Hands on Lab:
Custom Table: Create and Display

July, 2012

Twitter: @trobbins
Email: Thomasr@Kentico.com
Blog: http://devnet.kentico.com/Blogs/Thomas-Robbins.aspx

Kentico CMS Hands on Lab

For more information http://www.kentico.com

Table of Contents

Hands on Lab:
Custom Table: Create and Display

Lab overview

System Requirements:

- Kentico CMS 6(SP1) with the Sample Corporate Site

Intended Lab Audience:

- CMS Designers
- CMS Developers

Lab introduction

Kentico CMS provides custom tables as a way to store and manage custom data. Technically, they are SQL tables and coupled stored procedures. Once a custom table is created and contains data, the next step is to provide a display for site visitors. In this lab we will add a custom table and then use the Basic Repeater web part to display it on a page. We will also look at how to use the Kentico CMS API to add and edit an existing item. In completing this set of labs you will perform the following tasks:

- Create a custom table
- Explore the Data Source web part
- Explore the Basic Repeater web part
- Edit a transformation for better visual display
- Add a new item to the custom table using the Kentico CMS API
- Edit an existing item using the Kentico CMS API

1

Lab: Custom Table: Create and Display

Lab 1: Creating a custom table

1. As **Administrator** log into **CMS Site Manager**, select the **Development** tab, then select **Custom tables**, and click **New custom table** as shown in the following screenshot.
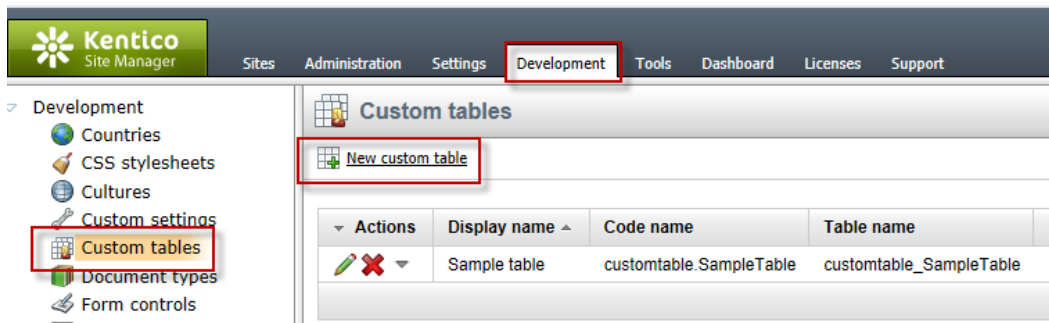
2. In **Step 1- General** update the following fields and then click **Next.**

Field	Value
Custom table display name	**Customer Contacts**
Custom table code name (custom table)	**custom**

Kentico CMS Hands on Lab

3. In **Step 2 – Data type** click **Next** as shown in the following screenshot.

4. In **Step 3 – Fields** click the **New attribute (+)** button as shown in the following screenshot.

3

5. In the **New attribute (+)** dialog update the following fields and then click **Save field**.

Section	Field	Value
Database	Attribute name	Name
Database	Attribute size	100
Field appearance	Field caption	Name
Field appearance	Form control	Textbox

6. Select the **New attribute (+)** button, update the following fields and then click **Save field.**

Section	Field	Value
Database	Attribute name	Title
Database	Attribute size	100
Field appearance	Field caption	Title
Field appearance	Form control	Textbox

7. Select the **New attribute (+)** button, update the following fields, and then click **Save field.**

Section	Field	Value
Database	Attribute name	Email
Database	Attribute size	100
Field appearance	Field caption	Email

Field appearance	Form type	E-mail

8. In **Step 2 – Fields** validate that you see the new fields and then click the **Next** button as shown in the following screenshot.

9. In **Step 4 – Sites** click the **Next** button as shown in the following screenshot.

10. In **Step 5 – Search options** click the **Next** button as shown in the following screenshot.
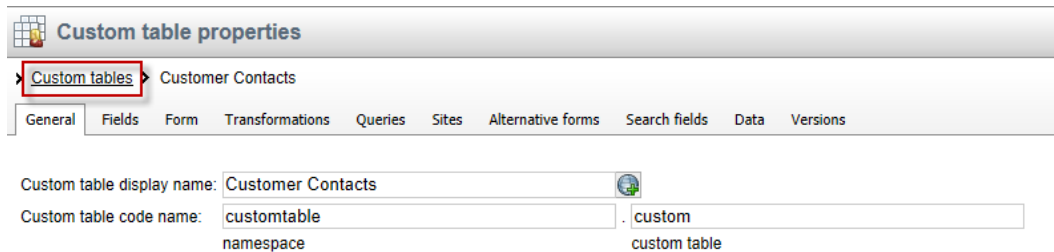
11. In **Step 6 – The wizard has finished** click the **Finish** button as shown in the following screenshot.
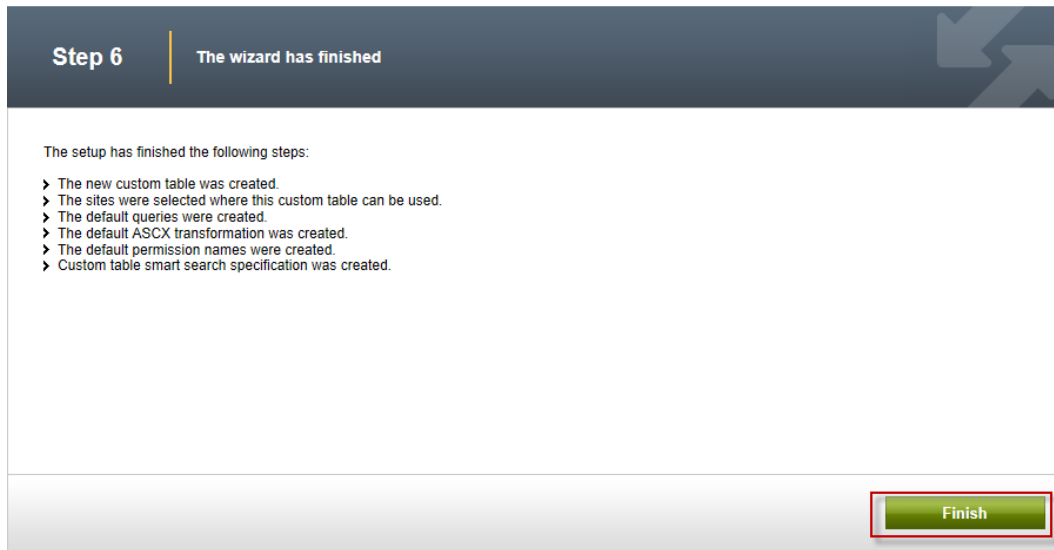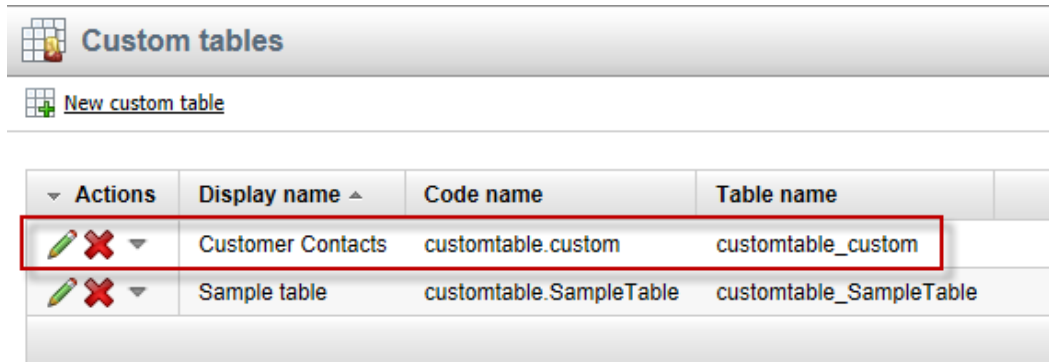

12. In the **Custom table** properties screen click the **Custom tables** menu item as shown in the following screenshot.

