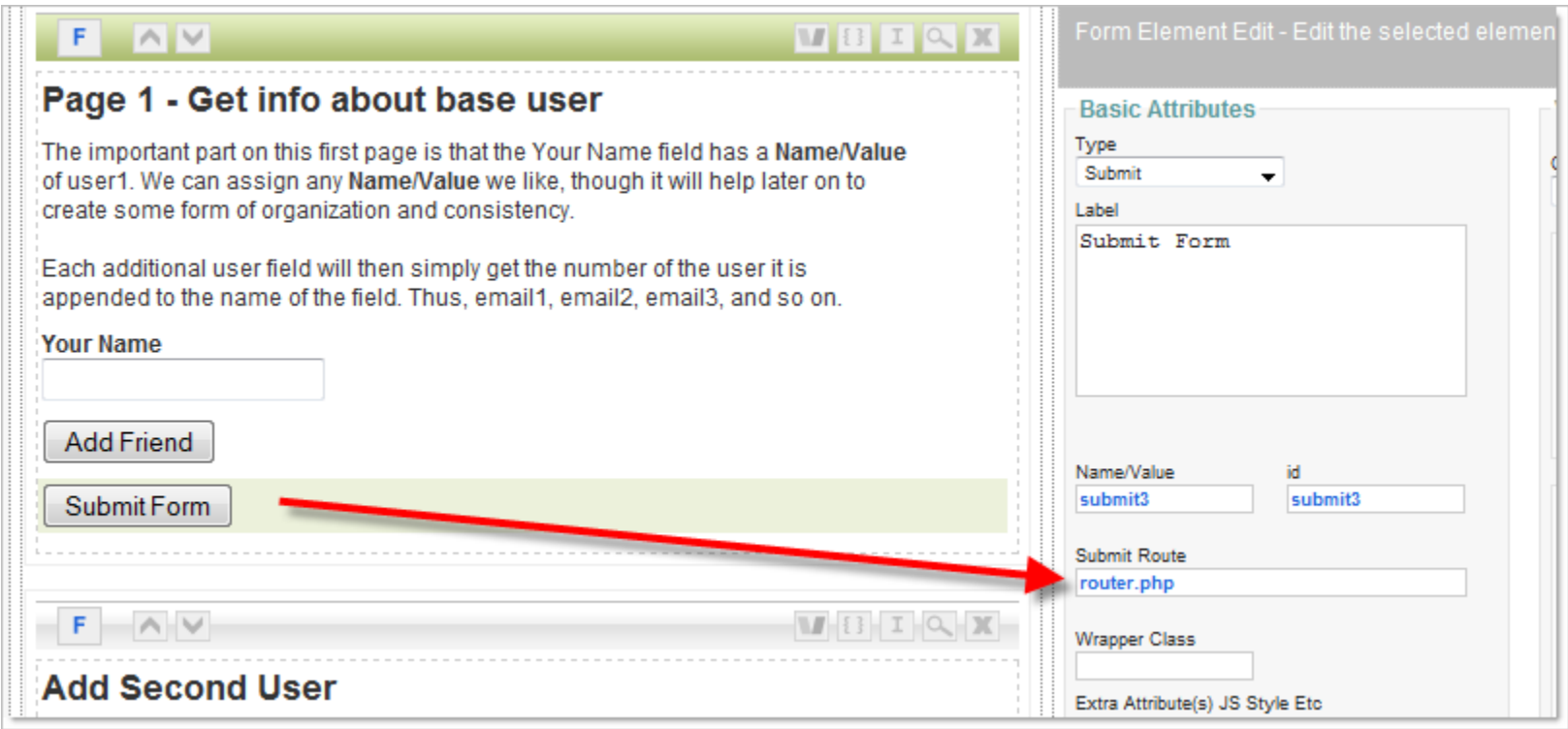
Please also note that to make this custom routing work, we need to be heavily involved with our page names. So as shown above, in that custom route code we go to pages proc1 – proc4.php.:



What this form is really doing, despite having a few pages, is the same thing 4 times.

When a user comes to the form they can skip directly to the ecomm page or add a friend.

If they go right to the ecomm, we go to the **router.php** page, as the Submit Route dictates:



On the router page then, we:

CUSTOM ROUTER

This page is not really seen by the end user, it is simply used by us to route to the correct payment page. We get to this page by the 'Submit Route' attribute of all of those 'Submit Form' buttons.

PHP Top Code

```
// Your Code Below...

if(isset($_user4) && $_user4 != ''){
header("Location: proc4.php");
exit;
}

if(isset($_user3) && $_user3 != ''){
header("Location: proc3.php");
exit;
}

if(isset($_user2) && $_user2 != ''){
header("Location: proc2.php");
exit;
}

if(isset($_user1) && $_user1 != ''){
header("Location: proc1.php");
exit;
}
```

Add 1 User

Add 1 User To This Page

☐ Single User - \$100

Because the user only filled out the first page name field--which again, is called user1--we send the user, via the header command, to the proc1.php page.

Which as we can see, is this page:

The screenshot displays the Form Builder interface with two form pages and their respective properties.

Form Page 1: Add 1 User

- Title: Add 1 User To This Event
- Options: Single User - \$1.00
- Submit Button: Submit

Form Page 2: Confirm 1 User

- Title: Confirm 1 User
- Image: e-comm
- Text: E-Commerce Delivery, PayPal, ecomm1User

Form Style Properties

- Form Name: page5
- Name: page5
- Encode Type: application/x-www-form-urlencoded
- Allow Direct Access To This Page: ☒
- Form Target: _self
- Font Family: Arial
- Center Form?: ☐
- Form BG Color: #ffffff
- Body BG Color: #ffffff
- Use Form BG Color?: ☐
- Use Body BG Color?: ☐
- Form Error Color: #ffffdd
- Form Link Color: #0000AE
- Underline Links?: ☐
- Page Margin In Pixels: 5
- Grid Overlay: ☐

Page Properties

- Page Title: Page Title
- Page Meta Description:
- Page Meta Keywords:
- Page Name (must include .php): proc1.php
- Page Action (must include .php): proc1_process.php
- On Fail (must include .php): proc1.php
- On Success (must include .php): page6.php
- Use Background: ☐
- BG Layout F: No Repeat

A red box highlights the text: "How do we know which page this is, because we called it proc1.php, that's how :)". A red arrow points from this box to the "Page Name" field in the "Page Properties" section, which contains the value "proc1.php".

When they click submit:

The screenshot shows the Visual Studio Code editor with two route files: 'Add 1 User' and 'Confirm 1 User'. The 'Add 1 User' route has a 'Submit' button. The 'Confirm 1 User' route has a 'Submit Route' field set to 'confirm1.php'. Red arrows point from text boxes to these elements.

Add 1 User

Add 1 User To This Event

☐ Single User - \$1.00

Confirm 1 User

 E-commerce Delivery
PayPal
ecom1User

When we click submit our Submt Route says guess what, we're going to confirm1.php, which is the guy directly below us...

Which becuase it is a confirmation page, means we'll process the single user e-commerce item.

Submit

Name/Value id

Submit Route

Wrapper Class

Extra Attribute(s) JS Style Etc

reference container with: fb_fid-

Conditional Processing

Field Initially Visible? ☒

Conditional Route Statement

Of course this process then is simply repeated for each possible user.

