function form\_example\_states\_form

|  |  |
| --- | --- |
| **7.x-1.x form\_example\_states.inc** | form\_example\_states\_form($form, &$form\_state) |

States demo form.

This form shows off the #states system by dynamically showing parts of the form based on the state of other parts.

The basic idea is that you add a #states property to the element which is to be changed based on some action elsewhere on the form. The #states property lists a change which is to be made, and under what conditions that change should be made.

For example, in the 'tests\_taken' form element below we have:

'#states' => array(

'visible' => array(

':input[name="student\_type"]' => array('value' => 'high\_school'),

),

),

Meaning that the element is to be made visible when the condition is met. The condition is a combination of a jQuery selector (which selects the element we want to test) and a condition for that element. In this case, the condition is whether the return value of the 'student\_type' element is 'high\_school'. If it is, this element will be visible.

So the syntax is:

'#states' => array(

'action\_to\_take\_on\_this\_form\_element' => array(

'jquery\_selector\_for\_another\_element' => array(

'condition\_type' => value,

),

),

),

If you need an action to take place only when two different conditions are true, then you add both of those conditions to the action. See the 'country\_writein' element below for an example.

Note that the easiest way to select a textfield, checkbox, or select is with the [':input' jquery shortcut](http://api.jquery.com/input-selector/), which selects any any of those.

There are examples below of changing or hiding an element when a checkbox is checked, when a textarea is filled, when a select has a given value.

See [drupal\_process\_states](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_process_states/7.x)() for full documentation.

**See also**

[forms\_api\_reference.html](https://api.drupal.org/api/drupal/developer%21topics%21forms_api_reference.html/7.x)

**Related topics**

[**Example: Form API**](https://api.drupal.org/api/examples/form_example%21form_example.module/group/form_example/7.x-1.x)

Examples demonstrating the Drupal Form API.

1 string reference to 'form\_example\_states\_form'

**File**

**form\_example/**[**form\_example\_states.inc**](https://api.drupal.org/api/examples/form_example%21form_example_states.inc/7.x-1.x)**, line 63**

An example of how to use the new #states Form API element, allowing dynamic form behavior with very simple setup.

**Code**

function [**form\_example\_states\_form**](https://api.drupal.org/api/examples/form_example%21form_example_states.inc/function/form_example_states_form/7.x-1.x)($form, &$form\_state) {

$form['student\_type'] = array(

'#type' => 'radios',

'#options' => array(

'high\_school' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('High School'),

'undergraduate' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Undergraduate'),

'graduate' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Graduate'),

),

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('What type of student are you?'),

);

$form['high\_school'] = array(

'#type' => 'fieldset',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('High School Information'),

// This #states rule says that the "high school" fieldset should only

// be shown if the "student\_type" form element is set to "High School".

'#states' => array(

'visible' => array(

':input[name="student\_type"]' => array('value' => 'high\_school'),

),

),

);

// High school information.

$form['high\_school']['tests\_taken'] = array(

'#type' => 'checkboxes',

'#options' => [**drupal\_map\_assoc**](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_map_assoc/7.x)(array([**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('SAT'), [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('ACT'))),

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('What standardized tests did you take?'),

// This #states rule says that this checkboxes array will be visible only

// when $form['student\_type'] is set to t('High School').

// It uses the jQuery selector :input[name=student\_type] to choose the

// element which triggers the behavior, and then defines the "High School"

// value as the one that triggers visibility.

'#states' => array(

// Action to take.

'visible' => array(

':input[name="student\_type"]' => array('value' => 'high\_school'),

),

),

);

$form['high\_school']['sat\_score'] = array(

'#type' => 'textfield',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Your SAT score:'),

'#size' => 4,

// This #states rule limits visibility to when the $form['tests\_taken']

// 'SAT' checkbox is checked."

'#states' => array(

// Action to take.

'visible' => array(

':input[name="tests\_taken[SAT]"]' => array('checked' => TRUE),

),

),

);

$form['high\_school']['act\_score'] = array(

'#type' => 'textfield',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Your ACT score:'),

'#size' => 4,

// Set this element visible if the ACT checkbox above is checked.

'#states' => array(

// Action to take.

'visible' => array(

':input[name="tests\_taken[ACT]"]' => array('checked' => TRUE),

),

),

);

// Undergrad information.

$form['undergraduate'] = array(

'#type' => 'fieldset',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Undergraduate Information'),

// This #states rule says that the "undergraduate" fieldset should only

// be shown if the "student\_type" form element is set to "Undergraduate".