13. In the **Custom tables** screen validate that the **Customer Contacts** custom table is displayed as shown in the following screenshot.
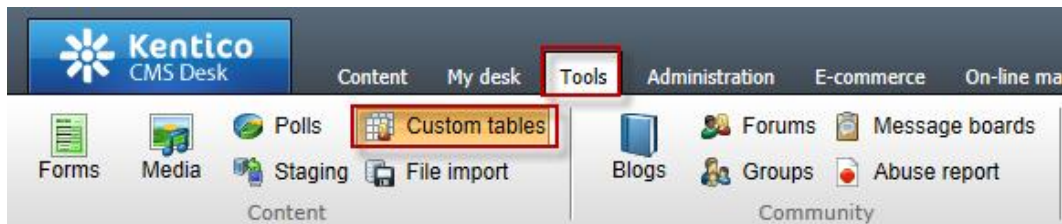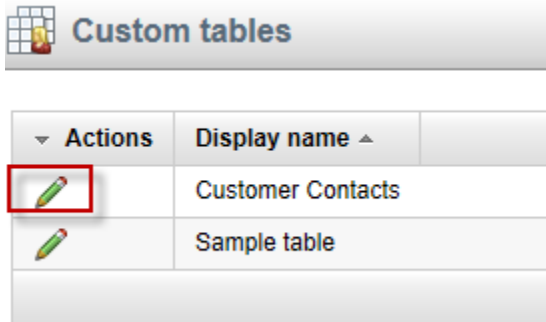

Lab 2: Adding custom table data

1. As **Administrator** log into **CMS Desk**, select the **Tools** tab, then select the **Custom tables** button as shown in the following screenshot.
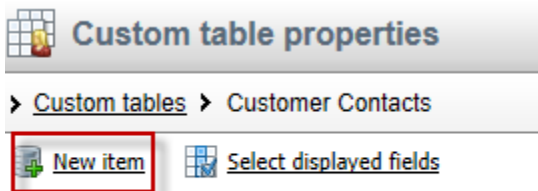
2. In the Custom tables screen select the **Edit (Pencil)** icon for **Customer Contacts** as shown in the following screenshot.

4. In the Custom table properties screen select the **New item** button as shown in the following screenshot.

3. In the **New item** screen update the following fields and then select **Ok.**

Field	Value
Name	**Joe Brown**
Title	**President**
Email	joe@company.com

4. Click the **Create another** button, update the following fields, and then click **Ok**.

Field	Value
Name	**Ted Thomas**
Title	**Marketing VP**
Email	**ted@company.com**

5. On the Edit item screen click the **Data** link as shown in the following screenshot.
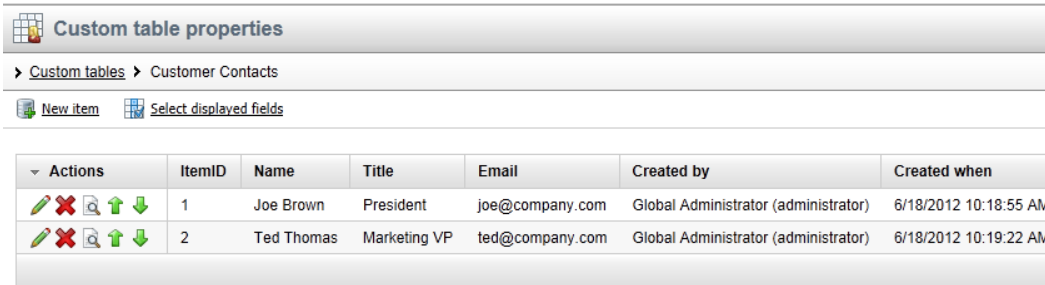
6. In the Custom table properties screen validate that the two items just entered are visible as shown in the following screenshot.

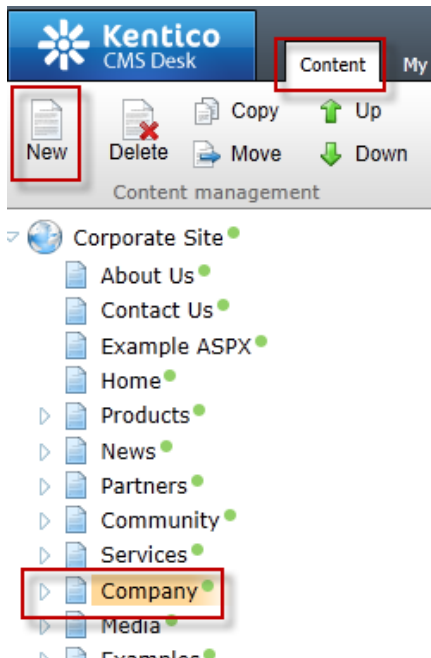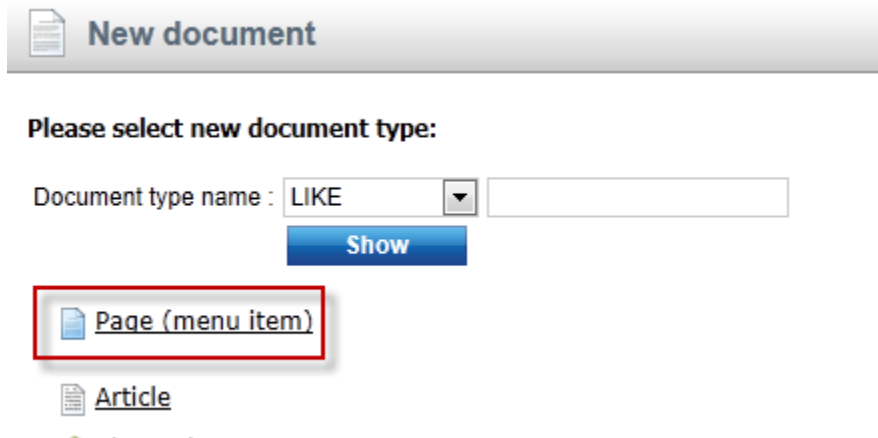## Lab 3: Creating the display page

1. In CMS Desk select the **Content** tab, in the Content tree select **Company** and then click the **New (Paper)** icon, as shown in the following screenshot.

