Registration Form Field Manipulations

This tutorial is for advanced users of the software who have html and Smarty code experience. Do not attempt unless you have a firm grasp of each and the ability to debug the changes you make if you run into a problem. Support cannot help with your php code or system template changes like the ones you would make below. One thing to know is if you follow the guidelines in "custom system template changes" wiki article here you will always be able to restore yourself to the default registration form.

First to get a feel for what variables that are available in this or any system template you can simply add the {debug} tag to that system template. This support wiki article explains the use of that Smarty supported tag in more detail.

The changes here will explore the system template you see in the registration form. Those changes specifically deal with the following system template:

```
/geo_templates/[your template set - not
default]/system/registration/registration_form.tpl
```

And as always follow these guidelines when making system/module template changes as the /default template set is over written in an update:

```
http://geodesicsolutions.com/support/geocore-wiki/doku.php/id,tutorials;desi
gn_adv;replace_system_templates/
```

If you find you get stuck or lost with your changes and want to "start over" or just scrap what you have you can always delete the:

```
/system/registration/registration_form.tpl
```

template in your custom template set. That will tell the system to use the same system template in the default template set that you didn't make your changes to (correct?).

If you wish to change the position of some general fields or control the display of all general fields in the registration form you would need to make changes to the first "foreach $fields" loop in the above template.

As of the writing of this tutorial this was that loop but this Smarty template can change a bit as new versions of the software are released.

```
        {foreach from=$fields item=f}
            <div class="{if $f.error}field_error_row{else}{cycle
values='row_odd,row_even'}{/if}">
                {if $f.label}<label for="{$f.name}" class="{if
$f.required}required{else}field_label{/if}">{$f.label|fromDB}{if
$f.required} *{/if}</label>{/if}
                {if $f.type == "text"}
                    <input type="text" id="{$f.name}" name="{$f.name}"
value="{$f.value}" size="{if !$f.size or $f.size >
30}30{else}{$f.size}{/if}" maxlength="{if !$f.size}100{else}{$f.size}{/if}"
```

```
class="field" />
                {elseif $f.type == "radio"}
                    {foreach from=$f.options item=radio key=k
name=radioLoop}
                        <label><input id="{$f.name}" type="radio"
name="{$f.name}" value="{$k}" {if $radio.checked}checked="checked"{/if} />
{$radio.text}</label>
                    {/foreach}
                {else}
                    {* no field type explicity set by PHP. Probably
something like Regions that has its own HTML *}
                    {$f.value}
                {/if}
                {if $f.error}
                    <span class="error_message">{$f.error|fromDB}</span>
                {/if}
            </div>
        {/foreach}
```

You'll see that the loop cycles through the contents of $fields array to display each field. If you inserted the {debug} tag as instructed above you would see the "fields" array inside the $body_vars variable. You would see all of the registration fields you have turned on within the following admin tool:

REGISTRATION SETUP > GENERAL SETTINGS

If you want to just move a specific field around in that form you would need to make sure that the above loop skips the display of that field. If for example you wanted to display the first name and last name fields to display somewhere else you would do the following:

```
        {foreach from=$fields item=f}
            {if $f.name neq "c[firstname]" || $f.name neq "c[lastname]"}
                <div class="{if $f.error}field_error_row{else}{cycle
values='row_odd,row_even'}{/if}">
                    {if $f.label}<label for="{$f.name}" class="{if
$f.required}required{else}field_label{/if}">{$f.label|fromDB}{if
$f.required} *{/if}</label>{/if}

                    {if $f.type == "text"}
                        <input type="text" id="{$f.name}" name="{$f.name}"
value="{$f.value}" size="{if !$f.size or $f.size >
30}30{else}{$f.size}{/if}" maxlength="{if !$f.size}100{else}{$f.size}{/if}"
class="field" />
                    {elseif $f.type == "radio"}
                        {foreach from=$f.options item=radio key=k
name=radioLoop}
                            <label><input id="{$f.name}" type="radio"
name="{$f.name}" value="{$k}" {if $radio.checked}checked="checked"{/if} />
{$radio.text}</label>
                        {/foreach}
                    {else}
                        {* no field type explicity set by PHP. Probably something like Regions that has its own HTML *}
                        {$f.value}
                    {/if}

                    {if $f.error}
                        <span class="error_message">{$f.error|fromDB}</span>
                    {/if}
                </div>
            {/if}
        {/foreach}
```

