The CMSDataList control displays a list of formatted page data.

- Provides a built-in data source for loading Kentico pages. You do not need to write code for retrieving and binding the data.
- Supports Kentico transformations.
- Can display data in multiple columns.
- Supports nested controls (See Using nested controls).

**Inherits from**: BasicDataList
**Web part equivalent (portal engine)**: Datalist

## Getting started

The following tutorial shows how to use the CMSDataList to display a list of all smartphones, laptops, and tablets (CMS. Smartphone and CMS.Laptop pages) from the products section of the sample Corporate Site:

1. Create a new **Web form** in your web project.
2. Drag the **CMSDataList** control from the toolbox onto the form.
3. Set the following properties for the control:

    - **Path**: /Products/%
    - **ClassNames**: cms.smartphone;cms.laptop
    - **OrderBy**: SKUName
    - **RepeatColumns**: 4
    - **TransformationName**: CorporateSite.Transformations.ProductList

```
<cms:CMSDataList ID="CMSDataList1" runat="server" ClassNames="cms.smartphone;cms.
laptop" OrderBy="SKUName" Path="/products/%" RepeatColumns="4"
TransformationName="CorporateSite.Transformations.ProductList" />
```

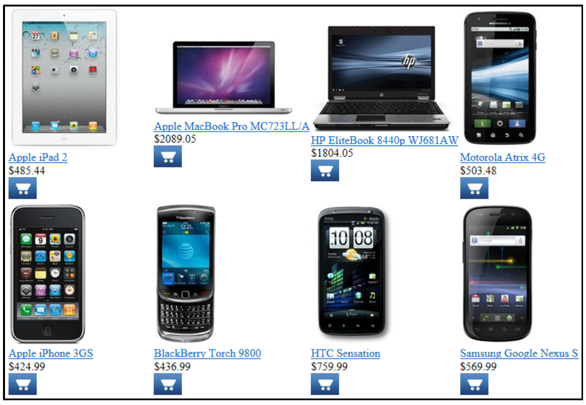
4. Add the **asp:ScriptManager** control before the CMSDataList.

```
<asp:ScriptManager ID="manScript" runat="server" ScriptMode="Release"
EnableViewState="false" />
```

The **ScriptManager** control included at the top is required by the transformation used to display products. It is only there to ensure that the web form is functional as a standalone example. Typically, the *ScriptManager* is included on the website's master page.

5. Save the web form.
6. Right-click the web form in the Solution explorer and select **View in Browser**.

The resulting page displays a list of products.



Apple iPad 2
$485.44

Apple MacBook Pro MC723LL/A
$2089.05

HP EliteBook 8440p WJ681AW
$1804.05

Motorola Atrix 4G
$503.48

Apple iPhone 3GS
$424.99

BlackBerry Torch 9800
$436.99

HTC Sensation
$759.99

Samsung Google Nexus S
$569.99

## Configuration

You can set the following properties for the CMSDataList control:

| CMSDataList properties | Description | Sample value |
|---|---|---|
| AlternatingTransformationName | Name of the transformation applied to alternating items.<br>Enter the transformation name in format *<page type code name>.<transformation name>*. | |
| DataBindByDefault | Indicates whether the control automatically performs data binding during the **Init** event. (Inherited from BasicDataList) | |
| DelayedLoading | Indicates whether the control loads data during the **Load** event instead of the default **Init** event. | |
| EnablePaging | Indicates whether the built-in DataPager control is used to page the list.<br>If you wish to use a UniPager, set this property to false and add a separate UniPager control to the page. | |
| HideControlForZeroRows | Indicates whether the control should be hidden when no data is loaded. The default value is False. (Inherited from BasicDataList) | |
| NestedControlsID | Enter the IDs of any nested controls (CMSRepeater, CMSDataList...), separated by semicolons. | "CMSRepeaterNested;<br>CMSDataListNested" |
| PageSize | Sets the number of displayed items per page. | |
| PagerControl | Can be used to set or get the pager control and its properties. | |
| PagerDataItem | Gets or sets the pager data item object. (Inherited from BasicDataList) | |
| PagerForceNumberOfResults | If set, the pager does not modify the DataSet containing paged items, but the pager itself behaves as if the amount of paged items were identical to this value.<br>The value must be set to *-1* for the property to be disabled.<br>(Inherited from BasicDataList) | |
| RelatedData | Custom data connected to the object. (Inherited from BasicDataList) | |

| | | |
|---|---|---|
| SelectedItemTransformationName | Name of the transformation applied to the selected item.<br>Enter the transformation name in format <*page type code name*>.<*transformation name*>. | |
| ShowEditDeleteButtons | Indicates if the control renders edit and delete buttons next to the displayed pages in editing modes (Page tab in the Pages application and On-site editing mode). | |
| TransformationName | Name of the transformation applied to standard items.<br>Enter the transformation name in format <*page type code name*>.<*transformation name*>. | |
| ZeroRowsText | Text shown if no records are found. This text is not visible when the control is hidden by the **HideControlForZeroRows** property. (Inherited from BasicDataList) | "No records found." |

