Kentico CMS Controls 4.0

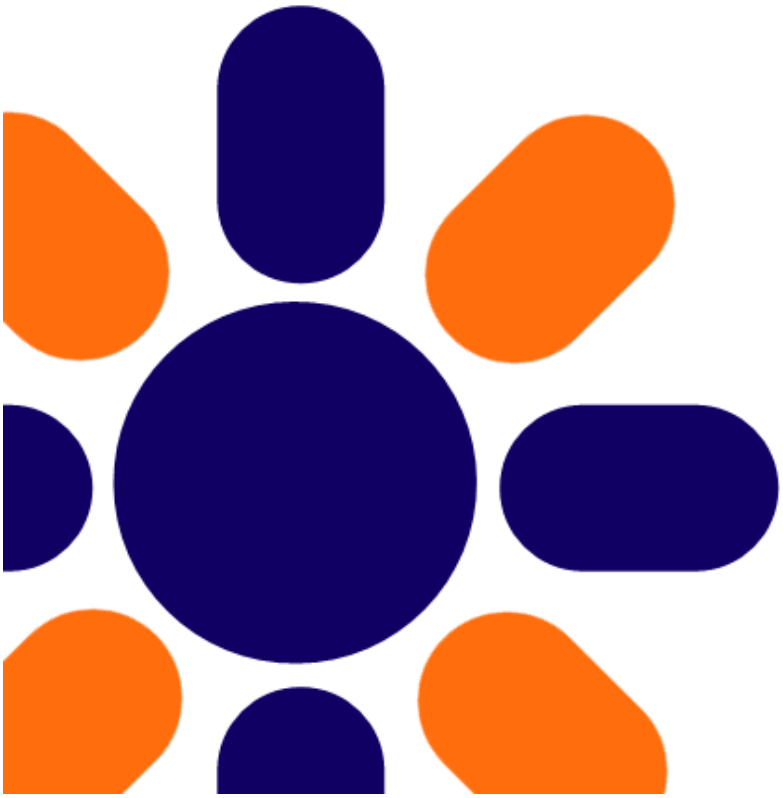
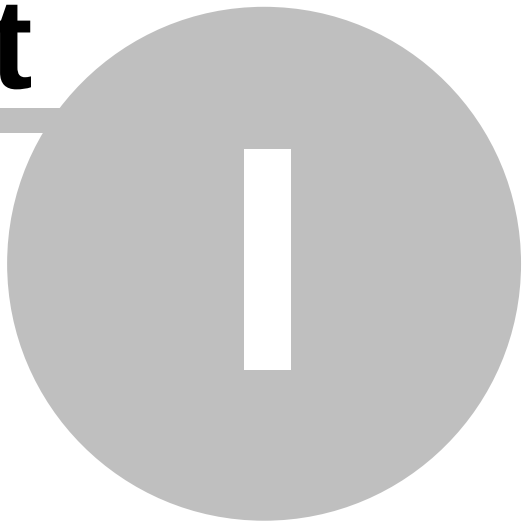


Table of Contents

© 2006-2008 Kentico Software

© 2006-2008 Kentico Software

Part



1 Kentico CMS Controls

1.1 Overview

Kentico CMS Controls are standard ASP.NET 2.0 server controls that can be used in Visual Studio 2005 or 2008. You can place them on your custom webparts or on page templates and pages that do not use the portal engine. Some of them can also be used outside Kentico CMS.

1.2 Path specification in web parts and controls

Many web parts and controls use a Path property that allows you to specify which documents should be displayed. This is the AliasPath property of the document. You can use either an exact path or you can use special characters for specifying multiple selection or relative paths:

Leaving the Path value empty

In many cases, **you can leave the Path value empty**. In this case, the Path value is set to the alias path of the currently displayed document.

In case of list controls/webparts, such as CMSRepater/Document repeater or CMSDataGrid/Document datalist, the path is set to <current alias path>/% if the current document is not of the same type as the required document in the ClassNames (document types) property. Otherwise, the path is set to the current alias path which leads to automatic selection of the current document.

Using wildcard characters % and _

You can use **% as a wildcard character for any number of characters**, which allows you to select all documents under specified site section.

Examples:

/ - only root
/% - all documents

/products - only the Products document.
/products/% - all documents under the Products document.

You can also use **_ as a wildcard character for a single character**.

Examples:

/product_ - selects documents /productA, /product1, etc.

© 2006-2008 Kentico Software

Using formatting string to get parts of the path

You can also use special expressions that **extract parts of the current path**, such as this.

Examples:

/{0}/{1}/% - all documents under the second level of the current path
/{0}/{1}/details - document Details under the second level of the current path

Using relative paths

You can use relative paths expressions to specify **sub-documents or parent documents**:

Examples:

./product - document product under the current path
../product - document product under the parent document of the current path
. - current path
.. - parent document of the current path
./% - all documents under the current path
../% - all documents under the parent document of the current path

1.3 Caching

What is Caching

Caching allows you to minimize the number of performed database queries. The server can store the data in memory and next time a user requests the content, the server returns content from memory instead of performing a resource-intensive database query. Caching can improve the performance of your Web site typically 10- to 100-times depending on your application.

The content **expires** after specified time span and must be retrieved from the database again. Each cached item has its name and the cache memory is common for all pages in your Web application.

Caching Support in Kentico CMS

You can manage the caching either by yourself in your code (please see the .NET Framework SDK documentation for more details) or you can leverage caching features of the following Kentico CMS Controls that are also used in CMS web parts:

- CMSBreadCrumbs
- CMSDataGrid
- CMSDataList
- CMSMenu
- CMSRepeater
- CMSSiteMap
- CMSTabControl
- CMSTreeMenu
- CMSViewer
- QueryDataGrid
- QueryRepeater
- QueryDataList

© 2006-2008 Kentico Software

All of these controls offer the following properties, that allow you to set up caching:

Property Name	Description	Sample Value
CacheItemName	Name of the cache item the control will use.	"products_" & request.querystring("categoryid")
CacheMinutes	Number of minutes the retrieved content is cached for. Zero indicates that the content will not be cached. -1 indicates the site-level settings should be used.	10

Cache Expiration Time

By setting the **CacheMinutes** property to a value higher than zero, the control starts to cache its source data. You can configure caching for all Kentico CMS content using the Cache content parameter in **Site Manager -> Settings -> Web site** section. If you set any particular value to the CacheMinutes property of control/web part, it overrides the global settings. If you leave the value empty or set to -1 (minus one), the global settings apply.

The caching mechanism of Kentico CMS Controls uses absolute expiration time instead of sliding expiration. It means that the cache item expires after specified period of time regardless if it was requested or not. It ensures that content is updated from the database in a regular interval.

Overriding the site-level caching settings

If you need to cache most of the content on your web site, but still want to have a single control/web part that doesn't use cache, you can configure caching as described in the previous paragraph and set value 0 (zero) to the CacheMinutes property of the particular control. It will override the site-level settings and disable caching for the single control/web part.

Cache Item Name

It's important to understand the **CacheItemName** property: Since the cache is common for all pages in your application, the cache item name should be unique not only for all pages, but also inside one page (for case you use several Kentico CMS Controls with caching on one page).

When you leave the CacheItemName property empty, the control automatically generates a name in this form:

URL including parameters|*control ID*

If the content displayed on the page depends on some parameter, such as URL parameter or role of the current user, you need to adjust the CacheItemName value accordingly.

© 2006-2008 Kentico Software

Example:
Your page products.aspx displays products according to the category that is passed through the URL parameter "category". You will need to use a code like this to ensure that the content will be cached "per category":

[C#]

```
CMSDataGrid.CacheItemName = "products_grid1_" + Request.QueryString["category"];
```

[VB.NET]

```
CMSDataGrid.CacheItemName = "products_grid1_" & Request.QueryString("category")
```

1.4 Displaying related documents

If you specify the related documents in the Relationships dialog of the Content module, you can display them using one of the following controls:

- CMSDataGrid
- CMSDataList
- CMSRepeater
- CMSViewer

All of them have three properties that need to be used in order to display only related documents (beside other properties, such as SelectNodesPath, SelectNodesClassName, etc.):

- RelationshipWithNodeGUID / Main document - NodeGUID value of the main document - it's typically the currently displayed document.
- RelationshipName / Relationship name - code name of the relationship.
- RelatedNodeIsOnTheLeftSide / Main document is on the left side - indicates if the main document is on the left or right side of the relationship.

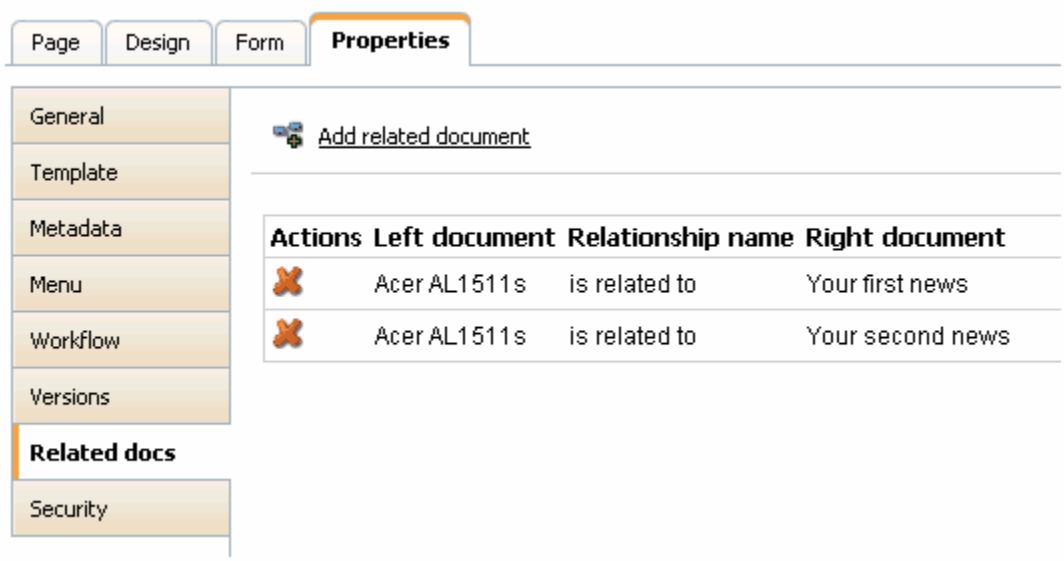
Example

The following example shows how to display news items related to the product.

1. Go to CMS Desk -> Content and click /Products/LCD Displays/Acer AL1511s, click Properties -> Related docs.

© 2006-2008 Kentico Software

2. Add two relationships with name **is related to** with documents /News/Your first news and /News/Your second news.



3. Edit the Products.aspx template and add a CMSRepeater control.

4. Set the following control properties:
   - Path: /% (we want to display related news items from the whole web site)
   - ClassNames: cms.news
   - TransformationName: cms.news.preview
   - RelationshipName: choose **Display documents related to the current document**, check the box **Main document is on the left side** and choose relationship name **is related to**.
   Save the changes.

© 2006-2008 Kentico Software

5. Click /Products/LCD Displays/Acer AL1511s and click Live site. You will see the list of related news items below the product displayed using the cms.news.preview transformation:

© 2006-2008 Kentico Software

1.5 Dynamic insertion of parameters in data web parts

The following controls support dynamically inserted parameters in the **WhereCondition** / **WHERE condition** property:

- QueryDatalist
- QueryDataGrid
- QueryRepeater

You can use the following expressions that are resolved at run-time:

Expression	Description
{%currentaliaspath%}	Alias path of the current page.
{%currentculturecode%}	Culture code of the user's preferred content culture.
{%currentsiteid%}	SiteID value of the current site.

##WHERE## macro expression

The custom query you are using must contain the **##WHERE##** expression that is dynamically replaced with value of the **WHERE condition** property value at run-time.

1.6 CMS Controls

Kentico CMS Controls is a set of ASP.NET **server controls** that allow you to **publish content from the Kentico CMS** database on your Web site. Some of them also provide additional features, such as search dialog.

Before you can use them, please make sure you configured your project as described in Configuring your project for Kentico CMS Controls.

Corporate Site sample required

The examples in this chapter assume that your Kentico CMS database contains data for the sample Corporate Site web site.

© 2006-2008 Kentico Software

Source Code

You can find the source code of the examples in the CMSControlsExamples subfolder under your project.

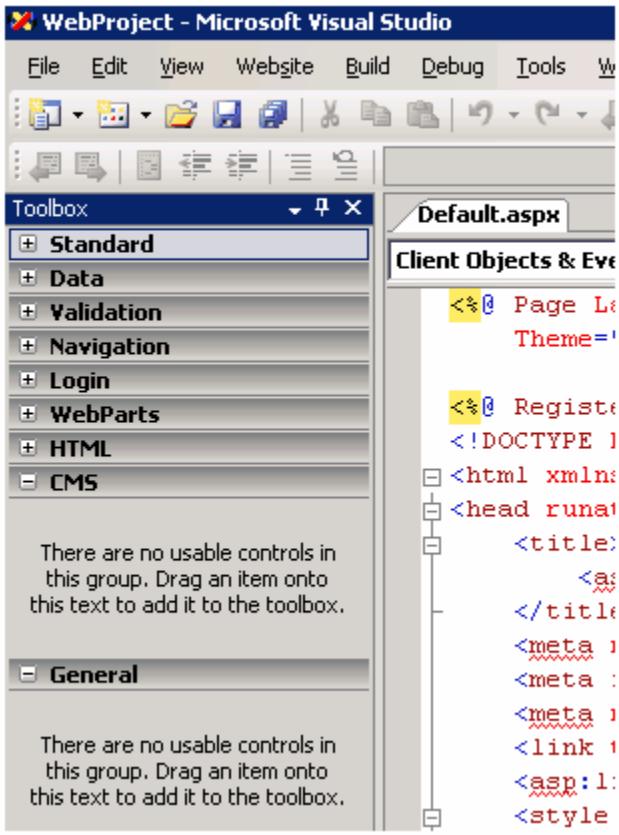
1.6.2 Configuring your project for Kentico CMS Controls

Before you start using Kentico CMS Controls in your ASP.NET project, you need to add the controls to the **Toolbox**:

1. Open the web site project in Visual Studio and open some ASPX page.

2. Right-click the **Toolbox** and choose **Add tab** from the context menu.

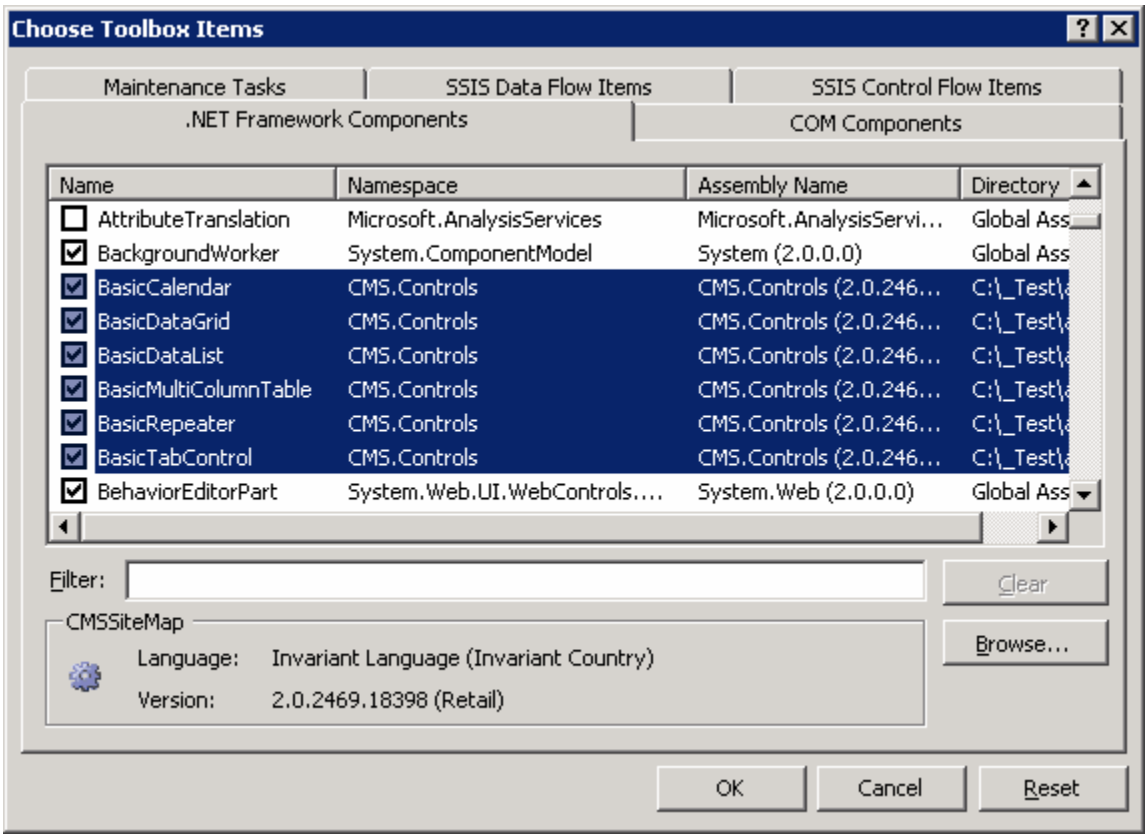
3. Type the name of the new tab (e.g. CMS) and press Enter:



4. Right-click the new tab and choose **Choose items...** from the context menu.

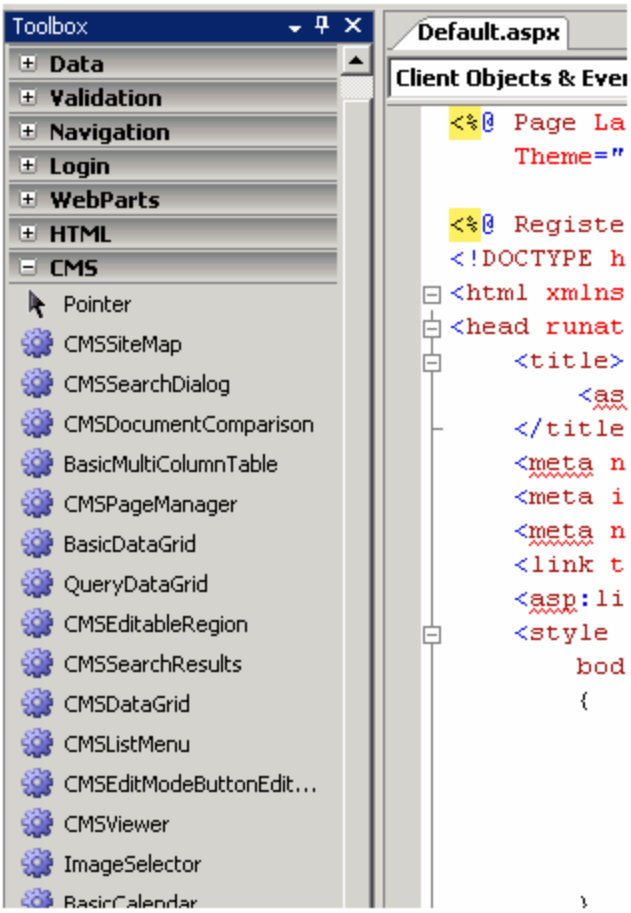
© 2006-2008 Kentico Software

5. In the **Choose Toolbox Items** dialog, click Browse and locate the **CMS.Controls.DLL** library in the **bin** folder under your web site. Click **Open** and then click **OK**.



© 2006-2008 Kentico Software

6. **The controls are now added to the Toolbox:**



7. Now you can easily drag and drop the controls on your Web form.

1.6.3 Using macro expressions in menu items

You can use macro expressions in menu item URL or JavaScript command (when you're editing the menu item in Kentico CMS Desk). These macros allow you to dynamically replace macro commands with specified values of the current menu item, such as alias path, id path, node name.

You only need to put **{%ColumnName%}** macros in the menu item URL or JavaScript attribute. The URL will look like this:

```
products.aspx?show=brand&aliaspath={%AliasPath%}
```

The menu will redirect user e.g. to URL products.aspx?show=brand&aliaspath=/MobileStore/Products/Nokia.

Please note: All apostrophes (') in source data are escaped to \' so that they do not break JavaScript.

© 2006-2008 Kentico Software

1.6.4 Using the CSSPrefix property for design of sub-menus

The CSSPrefix property allows you not only to use several menu controls with different styles on one page, but also to **specify style of menu sub-items at any chosen level**.

Here's an example of how to specify various style for particular menu levels (it can be used for both CMSMenu and CMSTreeMenu):

First, you need to specify the list of prefixes for particular levels using the CSSPrefix property:

CMSMenu1.CSSPrefix = "MainMenu;MainMenuSubMenu;MainMenuOtherLevels"

... now you define the following styles:

```
.MainMenuCMSMenu
... for menu control

.MainMenuCMSMenuItem
.MainMenuCMSMenuItemMouseUp
... etc. for the first level of the menu (level 0)

.MainMenuSubMenuCMSMenu
.MainMenuSubMenuCMSMenuItemMouseUp
... etc. for the second level of the menu (level 1)

.MainMenuOtherLevelsCMSMenu
.MainMenuOtherLevelsCMSMenuItemMouseUp
... etc. for all underlying levels of the menu (level 2 and all higher
levels)
```

© 2006-2008 Kentico Software

1.6.5 CMSBaseProperties - common properties

Property Name	Description	Sample Value
CacheItemName	Name of the cache item the control will use. By setting this name dynamically, you can achieve caching based on URL parameter or some other variable - simply put the value of the parameter to the CacheItemName property. If no value is set, the control stores its content to the item named "URL|ControlID".	"mycachename" + Request.QueryString["id"].ToString()
CacheMinutes	Number of minutes the retrieved content is cached for. Zero indicates that the content will not be cached. -1 indicates the site-level settings should be used. This parameter allows you to set up caching of content so that it's not retrieved from the database each time a user requests the page. The default value is retrieved from Site Settings.	10
ControlTagKey	Overrides the generation of the SPAN tag with custom tag. The default value is retrieved from Site Settings.	
OrderBy	ORDER BY part of the SQL statement.	"NewsReleaseDate DESC"
SiteName	Site code name.	"mywebsite"
StopProcessing	Indicates if processing of the control should be stopped and the control should not retrieve or display any data.	
WhereCondition	WHERE part of the SQL statement.	"ProductPrice > 100"

© 2006-2008 Kentico Software

1.6.6 CMSControlProperties - common properties

Inherits: CMSBaseProperties - common properties

Property Name	Description	Sample Value
CheckPermissions	Allows you to specify whether to check permissions of the current user. If the value is 'false' (default value) no permissions are checked. Otherwise, only nodes for which the user has read permission are displayed.	true
ClassNames	Classname value or several values separated by semicolon.	"cms.news" or "cms.news;cms.article"
CombineWithDefaultCulture	Indicates if the documents from the default culture version should be alternatively used.	true
CultureCode	Culture code, such as en-us. If not specified, it's read from the user's session or the default value is used.	"en-us"
MaxRelativeLevel	Relative level of child documents that should be selected. -1 selects all child documents.	
Path	Path of the documents to be displayed. See Path specification in web parts and controls for details.	See Path specification in web parts and controls for examples.
SelectOnlyPublished	Indicates if only published documents should be displayed.	
TreeProvider	Tree provider instance used to access data. If no TreeProvider is assigned, a new TreeProvider instance is created.	
WordWrap	Indicates if text displayed by the (navigation) control can use word wrapping.	

© 2006-2008 Kentico Software

1.6.7 CMSDataProperties - common properties

Inherits: CMSControlProperties - common properties

Property Name	Description	Sample Value
RelatedNodeIsOnTheLeftSide	If true, the returned nodes are on the right side of the relationship.	
RelationshipName	Name of the relationship.	"isrelatedto"
RelationshipWithNodeGUID	Select nodes with given relationship with given node. If you use 00000000-0000-0000-0000-000000000000 value, the relationships are not applied. If you use 11111111-1111-1111-1111-111111111111 value, the current node NodeGUID value is used automatically.	
SelectedItemTransformationName	Transformation name for selected item in format application.class.transformation.	"cms.news.preview"
SelectTopN	Select top N rows.	5

1.6.8 CMSMenuProperties - common properties

Inherits: CMSControlProperties - common properties

Property Name	Description	Sample Value
CSSPrefix	Specifies prefix of standard CMSMenu CSS classes. You can also use several values separated with semicolon (;) for particular levels.	"main;submenu1; submenu2"
HighlightAllItemsInPath	Indicates if all items in the unfolded path should be displayed as highlighted.	
SubmenuIndicator	Contains a path to image that will be used on the right of every item that contains subitems.	
UseAlternatingStyles	Indicates if alternating styles should be used for even and odd items in the same level of the menu.	

© 2006-2008 Kentico Software

1.6.9 CMSQueryProperties - common properties

Inherits: CMSBaseProperties - common properties

Property Name	Description	Sample Value
GeneralConnection	GeneralConnection instance. If it's not specified, a new instance of the GeneralConnection is created and returned.	
QueryName	Query name in format application.class.query.	"cms.news.selectlatest"
QueryParameters	Query parameters. The first dimension contains the name of the parameter in format @param, the second dimension contains its value.	
SelectTopN	Select top N rows.	5

1.6.10 BasicCalendar

The BasicCalendar control allows you to display a calendar with events, news and other date-based documents specified in the DataSource property value. It's inherited from the standard ASP.NET Calendar control, which means it provides advanced formatting capabilities and it allows you to display additional information for appropriate days.

Sea also: CMSCalendar

Inherits

Calendar (ASP.NET control)

Properties

Property Name	Description	Sample Value
DataMember	Name of the table when DataSet is used as a DataSource.	"MyTable"
DataSource	Data source with calendar events - either DataSet or DataTable object.	
DayField	Name of the field in the DataSource that contains the datetime value.	"NewsReleaseDate"
DayWithEventsStyle	Style of the day with some event.	
ItemTemplate	Template for displaying a day with event.	
NoEventsTemplate	Template for displaying a day without any event.	

© 2006-2008 Kentico Software

Design

You can modify the design of the calendar control by setting the standard properties of the ASP.NET Calendar control. You can find more details on particular properties in the .NET Framework documentation.

Example

This example will show you how to display a calendar with news items released on particular date.

- Create a new Web form.
- Drag and drop the BasicCalendar control on the form.
- Switch to the HTML mode and add the following code inside the BasicCalendar element. The **ItemTemplate** section specifies the look of the event that will be displayed in the calendar control. The **NoEventsTemplate** section specifies the look of the day without any event.

[C#], [VB.NET]

```
<ItemTemplate>
<br/>
  <a href='<%#
  ResolveUrl(CMS.CMSHelper.CMSContext.GetUrl(Convert.ToString(Eval("NodeAliasPath")),
     Convert.ToString(Eval("DocumentUrlPath")))) %>'>
  <%#Eval("NewsTitle")%>
  </a>
</ItemTemplate>
<NoEventsTemplate>
<br>
No Event
</NoEventsTemplate>
```

- Switch to the code behind and add the following code at the beginning of the code:

[C#]

```
using CMS.CMSHelper;
```

[VB.NET]

```
Imports CMS.CMSHelper
```

- Add the following code to the Page_Load method:

[C#]

```
BasicCalendar1.DataSource = TreeHelper.SelectNodes("/%", false, "cms.news", null,
"NewsReleaseDate", -1, true);
BasicCalendar1.DayField = "NewsReleaseDate";
BasicCalendar1.DataBind();
```

© 2006-2008 Kentico Software

[VB.NET]

```
BasicCalendar1.DataSource = TreeHelper.SelectNodes("/%", False, "cms.news", nothing,
"NewsReleaseDate", -1, True)
BasicCalendar1.DayField = "NewsReleaseDate"
BasicCalendar1.DataBind()
```

What you did

You have added the code that retrieves all news items from the Kentico CMS database and assigns them to the BasicCalendar control. You have specified the DayField that contains the date/time value. Then you called the BasicCalendar.DataBind method.

- Run the project. You should see a page like this:

July			August 2006			September
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30 No Event	31 No Event	1 No Event	2 No Event	3 Your first news	4 Your second news	5 No Event
6 No Event	7 No Event	8 No Event	9 No Event	10 No Event	11 No Event	12 No Event
13 No Event	14 No Event	15 No Event	16 No Event	17 No Event	18 No Event	19 No Event
20 No Event	21 No Event	22 No Event	23 No Event	24 No Event	25 No Event	26 No Event
27 No Event	28 No Event	29 No Event	30 No Event	31 No Event	1 No Event	2 No Event
3 No Event	4 No Event	5 No Event	6 No Event	7 No Event	8 No Event	9 No Event

1.6.11 BasicDataGrid

The BasicDataGrid control is inherited from the standard ASP.NET DataGrid control. It automatically ensures data binding, paging and sorting. You can use the standard DataGrid designer to set up BasicDataGrid style and behavior.

BasicDataGrid can be used with any bindable data source - it doesn't use Kentico CMS database or API.

See also: CMSDataGrid.

© 2006-2008 Kentico Software

Properties

Property Name	Description	Sample Value
DataBindByDefault	Indicates whether data binding should be performed by default.	
HideControlForZeroRows	Hides the control when no data is loaded. Default value is False.	
ProcessSorting	Indicates if sorting should be processed in DataView instead of sorting on the SQL level.	
SortAscending	Direction of sorting. Default value is True.	
SortField	Gets or sets the sort field. It can be used for setting the default sort field.	"NewsReleaseDate"
ZeroRowsText	Text to be shown when control is hidden by HideControlForZeroRows.	"No records found"
SetFirstPageAfterSortChange	Indicates if the current page should be set to the first page when the sorting is changed.	

Design

The design can be modified in the same way as the standard DataGrid control.

Example

This example will show you how to read the list of products and display it in the grid.

1. Create a new Web form.
2. Drag and drop the **BasicDataGrid** control on the form.
3. In the **Properties** window, click **Auto Format...** and choose some color schema.
4. In the **Properties** window, click **Property Builder...**, the **BasicDataGrid1 Properties** dialog appears.
   - On the **General** tab, check the **"Allow sorting"** box.
   - Now we will specify the columns that will be displayed. On the **Columns** tab:
     - Uncheck the **Create columns automatically at run time** box.
     - Add a new Bound Column from the **Available columns** list to the **Selected columns** list. Enter the following values in the appropriate fields:
       - Header text: Name
       - Data field: ProductName
       - Sort expression: ProductName
     - Add another Bound Column from the **Available columns** list to the **Selected columns** list. Enter the following values in the appropriate fields:
       - Header text: Price
       - Data field: ProductPrice
       - Sort expression: ProductPrice
- On the **Paging** tab check the box **Allow Paging**. Click **OK**.

© 2006-2008 Kentico Software

5. Add the following code at the beginning of the Web form code-behind:

[C#]

```
using CMS.CMSHelper;
```

[VB.NET]

```
Imports CMS.CMSHelper
```

What you did
You have included namespaces we will use.

6. Add the following code to the **Page_Load** method:

[C#]

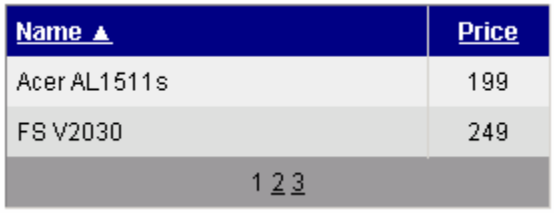
```
DataSet ds = TreeHelper.SelectNodes("/%", false, "CMS.Product", "", "ProductName", -1,
true);
BasicDataGrid1.DataSource = ds;
BasicDataGrid1.DataBind();
```

[VB.NET]

```
  Dim ds As DataSet = TreeHelper.SelectNodes("/%", False, "CMS.Product", "",
  "ProductName", -1, true)
  BasicDataGrid1.DataSource = ds
  BasicDataGrid1.DataBind()
```

What you did
You have added code that reads data from the database and provides them to the BasicDataGrid control.

7. Compile and run the project. You should see a page like this:



1.6.12 BasicDataList

The BasicDataList control is inherited from the standard ASP.NET DataList control. It automatically ensures data binding. You can use the common ASP.NET DataList tags to set up BasicDataList style and behavior - please see the .NET Framework documentation for details.

BasicDataList can be used with any bindable data source - it doesn't use Kentico CMS database or API.

© 2006-2008 Kentico Software

Unlike BasicRepeater, BasicDataList allows you to display data in several columns.

Please note: If you want to display **data from Kentico CMS**, please use the CMSDataList control that provides a more convenient way.

Please note

If you want to display **data from Kentico CMS**, please use the CMSDataList control that provides a more convenient way.

Inherits: DataList (ASP.NET control)

Properties

Property Name	Description	Sample Value
DataBindByDefault	Indicates whether data binding should be performed by default.	
HideControlForZeroRows	Hides the control when no data is loaded. The default value is False.	
ZeroRowsText	Text to be shown when the control is hidden by HideControlForZeroRows.	"No records found."

Design

The design can be modified in the same way as for the standard DataList control.

Example

This example will show you how to read a list of products (from the sample Corporate Site) and display it using the BasicDataList control.

1. Create a new Web form.
2. Drag and drop the BasicDataList control on the form. Change its **RepeatColumns** property value to 3.
3. Switch to the HTML mode and add the following code inside the BasicDataList tags:

© 2006-2008 Kentico Software

[C#], [VB.NET]

```
<itemtemplate>
<div style="width: 100%">
<h3>
<%# Eval("ProductName") %>
</h3>
<img alt="<%# Eval("ProductName") %>_img" src='<%#
ResolveUrl("~/CMSPages/GetFile.aspx?guid=" + Eval("ProductPhoto").ToString()) %>' />
</div>
</itemtemplate>
```

What you did

You have defined a template for one product displayed by the DataList control. The control dynamically replaces the <%# ... %> tags with values of the current record. This is repeated for each record in the SQL query result. The ResolveUrl("~") command generates the path to the root of your web project.

4. Add the following code at the beginning of the Web form code-behind:

[C#]

```
using CMS.CMSHelper;
```

[VB.NET]

```
Imports CMS.CMSHelper
```

What you did
You have included namespaces we will use..

5. Add the following code to the **Page_Load** method:

[C#]

```
DataSet ds = TreeHelper.SelectNodes("/%", false, "CMS.Product", "", "ProductName", -1, true);
BasicDataList1.DataSource = ds;
BasicDataList1.DataBind();
```

[VB.NET]

```
Dim ds as DataSet = TreeHelper.SelectNodes("/%", false, "CMS.Product", "", "ProductName", -1, true)
BasicDataList1.DataSource = ds
BasicDataList1.DataBind()
```

What you did
You have added code that reads data from the database and provides them to the BasicDataList control.

6. Compile and run the project. You should see a page like this:

© 2006-2008 Kentico Software

Acer AL1511s

HP nx6110

FS V2030

IPAQ HW6510

© 2006-2008 Kentico Software

1.6.13 BasicMultiColumnTable

The BasicMultiColumnTable control allows you to display a table where documents are rendered column-by-column instead of common row-by-row model used in the DataGrid and other controls. This control is useful if you want e.g. compare parameters of various products.

Sea also: CMSDocumentComparison

Properties

Property Name	Description	Sample Value
DataSource	DataTable object containing rows to be displayed.	
RenderFalseValueUsingEmptyTemplate	Indicates if false value of boolean type-columns should be displayed using the template for empty value.	
TableParams	Params of the table to be displayed. A string array of size (x, 4) where x is the number of rows of the table and the second parameter is: 0 - Row caption 1 - Row source column 2 - Non-empty-column template (if omitted, the value of the source column is used) 3 - Empty-column template (if omitted, the " " value is used) When specifying the template, you can use {%fieldname%} tags to insert value of any field of the source data. If you need to create a separator row, set only the (x, 0) value to the separator text and do not set the other values (x, 1), (x, 2) and (x, 3) - they must be nothing (null).	Standard row: (0, 0) = "Product weight:" (0, 1) = "Weight" (0, 2) = "{%weight%} kg" (0, 3) = "N/A" Separator row: (0, 0) = "This is separator"

© 2006-2008 Kentico Software

Design

You can modify the design of the table control by setting the following properties:

Property Name	Description	Sample Value
TableCellSpacing	Table cell spacing.	0
TableCellPadding	Table cell padding.	0
TableCellCssClass	Table cell (TD) CSS class.	"MulticolumnTableCell"
TableCssClass	Table CSS class.	"MulticolumnTable"
TableFirstColumnCellCssClass	CSS class of the first table columns' cells (TD).	"MulticolumnTableFirstColumnCell"
TableFirstRowCellCSSClass	Table cell (TD) CSS class for the first row of the table.	"MulticolumnTableFirstRowCell"
TableFirstRowCssClass	First table row (TR) CSS class.	"MulticolumnTableFirstRow"
TableRowCssClass	Table row (TR) CSS class.	"MulticolumnTableRow"
TableSeparatorCellCssClass	CSS class of the separator cell (TD).	"MulticolumnTableSeparatorCell"
TableSeparatorRowCssClass	CSS class of the separator row (TR).	"MulticolumnTableSeparatorRow"

Example

This example will show you how to display a product comparison:

1. Create a new web form.
2. Drag and drop the **BasicMultiColumnTable** control on the form.
3. Switch to the code behind and add the following code at the beginning of the code:

[C#]

```
using CMS.CMSHelper;
```

[VB.NET]

```
Imports CMS.CMSHelper
```

© 2006-2008 Kentico Software

4. Add the following code to the Page_Load method:

[C#]

```
string[,] tableParams = new string[3, 4];
DataSet ds = null;

// set data source
ds = TreeHelper.SelectNodes("/%", false, "cms.product", "", "ProductName", -1, true);
BasicMultiColumnTable1.DataSource = ds.Tables[0];

// define table header with product names
tableParams[0, 0] = "Product:";
tableParams[0, 1] = "productName";
// add row with product description
tableParams[1, 0] = "Description:";
tableParams[1, 1] = "ProductDescription";
// add row with product price
tableParams[2, 0] = "Price:";
tableParams[2, 1] = "productPrice";
tableParams[2, 2] = "USD {%ProductPrice%}";
tableParams[2, 3] = "N/A";
BasicMultiColumnTable1.TableParams = tableParams;
```

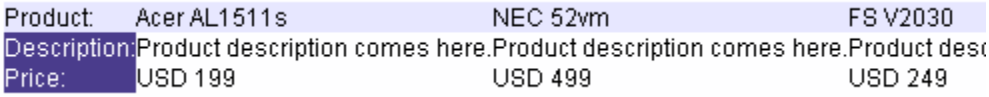
[VB.NET]

```
Dim tableParams(2, 3) As String
Dim ds As DataSet

'set data source
ds = TreeHelper.SelectNodes("/%", False, "cms.product", "", "ProductName", -1, True)
BasicMultiColumnTable1.DataSource = ds.Tables(0)

'define table header with product names
tableParams(0, 0) = "Product:"
tableParams(0, 1) = "ProductName"
'add row with product description
tableParams(1, 0) = "Description:"
tableParams(1, 1) = "ProductDescription"
'add row with product weight
tableParams(2, 0) = "Price:"
tableParams(2, 1) = "ProductPrice"
tableParams(2, 2) = "USD {%ProductPrice%}"
tableParams(2, 3) = "N/A"
BasicMultiColumnTable1.TableParams = tableParams
```

What you did

You have added the code that retrieves all products from the Kentico CMS database and assigns them to the **BasicMultiColumnTable** control. The tableParams array defines how the data should be displayed.

5. Compile and run the project. You should see a page like this:



© 2006-2008 Kentico Software

1.6.14 BasicRepeater

The BasicRepeater control is inherited from the standard ASP.NET Repeater control. It ensures data binding automatically. You can use the common ASP.NET Repeater tags to set up BasicRepeater style and behavior - please see the .NET Framework documentation for details.

BasicRepeater can be used with any bindable data source - it doesn't use Kentico CMS database or API.

Please note: If you want to display documents from Kentico CMS, please use the CMSRepeater control that provides a more convenient way.

Properties

Property Name	Description	Sample Value
DataBindByDefault	Indicates whether data binding should be performed by default.	
HideControlForZeroRows	Hides the control when no data is loaded. Default value is False.	
ZeroRowsText	Text to be shown when the control is hidden by HideControlForZeroRows.	"No records found."

Design

The design can be modified in the same way as for the standard Repeater control.

Example

This example will show you how to read a list of products and display it using the repeater.

- Create a new Web form.
- Drag and drop the **BasicRepeater** control on the form.
- Switch to the Source mode and add the following code inside the BasicRepeater tags:

[C#], [VB.NET]

```
<itemtemplate>
  <h3>
  <%# Eval("ProductName") %>
  </h3>
  <img src='<%# ResolveUrl("~/CMSPages/GetFile.aspx?guid=" +
Eval("ProductPhoto").ToString()) %>' />
</itemtemplate>
```

What you did

You have defined a template for one record displayed by the repeater control. The control dynamically replaces the <%# ... %> tags with values of the current record. This is repeated for each record in the datasource.

© 2006-2008 Kentico Software

- Add the following code at the beginning of the Web form code-behind:

[C#]

```
using CMS.CMSHelper;
```

[VB.NET]

```
Imports CMS.CMSHelper
```

What you did

You have included namespaces we will use.

- Add the following code to the Page_Load method:

[C#]

```
DataSet ds = TreeHelper.SelectNodes("/%", false, "CMS.Product", "", "ProductName", -1, true);
this.BasicRepeater1.DataSource = ds;
this.BasicRepeater1.DataBind();
```

[VB.NET]

```
Dim ds As DataSet = TreeHelper.SelectNodes("/%", False, "CMS.Product", "", "ProductName", -1, True)
BasicRepeater1.DataSource = ds
BasicRepeater1.DataBind()
```

© 2006-2008 Kentico Software

What you did

You have added code that reads documents from the database and provides them to the BasicRepeater control.

- Compile and run the project. You should see a page like this:

Acer AL1511s

FS V2030

1.6.15 BasicTabControl

The BasicTabControl control displays several tabs according to provided data. BasicTabControl doesn't use Kentico CMS database or API.

Please note: If you want to display a tab menu based on the data from Kentico CMS, please use the CMSTabControl control.

Properties

Property Name	Description	Sample Value
SelectedTab	Index of the selected tab.	2
TabControlLayout	Horizontal or vertical layout.	TabControlLayoutEnum.Horizontal
Tabs	4- or 7-dimensional array of tabs: tabs[0, 0] = title tabs[0, 1] = OnClick JavaScript tabs[0, 2] = URL tabs[0, 3] = tooltip tabs[0, 4] = left image URL tabs[0, 5] = center image URL tabs[0, 6] = right image URL **Please note:** 1. The image URLs in dimensions 4, 5 and 6 are optional. 2. If you specify the center image URL, the image is displayed instead of the title.	tabs[0, 0] = " Home "; tabs[0, 1] = "alert('It is very simple!');"; tabs[0, 2] = "http://www.kentico.com"; tabs[0, 3] = "Some tooltip"; tabs[0, 4] = "leftimage.gif"; tabs[0, 5] = "centerimage.gif"; tabs[0, 6] = "rightimage.gif";
UrlTarget	If URL for tab items is set, this property specifies target frame for all URLs.	"_blank"
UseClientScript	Indicates if client script should be generated for each tab.	
TabControlIdPrefix	Prefix that will be used for all IDs in the HTML code rendered by the BasicTabControl. It's useful if you need to place multiple tab controls on the same page.	"FirstTab"

© 2006-2008 Kentico Software

Design

You can modify the design using the following CSS Classes:

Class Name	Description
TabControlTable	Table that contains the tabs (<TABLE> tag).
TabControlRow	Table row (<TR> tag).
TabControl	Tab item (<TD> tag).
TabControlSelected	Selected tab item (<TD> tag).
TabControlLink	Tab item link - for case a URL is specified (<A> tag).
TabControlLinkSelected	Selected tab item link - for case a URL is specified (<A> tag).
TabControlLeft	Left side of the tab item (<TD> tag).
TabControlRight	Right side of the tab item (<TD> tag).
TabControlSelectedLeft	Left side of the selected tab item (<TD> tag).
TabControlSelectedRight	Right side of the selected tab item (<TD> tag).

Example

This example will show you how to display a simple tab-menu.

- Create a new Web form.
- Drag and drop the BasicTabControl control on the form.
- Switch to the Source mode and add the following CSS styles inside the <BODY> tag. It will modify the appearance of the tabs.

[C#], [VB.NET]

```
<style>
.TabControlTable { FONT-SIZE: 14px; FONT-FAMILY: Arial,Verdana }
.TabControlRow { }
.TabControl { BORDER-RIGHT: black 1px solid; BORDER-TOP: black 1px solid; FONT-WEIGHT:
bold; BACKGROUND: #e7e7ff; BORDER-LEFT: black 1px solid; CURSOR: hand; COLOR: black }
.TabControlSelected { BORDER-RIGHT: black 1px solid; BORDER-TOP: black 1px solid;
FONT-WEIGHT: bold; BACKGROUND: #4a3c8c; BORDER-LEFT: black 1px solid; CURSOR: default;
COLOR: white }
.TabControlLinkSelected { COLOR: white; TEXT-DECORATION: none }
.TabControlLink { COLOR: black; TEXT-DECORATION: none }
.TabControlLeft { WIDTH: 1px }
.TabControlRight { WIDTH: 0px }
.TabControlSelectedLeft { WIDTH: 1px }
.TabControlSelectedRight { WIDTH: 0px }
</style>
```

© 2006-2008 Kentico Software

- Add the following table code just after the BasicTabControl tags. It will display a stripe under the tabs.