And then you would place the following where you want that field to appear in that same smarty template. Here we are just using the first name field and you should be able to extrapolate to other fields:

```
                <div class="{if $fields.firstname.error}field_error_row{else}{cycle values='row_odd,row_even'}{/if}">
                    {if $fields.firstname.label}<label for="{$fields.firstname.name}" class="{if $fields.firstname.required}required{else}field_label{/if}">{$fields.firstname.label|fromDB}{if $fields.firstname.required} *{/if}</label>{/if}

                    {if $fields.firstname.type == "text"}
                        <input type="text" id="{$fields.firstname.name}" name="{$fields.firstname.name}" value="{$fields.firstname.value}" size="{if !$fields.firstname.size or $f.size > 30}30{else}{$fields.firstname.size}{/if}" maxlength="{if !$fields.firstname.size}100{else}{$f.size}{/if}" class="field" />
                    {elseif $fields.firstname.type == "radio"}
                        {foreach from=$fields.firstname.options item=radio key=k name=radioLoop}
                            <label><input id="{$fields.firstname.name}" type="radio" name="{$fields.firstname.name}" value="{$k}" {if $radio.checked}checked="checked"{/if} /> {$radio.text}</label>
                        {/foreach}
                    {else}
                        {* no field type explicity set by PHP. Probably something like Regions that has its own HTML *}
                        {$fields.firstname.value}
                    {/if}

                    {if $fields.firstname.error}
                        <span class="error_message">{$f.error|fromDB}</span>
                    {/if}
                </div>
```

Seeing the above you can see where "lastname" could be substited for "firstname" to get the code to display for the last name field. The name of the specific field to use in the $field array can be gotten from the debug display performed at the top of this page. Using the above code gives you the same controls over that field that you already have from the admin tool. As you can see the above covers all of the fields configurations that can be made to those fields in the admin tool. Seeing this is the firstname field which will ONLY be a text box you could use the following: Also using the above you could "scrap the loop" all to tegether and just place each individual field within the registration_form.tpl template where ever you wanted the to display.

```
                <div class="{if
$fields.firstname.error}field_error_row{else}{cycle
values='row_odd,row_even'}{/if}">
                    {if $fields.firstname.label}<label
for="{$fields.firstname.name}" class="{if
$fields.firstname.required}required{else}field_label{/if}">{$fields.firstnam
e.label|fromDB}{if $fields.firstname.required} *{/if}</label>{/if}

                    <input type="text" id="{$fields.firstname.name}"
name="{$fields.firstname.name}" value="{$fields.firstname.value}" size="{if
!$fields.firstname.size or $f.size >
30}30{else}{$fields.firstname.size}{/if}" maxlength="{if
!$fields.firstname.size}100{else}{$f.size}{/if}" class="field" />
                    {if $fields.firstname.error}
                        <span class="error_message">{$f.error|fromDB}</span>
                    {/if}
                </div>
```

And you can see that this could be used for any "text" field by "switching out" the variable names. This immediately code above could only be used for text type fields of the registration form and wouldn't work for the account type field.

Those are the basics but you can extrapolate the same for the 10 optional registration fields loop if you are wanting to change their placement in that form.

To emphasize the point changes like these are beyond the scope of support that we provide through our support staff. If you can temporarily remove the use of your system template (renaming it) and the base registration form works with the default template in use your changes would be the problems if there are any issues when your custom template is in use. If ours default template works and your custom template doesn't you're beyond support our support staff can provide. If you need help from Geodesic you would need to purchase developer time through sales or consult with an external developer familiar with html and Smarty code.