2. In the New document screen click the **Page (menu item)** as shown in the following screenshot.
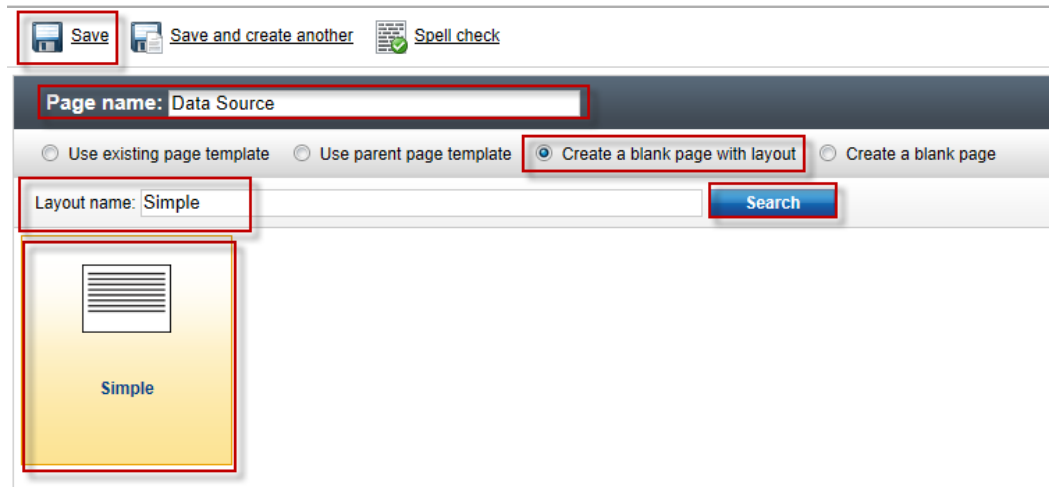
3. In the **New Page (menu item)** screen in the **Page name** enter **Data Source**, click **Create a blank page with layout**, in the **Layout name** enter **Simple**, then click the **Search** button, select the **Simple** page template, and finally click the **Save** button as shown in the following screenshot.
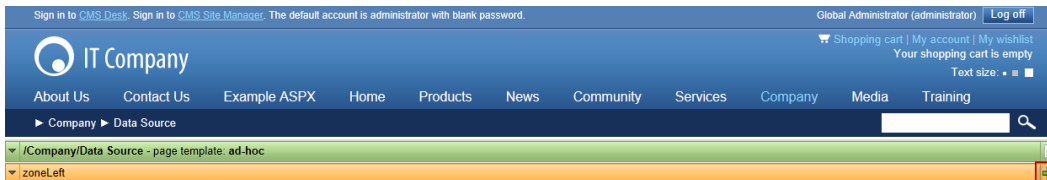
4. Validate that you see the new Data Source page as shown in the following screenshot.
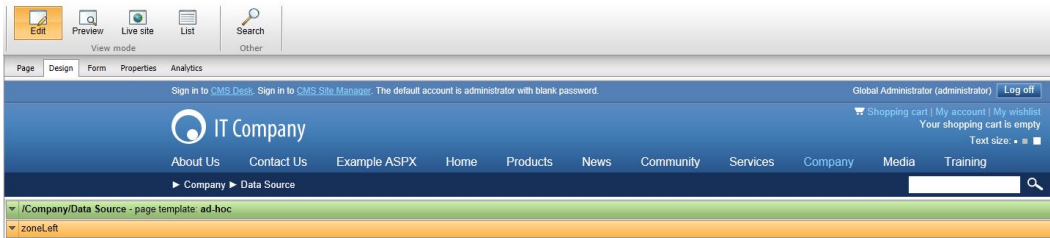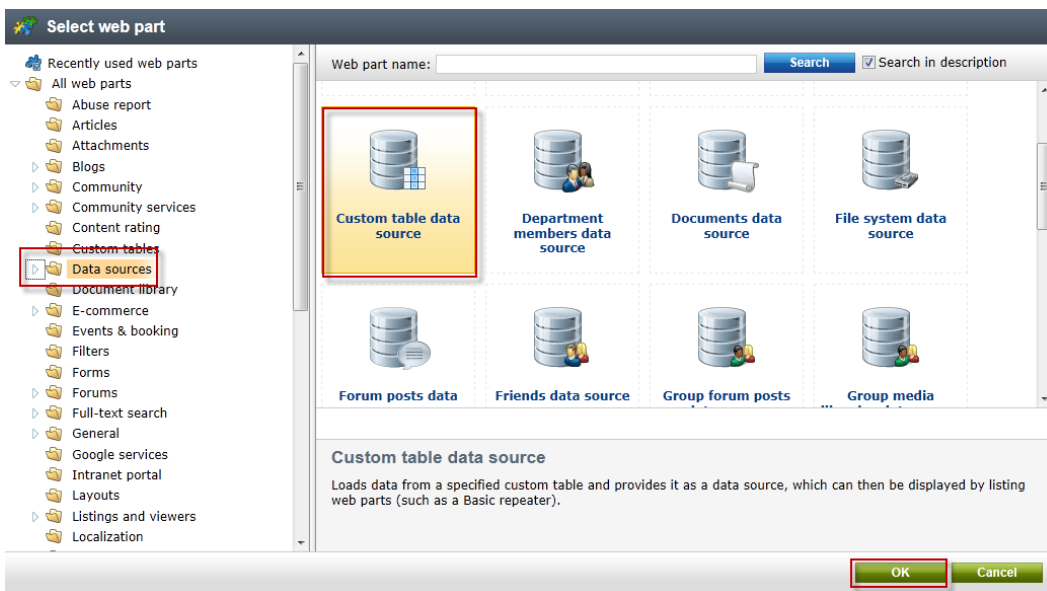
Lab 4: Displaying custom table data

1. In the **zoneLeft** web part zone, click the **Add web part icon (+)** as shown in the following screenshot.
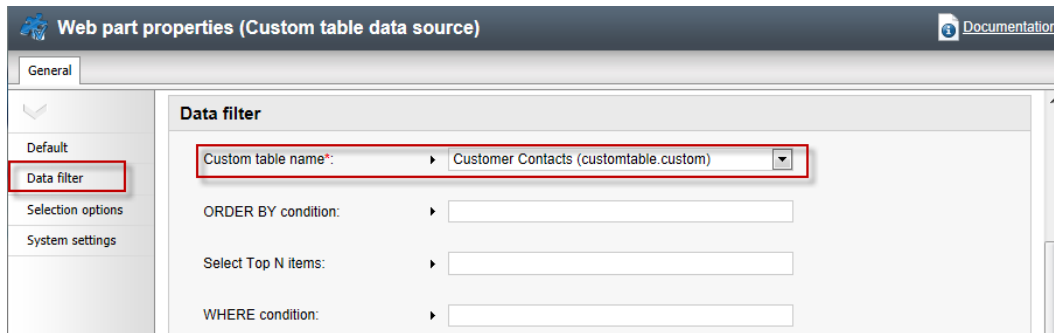

2. In the **Select web part** screen select **Data sources**, then select the **Custom table data source** web part, and then click **OK** as shown in the following screenshot.

3. In the **Web part properties (Custom table data source)** screen select the **Data filter** tab, in the **Custom table name** field select the **Customer Contacts (customtable.custom)** and click **Ok** as shown in the following screenshot.


What is a data sources web part?

Data source web parts are used to connect and retrieve data from back end data sources like custom tables. They retrieve data and pass them to display web parts. This separation is designed to provide page based flexibility in data retrieval and display. Data sources provide reusable sources of data. They reduce page resources as data is retrieved just once and then can be used by different web parts on the same template to display content.

4. In the **zoneleft** web part zone select the **Add web part icon (+)** as shown in the following screenshot.
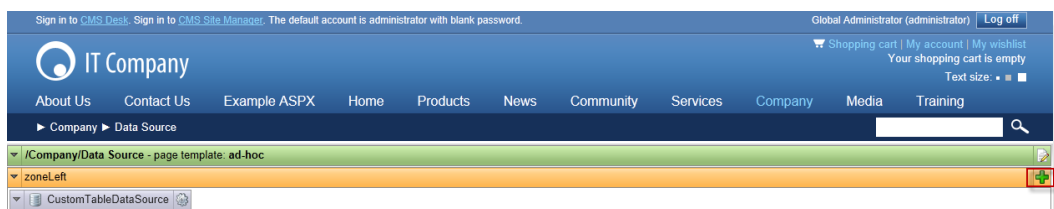
5. In the **Select web part** screen select the **Listings and viewers** node, click the **Basic repeater** web part and click **Ok** as shown in the following screenshot.
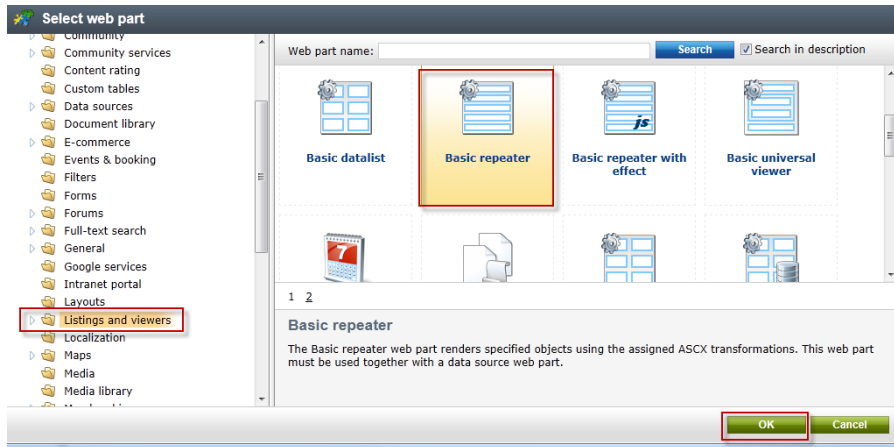
Listings and Viewers

The listings and viewers node contains web parts that can be used to provide display for the data source web parts.

6. In the **Web part properties (Basic repeater)** select the **Content** tab, and in the **Data source name** field enter **CustomTableDataSource** as shown in the following screenshot.
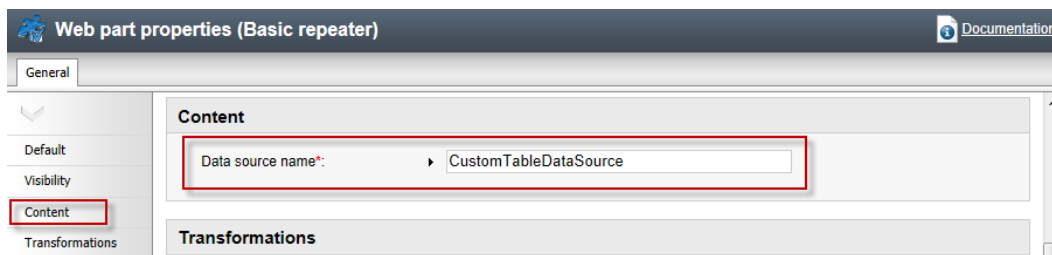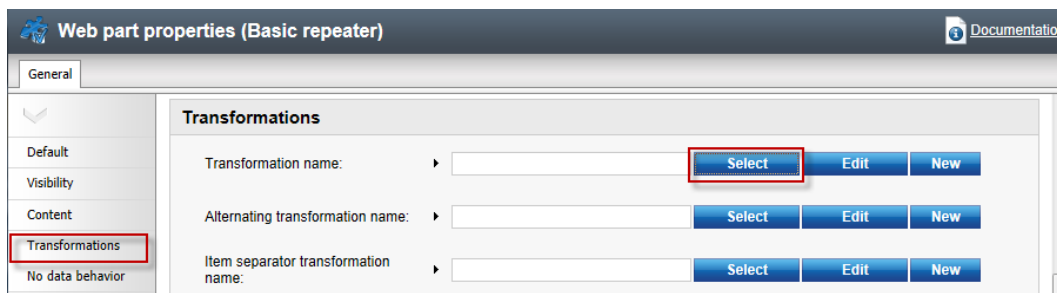
Connecting to Data Sources

The Data source name field is the connection between this web part and the data source web part we added in Step 2 and must be an exact match.

7. In the **Web part properties (Basic repeater)** screen select the **Transformation** tab, and in the **Transformation name** field click the **Select** button as shown in the following screenshot.
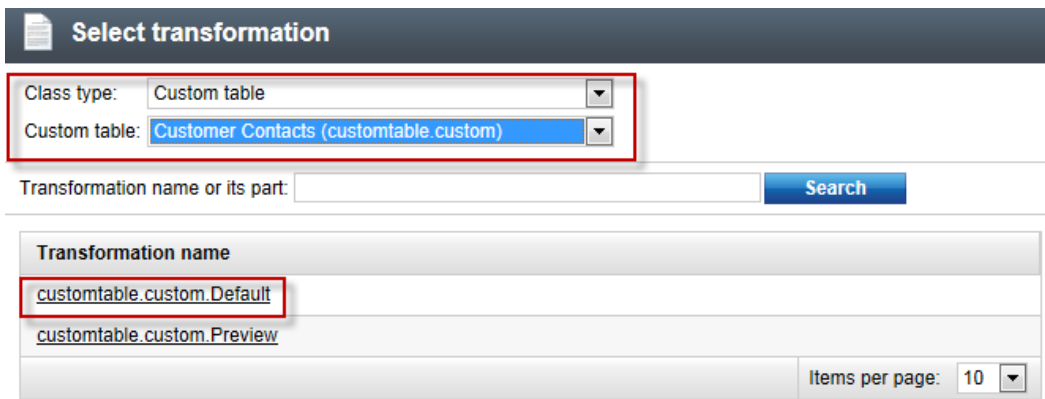

What is a transformation?

When data is passed from the data source web part to the Basic repeater a set of formatting code (transformation) is applied to define the output that is displayed.

8. In the **Select transformation** dialog in the **Class type** drop down select **Custom table**, in the **Custom table** drop down select **Customer Contacts (customtable.custom)**, then click the **customtable.custom Default** transformation as shown in the following screenshot.
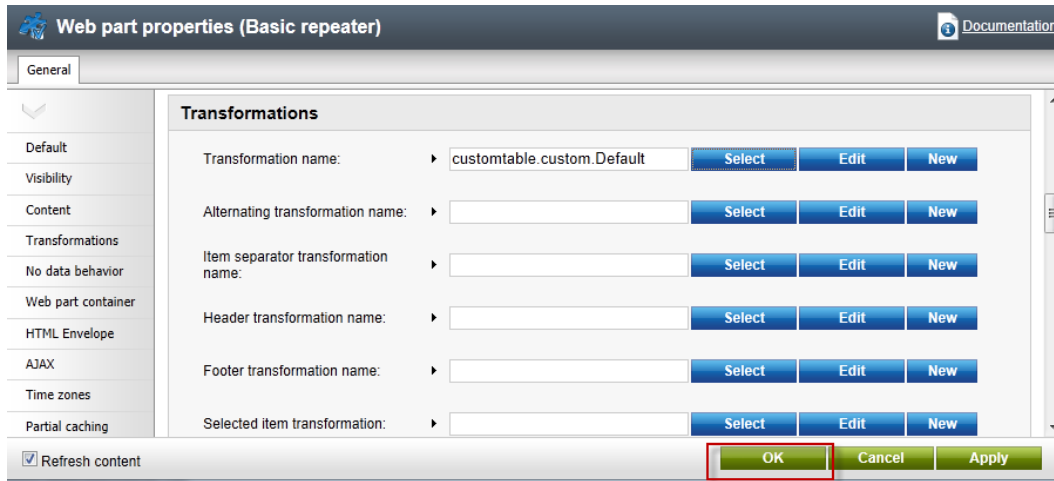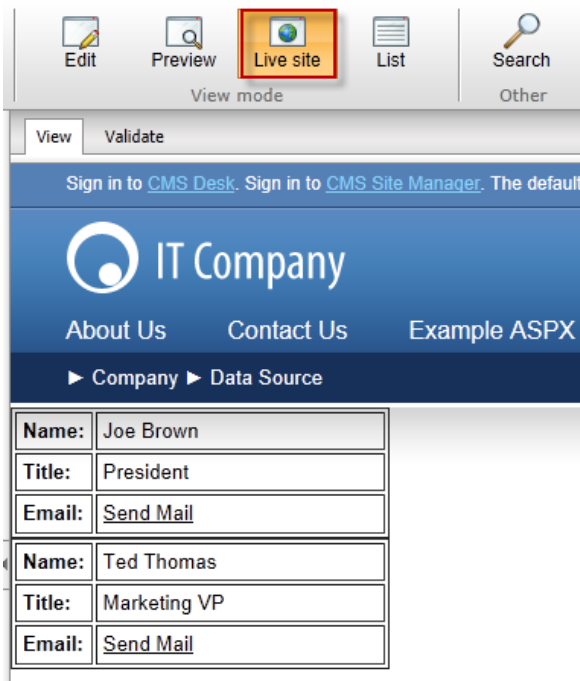

Where did these come from?

When we created the custom table the system automatically creates a Default and Preview transformation

9. In the **Web part properties (Basic repeater)** screen select **Ok** as shown in the following screenshot.

10. Validate that the data is displayed as shown in the following screenshot.

Lab 5: Editing the default Transformation

1. In the Design tab for the **BasicRepeater** web part select the **Configure** button as shown in the following screenshot.
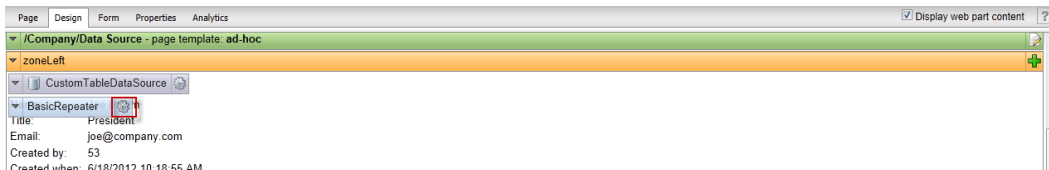
2. In the **Web part properties (Basic repeater)** screen select the **Transformation** tab, in the **Transformation name** select the **Edit** button as shown in the following screenshot.
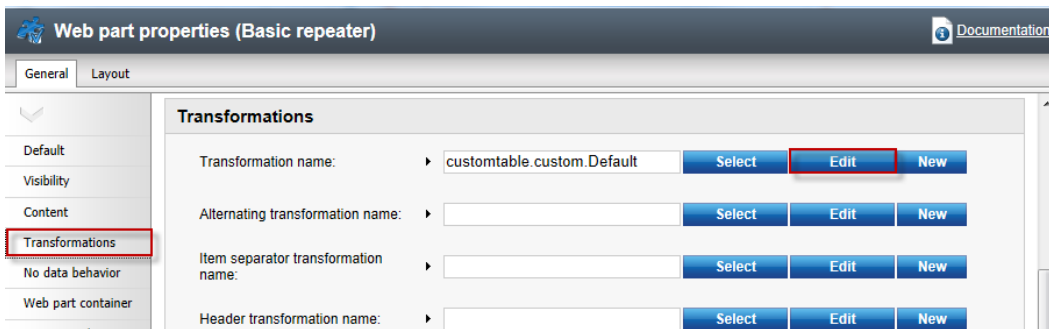
3. In the **Edit transformation** dialog within the **Code** window replace the existing code with the following, then select **Save & close**, and in the **Web part properties (basic repeater)** window click **OK.**

```
<table border = "1" cellpadding="4">
  <tr>
    <td><b>Name:</b></td>
    <td width="180"><%# Eval("Name") %></td>
  </tr>
  <tr>
    <td><b>Title:</b></td>
    <td><%# Eval("Title") %></td>
  </tr>
  <tr>
    <td><b>Email:</b></td>
    <td><a href="mailto:<%#
Eval("Email")%>?Subject=Contact%20Request">Send Mail</a></td>
  </tr>
  </table>
```

4. Select the **Live Site** button to view the edited transformation as shown in the following screenshot.

5. On the Live site click the **Send Mail** link for **Joe Brown** and validate that you see the email as shown below.
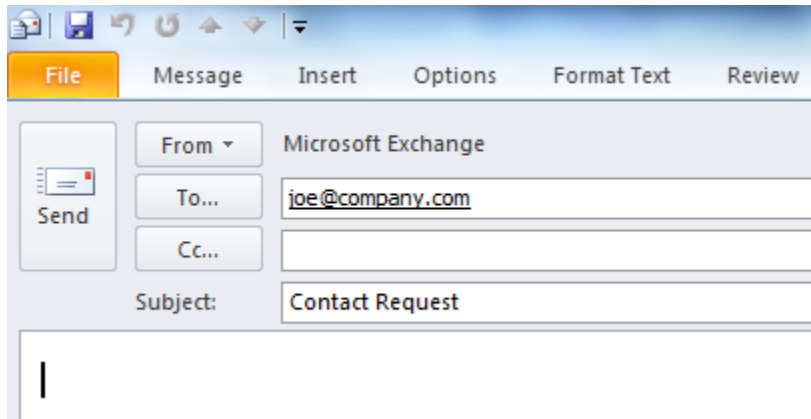
Lab 6: Adding custom table data with the API