[C#], [VB.NET]

```
<hr style="width:100%; height:2px; margin-top:0px;" />
```

- Switch to the page code-behind and add the following code to the Page_Load method:

[C#]

```
string[,] tabs = new string[3, 7];

tabs[0, 0] = " Home ";
tabs[0, 1] = "alert('It is very simple!');";
tabs[0, 2] = "http://www.kentico.com";
tabs[1, 0] = " Products ";
tabs[1, 2] = "http://www.comparesql.com";
tabs[2, 0] = " Contact ";
tabs[2, 2] = "http://www.syncserverfiles.com";
tabs[2, 3] = "Some tooltip";

BasicTabControl1.Tabs = tabs;
BasicTabControl1.SelectedTab = 0;
BasicTabControl1.UrlTarget = "_blank";
BasicTabControl1.UseClientScript = true;
```

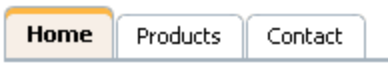
[VB.NET]

```
Dim tabs(2, 6) As String

tabs(0, 0) = " Home "
tabs(0, 1) = "alert('It\'s very simple!');"
tabs(0, 2) = "http://www.kentico.com"
tabs(1, 0) = " Products "
tabs(1, 2) = "http://www.comparesql.com"
tabs(2, 0) = " Contact "
tabs(2, 2) = "http://www.syncserverfiles.com"
tabs(2, 3) = "Some tooltip"

BasicTabControl1.Tabs = tabs
BasicTabControl1.SelectedTab = 0
BasicTabControl1.UrlTarget = "_blank"
BasicTabControl1.UseClientScript = True
```

What you did

You have added code that creates an array of tab items and assigns it to the BasicTabControl control. It also selects the first tab and sets the target frame to "_blank".

- Compile and run the project. You will see a page like this:



© 2006-2008 Kentico Software

1.6.16 CMSBreadCrumbs

The CMSBreadCrumbs control allows you to display current user's position within the Web site in format

Item 1 > Item 2 > Item 3

where Item X is name of the document in the path.

Inherits: CMSControlProperties - common properties

Properties

Property Name	Description	Sample Value
BreadCrumbSeparator	Character(s) that separate the bread crumbs. You can use HTML code here.	">>"
DefaultPath	Path to the node whose path should be displayed. This value is used in case no Path value is provided and no alias path is provided through friendly URL.	"/home"
ShowCurrentItem	Indicates if the current (last) item should be displayed.	
ShowCurrentItemAsLink	Indicates if the current (last) item should be displayed as a link.	
StartingPath	Starting part of the path.	"/products"
UrlTarget	Specifies target frame for all URLs.	"_blank"
LoadDataAutomaticaly	Indicates if data for the control should be loaded automatically. By default, the data is loaded automatically. If you set this property to false, you can supply custom DataSet to the DataSource property and then call the ReloadData(false) method.	
RenderedHTML	Allows you to get or set the HTML code rendered by the control. You need to set this property before the Render event - e.g. in the OnLoad event.	

© 2006-2008 Kentico Software

Methods

Method Name	Description
ReloadData	Reloads the data. If the **forceLoad** parameter is set to false and the custom value is assigned to the DataSource property, the properties of the CMSListMenu control are not used and only the data from the DataSource are used.

Design

You can modify the design using the following CSS styles:

Class Name	Description
CMSBreadCrumbsLink	Link (A element) in the bread crumbs path.
CMSBreadCrumbsCurrentItem	Style of the last item representing the current location (it's not a link).

CMSBreadCrumbs Example

This example will show you how to display user's current location within Web site structure. It assumes that you have configured your project for Kentico CMS Controls.

- Create a new Web form.
- Drag and drop the **CMSBreadCrumbs** control on the form.
- In the Properties window, set the following property value:
    - Path: /Products/Notebooks

Compile and run the project. You should see a page like this:

1.6.17 CMSCalendar

The CMSCalendar control allows you to display a calendar with events, news and other date-based documents from the Kentico CMS database. It inherits the BasicCalendar control.

Data Source

Data retrieved using the SelectDocuments query of the specified document type.

Inherits: BasicCalendar, CMSControlProperties - common properties

© 2006-2008 Kentico Software

Own Properties

Property Name	Description	Sample Value
DataSource	Gets or sets a DataSet containing values used to populate the items within the control. This value needn't be set.	
NoEventTransformationName	No event transformation name in format application.class.transformation.	"cms.news.noevent"
RelatedNodeIsOnTheLeftSide	If true, the returned nodes are on the right side of the relationship.	
RelationshipName	Name of the relationship.	"isrelatedto"
RelationshipWithNodeId	Select nodes with given relationship with given node.	10
TransformationName	Transformation name in format application.class.transformation.	"cms.news.event"
TreeProvider	Tree provider instance used to access data. If no TreeProvider is assigned, a new TreeProvider instance is created.	

Design

You can modify the design of the calendar control by setting the standard properties of the ASP.NET Calendar control. You can find more details on particular properties in the .NET Framework documentation.

Example

This example will show you how to display a calendar with news items released on particular date.

1. Create a new Web form.
2. Drag and drop the CMSCalendar control on the form.
3. Set the following properties - they specify what should be displayed.

   - **DayField** = "NewsReleaseDate"
   - **ClassNames** = "cms.news"
   - **Path** = "/%"
   - **TransformationName** = "cms.news.calendarevent"
   - **NoEventTransformationName** = "cms.news.calendarnoevent"

© 2006-2008 Kentico Software

4. Run the project. You should see a page like this:

Jul	August 2006					Sep
Sun	Mon	Tue	Wed	Thu	Fri	Sat
30 No event	31 No event	1 No event	2 No event	3 Your first news (8/3/2006) *Summary comes here.*	4 Your second news (8/4/2006) *Summary comes here.*	5 No event
6 No event	7 No event	8 No event	9 No event	10 No event	11 No event	12 No event
13 No event	14 No event	15 No event	16 No event	17 No event	18 No event	19 No event
20 No event	21 No event	22 No event	23 No event	24 No event	25 No event	26 No event
27 No event	28 No event	29 No event	30 No event	31 No event	1 No event	2 No event
3 No event	4 No event	5 No event	6 No event	7 No event	8 No event	9 No event

1.6.18 CMSDataGrid

The CMSDataGrid displays a table based on Kentico CMS data. It inherits the BasicDataGrid control. It automatically ensures databinding, paging and sorting.

It allows you to display Kentico CMS documents specified by path, depth, document type and WHERE condition. The CMSDataGrid control displays content without writing any additional code.

You can use the common DataGrid designer to set up CMSDataGrid style and behavior.

Please note: If you want to display data using **custom query**, please use the QueryDataGrid control.

Data Source

Data retrieved using the SelectDocuments query of the specified document type.

Inherits: CMSDataProperties - common properties, BasicDataGrid

Design

The design can be modified in the same way as the standard ASP.NET DataGrid control.

Example

This example will show you how to display a list of all mobile phones in the grid using the CMSDataGrid control.

© 2006-2008 Kentico Software

1. Create a new Web form.

2. Drag and drop the CMSDataGrid control on the form.

3. In the **Properties** window, click **Auto Format...** and choose some color schema.

4. In the **Properties** window, click **Property Builder...**, the **CMSDataGrid1 Properties** dialog appears.
- On the **General** tab check the **"Allow sorting"** box.
- Now we will specify the columns that will be displayed. On the **Columns** tab:
- Uncheck the **Create columns automatically at run time** box.
- Add a new Bound Column from the **Available columns** list to the **Selected columns** list. Enter the following values in the appropriate fields:
- Header text: Product Name
- Data field: ProductName
- **Sort expression:** ProductName
- Add a new Bound Column from the **Available columns** list to the **Selected columns** list. Enter the following values in the appropriate fields:
- Header text: Price
- Data field: ProductPrice
- Sort expression: ProductPrice
- On the **Paging** tab check the box **Allow Paging**. Click **OK**.

5. In the **Properties** window set the following property values:
- ClassNames: cms.product
- MaxRelativeLevel: -1
- Path: /products/%

6. Compile and run the project. You should see a page like this:

Name ▼	Price
Acer AL1511s	199
FS V2030	249
HP nx6110	349
IPAQ HW6510	249
IPAQ RX1950	299
NEC 52vm	499
1	

1.6.19 CMSDataList

The **CMSDataList** control is inherited from the BasicDataList control.

It allows you to display part of the CMS content specified by its path, depth, document template, WHERE condition and ORDER BY clause. The CMSDataList control displays content without writing any additional code. Unlike CMSRepeater, the CMSDataList control allows you to display content in several columns.

Please note: If you want to display data using **custom query**, please use the QueryDataList control.

© 2006-2008 Kentico Software

Data Source

Data retrieved using the SelectDocuments query of the specified document template.

Inherits: CMSDataProperties - common properties, BasicDataList
See also: DataPager

Properties

Property Name	Description	Sample Value
AlternatingTransformationName	Transformation name in format application.class.transformation applied to alternating items.	"cms.product.alternatepreview"
EnablePaging	Enables paging.	
PagerControl	Represents pager control. See DataPager for more details.	
TransformationName	Transformation name in format application.class.transformation.	"cms.product.preview"
NestedRepeaterID	ID of the nested CMSRepeater control.	"CMSRepeaterNested"
NestedDataListID	ID of the nested CMSDataList control.	"CMSDataListNested"

Please note: you can find an example of datalist/repeater nesting in CMSRepeater chapter.

Design

The design can be modified in the same way as for the standard ASP.NET DataList control and by the transformations defined in the **`AlternatingTransformationName`**, **`TransformationName`** and **`SelectedItemTransformationName`** properties.

Example

This example will show you how to read a list of documents and display it using the repeater.

1. Create a new Web form.

2. Drag and drop the **CMSDataList** control on the form.

3. In the **Properties** window set the following property values:
- ClassNames: cms.product
- MaxRelativeLevel: -1
- OrderBy: ProductName
- Path: /products/%
- RepeatColumns: 2
- TransformationName: cms.product.preview
- SelectedItemTransformationName: cms.product.default

Please note: as you defined TransformationName property, it's not necessary to define the standard ItemTemplate element of the DataList control.

4. Compile and run the project. You should see a page like this:

© 2006-2008 Kentico Software

1.6.20 CMSDocumentComparison

The CMSDocumentComparison control allows you to display several documents in table columns for comparison. This control is useful if you want to compare parameters of various products. The user can add the available documents to the comparison table and remove selected displayed documents from the comparison.

CMSDocumentComparison control uses the BasicMultiColumnTable control.

Data Source

Kentico CMS documents specified using standard properties (Path, ClassNames, etc.).

Inherits: CMSControlProperties - common properties

Properties

Property Name	Description	Sample Value
AddButton	Add button control.	
BasicMultiColumnTable	BasicMultiColumnTable control instance.	
DocumentList	Drop-down list with all comparable documents.	
DropDownListColumn	Column to be displayed in the drop-down list.	"ProductName"
RemoveAllText	Text of the link for removing all compared documents.	"Remove all"
RemoveText	Text of the link for removing selected document.	"Remove"
TableParams	See BasicMultiColumnTable	
TagKey	Overrides the generation of the SPAN tag with custom tag.	

Design

© 2006-2008 Kentico Software

See BasicMultiColumnTable for details on design.

You can also modify the properties of the appropriate controls (BasicMultiColumnTable, AddButton, DocumentList).

Example

This example will show you how to display a product comparison.

1. Create a new web form.

2. Drag and drop the **CMSDocumentComparison** control on the form. Set the following properties:
   - ClassNames="cms.product"
   - Path="/Products/%"
   - DropDownListColumn="ProductName"

3. Add the following code to the Page_Load method:

[C#]

```
// define table rows
string[,] tableParameters = new string[4, 4];
tableParameters[0, 0] = "Product:";
tableParameters[0, 1] = "ProductName";
tableParameters[1, 0] = "Description:";
tableParameters[1, 1] = "ProductDescription";
tableParameters[2, 0] = "Price:";
tableParameters[2, 1] = "ProductPrice";
tableParameters[2, 2] = "USD {%ProductPrice%}";
tableParameters[2, 3] = "N/A";
this.CMSDocumentComparison1.TableParams = tableParameters;

// set CSS styles
this.CMSDocumentComparison1.BasicMultiColumnTable.TableFirstRowCellCSSClass =
"MulticolumnTableFirstRow";
this.CMSDocumentComparison1.BasicMultiColumnTable.TableFirstColumnCellCssClass =
"MulticolumnTableFirstColumnCell";
```

© 2006-2008 Kentico Software

[VB.NET]

```
'define table rows
Dim tableParameters(3, 3) As String
tableParameters(0, 0) = "Product:"
tableParameters(0, 1) = "ProductName"
tableParameters(1, 0) = "Description:"
tableParameters(1, 1) = "ProductDescription"
tableParameters(2, 0) = "Price:"
tableParameters(2, 1) = "ProductPrice"
tableParameters(2, 2) = "USD {%ProductPrice%}"
tableParameters(2, 3) = "N/A"
Me.CMSDocumentComparison1.TableParams = tableParameters

'set CSS styles
CMSDocumentComparison1.BasicMultiColumnTable.TableFirstRowCellCSSClass =
"MulticolumnTableFirstRow"
CMSDocumentComparison1.BasicMultiColumnTable.TableFirstColumnCellCssClass =
"MulticolumnTableFirstColumnCell"
```

What you did

You have specified the rows of the BasicMultiColumnTable control and styles.

4. Compile and run the project. You should see a page like this:

1.6.21 CMSDocumentValue

This control displays the specified value of the currently displayed document. It's useful if you need to display e.g. the current document name on the page.

Properties

Property Name	Description	Sample Value
AttributeName	Name of the field to be displayed.	"DocumentName"
ClassNames	List of document types for which the value should be displayed, separated with a semicolon (;).	"cms.article;cms. menuitem"
FormattingString	.NET formatting expression for displaying the value.	"Name: {0}"

You can place this control into web parts, ASPX page templates or layouts.

© 2006-2008 Kentico Software

Example

You only need to add the following code inside your ASPX page or web part:

[C#], [VB.NET]

```
<cc1:CMSDocumentValue ID="docVal" runat="server" AttributeName="DocumentName"
FormattingString="Document name: {0}" />
```

1.6.22 CMSEditModeButtonAdd

The CMSEditModeButtonAdd control displays a button that is shown in the editing mode and allows content editors to add a new document when they click it. It provides an intuitive way of adding new documents.

Properties

Property Name	Description	Sample Value
ClassName	Document type in format cms.article that specifies the type of the document that should be created.	"cms.article"
Path	Alias path of the parent document under which the new document is created. If omitted, the documents are added under the currently selected document.	"/whitepapers"
Text	Custom caption of the button. If it's not set, the default text "Add new" is displayed.	"Add new article"

Design

You can modify the look of the control using the following CSS styles:

Class Name	Description
CMSEditModeButtonAdd	CSS style of the <A> tag.

Example

You only need to add the following code inside your ASPX page or web part:

[C#], [VB.NET]

```
<cc1:CMSEditModeButtonAdd ID="CMSEditModeButtonAdd1" runat="server"
ClassName="cms.article" />
```

The displayed button looks like this:

© 2006-2008 Kentico Software

1.6.23 CMSEditModeButtonEditDelete

The CMSEditModeButtonEditDelete control displays a button that is shown in the editing mode and allows content editors to edit or delete document when they click it. It provides an intuitive way of editing/deleting documents.

Properties

Property Name	Description	Sample Value
Path	Alias path of the document to be edited/deleted.	"/whitepapers/myfirstpaper"
EditText	Custom caption of the button. If not set, the default text "Edit" is displayed.	"Edit article"
DeleteText	Custom caption of the button. If not set, the default text "Delete" is displayed.	"Delete article"

Design

You can modify the look of the control using the following CSS styles:

Class Name	Description
CMSEditModeButtonEdit	CSS style of the <A> tag.
CMSEditModeButtonDelete	CSS style of the <A> tag.