| Page filtering properties | Description | Sample value |
|---|---|---|
| CheckPermissions | Indicates if the control checks the permissions of the user viewing the page. If the value is *false* (default value) no permissions are checked.<br>If true, the control only loads pages for which the user viewing the page has read permissions. | |
| ClassNames | Specifies which page types the control loads and displays. Identify page types through their code names, separated by semicolons (;).<br>You can use the * wildcard as a substitute for any number of characters. For example *Product.** includes the page types *Product.Camera*, *Product.CellPhone*, *Product.Computer* etc.<br>If the property is left empty, the control retrieves all page types by default. In the case of menu and navigation controls, only *CMS.MenuItem pages* are loaded by default.<br>**Note**: If the control loads all page types (empty value), only the data from the *View_CMS_Tree_Joined* and the *COM_SKU* table (for product pages) are available in the retrieved data. The specific fields of individual page types are not included. You need to keep this in mind when writing the code of transformations, WHERE conditions, ORDER BY expressions etc. | "cms.news"<br>"cms.news;cms.article" |
| CombineWithDefaultCulture | Indicates whether the control loads pages from the website's default culture version if the required pages are not available in the user's selected culture.<br>Only applies if you do not set the **TreeProvider** property manually. | |
| CultureCode | Specifies the culture code of the pages that the control loads. If not specified, the control automatically uses the preferred culture of the user viewing the page. | "en-us" |
| DataSource | Allows you to manually assign a DataSet or DataTable containing the pages that the control displays. You do not need to set this property for standard scenarios. | |
| FilterOutDuplicates | Indicates if the control filters out duplicated (linked) pages from the data. | |
| MaxRelativeLevel | Specifies the maximum number of content tree sub-levels from which the control displays pages. This number is relative, i.e. counted from the location of the page where the control is placed, not from the root of the website.<br>Enter *-1* to load all child pages. | |

| | | |
|---|---|---|
| Path | Path of the pages that the control loads.<br>**See**: Writing page path expressions | |
| SelectOnly Published | If enabled, the control only loads published pages. | |
| TreeProvider | Gets or sets the *TreeProvider* object used by the control to access page data. If you do not assign a TreeProvider object, the control automatically creates a new instance. | |

| Page relationship properties | Description | Sample value |
|---|---|---|
| RelationshipWithNodeGUID | If set, the control only loads pages that are related to the page with the specified **Node GUID**. You can find the Node GUID of pages on the **Properties -> General** tab in the Pages application.<br>Enter *"11111111-1111-1111-1111-111111111111"* to dynamically load pages related to the **current** page. | "36f8c4bc-f702-4736-8a25-a82295668794" |
| RelatedNodeIsOnTheLeftSide | Determines whether the page specified through the **RelationshipWithNodeGUID** property is on the left or right side of the relationship.<br>• If true, the control displays pages on the right side of the relationship.<br>• If false, the control displays pages on the left side of the relationship. | |
| RelationshipName | Specifies the type of the page relationship. Enter the code name of the relationship. | "isrelatedto" |

| CMS Base control properties | Description | Sample value |
|---|---|---|
| CacheDependencies | List of the cache keys on which the control's cached data depends. When the specified cache items change, the control clears its cache.<br>Each item (dependency) must be on one line.<br>If you leave this property empty, the control uses default dependencies.<br>**See also**: Setting cache dependencies, Configuring caching | cms. user\|all |
| CacheItem Name | Sets the name of the cache key used to store the control's content. If you leave the value empty, the system generates a default name containing variables, such as the control ID, the selected culture and the name of the user who loaded the page.<br>The system cache is shared by all pages in your application, so cache item names representing different data must be unique globally. If you have multiple controls that load the same data, you can share the cache keys between the controls (optimizes loading of content and avoids redundant data in the cache).<br>If the content displayed by the control depends on variables, such as URL parameters, you can set a custom name dynamically in the page's code behind.<br>**See also**: Caching the data of page components, Configuring caching | "CMSRepeaterNews" + Request. QueryString["id"]. ToString() |

| | | |
|---|---|---|
| CacheMinutes | Sets the number of minutes for which the control caches content retrieved from the database.<br>- 0 indicates that control does not cache content<br>- -1 indicates that the control uses the site-level content caching settings<br>Allows you to set up caching of content so that the control doesn't have to retrieve content from the database on each request.<br>The caching mechanism uses absolute expiration time. This means that cache items expire after a specified time period even if the page containing the control wasn't requested.<br>**See also**: Caching the data of page components, Configuring caching | |
| FilterControl | Gets or sets the filter control used to limit the data read by the control. | |
| FilterName | Gets or sets the code name of the filter control used to limit the data read by this control. | |
| OrderBy | Gets or sets the ORDER BY clause of the SQL query that the control uses to load data. | "NewsReleaseDate DESC" |
| SelectedColumns | Database table columns that the control loads for pages, separated by commas ( , ). If null or empty, the control loads all available columns. | |
| SiteName | Specifies the code name of the Kentico website for which the control loads data. | |
| StopProcessing | If true, the control stops all processing — does not load or display any data or other HTML output. | |
| TopN | Specifies the maximum number of database records that the control loads. | |
| WhereCondition | Gets or sets the WHERE clause of the SQL query that the control uses to loads data. | "Product Price > 100" |

⚠ **Note**: The CMSDataList is derived from the ASP.NET DataList control, so you can also set any of the base properties.

## Appearance and styling

You can modify the appearance of the CMSDataList control by setting the standard properties available for the ASP.NET DataList control (inherited through the BasicDataList).

The design of the items in the list is determined by the transformations specified through the **AlternatingTransformationName**, **TransformationName** and **SelectedItemTransformationName** properties, or by the code of item templates.