1. In Visual Studio open the Kentico CMS Web project in the **Solution Explorer** right click on the project root and select **Add new item** as shown in the following screenshot.
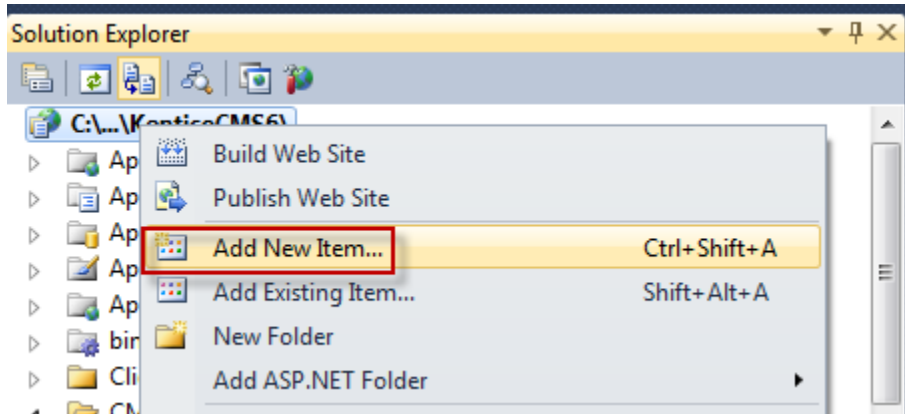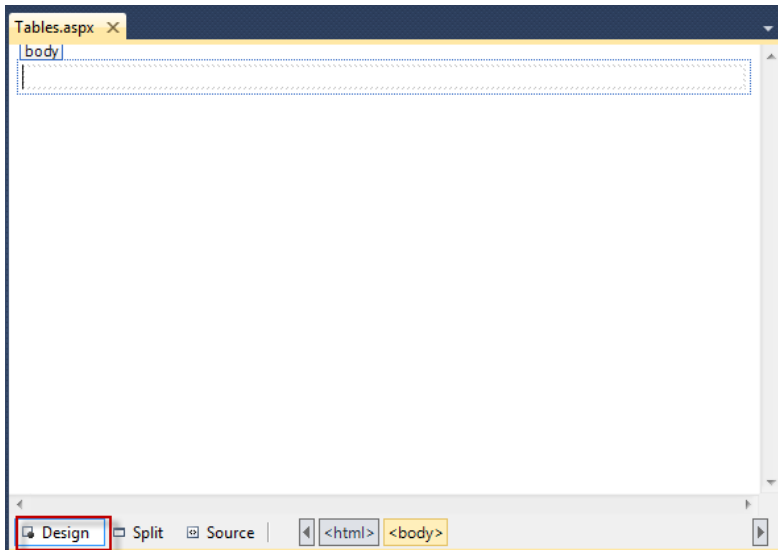

2. In the **Add new item** dialog select **Web form**, in the **Name** field enter **Tables.aspx**, and then click the **Add** button as shown in the following screenshot.
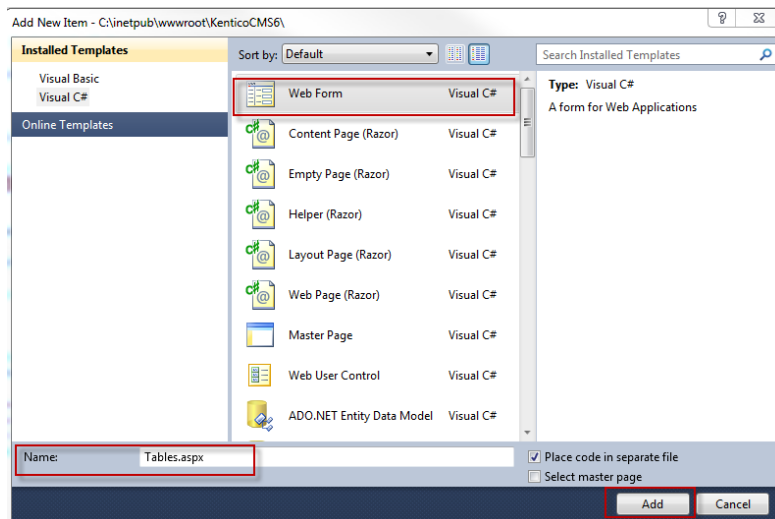
3. In the Tables.aspx form click the **Design** button as shown in the following screenshot.

4. From the Toolbox drag the **Button** control onto **Tables.aspx** as shown in the following screenshot.
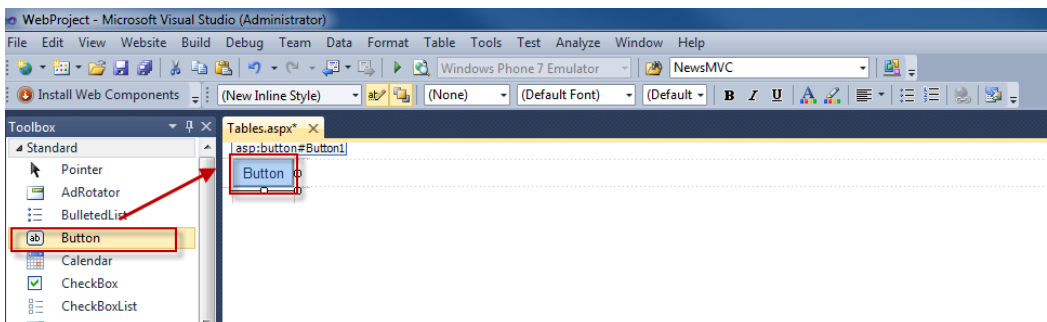
5. In the **Button1 Properties** window update the following fields

Field	Value
ID	**CreateItem**
Text	**Create new item**

6. On the Design tab double click the **Create new item** button as shown in the following screenshot.
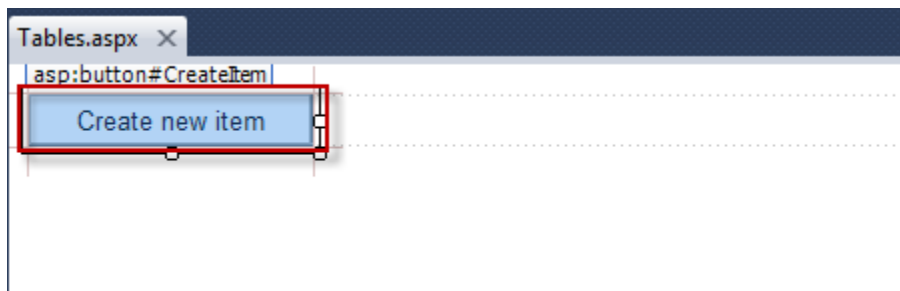
7. At the top of the Tables.aspx web form add the following **Using** statements.

```
using CMS.SettingsProvider;
using CMS.SiteProvider;
using CMS.DataEngine;
using CMS.CMSHelper;
```

