

# Web Forms

---

The page editor in Cascade Server does not include tools for creating web forms, so they must be created directly in the HTML code. If you need to create a web form as part of your site in Cascade Server, the best thing to do is to look at some examples and find one that's close to what you want. For a list of examples, log in to mySNC and in the lower left corner under "Web Site:" – "Forms;," click on "Examples."

## The form creation process

The order of these steps is important because the form and its "thank you" page (the page people see after they submit the form) must be published to the live web server before setting up the configuration.

1. Create the form web page in Cascade and publish it. At this point, it won't have a form id, but that's okay; you'll go back and add that later. Stay logged in to Cascade.
2. Create the form's thank you page in Cascade and publish it. Stay logged in to Cascade.
3. Use the Form Configurator to define how your form is handled (details below). When you save the configuration, you will be issued a line of code containing a new formid, e.g., `<input name="cfg_formid" type="hidden" value="mydepartment-1" />`. Highlight and copy that entire line of code so that you can paste it into your form.
4. Go back to Cascade and edit the form again. Go into HTML mode. Locate the code `<form action="/cgi-bin/sncform2.pl" method="post">`. Right after that line, paste the form id code you copied from the Configurator in the previous step, then update, save, and publish the form again.
5. Test the form. Fill it out to make sure it works. Look at the form recipient's e-mail to see the results.

## The Form Configurator

When a form is submitted on the web site, a program on the server receives the form input and sends it as an e-mail message to someone. The Form Configurator is the tool you use to create a configuration file for each form which tells the form-handling program how to process the form data. This program can also do extra things, like make sure certain fields are filled in or make sure an e-mail address is properly formatted. For complete Form Configurator instructions, log in to mySNC and in the lower left corner under "Web Site:" – "Forms;," click on Configurator, then click on "Documentation" at the top of the page.

## Form Configurator Prompts

### Site

This refers to the web content area where the form will be used, and determines who will have access to modify the form configuration. For example:

futurestudents

### Page Name

This is how the page is referred to when error messages are displayed (e.g., if a visitor leaves a required field blank). If a visitor fills out the form properly, the page name is not used. For example:

Alumni Contact Form

### Form URL

This is the address of the web page that contains the form. It is used by the program internally for verification purposes, so it must already exist on the live web site. For example:

<http://www.snc.edu/webteam/testing/askstudent.html>

### Thank You URL

This is the address of the web page where the visitor will be directed to after they have filled out the form. For internal validation purposes, this page must already exist on the live web site. This web page should thank the visitor and tell them what to expect next. Statements like "we will contact you within 24 hours" or "your event will appear in the next issue" are appropriate. For example:

<http://www.snc.edu/webteam/testing/askstudent-thankyou.html>

### **Required Field Names**

This is a comma delimited list of fields in your form that require an entry. If these fields are left blank by the visitor, an error message will appear. For example:

name,phone,email

### **Required Field Labels**

This is a comma delimited list of field labels that correspond one to one to the Required Field Names listed above. These labels are used when displaying error messages. For example:

Name,Telephone Number,E-mail

### **Validate E-mail**

This is a comma delimited list of fields on the form that must contain a valid e-mail address. The form-handling program will check that text entered into these fields is formatted as a valid e-mail address. If the values of these fields are not valid e-mail addresses, an error message will appear. For example:

email,workemail

### **From**

This is the e-mail address that the e-mails will come from. You cannot change this. All e-mails come from webform@snc.edu. You may want to setup a filter in your e-mail program to put all messages received from this address into their own folder.

### **Reply To**

This is the e-mail address your messages will go to if you reply to a form submission e-mail. Typically, you would want this to be the e-mail address of the person who filled out the form. Because you don't know that e-mail address in advance, you can't specify it, but you can tell the program to substitute the value of a field on your form (what the user entered in a given field). For example:

%email\_addr%

This tells the program to use the contents of the 'email\_addr' form field as the reply-to e-mail address.

### **To**

This is the e-mail address that will receive all e-mails from the form submission. It is suggested that you use a department e-mail address here, so that form submissions can be handled even if the usual person is on vacation or sick. For example:

mydept@snc.edu

To indicate multiple recipients, enter a comma-separated list like this:

person.one@snc.edu,person.two@snc.edu

### **Cc**

This is an e-mail address that will receive a copy of all e-mail form submissions. For example:

other.person@snc.edu

### **Subject**

This is the subject of the e-mail form submissions. For example:

Contact Web Form

### **Secure data transmission**

If this is turned on, all form data will be encrypted and stored on the server. An e-mail notification will be sent (containing no form data), allowing the user to click a link to view the form submission data securely. This feature is for forms that contain secure data which cannot be e-mailed, like credit card numbers.

### **Include geeky data**

If this is turned on, the e-mail will include the values of certain environment variables which may (or may not) be helpful. Variables included are the visitor's web browser name and version, the language setting from the visitor's computer, and the visitor's IP address.

## Securing/restricting forms

As with all SNC web files, any page in a “login” folder requires the user to log in with a valid SNC login ID and password, anything in a “loginfs” directory requires the user to log in with a valid SNC faculty or staff login ID and password, and anything in a “loginstu” directory requires the user to log in with a valid SNC student login ID and password. If a form is in a login, loginfs, or loginstu folder, the information about the person who was logged in to the web site when the form was submitted will be sent with the form data.

Anything in a “secure” folder encrypts transmitted data. This should be used when requesting confidential information via the web (e.g., credit card numbers). Secure web addresses start with “https.”

