

There are two basic ways to change or extend the functionality of existing [web parts](#):

- [Customize the web part code](#) - edit the ASPX markup or code behind of the user control that implements the web part.
- [Use macro expressions in web part properties](#) - by inserting [macro expressions](#) into the values of web part properties, you can implement many types of custom functionality without modifying the web part code.



Important: We do not recommend customizing the code of the default web parts directly. Instead, create a *clone* of the required web part and implement the changes there. Using clones ensures that the system does not overwrite your code when applying hotfixes or upgrades to newer versions of Kentico.

Alternatively, you can create:

- [Inherited web parts](#) - copies of existing web parts that share the code files, but allow you to modify the definitions of [properties](#) and their default values.
- [Custom web part layouts](#) - layouts determine the appearance and design of web parts (allow you to edit or extend the web part's ASPX markup). You can select a different layout for each web part instance.

Example - Customizing the On-line form web part


The following example creates a modified version of the **On-line form** web part. The customized web part sends an email whenever visitors submit the form and displays a confirmation message. You can use the same approach to alter the functionality of any of the default web parts to fit your specific requirements.



Note: This example is only a demonstration of the web part customization process. [Forms](#) provide built-in functionality for setting up email notifications, autoresponders and confirmation messages.

Cloning the original web part

Before customizing one of the web parts in Kentico, it is recommended to first create a clone. Working with a separate copy of the original web part allows you to use the default version as a reference point and backup.

1. Open the **Web parts** application.
2. Select the **Forms -> On-line form** web part in the tree.
3. Click **Clone web part** () above the tree and enter the following values:
 - **New object display name:** Form with custom email
 - **New object code name:** FormWithEmail
 - **Clone web part to category:** Forms
 - **Clone web part files:** yes (checked)
 - **Cloned web part file name:** BizForms/formwithemail.ascx
4. Click **Clone**.

The system creates a copy of the existing web part and its code files (ASCX and CS) using the specified names.

Modifying the web part code

1. Open your web project in Visual Studio using the **WebSite.sln** (or **WebApp.sln**) file.
2. Edit `~/CMSWebParts/BizForms/formwithemail.ascx`.



Important

If you installed Kentico as a web application, the clone's code files will not be visible right away, since they are not included in the project. In this case:

- a. Click **Show all files** at the top of the Solution Explorer.
- b. Right-click the **formwithemail.ascx** file in the BizForms folder.
- c. Click **Include in Project**.

3. Drag a **Label** control onto the form and place it under the *BizForm* control.
4. Set the label control's **ID** to *lblConfirmationMessage* and clear its **Text** property.

```
<cms:BizForm ID="viewBiz" runat="server" IsLiveSite="true" />
<asp:Label ID="lblConfirmationMessage" runat="server"></asp:Label>
```

5. Edit the code behind file (*formwithemail.ascx.cs*) and insert the following code underneath the default content in the **viewBiz_OnAfterSave** handler:

```
// Creates a new e-mail message
CMS.EmailEngine.EmailMessage msg = new CMS.EmailEngine.EmailMessage();

msg.From = "mail@localhost.local"; // Enter any valid e-mail address
msg.Recipients = "mail@localhost.local"; // Use a valid e-mail address that you
can access
msg.Subject = "Custom form e-mail";
msg.Body = "The value of the FirstName field: "
          + CMS.Helpers.ValidationHelper.GetString(viewBiz.GetDataValue
("FirstName"), "N/A");

// Sends out the custom e-mail notification
CMS.EmailEngine.EmailSender.SendEmail(msg);

// Sets the confirmation message shown in the label
lblConfirmationMessage.Text = "The e-mail has been sent.";
```



The web part executes the **viewBiz_OnAfterSave** handler method whenever a user saves the form.

The custom code creates a new email message, sends it out, and adds text into the confirmation label. The content of the email dynamically retrieves a field value from the form using the **GetDataValue(string fieldName)** method of the BizForm control.

6. Fill in valid email addresses into the **From** and **Recipients** properties of the **EmailMessage** object in the code.
7. Save the web part's source files.
8. **Build** the project if it was installed as a web application.

The customized form web part is now ready.

Result

You can try out the functionality of the custom web part by adding it to a page, for example on the sample *Corporate site*:



Note: You need to have a valid [SMTP server](#) set up in the system to send the email.

1. Log in to the Kentico administration interface and open the **Pages** application.

2. Select a page in the content tree.
3. Switch to the **Design** tab and [add](#) the **Form with custom email** web part.
4. Set the web part's **Form name** property to the **Contact Us** form (using the **Select** button).
 - The *Contact Us* form is available by default if you are working with the sample Corporate Site.
 - This form includes the **FirstName** field used in the custom code.
5. Click **OK**.
6. View the page on the live site.
7. Enter some values into the form and submit it.

The web part displays the additional confirmation message ("The email has been sent.") and sends an email message to the specified address.

Example - Using a WHERE condition macro

The following example shows how to dynamically set the **WHERE condition** of a Repeater web part through a [macro expression](#). The condition causes the repeater to display a full list of news pages for logged in users, and a limited sub-set for public users (unauthenticated visitors).

The example is designed to work on the sample Corporate site. If you are using it on your site, you will have to perform some of the steps, such as choosing the transformations, according to how your site is set up.

Adding the custom News page field

The example uses a custom field for news pages to determine whether news items are visible for public users.

1. Open the **Page types** application.
2. Edit (✎) the **News** page type.
3. Select the **Fields** tab.
4. Click **New field**.
5. Define the new field:
 - **Field type**: Standard field
 - **Field name**: ShowToPublicUsers
 - **Data type**: Boolean (Yes/No)
 - **Field caption**: Show to public users
 - **Form control**: Check box
6. Click **Save**.

You can now set the **Show to public users** field on the **Form** tab of *News* pages. Switch to the **Pages** application and enable the flag for some of the pages in the **/News** section of the website.

Setting the dynamic WHERE condition

1. In the **Pages** application, [create a new page](#) in the website's content tree.
2. Open the **Design** tab and add a **Repeater** web part.
3. Set the following properties for the Repeater:

- **Path**: /News/%
- **Page types**: CMS.News
- **WHERE condition**:

```
{% CurrentUser.IsAuthenticated ? " " : "ShowToPublicUsers = 1" %}
```

- **Transformation**: cms.news.preview
4. Click **OK**.

The repeater displays all news pages from the */News* section of the website. If you log out and view the page as a public visitor, the repeater only displays the news pages that you marked with the **Show to public users** flag.



Macro details

The macro in this example dynamically sets the **WHERE condition** property of the Repeater web part. The system resolves the macro when rendering the page.

- The *CurrentUser.IsAuthenticated* expression is true if the user viewing the page is logged in on the website under a registered account.
- The ternary operator (*<condition> ? <true result> : <false result>*) returns:
 - An empty WHERE condition value if the user is authenticated.
 - *ShowToPublicUsers = 1* if the user is public (i.e. the repeater loads and displays only pages whose **Show to public users** field is checked).