

The CMSSearchResults control displays search results according to parameters provided from the [CMSSearchDialog](#) control. The controls use the SQL search engine, which utilizes standard queries to find results in the Kentico database.



Note: Current versions of Kentico provide an index-based [smart search engine](#). The smart search has significantly better performance than the SQL search.

The SQL search uses the following queries:

- Automatically generated queries for the data of individual [page types](#). To override the search query for a page type, create a new query named **searchtree** in *Page types -> Edit page type -> Queries*.
- The **searchdocuments** query of the **Root** page type for fields that are shared by all pages (such as the document name).
- The **searchattachments** query of the **Root** page type for files uploaded as page attachments. To search attachments, you need to configure the system as described in [Configuring SQL search for attachment files](#).

The control combines the results of all relevant queries into a single data source and displays the data.



Web part equivalent (portal engine): SQL Search results

Configuration

You can set the following properties for the CMSSearchResults control:

Property name	Description	Sample value
CacheDependencies	<p>List of the cache keys on which the control's cached data depends. When the specified cache items change, the control clears its cache.</p> <p>Each item (dependency) must be on one line.</p> <p>If you leave this property empty, the control uses default dependencies.</p> <p>See also: Setting cache dependencies, Configuring caching</p>	cms. user all
CacheItemName	<p>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.</p> <p>The system cache is shared by all pages in your application, so cache item names representing different data must be unique globally. 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.</p> <p>See also: Caching the data of page components, Configuring caching</p>	"mycache name" + Request. QueryString["id"]. ToString()
CacheMinutes	<p>Sets the number of minutes for which the control caches content retrieved from the database.</p> <ul style="list-style-type: none"> • 0 indicates that control does not cache content • -1 indicates that the control uses the site-level content caching settings <p>Allows you to set up caching of content so that the control doesn't have to retrieve content from the database on each request.</p> <p>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.</p> <p>See also: Caching the data of page components, Configuring caching</p>	

CheckPermissions	<p>Indicates if the control checks the permissions of the user viewing the page. If the value is <i>false</i> (default value) no permissions are checked.</p> <p>If true, the control only loads pages for which the user viewing the page has read permissions.</p>	
ClassNames	<p>Specifies which page types the control loads and displays. Identify page types through their code names, separated by semicolons (;).</p> <p>You can use the * wildcard as a substitute for any number of characters. For example <i>Product.*</i> includes the page types <i>Product.Camera</i>, <i>Product.CellPhone</i>, <i>Product.Computer</i> etc.</p> <p>If the property is left empty, the control retrieves all page types by default. In the case of menu and navigation controls, only <i>CMS.MenuItem</i> pages are loaded by default.</p> <p>Note: If the control loads all page types (empty value), only the common data columns from the <i>View_CMS_Tree_Joined</i> view 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.</p>	"cms. news" "cms. news; cms. article"
CMSSearchDialogID	Specifies the ID of the CMSSearchDialog control that provides the search expression and parameters.	"CMSSearchDialog1"
CombineWithDefaultCulture	<p>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.</p> <p>Only applies if you do not set the TreeProvider property manually.</p>	
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	<p>Allows you to manually assign a DataSet or DataTable containing pages for the control to display.</p> <p>You can retrieve the search data in your code using the CMS.DocumentEngine.TreeProvider.Search method:</p> <pre>DataSet results = TreeProvider.Search(...); CMSSearchResults1.DataSource = results; CMSSearchResults1.DataBind();</pre>	
EnablePaging	Enables the paging of search results. True by default.	
FilterControl	Gets or sets the filter control used to limit the data read by the control.	
FilterName	Gets or sets the ID of the filter control used to limit the data read by this control.	
FilterOutDuplicates	Indicates if the control filters out duplicated (linked) pages from the data.	
IgnoreTransformations	Indicates whether the control ignores the TransformationName property and uses the item templates instead.	
NoResultsLabel	The label control that the control displays when there are no matching search results.	
OrderBy	Gets or sets the ORDER BY clause of the SQL query that the control uses to load data.	"NewsReleaseDate DESC"