'#states' => array(

'visible' => array(

':input[name="student\_type"]' => array('value' => 'undergraduate'),

),

),

);

$form['undergraduate']['how\_many\_years'] = array(

'#type' => 'select',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('How many years have you completed?'),

// The options here are integers, but since all the action here happens

// using the DOM on the client, we will have to use strings to work with

// them.

'#options' => array(

1 => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('One'),

2 => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Two'),

3 => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Three'),

4 => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Four'),

5 => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Lots'),

),

);

$form['undergraduate']['comment'] = array(

'#type' => 'item',

'#description' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)("Wow, that's a long time."),

'#states' => array(

'visible' => array(

// Note that '5' must be used here instead of the integer 5.

// The information is coming from the DOM as a string.

':input[name="how\_many\_years"]' => array('value' => '5'),

),

),

);

$form['undergraduate']['school\_name'] = array(

'#type' => 'textfield',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Your college or university:'),

);

$form['undergraduate']['school\_country'] = array(

'#type' => 'select',

'#options' => [**drupal\_map\_assoc**](https://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_map_assoc/7.x)(array([**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('UK'), [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Other'))),

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('In what country is your college or university located?'),

);

$form['undergraduate']['country\_writein'] = array(

'#type' => 'textfield',

'#size' => 20,

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Please enter the name of the country where your college or university is located.'),

// Only show this field if school\_country is set to 'Other'.

'#states' => array(

// Action to take: Make visible.

'visible' => array(

':input[name="school\_country"]' => array('value' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Other')),

),

),

);

$form['undergraduate']['thanks'] = array(

'#type' => 'item',

'#description' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Thanks for providing both your school and your country.'),

'#states' => array(

// Here visibility requires that two separate conditions be true.

'visible' => array(

':input[name="school\_country"]' => array('value' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Other')),

':input[name="country\_writein"]' => array('filled' => TRUE),

),

),

);

$form['undergraduate']['go\_away'] = array(

'#type' => 'submit',

'#value' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Done with form'),

'#states' => array(

// Here visibility requires that two separate conditions be true.

'visible' => array(

':input[name="school\_country"]' => array('value' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Other')),

':input[name="country\_writein"]' => array('filled' => TRUE),

),

),

);

// Graduate student information.

$form['graduate'] = array(

'#type' => 'fieldset',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Graduate School Information'),

// This #states rule says that the "graduate" fieldset should only

// be shown if the "student\_type" form element is set to "Graduate".

'#states' => array(

'visible' => array(

':input[name="student\_type"]' => array('value' => 'graduate'),

),

),

);

$form['graduate']['more\_info'] = array(

'#type' => 'textarea',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Please describe your graduate studies'),

);

$form['graduate']['info\_provide'] = array(

'#type' => 'checkbox',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Check here if you have provided information above'),

'#disabled' => TRUE,

'#states' => array(

// Mark this checkbox checked if the "more\_info" textarea has something

// in it, if it's 'filled'.

'checked' => array(

':input[name="more\_info"]' => array('filled' => TRUE),

),

),

);

$form['average'] = array(

'#type' => 'textfield',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Enter your average'),

// To trigger a state when the same controlling element can have more than

// one possible value, put all values in a higher-level array.

'#states' => array(

'visible' => array(

':input[name="student\_type"]' => array(

array('value' => 'high\_school'),

array('value' => 'undergraduate'),

),

),

),

);

$form['expand\_more\_info'] = array(

'#type' => 'checkbox',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Check here if you want to add more information.'),

);

$form['more\_info'] = array(

'#type' => 'fieldset',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Additional Information'),

'#collapsible' => TRUE,

'#collapsed' => TRUE,

// Expand the expand\_more\_info fieldset if the box is checked.

'#states' => array(

'expanded' => array(

':input[name="expand\_more\_info"]' => array('checked' => TRUE),

),

),

);

$form['more\_info']['feedback'] = array(

'#type' => 'textarea',

'#title' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('What do you have to say?'),

);

$form['submit'] = array(

'#type' => 'submit',

'#value' => [**t**](https://api.drupal.org/api/drupal/includes%21bootstrap.inc/function/t/7.x)('Submit your information'),

);

return $form;

}

Comments

[](https://api.drupal.org/user/253204)

[**one question**](https://api.drupal.org/comment/16959#comment-16959)

*nbie commented 6 years ago*

In case I have two select list for e.g. state and city, where city list depends on selection of a value from state list. How do I do that with #states property?