8. In the **Createitem_click** event replace any existing code with the following and then select **Save (CTRL+S).**

```
        CustomTableItemProvider ctiProvider = new
CustomTableItemProvider(CMSContext.CurrentUser);
        string customtablename = "customtable.custom";


        DataClassInfo customTableClassInfo =
DataClassInfoProvider.GetDataClass(customtablename);

        if (customTableClassInfo == null)
        {
            throw new Exception("table does not exist.");
        }
        else
        {
            // Create item
            CustomTableItem item = new
CustomTableItem(customTableClassInfo.ClassName, ctiProvider);
            // Set values
            item.SetValue("Name", "Thom Robbins");
            item.SetValue("Email", "thomasr@kentico.com");
            item.SetValue("Title", "Dish washer");
            // Insert the item
            item.Insert();
```

9. With Tables.aspx selected in the **Solution Explorer** press the **Start (Play)** button as shown in the following screenshot.
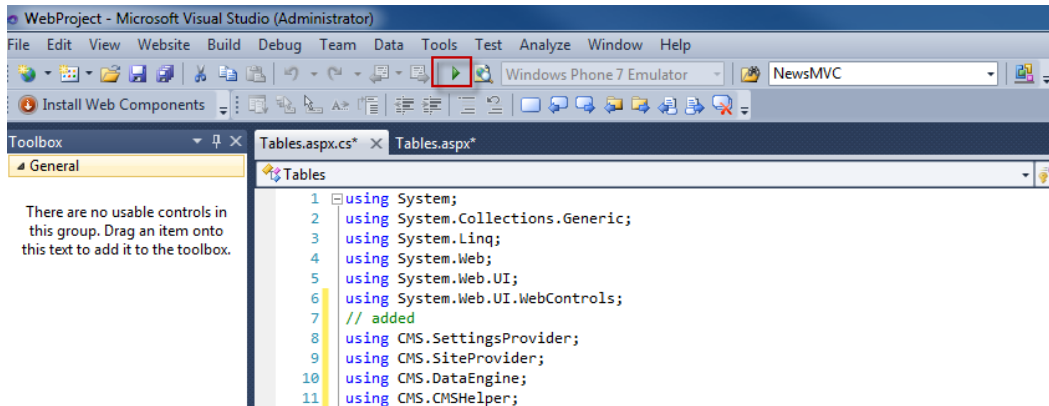
10. In the running application click the **Create new item** button as shown in the following screenshot.
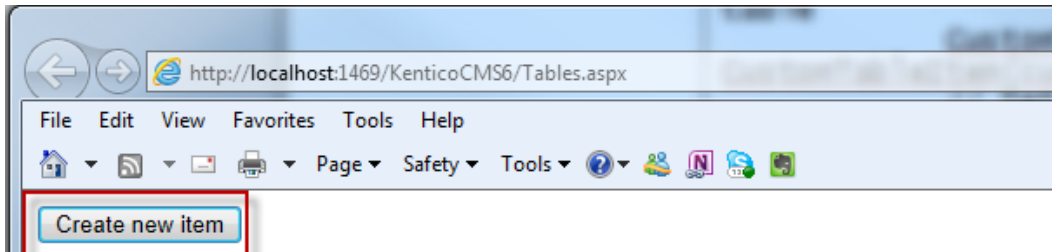
11. Log into CMS Desk then select the **Tools** tab and then select **Custom tables** as shown in the following screenshot.
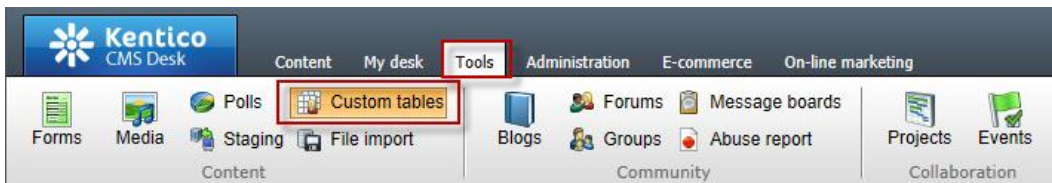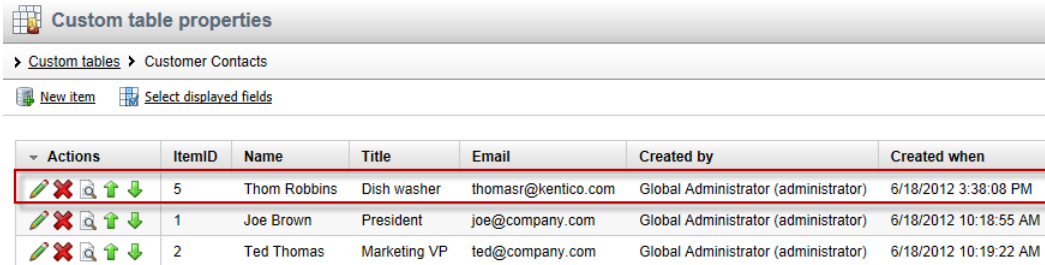
12. In the Custom tables menu select the **Edit (Pencil)** icon for the **Customer Contacts** table as shown in the following screenshot.
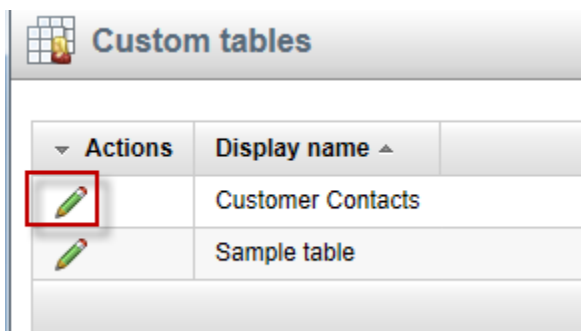
13. In the Custom table properties validate that the new custom table item is available as shown in the following screenshot.
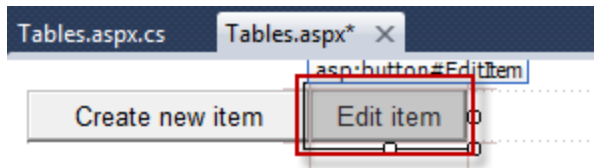
Lab 7: Editing custom table data using the API

1. On the Tables. aspx form on the Design tab from the **Toolbox** drag a **Button** control onto the form as shown in the following screenshot.
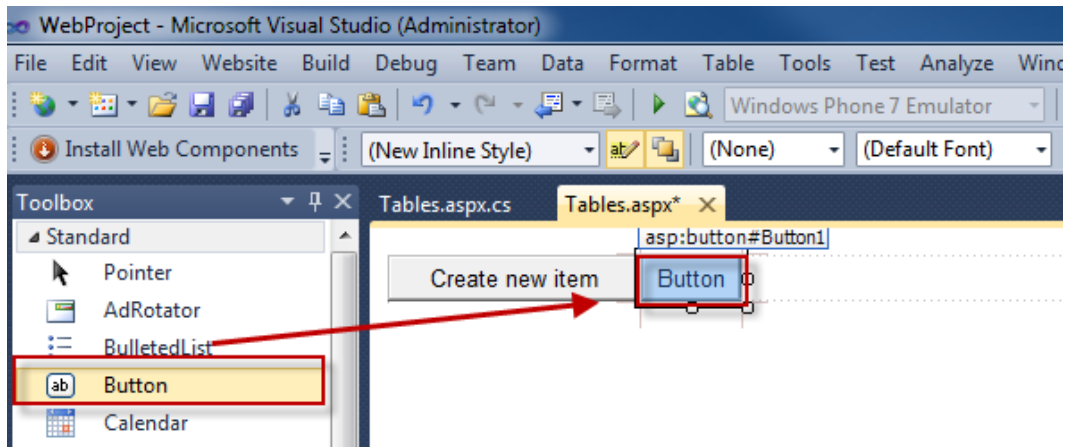
2. In the **Button 1 Properties** update the following fields