PagerControl	DataPager object used for the paging of search results.	
Path	Limits the content tree path where the control searches. See: Writing page path expressions	
QueryStringKey	Name of the query string parameter that contains the current page number (if paging is used).	"pagenumber"
SearchExpression	Contains the words of the entered search expression.	
SearchMode	Search mode - any word, all words or exact phrase.	"AllWords" "AnyWord" "ExactPhrase"
SearchScope	Indicates whether the control searches: <ul style="list-style-type: none"> All pages on the website (<i>SearchAllContent</i>) Only pages that are placed under the page containing the control in the content tree (<i>SearchCurrentSection</i>) Only applies if the Path property is empty.	"SearchAllContent" "SearchCurrentSection"
SelectOnlyPublished	If true, the control only loads published pages.	
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.	
TagKey	Overrides the SPAN tags generated by the control with a custom tag.	
TransformationName	Name of the transformation applied to displayed search results. Enter the transformation name in format <code><page type code name>.<transformation name></code> . The default transformation is cms.root.searchresults .	"cms.searchresults"
WhereCondition	Gets or sets the WHERE clause of the SQL query that the control uses to load data.	"DocumentModifiedWhen > '1/1/2007' "

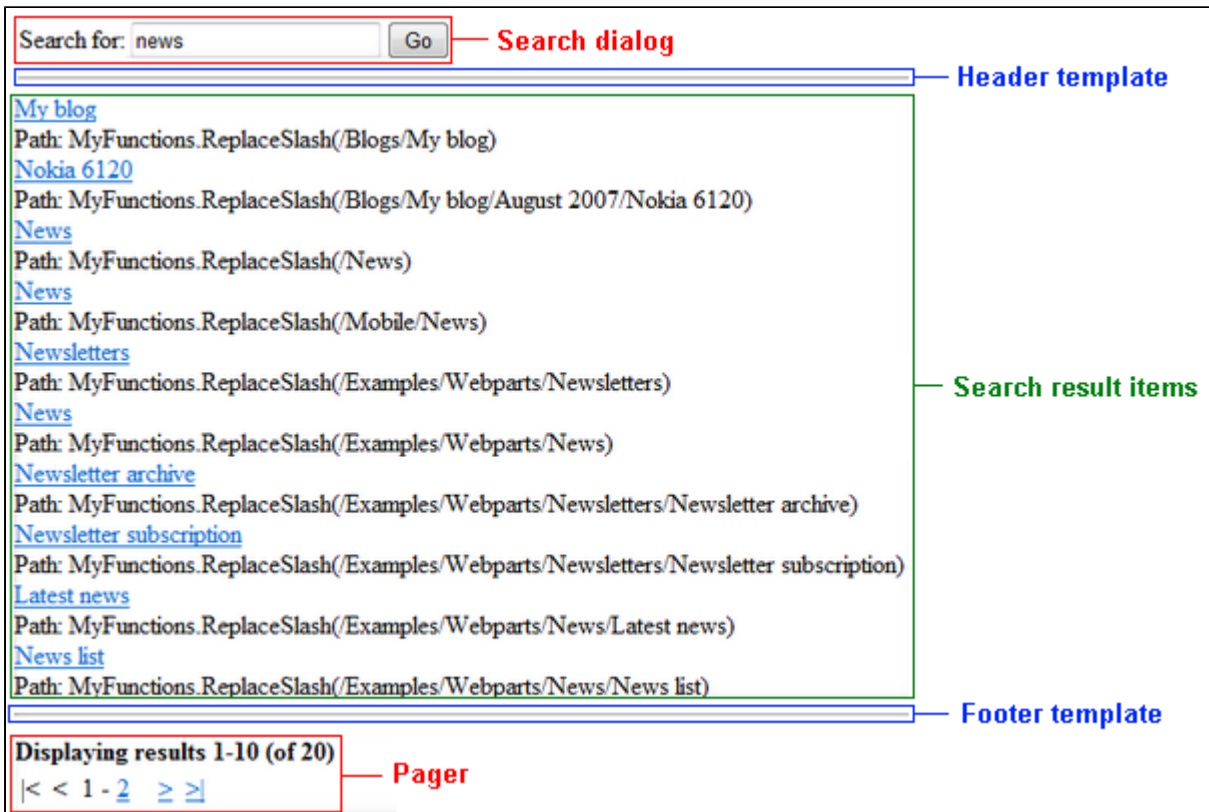
The CMSSearchResults control accepts the following query string (URL) parameters:

Parameter name	Description	Sample value
searchtext	Contains the search text.	products
searchmode	Sets the search mode.	allwords exactphrase anyword (default value)

Appearance and styling

The appearance of the CMSSearchResults control is determined by the code of the [transformation](#) used to format the result. