The CMSSiteMap control displays the entire navigation structure of the website or a specified sub-section (reads Kentico pages and renders their structure as a site map).

The control only displays pages that have the **Show in sitemap** setting enabled in the Kentico administration interface (**Pages -> Edit -> Properties -> Navigation**).

Web part equivalent (portal engine): Site map

Getting started

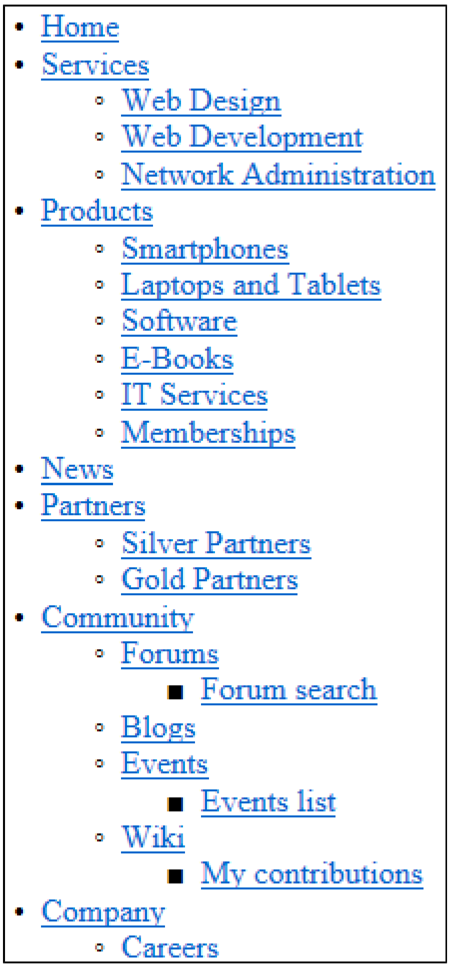
The following is a step-by-step tutorial that shows how to display a site map based on the content of a Kentico website using the CMSSiteMap control:

1. Create a new **Web form** somewhere in your web project.
2. Drag the **CMSSiteMap** control from the toolbox onto the form.
    - The code of the CMSSiteMap looks like this:

```
<cms:CMSSiteMap ID="CMSSiteMap1" runat="server" />
```

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

The resulting page displays a tree containing the website's pages.



- Home
- Services
    - Web Design
    - Web Development
    - Network Administration
- Products
    - Smartphones
    - Laptops and Tablets
    - Software
    - E-Books
    - IT Services
    - Memberships
- News
- Partners
    - Silver Partners
    - Gold Partners
- Community
    - Forums
        - Forum search
    - Blogs
    - Events
        - Events list
    - Wiki
        - My contributions
- Company
    - Careers

Configuration

You can set the following properties for the CMSSiteMap control:

CMSSiteMap properties	Description	Sample value
ApplyMenuInactivation	If true, the sitemap does not render links for pages set as **Inactive menu items**. You can configure the navigation settings for individual pages in **Pages -> Edit -> Properties -> Navigation**.	
EncodeMenuCaption	Indicates whether the control HTML encodes the captions of site map items. Enable if you need to display pages whose names contain HTML code.	
LoadDataAutomaticaly	Indicates whether the control automatically loads page data. True by default. If you set this property to false, you need to assign a custom DataSet into the **DataSource** property and then call the control's **ReloadData** method.	
OrderBy	Gets or sets the ORDER BY clause of the SQL query that the control uses to load data. **Important**: The root of the displayed page tree (or sub-tree) must be **first** in the resulting order, otherwise the control may not display all pages correctly. You can ensure this by starting the Order by value with the **NodeLevel** column.	"NodeLevel, NodeOrder"
RenderedHTML	Gets or sets the HTML code rendered by the control. If you need to set this property manually, do so before the *Render* event (for example in the *OnLoad* event).	
RenderLinkTitle	Indicates whether the sitemap renders page names as **Title** attributes (tooltips) for links. This can improve the sitemap's accessibility.	
UrlTarget	Specifies the target frame for all links in the site map.	"_blank"

Common navigation properties	Description	Sample value
ApplyMenuDesign	Indicates whether the control applies page menu design settings. True by default.	
Columns	Contains the names of columns that the control loads for pages (menu items) in addition to the default columns. If you need data from other columns, type their names separated by commas. **Note**: To find a full list of the default navigation columns, use the SQL queries debugging tool and inspect the query performed by your navigation control.	"DocumentPageTitle, DocumentPageKeywords"
HideControlForZeroRows	Indicates whether the control should be hidden when no data is loaded. Default value is False.	
HighlightAllItemsInPath	Indicates whether the control highlights all items on the visitor's current path.	
SubmenuIndicator	Path to an image displayed next to every menu item that contains sub-items.	
UseAlternatingStyles	Indicates whether the control uses alternating styles for even and odd items on the same menu level.	

UseItemImages ForHiglightedItem	Indicates whether the control uses the item image if the highlighted image is not specified.	
WordWrap	Indicates whether text displayed by the control uses word wrapping. If disabled, text that is too long is replaced by 'nbsp'.	
ZeroRowsText	Text shown if no records are found. This text is not visible when the control is hidden by the **HideControlForZeroRows** property.	"No records found."

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

TreeProvider	Gets or sets the *TreeProvider* object used by the control to access page data. If you do not assign a TreeProvider object, the control automatically creates a new instance.	

CMS Base control properties	Description	Sample value
CacheDependencies	List of the cache keys on which the control's cached data depends. When the specified cache items change, the control clears its cache. Each item (dependency) must be on one line. If you leave this property empty, the control uses default dependencies. **See also**: Setting cache dependencies, Configuring caching	cms.user|all
CacheItemName	Sets the name of the cache key used to store the control's content. If you leave the value empty, the system generates a default name containing variables, such as the control ID, the selected culture and the name of the user who loaded the page. The system cache is shared by all pages in your application, so cache item names representing different data must be unique globally. If you have multiple controls that load the same data, you can share the cache keys between the controls (optimizes loading of content and avoids redundant data in the cache). If the content displayed by the control depends on variables, such as URL parameters, you can set a custom name dynamically in the page's code behind. **See also**: Caching the data of page components, Configuring caching	"CMSRepeaterNews" + Request.QueryString["id"].ToString()
CacheMinutes	Sets the number of minutes for which the control caches content retrieved from the database. ● 0 indicates that control does not cache content ● -1 indicates that the control uses the site-level content caching settings Allows you to set up caching of content so that the control doesn't have to retrieve content from the database on each request. The caching mechanism uses absolute expiration time. This means that cache items expire after a specified time period even if the page containing the control wasn't requested. **See also**: Caching the data of page components, Configuring caching	
FilterControl	Gets or sets the filter control used to limit the data read by the control.	
FilterName	Gets or sets the code name of the filter control used to limit the data read by this control.	
OrderBy	Gets or sets the ORDER BY clause of the SQL query that the control uses to load data.	"NewsReleaseDate DESC"
SelectedColumns	Database table columns that the control loads for pages, separated by commas (,). If null or empty, the control loads all available columns.	
SiteName	Specifies the code name of the Kentico website for which the control loads data.	
StopProcessing	If true, the control stops all processing — does not load or display any data or other HTML output.	

TopN	Specifies the maximum number of database records that the control loads.	
WhereCondition	Gets or sets the WHERE clause of the SQL query that the control uses to loads data.	"Product Price > 100"

Appearance and styling

The appearance of the CMSSiteMap control is determined by its properties and CSS classes. You can use the following CSS classes to modify the design:

CSS class name	Applies to
CMSSiteMapList	UL elements in the site map.
CMSSiteMapListItem	LI elements in the site map.
CMSSiteMapLink	Links (A elements) in the site map.

The recommended place to define these classes is in a Kentico stylesheet using the **CSS stylesheets** application.

You can apply stylesheets to:

- Entire websites
- Individual pages that contain the control

Example - Applying styles

The following example demonstrates how to apply CSS styles to the CMSSiteMap control. To implement the example, please follow the steps in the Getting started section and then continue with the following steps:

1. Add the following style definitions inside the page's *<head>* element:

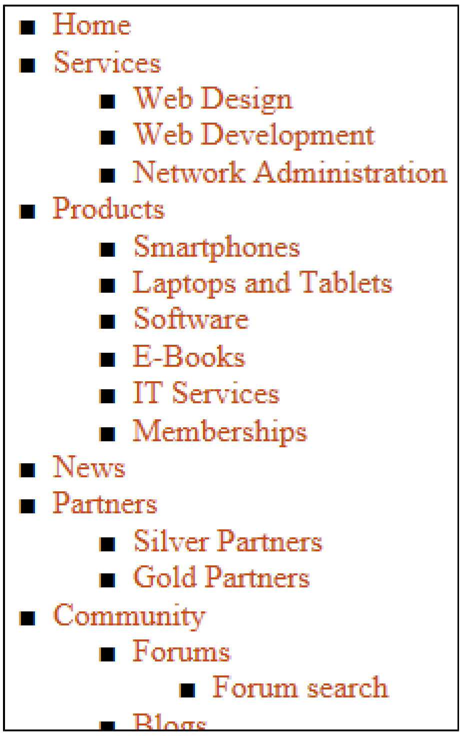
```
<style type="text/css">

/* Site map class definitions */
.CMSSiteMapList { }
.CMSSiteMapListItem { list-style-type: square; }
.CMSSiteMapLink { color: #C34C17; text-decoration:none; }

</style>
```

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

The sitemap now has a different appearance.



- Home
- Services
  - Web Design
  - Web Development
  - Network Administration
- Products
  - Smartphones
  - Laptops and Tablets
  - Software
  - E-Books
  - IT Services
  - Memberships
- News
- Partners
  - Silver Partners
  - Gold Partners
- Community
  - Forums
    - Forum search
  - Blogs