Example

You only need to add the following code inside your ASPX page or web part:

[C#], [VB.NET]

```
<cc1:CMSEditModeButtonEditDelete runat="server" id="btnEditDelete" Path="/news/news1" />
```

If you want to display the buttons in the transformation, you can use the following code:

[C#], [VB.NET]

```
<cc1:CMSEditModeButtonEditDelete runat="server" id="btnEditDelete" Path='<%#
Eval("NodeAliasPath") %>' />
```

The displayed button looks like this:

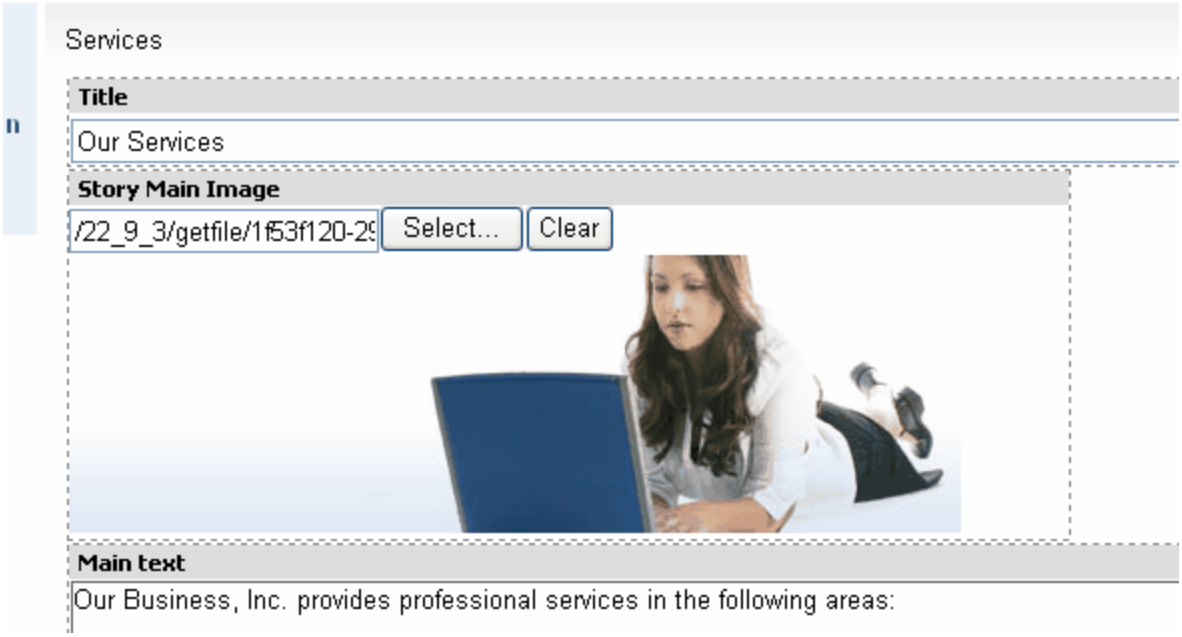
© 2006-2008 Kentico Software

1.6.24 CMSEditableImage

Please note:

Please note: This control is compatible only with ASPX page templates. On portal pages, use web part Editable image instead of this control

The **Editable image** web part displays an editable region that enables content editors to choose an image.



Content

Property Name	Description	Sample Value
ImageWidth	Image width in pixels - the image will be resized to this width.	400
ImageHeight	Image height in pixels - the image will be resized to this height.	300
ImageTitle	Title displayed in the editing mode.	"Main section image"
AlternateText	ALT text of the image displayed on the web site.	"Main image"
DisplaySelectorTextBox	Indicates if textbox with image path should be displayed in the editing mode.	

1.6.25 CMSEditableRegion

© 2006-2008 Kentico Software

Please note:

This control is compatible only with ASPX page templates. On portal pages, use web part Editable text instead of this control.

The CMSEditableRegion control defines the part of the page that should be editable. It ensures displaying in both edit and view mode. The flow of data to/from CMSEditableRegion control is managed by CMSPageManager control. There must be just one CMSPageManager control and any number of CMSEditableRegion controls.

Data Source

Data is set/read by the CMSPageManager control. The CMSEditableRegion control doesn't communicate with database directly.

Properties

Property Name	Description	Sample Value
DialogHeight	Height of the control.	400
DialogWidth	Width of the control.	500
HtmlAreaToolbar	HTML editor toolbar set name.	"default"
HtmlAreaToolbarLocation	HTML editor toolbar location.	
MaxLength	Maximum length of the content (in number of characters).	10
MinLength	Minimum length of the content (in number of characters).	2
RegionTitle	Control title which is displayed in the editable mode.	"Main text"
RegionType	Type of server control which is displayed in the editable region. It can be a textbox, textarea of HTML editor.	CMSEditableRegionTypeEnum.TextBox
WordWrap	Wrap the text if using text area field.	
InheritContent	Indicates if content of the editable region should be inherited from the parent page (menu item) document.	

Please note: the other properties are used by the CMSPageManager control only and shouldn't be modified in your code.

© 2006-2008 Kentico Software

CSS Styles

Class Name	Description
CMSEditableRegionEdit	Style of the main <TABLE> element.
CMSEditableRegionTitle	Style of the <TD> element containing the title.
CMSEditableRegionError	Style of the <TD> element containing the error message.

1.6.26 CMSListMenu

CMSListMenu control allows you to create a large variety of menus. It renders <UL> and <LI> tags and the design depends only on your CSS style sheet. This menu control provides several advantages:

- It's based only on CSS styles which makes it highly configurable.
- It renders shorter HTML code than CMSMenu.
- It's fully XHTML compliant.
- The list-based menu is better accessible.
- You can create drop-down menu using the list-based menu and CSS almost without any JavaScript.
- It automatically displays standard UL/LI listing with links if the browser does not support CSS styles so that the user can still navigate on the web site.

However, **it requires advanced knowledge of CSS** as it doesn't render any specific layout by itself.

It allows you to display part of the CMS content specified using its path, depth, document type and WHERE condition.

See also: Using macro expressions in menu items, Using the CSSPrefix property for design of sub-menus

Inherits: CMSMenuProperties - common properties

© 2006-2008 Kentico Software

Properties

Property Name	Description	Sample Value
DisplayHighlightedItemAsLink	Indicates if the highlighted item should be displayed as a link.	
DisplayOnlySelectedPath	Specifies whether all submenus should be displayed or just submenu under highlighted (selected) item.	
FirstItemCssClass	Specifies CSS class for the first item in every menu level.	"ListMenuFirstItem"
HighlightAllItemsInPath	Indicates if all items in the unfolded path should be displayed as highlighted.	
HighlightedNodePath	Path of the item that will be highlighted like it was selected. The path type must be the same as PathType. If you omit this value, the control automatically uses the current alias path from the "aliaspath" querystring parameter.	"/products/nokia"
LastItemCssClass	Specifies CSS class for the last item in every menu level.	"ListMenuLastItem"
OnMouseOutScript	OnMouseOutScript script for menu items. You can use macro expressions here.	"alert(this.innerText);" "alert('{%nodealiaspath%}')"
OnMouseOverScript	OnMouseOver script for menu items. You can use macro expressions here.	"alert(this.innerText);" "alert('{%nodealiaspath%}')"
RenderCssClasses	Indicates if CSS classes should be rendered for every element. If set to false, only first and last item of menu level will use CSS class.	
RenderItemID	Indicates if unique ID should be rendered for every menu item.	
ItemIdPrefix	Prefix placed before each item ID. You can use it to keep ID's unique if you have several CSS list menu web parts on the same page.	"submenu"
HoverCSSClassName	Name of the surrounding CSS class that is used to define styles for the hover effect if you want to render a drop-down menu.	Horizontal
UrlTarget	Specifies target frame for all URLs.	"_blank"
RenderLinkTitle	Specifieds if document name should be rendered as a TITLE tag of the link (for better accessibility).	
RenderImageAlt	Indicates if ALT attribute should be rendered for images used in the menu (for XHTML compatibility).	
LoadDataAutomaticaly	Indicates if data for the control should be loaded automatically. By default, the data is	

© 2006-2008 Kentico Software

	loaded automatically. If you set this property to false, you can supply custom DataSet to the DataSource property and then call the ReloadData(false) method.	
RenderedHTML	Allows you to get or set the HTML code rendered by the control. You need to set this property before the Render event - e.g. in the OnLoad event.	

Methods

Method Name	Description
ReloadData	Reloads the data. If the **forceLoad** parameter is set to false and the custom value is assigned to the DataSource property, the properties of the CMSListMenu control are not used and only the data from the DataSource are used.

Design

You can modify the design using the following CSS styles if the RenderCssClasses property is set to true:

Class Name	Description
CMSListMenuUL	UL element style
CMSListMenuLI	LI element style
CMSListMenuLink	A element style
CMSListMenuHighlightedLI	LI element style of a highlighted item

See also Using the CSSPrefix property for design of sub-menus to find out how to set different styles for particular menu levels.

Examples

Simple menu without styles

This example will show you how to display a simple menu based on the CMS content.Then, you will see how different CSS styles can render different menus.

- Create a new Web form.
- Drag and drop the CMSListMenu control on the form.
- Switch to the HTML mode and add the following line at the beginning of the page:

© 2006-2008 Kentico Software

- Set the following DOCTYPE just before the HTML element:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Switch back to the Design mode.
- Set the Path property value to /%
- Run the project. You should see a UL/LI-based menu like this:

- Home
- News
- Services
  - Service 1
  - Service 2
- Products
  - LCD Displays
  - Notebooks
  - PDAs
- Partners
- Contact

Creating a Horizontal Drop-Down Menu Using CSS Styles

Now we will modify the CSS style of the menu so that it displays a horizontal drop-down menu.

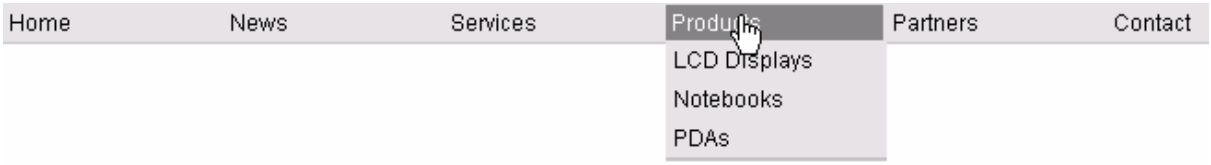
- Open the web page in VS.NET, in HTML mode and add the following style definition inside the <HEAD> element:

```
<STYLE type="text/css" media="screen">
.Horizontal { BORDER-RIGHT: #c2c2c2 1px solid; BORDER-TOP: #c2c2c2 1px solid; FONT-SIZE:
12px; FLOAT: left; BORDER-LEFT: #c2c2c2 1px solid; WIDTH: 100%; BORDER-BOTTOM: #c2c2c2 1px
solid; FONT-FAMILY: Arial; BACKGROUND-COLOR: #e2e2e2 }
.Horizontal UL { PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; MARGIN: 0px;
WIDTH: 100%; PADDING-TOP: 0px; LIST-STYLE-TYPE: none }
.Horizontal LI { BORDER-RIGHT: #e2e2e2 1px solid; PADDING-RIGHT: 0px; BORDER-TOP: #e2e2e2
1px solid; DISPLAY: inline; PADDING-LEFT: 0px; FLOAT: left; PADDING-BOTTOM: 0px;
BORDER-LEFT: #e2e2e2 1px solid; PADDING-TOP: 0px; BORDER-BOTTOM: #e2e2e2 1px solid }
.Horizontal A { PADDING-RIGHT: 3px; DISPLAY: block; PADDING-LEFT: 3px; PADDING-BOTTOM:
2px; MARGIN: 0px; WIDTH: 112px; COLOR: black; PADDING-TOP: 2px; BACKGROUND-COLOR: #e2e2e2;
TEXT-DECORATION: none }
.Horizontal A:hover { BACKGROUND: #808080 no-repeat right bottom; COLOR: white }
.Horizontal UL UL { Z-INDEX: 500; WIDTH: 120px; BORDER-BOTTOM: #c2c2c2 2px solid;
POSITION: absolute }
.Horizontal UL UL LI { CLEAR: left; DISPLAY: block; POSITION: relative }
.Horizontal UL UL UL { BORDER-RIGHT: #c2c2c2 2px solid; LEFT: 100%; BORDER-BOTTOM: white
0px solid; TOP: -1px }
.Horizontal UL UL { DISPLAY: none }
.Horizontal UL LI:hover UL UL { DISPLAY: none }
.Horizontal UL UL LI:hover UL UL { DISPLAY: none }
.Horizontal UL LI:hover UL { DISPLAY: block }
.Horizontal UL UL LI:hover UL { DISPLAY: block }
.Horizontal UL UL UL LI:hover UL { DISPLAY: block }
</STYLE>
```

- Set the HoverCSSClassName property value to: **Horizontal**

© 2006-2008 Kentico Software

- Save the page and display it in the web browser. You should see a menu like this



Creating a Vertical Drop-Down Menu Using CSS Styles

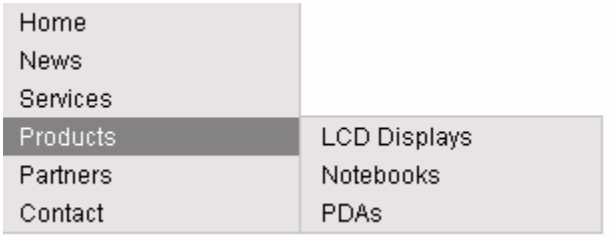
Now we will modify the CSS style of the menu so that it displays a vertical drop-down menu.

- Open the web page in VS.NET, in HTML mode and add the following style definition inside the <HEAD> element:

```
<STYLE type="text/css" media="screen">
.Vertical { BORDER-RIGHT: #c2c2c2 1px solid; BORDER-TOP: #c2c2c2 1px solid; FONT-SIZE:
12px; BORDER-LEFT: #c2c2c2 1px solid; WIDTH: 150px; BORDER-BOTTOM: #c2c2c2 1px solid;
FONT-FAMILY: Arial; BACKGROUND-COLOR: #e2e2e2 }
.Vertical UL { PADDING-RIGHT: 0px; PADDING-LEFT: 0px; PADDING-BOTTOM: 0px; MARGIN: 0px;
PADDING-TOP: 0px; LIST-STYLE-TYPE: none }
.Vertical LI { POSITION: relative; FLOAT: left; WIDTH: 100% }
.Vertical A { PADDING-RIGHT: 0px; BACKGROUND-POSITION: 0px 50%; DISPLAY: block;
PADDING-LEFT: 10px; PADDING-BOTTOM: 2px; MARGIN: 0px; WIDTH: 140px; COLOR: black;
PADDING-TOP: 2px; BACKGROUND-REPEAT: no-repeat; BACKGROUND-COLOR: #e2e2e2;
TEXT-DECORATION: none }
.Vertical A:hover { BACKGROUND: #808080 no-repeat 0px 50%; COLOR: white }
.Vertical UL UL { BORDER-RIGHT: #c2c2c2 1px solid; BORDER-TOP: #c2c2c2 1px solid; Z-INDEX:
100; LEFT: 100%; BORDER-LEFT: #c2c2c2 1px solid; WIDTH: 100%; BORDER-BOTTOM: #c2c2c2 1px
solid; POSITION: absolute; TOP: -1px }
#Vertical1 UL { DISPLAY: none }
#Vertical1 LI:hover UL UL { DISPLAY: none }
#Vertical1 UL LI:hover UL UL { DISPLAY: none }
#Vertical1 LI:hover UL { DISPLAY: block }
#Vertical1 UL LI:hover UL { DISPLAY: block }
#Vertical1 UL UL LI:hover UL { DISPLAY: block }
</STYLE>
```

- Set the HoverCSSClassName property value to: **Vertical**

- Save the page and display it in the web browser. You should see a menu like this:



© 2006-2008 Kentico Software

1.6.27 CMSMenu

The CMSMenu control allows you to display multi-level DHTML menu based on data from Kentico CMS. It encapsulates the skmMenu control. This is a free menu control for ASP.NET that can be downloaded from http://skmmenu.com.

