

Form Validation Contracts

The Form Validation Contracts are used to validate data entered by a user in an edit box. For instance, if you have a field that is supposed to be a US phone number and the user enters his or her name, a warning message will appear informing the user of the error.

The contracts that specify how to validate a control don't actually do anything by themselves. They must be combined with one of the contracts that execute the validations that have been applied. Most typically this is the contract "Validate when [source] is submitted" applied to the Form element in the Drumbeat basement. When the form is submitted the applied validations will execute. The form can be submitted either by a form button or another element with the contract "Submit [target] when [source] is clicked" where [target] is the form element in the Drumbeat basement.

You can enhance form validation by validating a particular control when the control loses focus. This is done with the activation "Validate [source] when it loses focus". This contract is only available for IE 4.0.

For purposes of this discussion there are two cases to consider: A simple form and a complex form.

Simple Form

A simple form is any form where there is a single function performed when the form is submitted. There is usually a single submit button or an image or text that you want a click event to submit the form. In some cases you may want one of the other submit contracts such as submit with the selection changes in a list box. These are all simple forms since the form performs a single function.

To construct a simple form start by applying validation contracts to controls on your page that you want to validate to specific types of data (for example, date, integer, etc.). Then apply the activation "Validate when [source] is submitted" to the form element in the Drumbeat basement. Finally place a submit button on the page.

If you want to submit the form with an image, then place an image on the page (or image button) and apply the interaction "Submit [target] when [source] is clicked" between the image and the form element. Note that there is a parameter on this contract named "Validate form". You should uncheck this since you applied form validation to the form element. It is equivalent to check this parameter and remove the validation contract from the form element. In either case form validations will be executed when the form is submitted.

Complex Form

A complex form is any form where there is more than one way to submit the form and where the various submits do different things. The common example of this is data drive pages. On the Insert page generated by the Drumbeat Data Form Wizard (DFW) there is an insert button and a back button. Form validation is well suited to inserting data to the database via the DFW insert page. However, when clicking the back button you do not want the form to validate as the user is

indicating that an insert should not occur, and there is no reason to validate the form controls. This is a case of two submit buttons where you want two different things to happen with respect to form validation.

On this type of form do not apply the contract “Validate when [source] is submitted” as this will execute validation contracts for every submit. Use the contract “Submit [target] when [source] is clicked” on the controls that you want form validation to occur on. In the example above, the Insert button should have this contract applied, and the Back button should not.

Contract Descriptions:

Most of the Form Validation contracts have these parameters in common:

1. **Required:** If this box is checked, the user will not be allowed to submit the form until something has been entered in this edit box. **Note:** Entering only spaces and/or carriage returns to try to get around the **Required** validation will not work.
2. **Error Msg:** Some of the contracts allow you to enter your own Error Message that is displayed when the user enters invalid data. If this parameter is left blank, a default error message is used.

Below is a description of each validation contract along with explanations of their specific parameters as well as some examples.

- **Email Address**

Valid if the value contains the “@” character.

- **Date**

Valid if the user enters a date using one of the following formats:

5/3/97

May 3, 1997

3 May 1997

Note: Any date that is recognizable by the JavaScript Date() function will be accepted.

- **Integer**

Valid if all the characters are digits. Leading spaces and zeros are removed. Trailing spaces are removed.

- **Credit Card Number**

Ignores spaces, dashes, periods, and carriage returns. Valid if all remaining characters are digits and the entire number passes a Mod 10 check (a mathematical formula where the value of the final digit is known when given the first part of the number)

- **Custom**

This contract allows you to validate the data by comparing it to a format that you supply.

Parameters:

1. **Format:** The format of the data you want the user to enter.
2. **Number Char:** The character in the Format that represents a digit ("0"- "9"). Default char: #
3. **Letter Char:** The character in the Format that represents a letter ("a"- "z" or "A"- "Z"). Default char: A
4. **AlphaNum Char:** The character in the Format that represents either a letter or number ("a"- "z" or "A"- "Z" or "0"- "9"). Default char: ?

For instance, if you want the field to be a California Driver's license, you would supply the value A##### in the Format parameter. This formatted string means that the value has to be one letter followed by seven digits. **Note:** Leading and trailing spaces are ignored.

Any character in the Format that is not one of the special characters above, needs to be entered exactly by the user.

Suppose you want the user to enter a US Phone Number. You could supply the Format (###)###-####. A valid entry would be "(101)555-1212". Notice how the parentheses and the dash must entered. An invalid entry would be "101 555 1212".

But, what if you want the field to be an apartment number such as #232? You can decide which characters are *special characters* that represent digits and letters in the Format. If you change the special character for digits to be, let's say "*", then the appropriate format for the apartment number would be #***. That is, a pound sign followed by three digits.

▪ **Entry Length**

You can put limitations on the number of characters entered by the user.

Parameters:

1. **Min Length:** The minimum length the value should have. A minimum length less than 1 is ignored.
2. **Max Length:** The maximum length the value should have. A maximum length less than 1 is ignored.
3. **Strip spaces:** If this box is checked, spaces and carriage returns are stripped out of the value before its length is measured.

▪ **Fixed Point**

Limits the number of decimal places a user can enter.

Parameters:

1. **Decimal Places:** The maximum number of decimal places allowed.
2. **Min:** The minimum value allowed.
3. **Max:** The maximum value allowed.

This contract is useful if you want validate a money value by setting the Decimal Places parameter to 2.

- **Floating Point**

Valid if the user enters a floating point value. A floating point value is a number with a decimal place such as 1.23 or 0.1. Any floating point number that is recognized by the JavaScript function `parseFloat()` is valid.

- **International Phone Number**

Ignores spaces, pluses, dashes, carriage returns, parentheses, slashes, underscores, and asterisks. Valid if all remaining characters are digits and there are at least 6 digits.

- **Non Blank**

Ignores spaces and carriage returns. Valid if there are any remaining characters.

- **Social Security Number**

Ignores spaces, dashes and periods. Valid if the remaining characters are 9 digits. The value will be reformatted to `###-##-####`.

- **Time**

Accepts common time values such as:

13:30:00

13:30

1:30 pm

Note: Any time that is recognizable by the JavaScript `Date()` function will be accepted.

- **US Phone Number**

Ignores spaces, dashes, carriage returns, parentheses, slashes, underscores, and asterisks. Valid if all remaining characters are 7 digits, 10 digits, or 10 digits preceded by a “1”. The value will be reformatted to one of the following:

1(###) ###-####

(###) ###-####

###-####

- **Zip Code**

Ignores dashes, spaces, and carriage returns. Valid if remaining characters are 5 digits or 9 digits. The value will be reformatted to `#####-####` or `#####`.

- **URL**

Accepts a string that begins with all of the protocols listed in the Drumbeat link dialog, such as `http://`, `ftp://`, etc.