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/167299)

[**How about "#required"?**](https://api.drupal.org/comment/24863#comment-24863)

*bartl commented 6 years ago*

If one of the fields depend visibility depends on the state of another element, and its "#required" parameter is set to true, is it possible to only have it required if the condition is met?

Since this "#states" thing only seems to apply to tricks using jQuery, thus, Javascript, I don't see how the built-in form validation in Drupal can take the state state into account.

The only way I can see around it is *not* make it *really* required, but

1. add HTML to appen the Drupal red star to the field title, and
2. in your custom form-validate routine, conditionally explicitly validate that a value for this field is present.

If I'm wrong, please tell me a better way.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/167299)

[**Digging deeper, I found a**](https://api.drupal.org/comment/24868#comment-24868)

*bartl commented 6 years ago*

Digging deeper, I found a comment thread about this exact same problem, here: [http://api.drupal.org/api/drupal/includes--common.inc/function/drupal\_pr...](http://api.drupal.org/api/drupal/includes--common.inc/function/drupal_process_states/7#comment-16909)

It looks like the current state is indeed as I saw it. Which is... unfortunate. This "#states" thing is only a half solution to a fairly common problem.

I'll take a good look at the implementation of the module Conditional Fields (<http://drupal.org/project/conditional_fields>) to see if, and how, it solves this problem. (Yet, for Drupal 7, the maintainers still strongly dissuade using this module for production sites. Hence, I can't just use that module.)

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/12268)

[**Still buggy**](https://api.drupal.org/comment/52793#comment-52793)

*Kristen Pol commented 4 years ago*

I found too many bugs in Conditional Fields so had to remove it and add custom #states code.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/390173)

[**One solution for #required**](https://api.drupal.org/comment/54298#comment-54298)

*dahousecat commented 4 years ago*

The red star can be added to the field that you want to be required by setting the required property in the states array.

E.g.

'#states' => array(

'visible' => array(

':input[name=field\_name]' => array('value' => 'your value'),

),

'required' => array(

':input[name=field\_name]' => array('value' => 'your value'),

),

),

To actually perform the validation a custom validation function is required however I tried to automate the function by inspecting the #states array in the validation function so rules only have to be specific once, not twice. This does rely on the name of the element being in the jQuery selector in the format :input[name="field\_name"] or it won't work.

This code is only tested in the specific scenario that I was using it in however I though it may prove useful to someone else.

function hook\_form\_validate($form, &$form\_state) {

// check for required field specified in the states array

foreach($form as $key => $field) {

if(is\_array($field) && isset($field['#states']['required'])) {

$required = false;

$lang = $field['#language'];

foreach($field['#states']['required'] as $cond\_field\_sel => $cond\_vals) {

// look for name= in the jquery selector - if that isn't there then give up (for now)

preg\_match('/name="(.\*)"/', $cond\_field\_sel, $matches);

if(isset($matches[1])) {

// remove language from field name

$cond\_field\_name = str\_replace('[und]', '', $matches[1]);

// get value identifier (e.g. value, tid, target\_id)

$value\_ident = key($cond\_vals);

// loop over the values of the conditional field

foreach($form\_state['values'][$cond\_field\_name][$lang] as $cond\_field\_val) {

// check for a match

if($cond\_vals[$value\_ident] == $cond\_field\_val[$value\_ident]) {

// now we know this field is required

$required = true;

break 2;

}

}

}

}

if($required) {

$field\_name = $field[$lang]['#field\_name'];

$filled\_in = false;

foreach($form\_state['values'][$field\_name][$lang] as $item) {

if(array\_pop($item)) {

$filled\_in = true;

}

}

if(!$filled\_in) {

form\_set\_error($field\_name, t(':field is a required field', array(':field' => $field[$lang]['#title'])));

}

}

}

}

}

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/123574)

[**Performance**](https://api.drupal.org/comment/26223#comment-26223)

*GaëlG commented 6 years ago*

For faster rendering of big forms, use IDs in your JQuery selectors : something like '#edit-field-name' instead of ':input[name=field\_name]'.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/21602)

[**Not a good idea if using AJAX**](https://api.drupal.org/comment/33958#comment-33958)

*sokrplare commented 5 years ago*

Adding here for reference. This is per the thread over on Randy's site (<http://randyfay.com/comment/2208#comment-2208>)-

The only problem with that, of course, is that the ID must be unique on the page. And you can't be using AJAX in any of those elements (which would be strange anyway... one should generally use either #states or AJAX) because AJAX messes with the ID on every load.

In D7, the HTML ID changes on every request, due to drupal\_html\_id(). In practice, it's always the same on the initial page load, but after that, any part replaced by an AJAX action will result in an unpredictable ID (which will be different).

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/206604)

[**container id to narrow down**](https://api.drupal.org/comment/50788#comment-50788)

*japo32 commented 4 years ago*

Perhaps a better way would be to combine selectors and use the ID of the form or the container of that form (i.e. "#myform :input[name=field\_name]").

as noted in the jQuery API site:

*Because :input is a jQuery extension and not part of the CSS specification, queries using :input cannot take advantage of the performance boost provided by the native DOM querySelectorAll() method. To achieve the best performance when using :input to select elements, first select the elements using a pure CSS selector, then use .filter(":input").*

since we can't use the filter function, I'd say combining the selectors would be a good solution.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/451948)

[**Drupal made it all so easy!**](https://api.drupal.org/comment/45133#comment-45133)

*vebs0205 commented 5 years ago*

Thanks to developers for making it all so easy to use and understand. Even a non programmer can make effective use of all this to do big tasks easily!

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/603213)

[**A How to assign Terms XOR and OR?**](https://api.drupal.org/comment/55258#comment-55258)

*Catsys commented 3 years ago*

It turns out that we can only use AND. And how to implement eg the following condition:  
See input "c" as long as the input "a" or input "b" are checked = true.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/439723)

[**See here a reasonable idea on**](https://api.drupal.org/comment/58903#comment-58903)

*bucefal91 commented 3 years ago*

See here a reasonable idea on how to do it: <https://api.drupal.org/comment/24708#comment-24708> Basically you'll have to apply some discrete math transformations to make your condition consist only of ANDs.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/11614)

[**CAST VALUES TO STRING!!! I(E will not work otherwise)**](https://api.drupal.org/comment/59225#comment-59225)

*Anybody commented 2 years ago*

Very very important note: Never forget to cast values to string, as the comment above already stated out:  
<https://api.drupal.org/comment/36623#comment-36623>

Otherwise it will simply NOT work in Internet Explorer (for numeric values)!

I've written a blog article about this very dangerous bug:  
[http://julian.pustkuchen.com/en/drupal-7-fapis-states-take-care-internet...](http://julian.pustkuchen.com/en/drupal-7-fapis-states-take-care-internet-explorer-ie)

Take care!

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/614418)

[**If you want to use OR condition**](https://api.drupal.org/comment/60215#comment-60215)

*belthazorjk commented 2 years ago*

I was searching for the OR condition and it's done this way.  
Example if you have **checkboxes** with options 1, 2, 3, 4 and 5 and you want to set the visibility only when option 1 **OR** 3 is checked.

<?php

...

'#states' => array(

'visible' => array(

array( ':input[name="XXX"]' => array("checked" => TRUE) ),

array( ':input[name="XXX"]' => array("checked" => TRUE) )

)

),

...

?>

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/683943)

[**Thanks. It is working.**](https://api.drupal.org/comment/63171#comment-63171)

*5418ryadav commented 3 months ago*

Thanks. It is working.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/163574)

[**Opposite land**](https://api.drupal.org/comment/60514#comment-60514)

*leex commented 2 years ago*

Applied states also apply their opposite, which is undocumented and unexpected.

For example, if you have 2 checkboxes, checkboxA and checkboxB and checkboxB should be unchecked when checkboxA is checked, checkboxB will also become checked when checkboxA is unchecked. Very frustrating.

See: <http://drupal.stackexchange.com/a/136964/3936>

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/1836)

[**.**](https://api.drupal.org/comment/61167#comment-61167)

*martin\_q commented 2 years ago*

Deleted. Issue 2680529 filed instead.

* [Log in](https://api.drupal.org/user/login?destination=node/185104%23comment-form) or [register](https://api.drupal.org/user/register?destination=node/185104%23comment-form) to post comments

[](https://api.drupal.org/user/578568)

[**how to get same field name working?**](https://api.drupal.org/comment/61203#comment-61203)

*dzy commented 2 years ago*

$form['high\_school\_one']['sat\_score'] = array(

............

);

$form['high\_school\_two']['sat\_score'] = array(

............

);

there two sat\_score filelds, not working like this

'#states' => array(

'visible' => array(

':input[name=high\_school\_two[sat\_score]]' => array('value' => 'your value'),

),