It allows you to display part of the CMS content specified using its path, depth, document type and WHERE condition. The items are sorted by MenuItemOrder and MenuItemCaption values.

See also: Using macro expressions in menu items, Using the CSSPrefix property for design of sub-menus

Inherits: CMSMenuProperties - common properties

Properties

Property Name	Description	Sample Value
MouseCursor	Represents Cursor for menu (skmMenu) control.	MouseCursor.Default MouseCursor.Cursor
ExternalScriptPath	Path of the external .js file with skmMenu scripts. The default path is ~/cmsscripts/skmmenu.js.	"~/myscripts/skmmenu.js"
HighlightedMenuItem	Highlighted menu item. You can use it to set its style.	
HighlightedNodePath	Path of the item that will be highlighted like it was selected.	"/products/nokia"
Layout	Represents Layout for menu (skmMenu) control.	MenuLayout.Vertical MenuLayout.Horizontal
MenuControl	Menu (skmMenu) control that ensures rendering.	
RenderItemName	Indicates if ItemName attribute should be rendered.	
Padding	CMSMenu table padding.	1
Spacing	CMSMenu table spacing.	2
SeparatorText	Text of the separator placed between menu items of the first menu level.	"|"
SeparatorCssClass	CSS class of the separator cell (TD element).	
RenderImageAlt	Indicates if ALT attribute should be rendered for images used in the menu (for XHTML compatibility).	
EnableMouseUpDownClass	Indicates if the menu should render different CSS classes for mouse-up and mouse-down events.	

Design

© 2006-2008 Kentico Software

You can modify the design using the following CMSMenu properties.

Class Name	Description
CMSMenu.MenuControl.CssClass	Style of the whole control.
CMSMenu.MenuControl.DefaultCssClass	Default style of menu item. This style is applied unless you specify another style for particular (highlighted) item.
CMSMenu.MenuControl.DefaultMouseDownCssClass	Default style of the menu item when it's clicked.
CMSMenu.MenuControl.DefaultMouseOverCssClass	Default style of the menu item when you move mouse over it.
CMSMenu.MenuControl.DefaultMouseUpCssClass	Default style of the menu item when mouse up event occurs.
CMSMenu.HighlightedMenuItem.CssClass	Style of the highlighted item.
CMSMenu.HighlightedMenuItem.MouseDownCssClass	Style of the highlighted item when it's clicked.
CMSMenu.HighlightedMenuItem.MouseOverCssClass	Style of the highlighted item when you move mouse over it.
CMSMenu.HighlightedMenuItem.MouseUpCssClass	Style of the highlighted item when mouse up event occurs.
CMSMenu.MenuControl.DefaultTarget	This is rather a "control behavior", but it's an important property - it allows you to specify the target frame of the menu item links.

You can also use the following **CSS classes**:

Class Name	Description
CMSMenu	CSS class of the menu table.
CMSMenuItem	CSS class of the menu item.
CMSMenuItemMouseDown	CSS class of the menu item when mouse button is down.
CMSMenuItemMouseOver	CSS class of the menu item when user moves mouse cursor over the menu item.
CMSMenuItemMouseUp	CSS class of the menu item when mouse button is released.
CMSMenuHighlightedMenuItem	CSS class of the highlighted menu item.
CMSMenuHighlightedMenuItemMouseDown	CSS class of the highlighted menu item when mouse button is down.
CMSMenuHighlightedMenuItemMouseOver	CSS class of the highlighted menu item when user moves mouse cursor over the menu item.
CMSMenuHighlightedMenuItemMouseUp	CSS class of the highlighted menu item when mouse button is released.

See also: Using the CSSPrefix property for design of sub-menus

Example

© 2006-2008 Kentico Software

This example will show you how to display a simple DHTML menu based on the CMS content. It assumes that you have configured your project for Kentico CMS Controls.

- Create a new Web form.
- Drag and drop the CMSMenu control on the form.
- In the HTML mode, add the following CSS styles inside the <BODY> tag. It will modify the appearance of the menu.

[C#], [VB.NET]

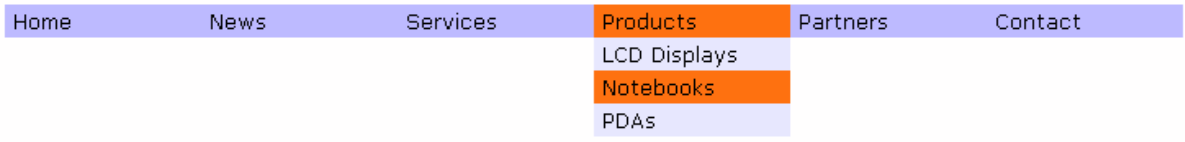
```
<style type="text/css">
/* horizontal menu - main menu and sub-menu*/
.MainCMSMenu { BORDER-RIGHT: 0px; TABLE-LAYOUT: fixed; BORDER-TOP: 0px; BORDER-LEFT: 0px;
WIDTH: 100px; BORDER-BOTTOM: 0px; BACKGROUND-COLOR: #b8bafe }
.MainCMSMenuItem { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.MainCMSMenuItemMouseUp { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.MainCMSMenuItemMouseOver { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: white; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: #4a3c8c }
.MainCMSMenuItemMouseDown { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.MainCMSMenuHighlightedMenuItem { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana;
BACKGROUND-COLOR: #ff7315 }
.MainCMSMenuHighlightedMenuItemMouseUp { PADDING-RIGHT: 15px; PADDING-LEFT: 5px;
FONT-SIZE: 10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black;
PADDING-TOP: 2px; FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.MainCMSMenuHighlightedMenuItemMouseOver { PADDING-RIGHT: 15px; PADDING-LEFT: 5px;
FONT-SIZE: 10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black;
PADDING-TOP: 2px; FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.MainCMSMenuHighlightedMenuItemMouseDown { PADDING-RIGHT: 15px; PADDING-LEFT: 5px;
FONT-SIZE: 10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black;
PADDING-TOP: 2px; FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.SubCMSMenuItem { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt; BACKGROUND:
#e7e7ff; PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY:
verdana }
.SubCMSMenuItemMouseUp { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.SubCMSMenuItemMouseOver { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: white; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: green }
.SubCMSMenuItemMouseDown { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.SubCMSMenuHighlightedMenuItem { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana;
BACKGROUND-COLOR: #ff7315 }
.SubCMSMenuHighlightedMenuItemMouseUp { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE:
10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.SubCMSMenuHighlightedMenuItemMouseOver { PADDING-RIGHT: 15px; PADDING-LEFT: 5px;
FONT-SIZE: 10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black;
PADDING-TOP: 2px; FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.SubCMSMenuHighlightedMenuItemMouseDown { PADDING-RIGHT: 15px; PADDING-LEFT: 5px;
FONT-SIZE: 10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black;
PADDING-TOP: 2px; FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }

/* specifying different styles for particular items */
.MainCMSMenuItemMouseOver#CMSMenu1-000 { BACKGROUND-COLOR: #4a3c8c }
.MainCMSMenuItemMouseOver#CMSMenu1-001 { BACKGROUND-COLOR: #5a4c9c }
.MainCMSMenuItemMouseOver#CMSMenu1-002 { BACKGROUND-COLOR: #6a5cac }
.MainCMSMenuItemMouseOver#CMSMenu1-003 { BACKGROUND-COLOR: #7a6cbc }
.MainCMSMenuItemMouseOver#CMSMenu1-004 { BACKGROUND-COLOR: #8a7ccc }
.MainCMSMenuItemMouseOver#CMSMenu1-005 { BACKGROUND-COLOR: #9a8cdc }

/* vertical menu - all menu levels */
.CMSMenu { BORDER-RIGHT: 0px; TABLE-LAYOUT: fixed; BORDER-TOP: 0px; BORDER-LEFT: 0px;
```

© 2006-2008 Kentico Software

```
WIDTH: 100px; BORDER-BOTTOM: 0px; BACKGROUND-COLOR: #b8bafe }
.CMSMenuItem { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt; PADDING-BOTTOM:
2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.CMSMenuItemMouseUp { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.CMSMenuItemMouseOver { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: white; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: #4a3c8c }
.CMSMenuItemMouseDown { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana }
.CMSMenuHighlightedMenuItem { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE: 10pt;
PADDING-BOTTOM: 2px; WIDTH: 100px; COLOR: black; PADDING-TOP: 2px; FONT-FAMILY: verdana;
BACKGROUND-COLOR: #ff7315 }
.CMSMenuHighlightedMenuItemMouseUp { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE:
10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.CMSMenuHighlightedMenuItemMouseOver { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE:
10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
.CMSMenuHighlightedMenuItemMouseDown { PADDING-RIGHT: 15px; PADDING-LEFT: 5px; FONT-SIZE:
10pt; PADDING-BOTTOM: 2px; WIDTH: 100px; CURSOR: hand; COLOR: black; PADDING-TOP: 2px;
FONT-FAMILY: verdana; BACKGROUND-COLOR: #ff7315 }
</style>
```

- Switch back to the Design mode.
- In the Properties window, set the following property values:
  - Path: /%

- Run the project. You should see a page like this:



© 2006-2008 Kentico Software

1.6.28 CMSPageManager

Please note:

This control is compatible only with ASPX page templates, do not use it on Portal engine templates. For portal pages, use Editable text and editable image web parts instead of CMSEditableRegion and CMSEditableImage controls.

The CMSPageManager control is used for pages with editable regions. It ensures loading and saving content to the database. It also displays the "Save" dialog. CMSPageManager manages the flow of data to/from CMSEditableRegions.

Data Source

Content is loaded from the nearest Page (menu item) document in the alias path specified through URL or through the DefaultPageAliasPath property. The content is stored in the following format:

```
<content>
<region id=”ID of the CMSEditableRegion control related to this content section”>
<!CDATA[ content of the editable region ]>
</region>
<region id=”...”>
<!CDATA[ content of the editable region ]>
</region>
</content>
```

© 2006-2008 Kentico Software

Properties

Property Name	Description	Sample Value
CacheItemName	Name of the cache item the control will use. By setting this name dynamically, you can achieve caching based on URL parameter or some other variable - simply put the value of the parameter to the CacheItemName property. If no value is set, the control stores its content to the item named "URL|ControlID".	"homepage_pagemanager"
CacheMinutes	Number of minutes the retrieved content is cached for. Zero indicates that the content will not be cached. -1 indicates the site-level settings should be used. This parameter allows you to set up caching of content so that it's not retrieved from the database each time a user requests the page.	10
CheckPermissionsForUserID	Allows you to specify the UserID of the current user. If the value is 0 (default value) no permissions are checked. Otherwise, only nodes for which the user has read permission are displayed.	126
CmsDeskPath	CMS Desk virtual path. If no value is provided, it's read from CMSDeskVirtualPath web.config parameter or it uses default value "~/cmsdesk".	"~/cmsdesk"
CombineWithDefaultCulture	Indicates if the results should be combined with default language versions in case the translated version is not available. This property is applied only if you do not set TreeProvider property manually.	False
CurrentPageAliasPath	Alias path that is used for retrieving editable regions content.	"/products/category1"
DefaultPageAliasPath	Default path that is used if no alias path is provided in the query string or through friendly URL.	"/home"
IsEditable	Indicates if the page is in editable mode. (read only)	true
PreferredCultureCode	Code of the preferred culture (en-us, fr-fr, etc.). If it's not specified, it is read from the CMSPreferredCulture session variable and then from the CMSDefaultCultureCode configuration key.	"en-us"
TreeProvider	Tree provider instance. If it's not provided, it's created automatically.	

© 2006-2008 Kentico Software

CSS Styles

Class Name	Description
CMSPageManagerError	Style of the error label.
CMSPageManagerLabel	Style of the standard label.
CMSPageManagerTextLink	Style of the link.
CMSPageManagerTDLabel	Style of the TD element that contains text with save confirmation message.

1.6.29 CMSRepeater

The CMSRepeater control inherits from the BasicRepeater control. It allows you to display part of the CMS content specified using its path, depth, document template, WHERE condition and ORDER BY clause. The CMSRepeater control displays content without writing any additional code.

Inherits: BasicRepeater, CMSDataProperties - common properties
See also: DataPager

Data Source

Data retrieved using the SelectDocuments query of the specified document type.

Properties

Property Name	Description	Sample Value
AlternatingTransformationName	Transformation name in format application.class.transformation applied to alternating items.	"cms.news.previewalternate"
ItemSeparator	Item separator between displayed records.	" "
PagerControl	DataPager control used for paging.	
TransformationName	Transformation name in format application.class.transformation.	"cms.news.preview"
NestedRepeaterID	ID of the nested CMSRepeater control.	"CMSRepeaterNested"
NestedDataListID	ID of the nested CMSDataList control.	"CMSDataListNested"

Design

The design can be modified using the transformations.

© 2006-2008 Kentico Software

Example

This example will show you how to read a list of news and display it using the repeater.

- Create a new Web form.
- Drag and drop the **CMSRepeater** control on the form.
- Switch to the HTML edit mode and add the following line at the beginning of the page:
- In the Properties window set the following property values:
    - ClassNames: cms.news
    - OrderBy: NewsReleaseDate DESC
    - Path: /%
    - SelectedItemTransformationName: cms.news.default
    - TransformationName: cms.news.preview

- Compile and run the project. You should see a page like this:

Displaying a nested (hierarchical) repeater/datalist

This example explains how you can display a hierarchical repeater/datalist. The hierarchical repeater consists of the main repeater and the nested repeater. You can use it for example to display a list of product categories and a preview of products in each category. You can combine nested repeaters and datalists as you need.

You can use this approach for both server controls and web parts.

- Add CMSRepeater/Repeater web part to your page.
- Set the following properties:
    - Path: /{0}/%
    - ClassNames/Document types: cms.menuitem
    - TransformationName/Transformation: CMS.MenuItem.Category
    - NestedDataListID/Nested datalist ID: CMSDataList1
- Go to Site Manager -> Development -> Document types -> Page (menu item) -> Transformation and define the following ASCX transformation with name **category**:

```
<h1><%# Eval("DocumentName") %></h1>
<p>
<cc1:CMSDataList ID="CMSDataList1" runat="server" ClassNames="cms.product"
       TransformationName="cms.product.preview" RepeatColumns="2" >
</cc1:CMSDataList>
</p>
```

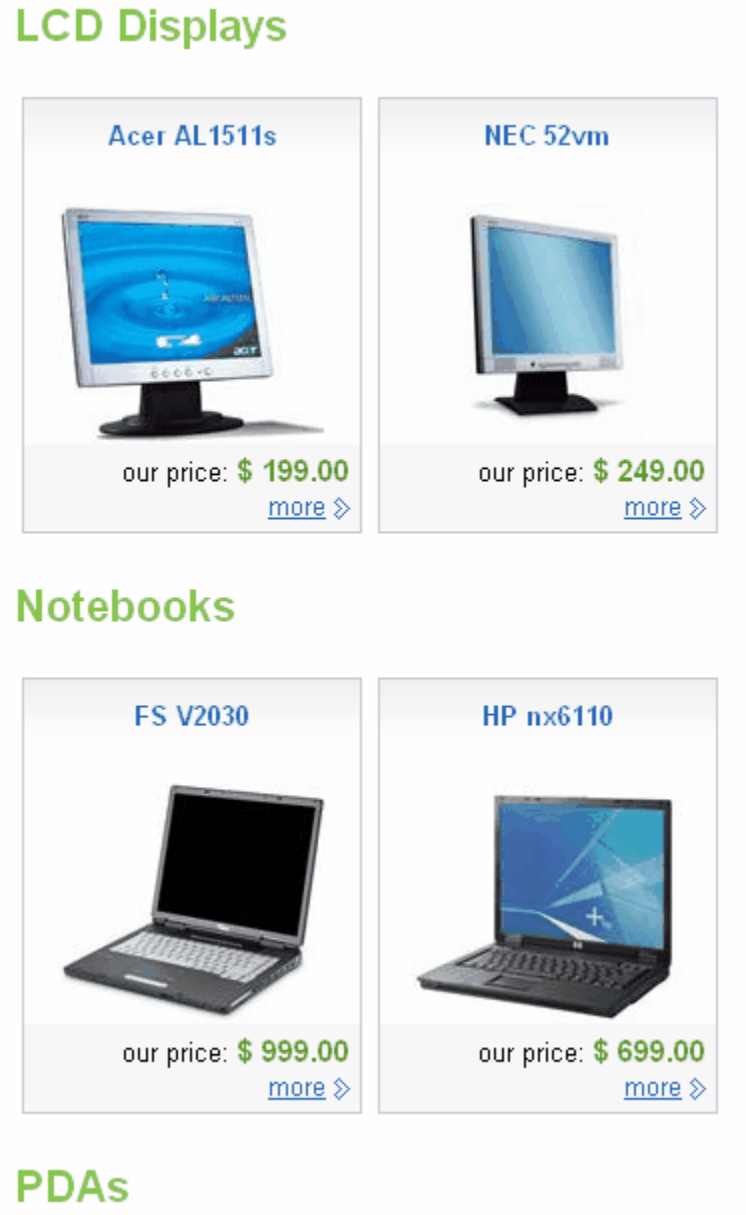
The transformation contains the nested datalist control that displays the documents of type **product** using the transformation **cms.product.preview**. Please note that the path is not specified - it's supplied dynamically by the parent repeater control/web part.

If you need to dynamically set properties of a nested control, you have to set its DelayedLoading

© 2006-2008 Kentico Software

property to 'True'. Please note that this setting can cause problems with ViewState.

- Go to the live site. You will see a page like this:



© 2006-2008 Kentico Software

1.6.30 CMSSearchDialog, CMSSearchResults

The **CMSSearchDialog** control allows users to enter searched words. The user can also (optionally) specify the search scope (where to search) and search mode (how to search).

CMSSearchDialog can be easily used with CMSSearchResults control that displays the search results according to the provided parameters from the CMSSearchDialog.

Note: Both controls can be used separately. Also, you can receive search results using the CMS.TreeProvider.Search() method.

Data Source

The CMSSearchDialog is not connected to any data source - it only communicates with user. The search method in the CMSSearchResults dialog uses the pre-defined queries "searchtree" in the document template definitions. It combines all results and returns them as one table.

CMSSearchDialog Properties

Property Name	Description	Sample Value
ShowSearchMode	Indicates if search mode settings should be displayed.	
ShowSearchScope	Indicates if search scope settings should be displayed.	
SearchExpression	Entered word(s) to be searched for.	"asp.net cms"
SearchMode	Search mode - any word, all words or exact phrase.	SearchModeEnum.AnyWord
SearchScope	Returns 0 for all content or 1 for the current section.	0

CMSSearchDialog Events

Event Name	Description
DoSearch	Occurs when user submits the dialog.

© 2006-2008 Kentico Software

CMSSearchDialog Design

The design can be modified by setting style of particular controls. All displayed controls can be accessed through the following properties:

Property Name	Description
SearchForLabel	SearchFor label control.
SearchForTextBox	SearchFor textbox.
SearchModeLabel	SearchMode label.
SearchModeList	SearchMode drop-down list.
SearchScopeLabel	SearchScope label.
SearchScopeList	SearchScope drop-down list.
SearchButton	Search button.

You can also use the following CSS classes:

Event Name	Description
CMSSearchDialogSearchForLabel	CSS class of the "Search for:" label.
CMSSearchDialogSearchForTextBox	CSS class of the search expression text box.
CMSSearchDialogSearchModeLabel	CSS class of the "Search Mode:" label.
CMSSearchDialogSearchModeDropDownList	CSS class of the search mode drop down list.
CMSSearchDialogSearchScopeLabel	CSS class of the "Search Scope:" label.
CMSSearchDialogSearchScopeDropDownList	CSS class of the search scope drop down list.
CMSSearchDialogSeachButton	CSS class of the search button.

© 2006-2008 Kentico Software

CMSSearchResults Properties

Inherits: CMSControlProperties - common properties
See also: DataPager

Property Name	Description	Sample Value
PagerControl	DataPager control used for paging of the search results.	
QueryStringKey	Query string key used for data pager URL parameter.	
SearchExpression	Word(s) to be searched for.	"asp.net cms"
SearchMode	Search mode - any word, all words or exact phrase.	SearchModeEnum.AnyWord
StartingPath	Starting path specifying the content to be searched.	"/products"
CMSSearchDialogID	Optionally, you can use this property to specify the ID of the source CMSSearchDialog control that provides search parameters.	"CMSSearchDialog1"
TransformationName	Transformation name in format application.class.transformation. This transformation is used for displaying the search results. The default transformation is "cms.root.searchresults".	"cms.searchresults"
WhereCondition	WHERE condition used for the SQL search queries.	" DocumentModifiedWhen > '1/1/2007' "
OrderBy	ORDER BY condition used for the SQL search queries.	" DocumentModifiedWhen DESC "

CMSSearchResults QueryString (URL) parameters

The CMSSearchResults control accepts the following URL parameters:

Parameter Name	Description	Sample Value
searchtext	Searched text.	products
searchmode	Search mode.	allwords exactphrase anyword (default value)

CMSSearchResults Design

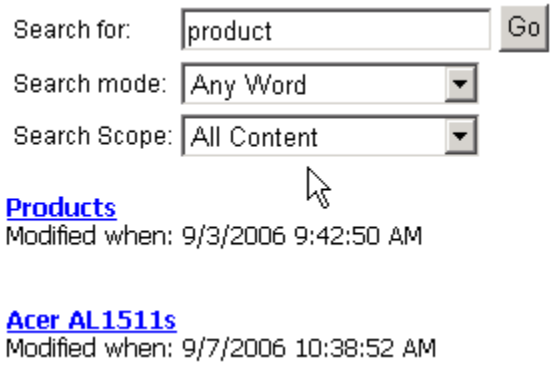
The search results are displayed using the transformation specified in the TransformationName property.

© 2006-2008 Kentico Software

Example of Using CMSSearchDialog and CMSSearchResults Controls

This example will show you how to create a search form and display the search results.

- Create a new Web form.
- Drag and drop the CMSSearchDialog control on the form.
- Drag and drop the CMSSearchResults control on the form.
- Set the following properties of the CMSSearchResults control:
  - CMSSearchDialogID: CMSSearchDialog1
- Compile and run the project. You should see a page like this:



1.6.31 CMSSiteMap

The **CMSSiteMap** control allows you to display the whole navigation structure of the Web site or just its specified part. It reads CMS.MenuItem documents and renders their structure as a site map.

It allows you to display part of the menu structure specified using its path, depth, document type and WHERE condition. The items are sorted by MenuItemOrder and MenuItemCaption values.

See also: Using macro expressions in menu items

Inherits: CMSMenuProperties - common properties

© 2006-2008 Kentico Software

Properties

Property Name	Description	Sample Value
UrlTarget	Specifies target frame for all URLs.	"_blank"
LoadDataAutomaticaly	Indicates if data for the control should be loaded automatically. By default, the data is loaded automatically. If you set this property to false, you can supply custom DataSet to the DataSource property and then call the ReloadData(false) method.	
RenderedHTML	Allows you to get or set the HTML code rendered by the control. You need to set this property before the Render event - e.g. in the OnLoad event.	

Methods

Method Name	Description
ReloadData	Reloads the data. If the **forceLoad** parameter is set to false and the custom value is assigned to the DataSource property, the properties of the CMSListMenu control are not used and only the data from the DataSource are used.

Design

You can modify the design using the following CSS styles:

Class Name	Description
CMSSiteMapList	The UL element in the site map.
CMSSiteMapListItem	The LI element in the site map.
CMSSiteMapLink	Link (A element) in the site map.

© 2006-2008 Kentico Software

Example

This example will show you how to display a site map based on the CMS content. It assumes that you have configured your project for Kentico CMS Controls.

- Create a new Web form.
- Drag and drop the CMSSiteMap control on the form.
- In the HTML mode, add the following CSS styles inside the <head> tag. It will modify the appearance of the menu.

```
<style type="text/css">
.CMSSiteMapList { }
.CMSSiteMapListItem { list-style-image: url(images/menuitem.gif); }
.CMSSiteMapLink { color: #C34C17; text-decoration:none; }
</style>
```

- Switch back to the Design mode.
- In the Properties window, set the following property values:
  -Path: /%
- Compile and run the project. You should see a page like this:

© 2006-2008 Kentico Software

1.6.32 CMSTabControl

The CMSTabControl control is inherited from the BasicTabControl control. It allows you to display one-level tab menu based on data from Kentico CMS. It reads specified documents and renders the menu according to their values.

It allows you to display menu items specified using their path, depth, document type and WHERE condition. The CMSTabControl control displays content without writing any additional code. The items are sorted by MenuItemOrder and MenuItemCaption values.

See also: Using macro expressions in menu items, Using the CSSPrefix property for design of sub-menus

Inherits: BasicTabControl, CMSMenuProperties - common properties

Properties

Property Name	Description	Sample Value
HighlightedNodePath	Path of the item that will be highlighted like it was selected.	"/products/notebooks"
RenderImageAlt	Indicates if ALT attribute should be rendered for images used in the menu (for XHTML compatibility).	
LoadDataAutomaticaly	Indicates if data for the control should be loaded automatically. By default, the data is loaded automatically. If you set this property to false, you can supply custom DataSet to the DataSource property and then call the ReloadData(false) method.	
RenderedHTML	Allows you to get or set the HTML code rendered by the control. You need to set this property before the Render event - e.g. in the OnLoad event.	

Methods

Method Name	Description
ReloadData	Reloads the data. If the **forceLoad** parameter is set to false and the custom value is assigned to the DataSource property, the properties of the CMSListMenu control are not used and only the data from the DataSource are used.

© 2006-2008 Kentico Software

Design

You can modify the design using the same classes as for the BasicTabControl control.

Example

This example will show you how to display a simple tab-menu based on the CMS content. It assumes that you have configured your project for Kentico CMS Controls.

- Create a new Web form.
- Drag and drop the **CMSTabControl** control on the form.
- In Source mode add the following CSS styles inside the <BODY> tag. It will modify the appearance of the tabs.

[C#], [VB.NET]

```
<style type="text/css">
.TabControlTable { FONT-SIZE: 14px; FONT-FAMILY: Arial,Verdana }
.TabControlRow { }
.TabControl { BORDER-RIGHT: black 1px solid; BORDER-TOP: black 1px solid; FONT-WEIGHT:
bold; BACKGROUND: #e7e7ff; BORDER-LEFT: black 1px solid; CURSOR: hand; COLOR: black }
.TabControlSelected { BORDER-RIGHT: black 1px solid; BORDER-TOP: black 1px solid;
FONT-WEIGHT: bold; BACKGROUND: #4a3c8c; BORDER-LEFT: black 1px solid; CURSOR: default;
COLOR: white }
.TabControlLinkSelected { COLOR: white; TEXT-DECORATION: none }
.TabControlLink { COLOR: black; TEXT-DECORATION: none }
.TabControlLeft { WIDTH: 1px }
.TabControlRight { WIDTH: 0px }
.TabControlSelectedLeft { WIDTH: 1px }
.TabControlSelectedRight { WIDTH: 0px }
</style>
```

- Add the following table code just after the CMSTabControl tags. It will display a stripe under the tabs.

[C#], [VB.NET]

```
<hr style="width:100%; height:2px; margin-top:0px;" />
```

- Switch back to the Design mode.
- In the **Properties** window set the following property values:
  - MaxRelativeLevel: 1
  - Path: /%
- Compile and run the project. You will see a page like this:

© 2006-2008 Kentico Software

1.6.33 CMSTreeMenu

The **CMSTreeMenu** control allows you to display multi-level tree menu based on data from Kentico CMS. It allows you to display part of the menu structure specified using its path, depth, document type and WHERE condition.

See also: Using macro expressions in menu items, Using the CSSPrefix property for design of sub-menus

Inherits: CMSMenuProperties - common properties

Properties

Property Name	Description	Sample Value
CellPadding	Cell padding of the table representing menu.	
CellSpacing	Cell spacing of the table representing menu.	
CollapseSelectedNodeOnClick	Indicates if the selected section of menu should be collapsed when it's clicked.	
DisplayHighlightedItemAsLink	Indicates if the highlighted item should be displayed as a link.	
GenerateIndentationInsideLink	Indicates if indentation spaces should be generated inside hyperlink (true) or outside (false). This applies only when you do not use images in the menu.	
GenerateOnlyOuterLink	Indicates if only one outer link should be generated per each menu item.	
HighlightAllItemsInPath	Indicates if all items in the currently selected path should be displayed as highlighted.	
HighlightedNodePath	Path of the item that will be highlighted like it was selected. If you omit this value, the control automatically uses the current alias path from the "aliaspath" querystring parameter.	"/products"
Indentation	Indentation of menu item levels. Number of spaces that will be placed before each level of menu items.	
MenuItemImageUrl	URL address of the image that is displayed next to menu items. It may start with "~/" representing the virtual path of the current application.	"~/images/menuitem.gif"
MenuItemOpenImageUrl	URL address of the image that is displayed next to open menu items. It may start with "~/" representing the virtual path of the current application.	"~/images/openmenuitem.gif"
OnMouseOutScript	OnMouseOutScript script for menu items. You can use macro expressions here.	
OnMouseOverScript	OnMouseOver script for menu items. You can use macro expressions here.	

© 2006-2008 Kentico Software

UrlTarget	Specifies target frame for all URLs.	"_blank"
UseAlternatingStyles	Indicates if the control should render different styles for odd and even items.	
RenderImageAlt	Indicates if ALT attribute should be rendered for images used in the menu (for XHTML compatibility).	
LoadDataAutomaticaly	Indicates if data for the control should be loaded automatically. By default, the data is loaded automatically. If you set this property to false, you can supply custom DataSet to the DataSource property and then call the ReloadData(false) method.	
RenderedHTML	Allows you to get or set the HTML code rendered by the control. You need to set this property before the Render event - e.g. in the OnLoad event.	

Methods

Method Name	Description
ReloadData	Reloads the data. If the **forceLoad** parameter is set to false and the custom value is assigned to the DataSource property, the properties of the CMSListMenu control are not used and only the data from the DataSource are used.