Field	Value
ID	**EditItem**
Text	**Edit item**

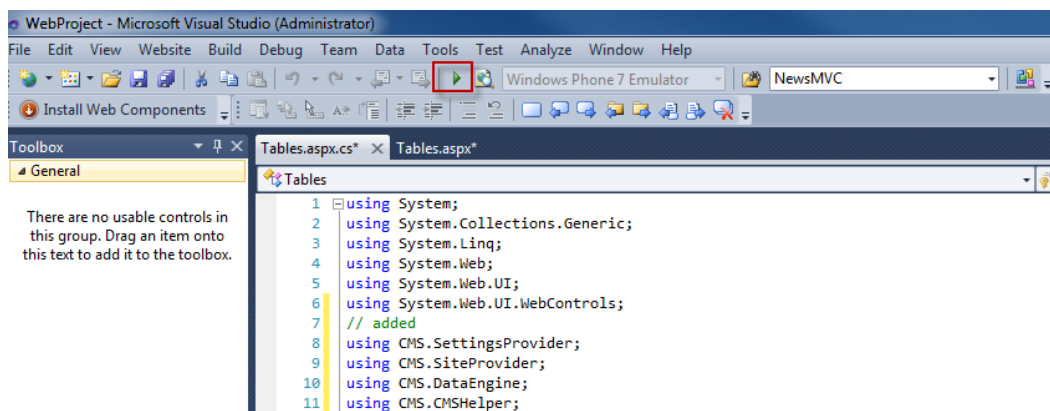3. Double click the **Edit item** button as shown in the following screenshot.

4. In the **EditItem_Click** event replace any code with the following and then select
   Save (CTRL+S).

```
  CustomTableItemProvider ctiProvider = new
CustomTableItemProvider(CMSContext.CurrentUser);
        string customtablename = "customtable.custom";


        DataClassInfo customTableClassInfo =
DataClassInfoProvider.GetDataClass(customtablename);

        if (customTableClassInfo == null)
        {
            throw new Exception("table does not exist.");
        }
        else
        {
            // Provide ID of item you want to edit
            int itemId = 5;
            // Get custom table item with given ID
            CustomTableItem item = ctiProvider.GetItem(itemId,
customTableClassInfo.ClassName);
            // Set value of the custom table item field
            item.SetValue("Name", "Frank Jones");
            // Update item
            item.Update();
        }
```
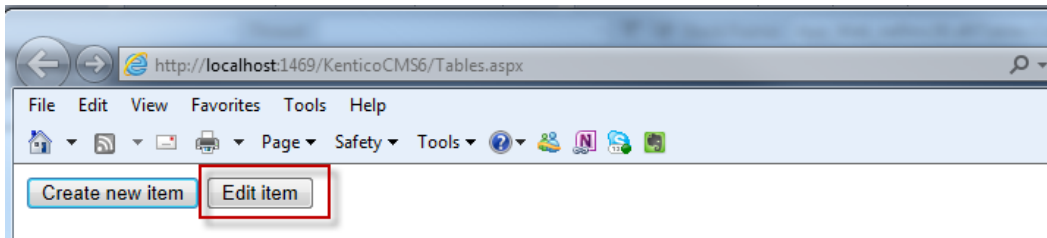
5. With Tables.aspx selected in the **Solution Explorer** press **Start (Play)** as shown in
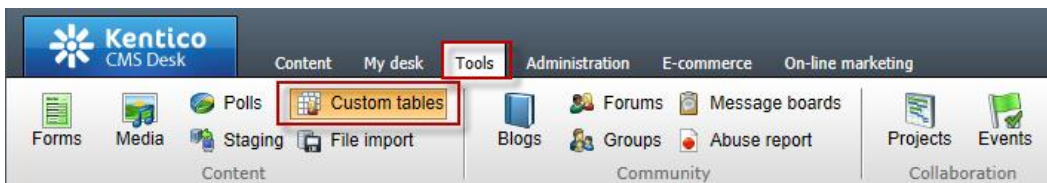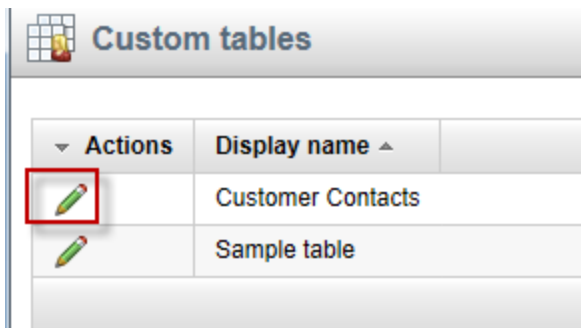   the following screenshot.

6. In the running application click the **Edit item** button as shown in the following screenshot.

7. Log into CMS Desk then select the **Tools** tab and then select **Custom tables** as shown in the following screenshot.
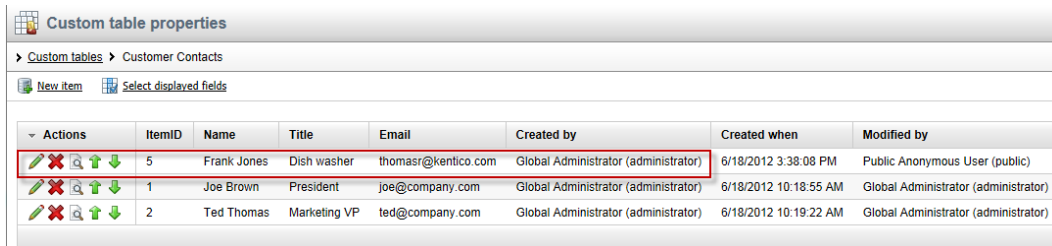

8. In the Custom tables menu select the **Edit (Pencil)** icon for the **Customer Contacts** table as shown in the following screenshot.

9. In the Custom table properties validate that the **Name** field contains **Frank Jones** as shown in the following screenshot.

Summary

In this set of Hands on Labs we looked at how to create a custom table and then display the data using the Basic Repeater web part. We also looked at how we can edit the default transformation and use the Kentico CMS API. Specifically, we covered the following.

- Creating a custom table
- Adding data to a custom data using CMS Site manager
- Creating a new CMS page
- Using Data source web parts
- Displaying data with the Basic repeater web part
- Editing the default transformation
- Adding items using the API
- Editing items using the API

Key Resources:

- http://devnet.kentico.com/Documentation.aspx
- http://devnet.kentico.com/Videos/Content-Management/How-To--Using-Data-Source-Web-Parts.aspx