

The ValidateAndSubmit 1.1 action.

Features:

The ValidateAndSubmit action offers a number of unique options like:

- The ValidateAndSubmit action validates text fields, password fields, radio button groups and popup lists.
- Custom error messages. Use your own words and/or your own language.
- Form elements with errors are color marked for easy detection.
- Use any color you like, or no color at all.
- The user will get an alert message with a list of errors and an indication of what caused them.
- The ValidateAndSubmit action uses regular expressions to perform tests on e-mail addresses.
- The ValidateAndSubmit action searches the form for empty fields.
- The ValidateAndSubmit action performs tests on phone numbers.
- Use lists of fields that you want to have checked. One field can be in several lists.
- The form is validated up on submission, a method that is much safer, and less prone to errors, than runtime validation.

Important:

Do not use a submit button with the ValidateAndSubmit action. The submit button has a built-in submission function that will interfere with the action. If you want to use GoLive's submit button, highlight the button and uncheck "submit" (set it to "normal"). You can use any type of graphic or text link to trigger submission. Please read the information about how the validation is done, it will help you avoid a lot of problems!! **Do not validate "List boxes"!** Read more about the list box problem under: "The validation:"

Form set up:

Create your form and give it a name. Give all the fields names. Remember that the names you use for the fields will be displayed in the error messages. Use easy to understand names like PhoneNumber, Name, etc. **Never use special characters or spaces in the field names.** This goes for all actions and all JavaScript applications.

Action set up:

If you have a large form, start by making a list over the fields and what validation groups they should be in.

Connect the action to the link (graphic or text) that you will be using to trigger the submission. Enter the fields names as a comma separated list, like this: Name,Phone,Address,Email, etc.

No spaces!

Add the fields you want for "empty test" to the "empty list" box, the phone number fields you want to test to the "phone #"list" box, the e-mail fields you want to test to the "e-mail list" box, the radio button groups you want to test to the "radio list" and the popup lists you want to check to the "popup list".

Check the color marking option if you want to change the background color for fields with errors. Then drag a color from the color palette and dropt it in the color box. **Colors are not displayed in Netscape 4 and Opera.**

If you want to use custom text messages fill in the last six fields: "Header error msg", "Empty error msg", "E-mail error msg", "Phone error msg", "Radio error msg", and "Popup error msg". The name of the fields with errors will automatically be placed **after** your text. If no text is entered the default English messages will be used.

Troubleshooting:

It is vital that the names of your fields (and the form) are correctly spelled. JavaScript is case sensitive so watch out for typos! If your form submits even if it hasn't passed the validation, highlight your submit button and uncheck the submit check box in the inspector. Set it to "normal".

The validation:

Empty:

Can't be simpler, if the field is empty an error message will be generated.

E-mail:

The address is searched for illegal characters, spaces and other typos but the validation is mostly based on the "structure" as a huge number of special characters are legal, even if we hardly ever see them in an e-mail address. For more information visit:

<http://www.w3.org/Protocols/rfc822/Overview.html#z8>

Radio buttons:

If none of the radio buttons in a radio button group is selected an error message will be displayed. A radio button group is a number of radio buttons that shares the same name.

Select lists:

First let me explain the difference between "popup lists" and "list boxes". Popup lists display only one item by default (before you click it), that options is also the default option (# 0). List boxes display a number of options (>1), there is no default alternative (# -1).

Popup lists works very well with this action. List boxes on the other hand give the coder huge problems. In NN4 you can't detect the difference between the two types, Opera goes against all other brands and uses the first option in the list box as default, and NN6 crash if you select an item from a list box with a set background color! **Due to those facts I have decided not to support list boxes.**

There is one important thing to think of when you set up a popup list. **Use the first option as a "header" with a text like "Pick a color"**. The next item will be the first actual choice you can make. Why? The answer is that the only way to find out if a user has made a selection is to detect if the current selection is the first item or not. If the first item is still active no selection has been made!

A few words from the author:

I have put in a lot of work to make this action as flexible as possible, and not too hard to understand. This action can't compete with tailor made scripts written for a specific form, but it should supply you with a lot of help for "standard" forms, and save you a lot of time! If you're not familiar with JavaScript this action will enable you to do things that would have been impossible without the action. I hope you will have a lot of use for it!

There is a special "Action Forum" at <http://www.ahgrens-dental.se/cgi-bin/yabb/YaBB.pl> where you can post questions, feature requests etc.

More Ahgren Actions can be found at <http://www.ahgrens-dental.se/pages/actions/>

/michael åhgren

Updates in version 1.1:

The email validation now accepts the new top domain names (names containing more than 3 characters) and the following examples will now pass validation:

myName@mySite.info
myName@mySite.name