For more about special folder names, see <http://www.snc.edu/techsupport/web/filenaming.html#special>. For instructions on other advanced form features, see <http://www.snc.edu/techsupport/web/forms.html>.

## Advanced form features

For information about advanced form features and fields, such as automatic field completion, credit card validation, year, month, and number lists, country and state/province lists, and file upload fields, see the Web Forms page at <https://www.snc.edu/techsupport/web/login/forms.html> (login required).

## HTML code for forms

Most of the HTML code shown below is in ALL CAPS to make it easier to read. It does not actually matter whether it's typed in upper or lower case; Cascade Server will change it to all lower case anyway.

### Form action and form id fields

The following two lines of code must be at the start of every form on the SNC web server:

```
<form action="/cgi-bin/sncform2.pl" method="post">
  <input name="cfg_formid" type="hidden" value="formid-#"/>
```

where **formid-#** is replaced by the actual form id from your form configuration. When you use the Form Configurator to create a configuration file, you'll be given the line of code to copy and paste into your form.

### Text boxes

A *text box* is a single-line text box where the user may type text.

e.g., `<INPUT NAME="address" TYPE="text" SIZE="30" MAXLENGTH="50">`

- **TYPE**: Defines the field as a text box.
- **NAME**: The name of the field. This name is appears in the e-mailed results and is used to make the field required. The name should be clear and descriptive, and must be unique within the form. The name should not contain any spaces or special characters, but may contain upper case letters if needed.
- **SIZE**: This is the width of the text box in number of characters.
- **MAXLENGTH** (optional): This is the maximum number of characters that can be entered into the box. Without this attribute, there is no limit. If this number is larger than the **SIZE** attribute, the contents of the box will scroll to allow the user to enter more data than can be seen at one time.

### Check boxes

*Checkboxes* are square boxes that can be checked and unchecked. More than one in a given set may be selected at a time.

e.g., `<INPUT NAME="FavoriteFood" TYPE="checkbox" VALUE="chocolate">chocolate  
<INPUT NAME="FavoriteFood" TYPE="checkbox" VALUE="pizza">pizza  
<INPUT NAME="FavoriteFood" TYPE="checkbox" VALUE="icecream">ice cream`

- **TYPE**: Defines the field as a checkbox.
- **NAME**: The name of this set of checkboxes. See discussion above under Text boxes.
- **VALUE**: The value returned in the e-mail if this box is checked.
- **CHECKED** (optional): This specifies that the box should be checked by default.

## Drop-down menus

A drop-down menu, where users can choose from a list of options, is called a *select list*.

```
e.g. , <SELECT NAME="semester">
      <OPTION SELECTED>Choose One...
      <OPTION VALUE="Fall107">Fall 2007
      <OPTION VALUE="Spring08">Spring 2008
      </SELECT>
```

Attributes for the SELECT tag:

- **NAME**: The name of the field. See discussion above under Text boxes.
- **SIZE** (optional): The number of options visible, or how long the list is, before it's clicked on. (default=1)
- **MULTIPLE** (optional): This allows users to select more than one item at a time from the list by holding down the Ctrl key when clicking.

Attributes for the OPTION tag:

- **VALUE**: The value returned in the e-mail if this option is selected.
- **SELECTED** (optional): This specifies the default option (without this, the first item will be selected).

Note that the select list must be ended with a </SELECT> tag.

## Radio buttons

*Radio buttons* are round buttons. Only one in a given set may be selected at a time.

```
e.g. , <INPUT NAME="OperatingSystem" TYPE="radio" VALUE="WindowsXP">Windows XP
      <INPUT NAME="OperatingSystem" TYPE="radio" VALUE="WindowsVista">Windows Vista
      <INPUT NAME="OperatingSystem" TYPE="radio" VALUE="MacOSX">Macintosh OS X
```

- **TYPE**: Defines the field as a radio button.
- **NAME**: The name of the set of buttons. See discussion above under Text boxes.
- **VALUE**: The value returned in the e-mail if this button is selected.
- **CHECKED** (optional): This specifies that the button should be selected by default.

## Large text boxes/textareas

A large text box, longer than one line, is called a *textarea*. Users can type large blocks of text.

```
e.g. , <TEXTAREA NAME="comments" ROWS="6" COLS="50" WRAP="physical"></TEXTAREA>
```

- **NAME**: The name of the field. See discussion above under Text boxes.
- **ROWS**: The number of rows of text that the box can display at one time, or its height in lines.
- **COLS**: The number of columns of text that the box can display at one time, or its width in characters.
- **WRAP**: Specifies that the text should wrap at the edge of the box ("physical" option recommended).

## Submit and Reset buttons

The submit button sends the form; the reset button resets the form to the default values. These go at the end of the form.

```
e.g. , <INPUT TYPE="submit" VALUE="Submit">
      <INPUT TYPE="reset" VALUE="Reset">
```

- **TYPE**: Defines the button as either a submit or a reset button and determines what happens when it's clicked.
- **VALUE**: The text that appears on the actual button. "Submit" and "Reset" are recommended.

## Ending the form

At the end of the form, you must close the form with a </FORM> tag.

## Assistance and questions

If you have any questions or need more information, contact the Help Desk at (920) 403-HELP (4357) or [helpdesk@snc.edu](mailto:helpdesk@snc.edu). The Help Desk is for St. Norbert College students and employees only.