

Custom fields may help you store and organize additional data about your contacts when you, for example, integrate Kentico with a third party system. If you need to gather data about contacts that is not collected by the default fields, you can add the following types of custom fields to contacts:

- [Static fields](#)
- [Reference fields](#)


Added custom fields can be displayed among other editable fields of the [contact editing interface](#) or on the separate **Custom data** tab of the contact managing interface.

Adding static custom fields to contacts

Static custom fields may store values such as text, date and time, and the boolean value. To define static custom fields:

1. Open the **Modules** application.
2. **Edit** (✎) the **Contact management** module.
3. Switch to the **Classes** tab.
4. **Edit** (✎) the **Contact management - Contact** class.
5. Switch to the **Fields** tab.
6. [Create a new category](#) below the last field in the fields listing (*ContactSalesForceLeadReplicationRequired* in the default implementation).
7. [Create a new field](#) based on your requirements and place it inside the new category.
 - Leave *Standard field* in **Field type**.
 - Leave the **Display field in the editing form** check box selected.
8. Click **Save**.


Now you have created a static custom field to gather details about your contacts. You can view and edit the values of your new custom field on the **Profile** tab of the contact managing interface.

 Custom fields on the **Profile** tab are visible only when they contain a value which is not empty for the given contact. Empty fields are hidden by default.

Adding reference custom fields to contacts

Reference custom fields store a reference to a database table. To define reference custom fields:

1. Open the **Modules** application.
2. **Edit** (✎) the **Contact management** module.
3. Open the **Classes** tab.
4. **Edit** (✎) the **Contact management - Contact** class.
5. Switch to the **Fields** tab.
6. [Create a new field](#) based on your needs.
 - Leave *Standard field* in **Field type**.
 - Select *Integer number* in **Data type**.
 - Leave the **Display field in the editing form** check box selected.

 Make sure that you also fill in the **Field caption** field, which sets the text displayed next to the field in the form.

Remember the value that you filled in as **Field name**, you will need to enter the value as a parameter in the application initialization method when defining the [binding between contacts and the referenced objects](#).

7. Click **Save**.

Now you have created a reference custom field to gather further details about your contacts.

Defining binding between contacts and referenced objects

To be able to view and edit the values of your new fields, you still need to define the binding between contacts and referenced objects.

1. Open your project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
2. Initialize your application.
 - For more information, see [Initializing modules to run custom code](#).
3. In the application initialization method, enter the value that you filled in as **Field name** of your reference custom field to the *ObjectDependency* first parameter.
 - In the following example, the filled in value is *ContactCoffeeCountryID*.

Example

```
private static void ApplicationInit(object sender, EventArgs e)
{
    var contactDependencies = (List<ObjectDependency>)ContactInfo.TYPEINFO.
DependsOn;
    var contactCoffeeCountryDependency = new ObjectDependency
("ContactCoffeeCountryID", CountryInfo.OBJECT_TYPE);

    contactDependencies.Add(contactCoffeeCountryDependency);
}
```

The reference between contacts and the referenced objects is defined and you are able to view and edit the values of your new reference fields on the **Profile** tab of the contact managing interface.

Moving fields in contact detail categories

When you have created multiple custom fields, you can organize the fields displayed on the contact's **Details** tab by moving them.

1. Open the **Modules** application.
2. **Edit** (✎) the **Contact management** module.
3. Open the **Classes** tab.
4. **Edit** (✎) the **Contact management - Contact** class.
5. Switch to the **Fields** tab.
6. Select the field in the list.
7. Move the item using the **Move up** (⬆) or **Move down** (⬇) buttons.

Moving fields in the list changes their positions in the resulting form.

Displaying custom data on a separate tab

When you have gathered a lot of custom data, you can distinguish between the custom data and the data collected by the default fields by displaying the custom data on a separate tab of the contact editing interface.

To be able to render the custom data on the separate tab, you first need to [create the custom page](#) and then [create the tab](#) for displaying the page's content.

Creating custom page with contact card

To create the custom page for displaying the contact card and contact's custom data:

1. Open your project in Visual Studio (using the **WebSite.sln** or **WebApp.sln** file).
2. Navigate to the **Details.aspx** file.

- The path is `~/CMSModules/ContactManagement/Pages/Contact/Details.aspx`.
3. While still in the **Contact** folder, copy and paste the **Details.aspx** file.
 4. Rename the file to, for example, **CustomData.aspx**.
 5. Change the class name in the **CustomData.aspx.cs** and **CustomData.aspx** files.
 6. Replace the content in the **CustomData.aspx** file with:

Example

```
<div data-ng-controller="app as $ctrl">
  <cms-card-wrapper contact-id="$ctrl.contactId"></cms-card-wrapper>
</div>

<!-- any content -->
```

7. You will get the following result:

Example

```
<asp:Content ID="cntBody" runat="server" ContentPlaceHolderID="cpAfterForm" ng-strict-di>
  <div data-ng-controller="app as $ctrl">
    <cms-card-wrapper contact-id="$ctrl.contactId"></cms-card-wrapper>
  </div>

  <!-- any content -->
</asp:Content>
```

8. Insert your content under the contact card section.

Now that you have created the custom page for displaying the contact card and inserted your custom data, you can create a new tab in the Kentico administration interface.

Creating a new tab in the user interface

To create a separate tab in the contact editing interface for displaying the content of your custom page:

1. Open the **Modules** application.
2. **Edit** (✎) the **Custom** module.
3. Switch to the **User interface** tab.
4. Navigate to the **Administration -> On-line marketing -> Contact management -> Contacts -> Contact properties** tree item.
5. Click **New element** (+).
6. Enter *Custom data* in the **Display name** field.
7. Select *URL* in the **Type** field.
8. In the **Target URL** field, enter the `~/CMSModules/ContactManagement/Pages/Contact/CustomData.aspx` path to the created custom page.
9. Click **Save**.

The custom page content is now displayed on the **Custom data** tab of the contact managing interface.