Cultiv Razor Contact Form – Documentation

A simple contact form, written in Razor. The Razor file is structured so that you can easily modify the existing fields or add new ones.

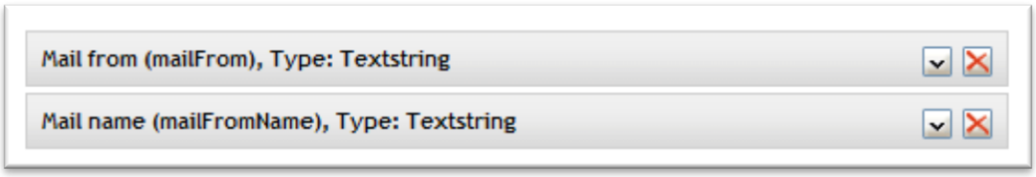
Basic usage

To put the contact form on your template, add this Umbraco macro:

```
<umbraco:Macro FileLocation="~/macroScripts/CultivContactForm.cshtml"
MailFrom="hi@test.com" runat="server" />
```

This is the simplest way to start and add the form to your site. It will give you a full contact form with validation. The MailFrom parameter is required, this is the e-mail address that will appear in the "from" field of the e-mails that are being sent out.
Additionally, you may add a parameter called MailFromName to specify the name that will be shown in the "from" field.

You could give control over the e-mail address and name that appears in the "from" field by adding them as properties your document type instead of in the macro parameters. If you do that, you can safely remove both the "MailFrom" and "MailFromName" macro parameters.
The document type properties should have the same alias: mailFrom and mailFromName:



Mail from (mailFrom), Type: Textstring

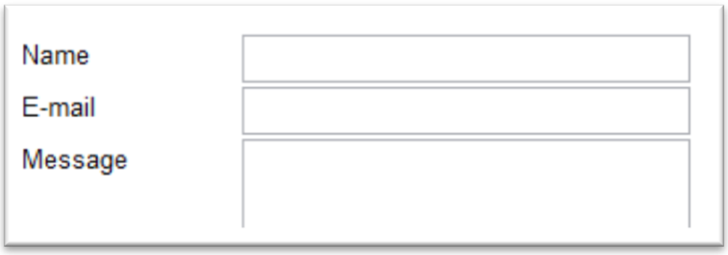
Mail name (mailFromName), Type: Textstring

Please note that without a valid, filled in, "mailFrom" address (either in the macro parameters or as a property on the document type), there will be no mail sent and any entries will be lost.

Change or translate text labels

If you use the macro as it is out of the box, it will give you default text labels, but you might want to overrule them, especially if you need them to be in a different language. You can do that with macro parameters, example:

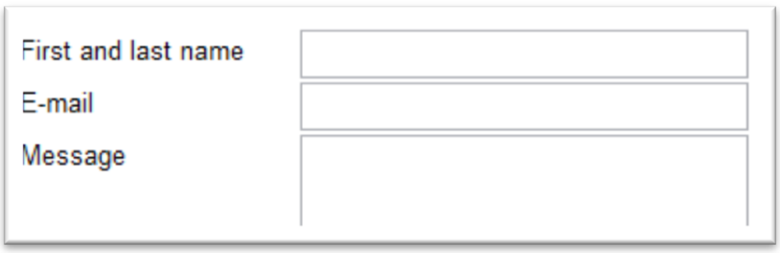
This is what the form might look like by default:



You can add a macro parameter, like this:

```
<umbraco:Macro FileLocation="~/macroScripts/CultivContactForm.cshtml" MailFrom="hi@test.com"
FormLabelName="First and last name" runat="server" />
```

As you can see here the "Name" label has been change. The form will pick this up and look like this now:



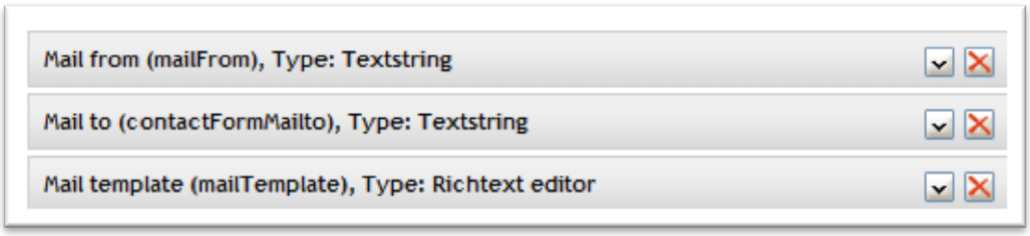
Here is a list of macro parameters that you can add.

- **MailFrom** – Used in the "from" field of the e-mail, also used to send the e-mail to.
- **MailFromName** – Used as the name from the MailFrom field.
- **FormLabelName** – The "name" label in the form.
- **FormLabelEmail** – The "e-mail" label in the form.
- **FormLabelMessage** – The "message" label in the form.
- **FormLabelSendCopy** – The "send me a copy of this mail" label in the form.
- **FormValidationError** – A general error message when validation fails.
- **FormGenericError** – A generic error message when something in the code fails.
- **MailSubject** – The subject of the contact mail.
- **MailIntroText** – An introductionary text for the contact mail.
- **MailSubjectCopyToSender** – The subject of the copy of the contact mail that is sent to the sender.
- **MailIntroTextCopyToSender** – An introductionary text for the copy of the contact mail that is sent to the sender.
- **FormSentConfirmation** – Confirmation message that the form was sent successfully.
- **RedirectUrl** – Specify a URL when you want to redirect to a different page after the form was sent.
- **EnableSsl** – If your SMTP server requires an SSL connection, specify either "true" or "1".
- **NameFieldError** – Error to be shown near the "name" field when there's a validation error.
- **EmailFieldError** – Error to be shown near the "e-mail" field when there's a validation error.
- **MessageFieldError** – Error to be shown near the "message" field when there's a validation error.

Tip: for any of these parameters you can also insert a dictionary item, like so: FormLabelName="[! labelName]" (where the dictionary key is: labelName).

Mail template

There is a default template that's included in the Razor file, which you can change however you want. It might be easier, though, to do that in a Rich text editor (or Simple Editor or Textbox Multiple or any data type that holds text). To do so, just add a new property to your document type called mailTemplate.



Mail from (mailFrom), Type: Textstring

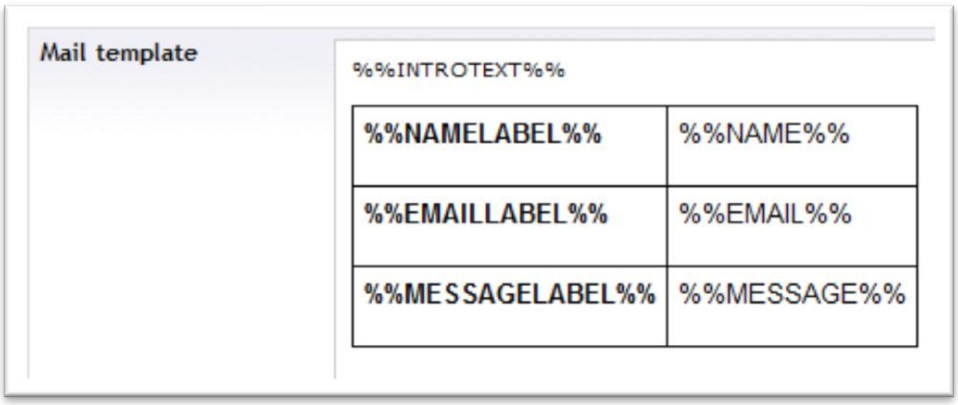
Mail to (contactFormMailto), Type: Textstring

Mail template (mailTemplate), Type: Richtext editor

By default, the following strings will be replaced with the values on the form:

- **%%INTROTEXT%%** - The intro text for the e-mail.
- **%%NAMELABEL%%** - The form label for "name".
- **%%EMAILLABEL%%** - The form label for "e-mail".
- **%%MESSAGELABEL%%** - The form label for "message".
- **%%NAME%%** - The name that was filled in.
- **%%EMAIL%%** - The e-mail address that was filled in.
- **%%MESSAGE%%** - The message that was entered.

Your template could look a little like this, for example:



Mail template

%%INTROTEXT%%

%%NAMELABEL%%	%%NAME%%
%%EMAILLABEL%%	%%EMAIL%%
%%MESSAGELABEL%%	%%MESSAGE%%

Please note that markup can appear very differently in various e-mail programs.

Changing and extending

You're free to change any part of the form as you wish.

Changing the HTML output is easily done by just changing the tags. Note that the classname for the form fields is controlled by calling the Render***Field helper (the last parameter).

If you want to add fields, you need to do that in a three places:

- RenderContactForm() – Add a call to a Render***Field helper.
    - If the field is required, the fieldname should end with "-req", for example:
      `@RenderTextField("Last name", "lastname-req", "Enter last name", "text")`
- GetMailContent() –
    - Add the extra field at the top, for example:
      ```
      var lastname = HttpUtility.HtmlEncode(HttpContext.Current.Request["lastname-
      req"]);
      ```
    - Add the field to the default mailContent template and/or to the mailTemplate replace strings under there.

You can do much more, like build in your own custom validation and so on, it's all up to you.

Final thoughts

Tip: If you have WebForms forms on your page, make sure that the form and it's field have a ValidationGroup defined, that way, you should not get conflicts between two forms.

Tip: To test locally, without having to set up a mail server change your SMTP server in the web.config to this:

```
<system.net>
  <mailSettings>
    <smtp deliveryMethod="SpecifiedPickupDirectory" from="">
      <network host="localhost" />
      <specifiedPickupDirectory pickupDirectoryLocation="C:\inetpub\mailroot" />
    </smtp>
  </mailSettings>
</system.net>
```

Make sure that the Application pool user (typically NETWORK SERVICE or ApplicationPoolIdentity) has write access to the specified pickupDirectoryLocation.

Tip: You can relay mail through Gmail if you don't have or don't want to set up an SMTP server, make sure to set the macro parameter EnableSsl to "true" if you do that. You can add this into your web.config file (replace the account name and password of course):

```
<system.net>
  <mailSettings>
    <smtp>
      <network
          host="smtp.gmail.com"
          port="587"
          userName="yourGmailAccount"
          password="yourGmailPassword"
          defaultCredentials="false"
        />
    </smtp>
  </mailSettings>
</system.net>
```

Note: All of the validation is done on the server side for now, but it wouldn't be too difficult to add client-side validation as well. If you're adding client-side validation, please consider giving it back to the community; Just request collaboration on the package page, thanks!