© 2006-2008 Kentico Software

Design

You can modify the design using the following CSS styles:

Class Name	Description
CMSTreeMenuTable	The main table (TABLE element).
CMSTreeMenuItem	Tree menu item (TD element).
CMSTreeMenuItemAlt	Alternating style of the menu item (TD element). It's used only when you set the UseAlternatingStyles property to true.
CMSTreeMenuSelectedItem	Selected tree menu item (TD element).
CMSTreeMenuLink	Link (A element).
CMSTreeMenuLinkAlt	Alternating style of the link (A element). It's used only when you set the UseAlternatingStyles property to true.
CMSTreeMenuSelectedLink	Link of the selected item (A element).
CMSTreeMenuNestedTable	Nested table (TABLE element). It's used only when CollapseSelectedNodeOnClick is true.

See also: Using the CSSPrefix property for design of sub-menus to find out how to set different styles for particular menu levels.

Example

This example will show you how to display a tree menu based on the CMS content.

- Create a new Web form.
- Drag and drop the CMSTreeMenu control on the form.
- In the HTML mode, add the following CSS styles inside the <HEAD> element. It will modify the appearance of the menu.

```
<style>
.CMSTreeMenuTable { PADDING-RIGHT: 5px; PADDING-LEFT: 5px; PADDING-BOTTOM: 2px;
PADDING-TOP: 2px; }
.CMSTreeMenuItem { BACKGROUND: #e7e7ff }
.CMSTreeMenuSelectedItem { BACKGROUND: #4a3c8c }
.CMSTreeMenuLink { COLOR: black; TEXT-DECORATION: none }
.CMSTreeMenuSelectedLink { COLOR: white; TEXT-DECORATION: none }
</style>
```

- Switch back to the Design mode.
- In the Properties window, set the following property values:
  - Path: /Products
  - MenuItemImageUrl: ~/images/menuitem.gif
  - MenuItemOpenImageUrl: ~/images/menuitem_selected.gif
  - Indentation: 3

© 2006-2008 Kentico Software

- Compile and run the project. You should see a page like this:

1.6.34 CMSTreeView

The **CMSTreeView** control allows you to display multi-level tree menu based on data from Kentico CMS. It allows you to display part of the menu structure specified using its path, depth, document type and WHERE condition.

See also: Using macro expressions in menu items

Inherits: CMSMenuProperties - common properties

This web part uses the standard ASP.NET TreeView control and enhances it with standard CMSMenuProperties set of properties. Please see the ASP.NET documentation for more details on the TreeView control properties, behavior and design.

1.6.35 CMSViewer

The CMSViewer control allows you to display part of the CMS content specified using its path, depth, document template, WHERE condition and ORDER BY clause. It uses XSLT transformations to display the content. The CMSViewer control displays content without writing any additional code.

Inherits: CMSDataProperties - common properties

Properties

Property Name	Description	Sample Value
HideControlForZeroRows	Hides the control when no data is loaded. The default value is False.	
TransformationName	Transformation name in format application.class.transformation.	"cms.article.preview"
ZeroRowsText	Text to be shown when the control doesn't display any data.	

Design

The displayed content is completely driven by your XSLT transformation.

Example

© 2006-2008 Kentico Software

This example will show you how to display specific news item using the CMSViewer control. It assumes that you have configured your project for Kentico CMS Controls.

- Create a new Web form.
- Drag and drop the CMSViewer control on the form.
- Switch back to the Design mode. In the Properties window set the following property values:
  - ClassNames: cms.news
  - Path: /News/Your-First-News
  - TransformationName: cms.news.default_xslt
- Compile and run the project. You should see a page like this:

© 2006-2008 Kentico Software

1.6.36 DataPager

The DataPager control ensures paging of some of the CMSControls:

- CMSDataList
- CMSRepeater
- CMSSearchResults
- QueryDataList
- QueryRepeater

It cannot be used separately.

See also: TemplateDataPager

Properties

Property Name	Description	Sample Value
CurrentPage	Current page index.	
BackText	Back button/hyperlink text.	
FirstText	First text.	
HideOnSinglePage	If true, the pager is hidden if only one page is displayed.	
IgnoreQueryString	Indicates if QueryString parameters should be ignored.	
LabelText	Label text.	
LastText	Last text.	
MaxPages	Maximum number of pages to be diplayed.	
NextText	Next button/hyperlink text.	
PageCount	Page count (read only).	
PageNumbersSeparator	Page numbers separator.	","
PageSize	Page size	
PagingMode	Determines if PostBack or QueryString should be used for the paging.	PostBack QueryString
QueryStringKey	Query parameter name for the page index.	"pagenumber"
RecordEnd	Index of the last record on current page.	
RecordStart	Index of the first record on current page.	
ResultsFormat	Results text format.	"Displaying results {0}-{1} (of {2})"
SliderSize	Slider size.	
TotalRecords	Total data source records.	

© 2006-2008 Kentico Software

Design

Property Name	Description	Sample Value
BackNextButtonStyle	Back/Next button style.	
BackNextDisplay	Back/Next display mode.	Buttons Hyperlinks
BackNextLinkSeparator	Back/Next link separator.	
BackNextLocation	Back/Next location.	Right Left Split None
BackNextStyle	Back/Next style.	
ControlCssClass	CSS Class of the pager control	
LabelStyle	Label style	
PageNumbersDisplay	Page numbers display mode.	Number Results
PageNumbersStyle	Page numbers style.	
PagerControlStyle	Pager control style.	
PagerPosition	Pager position.	Bottom Top
ResultsLocation	Results location.	Top Bottom None
ResultsStyle	Results style.	
SectionPadding	Section padding.	
ShowFirstLast	Show first last labels.	
ShowLabel	Show label.	
ShowPageNumbers	Show page numbers.	
UseSlider	Use slider.	

1.6.37 QueryDataGrid

The QueryDataGrid control is inherited from the BasicDatagrid control. It automatically ensures data binding, paging and sorting. Moreover, it allows you to specify query and the WHERE condition and it displays data without any additional code.

You can use the common DataGrid designer to set up QueryDataGrid style and behavior.

QueryDataGrid can be only used with pre-defined queries stored in the document type configuration. If you want to display only data from particular part of the content tree, please use CMSDataGrid.

Data Source

Data retrieved using the predefined query specified in the QueryName property.

Inherits: CMSQueryProperties - common properties, BasicDataGrid

© 2006-2008 Kentico Software

Design

The design can be modified in the same way as the standard DataGrid control.

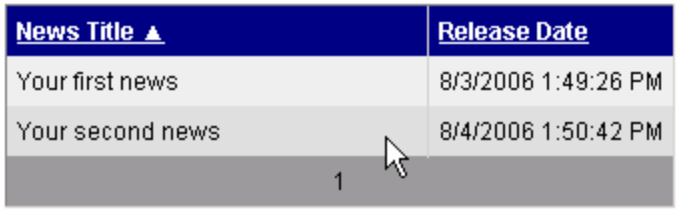
Example

This example will show you how to read list of news and display it in the grid using the QueryDataGrid control.

- Create a new Web form.
- Drag and drop the QueryDataGrid control on the form.
- In the Properties window, click **Auto Format...** and choose some color schema.
- In the Properties window, click **Property Builder...**, the QueryDataGrid1 Properties dialog appears.
  - On the General tab check the **Allow sorting** box.
  - Now we will specify the columns that will be displayed. On the Columns tab:
    - Uncheck the **Create columns automatically at run time** box.
    - Add a new **Bound Column** from the **Available columns** list to the **Selected columns** list.

Enter the following values in the appropriate fields:
    - Header text: News Title
    - Data field: NewsTitle
    - Sort expression: NewsTitle
  - Add a new **Bound Column** from the **Available columns** list to the **Selected columns** list.

Enter the following values in the appropriate fields:
    - Header text: Release Date
    - Data field: NewsReleaseDate
    - Sort expression: NewsReleaseDate
  - On the **Paging tab**, check the **Allow Paging** box and click **OK**.
- In the **Properties** window set the property **QueryName** to cms.news.selectlatest and set **ProcessSorting** to True (sorting will be ensured by the control).
- Compile and run the project. You should see a page like this:



1.6.38 QueryDataList

The QueryDataList control is inherited from the BasicDataList control. It automatically ensures data binding. Moreover, it allows you to specify query, ORDER BY clause and the WHERE condition and it ensures displaying data without any additional code.

QueryDataList can be used only with pre-defined queries stored in the document type configuration.

If you want to display data only from particular part of the content tree, please use CMSDataList.

Data Source

Data retrieved using the predefined query specified in the QueryName property.

Inherits: CMSQueryProperties - common properties, BasicDataList

© 2006-2008 Kentico Software

See also: DataPager

Properties

Property Name	Description	Sample Value
AlternatingTransformationName	Transformation name in format application.class.transformation applied to alternating items.	"cms.news.preview_alt"
EnablePaging	Enables paging and show the DataPager control.	
PagerControl	DataPager control.	
TransformationName	Transformation name in format application.class.transformation.	"cms.news.preview"

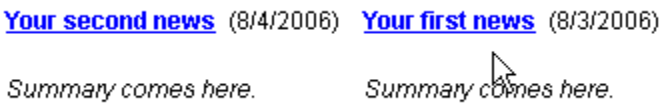
Design

The design can be modified in the same way as for the standard ASP.NET DataList control. The design of the displayed content depends on used transformations.

Example

This example will show you how to read a list of the latest news and display them using the QueryDataList control.

- Create a new Web form.
- Drag and drop the QueryDataList control on the form.
- In the **Properties** window set the following property values:
  - QueryName: cms.news.selectlatest
  - RepeatColumns: 2
  - TransformationName: cms.news.preview
- Compile and run the project. You should see a page like this:



1.6.39 QueryRepeater

The QueryRepeater control is inherited from the BasicRepeater control. It automatically ensures data binding. Moreover, it allows you to specify query, ORDER BY clause and the WHERE condition and it ensures displaying data without any additional code.

QueryRepeater can be only used with pre-defined queries stored in the document type configuration.

If you want to display only data from particular part of the content tree, please use CMSRepeater instead.

Data Source

© 2006-2008 Kentico Software

Data retrieved using the predefined query specified in the QueryName property.

Inherits: BasicRepeater, CMSQueryProperties - common properties
See also: DataPager

Properties

Property Name	Description	Sample Value
AlternatingTransformationName	Transformation name in format application.class.transformation applied to alternating items.	"cms.news.preview_alt"
ItemSeparator	Item separator between displayed records.	" "
EnablePaging	Enables paging a shows the DataPager control.	
PagerControl	DataPager control that ensured paging.	
TransformationName	Transformation name in format application.class.transformation.	"cms.news.preview"

Design

The design depends on the transformations you use to display content.

Example

This example will show you how to read list of news and display them using the QueryRepeater control.

- Create a new Web form.
- Drag and drop the **QueryRepeater** control on the form.
- In the Properties window set the following property values:
  - QueryName: cms.news.selectlatest
  - TransformationName: cms.news.preview
- Compile and run the project. You should see a page like this:

1.6.40 TemplateDataPager

The **TemplateDataPager** control can be used for custom format of data pager. It automatically renders the list of numbers, but you need to write some code to bind it to a control that ensures

© 2006-2008 Kentico Software

displaying of the content (e.g. CMSRepeater, CMSDataList or other).

Here's a simple example of how you can use it:

1. Add the following line to the beginning of your ASPX page:

[C#]

```
<%@ Register Assembly="CMS.Controls" Namespace="CMS.Controls" TagPrefix="cc1" %>
```

2. Add the following code to the page, inside the <form> element:

[C#]

```
<table style="border: solid 1px #CCCCCC; margin-left: auto; margin-right: auto;">
  <tr>
    <td style="border-bottom: solid 1px #CCCCCC; padding: 10px; text-align: center;">
      <cc1:CMSRepeater ID="CMSRepeater1" runat="server" Path="/%" ClassNames="CMS.Product"
      TransformationName="CMS.Product.preview">
      </cc1:CMSRepeater>
    </td>
  </tr>
  <tr>
     <td style="padding: 10px; background-color: #D9D9D9;">
       <cc1:TemplateDataPager ID="TemplateDataPager1" runat="server">
           <NumberTemplate>
               <a href="?Page=<%# Eval("PageNumber")  %>">
                    <%# Eval("PageNumber")  %>
               </a>
           </NumberTemplate>
           <SelectedNumberTemplate>
               <b>
                 <%# Eval("PageNumber")  %>
               </b>
           </SelectedNumberTemplate>
           <SeparatorTemplate>
               -
           </SeparatorTemplate>
           <FirstItemTemplate>
               <a href="?Page=1">First</a> | 
           </FirstItemTemplate>
           <LastItemTemplate>
                | <a href="?Page=<%# pageCount %>">Last</a>
           </LastItemTemplate>
           <PreviousItemTemplate>
               <a href="?Page=<%# previousPage %>">Previous</a>  | 
           </PreviousItemTemplate>
           <NextItemTemplate>
                |  <a href="?Page=<%# nextPage %>">Next</a>
           </NextItemTemplate>
       </cc1:TemplateDataPager>
     </td>
  </tr>
</table>
```

As you can see, the control uses the standard CMSRepeater control. The pager format is specified using the templates inside its definition.

3. Modify the code-behind so that it looks like this (the page class name and type may be different):

[C#]

© 2006-2008 Kentico Software

```
using CMS.GlobalHelper;

public partial class CMSControlsExamples_TemplatedDataPager : ControlsExamplesPage
{
    public string pageCount = "1";
    public string previousPage = "1";
    public string nextPage = "";


    /// <summary>
    /// OnInit override
    /// </summary>
    /// <param name="e"></param>
    protected override void OnInit(EventArgs e)
    {
        // Disable repeater pager and databindbydefault
        CMSRepeater1.EnablePaging = false;
        CMSRepeater1.DataBindByDefault = false;

        base.OnInit(e);
    }


    protected void Page_Load(object sender, EventArgs e)
    {
        // Get repeater datasource
        TemplateDataPager1.DataSource = CMSRepeater1.DataSource;

        // Set page size
        TemplateDataPager1.PageSize = 1;

        // Set current page from query string
        TemplateDataPager1.CurrentPage = ValidationHelper.GetInteger(Request.QueryString[
"Page"], 1);

        // Get page number for last link
        pageCount = ((int)(TemplateDataPager1.PageCount - 1)).ToString();

        // Set default next page link
        nextPage = pageCount;

        // Set previous link
        if ((TemplateDataPager1.CurrentPage - 1) >= 1)
        {
            previousPage = ((int)(TemplateDataPager1.CurrentPage - 1)).ToString();
        }

        // Set next link
        if ((TemplateDataPager1.CurrentPage + 1) <= (TemplateDataPager1.PageCount - 1))
        {
            nextPage = ((int)(TemplateDataPager1.CurrentPage + 1)).ToString();
        }

        // Set paged datasource to the repeater and databind it
        CMSRepeater1.DataSource = TemplateDataPager1.PagedData;
        if (!DataHelper.DataSourceIsEmpty(CMSRepeater1.DataSource))
        {
            CMSRepeater1.DataBind();
        }
    }
}
```

4. Save all changes and see the page on the live site. It will look like this:

© 2006-2008 Kentico Software

IPAQ RX1950

our price: $ 699.00
more

First | Previous | 1 - 2 - 3 - 4 - 5 | Next | Last

© 2006-2008 Kentico Software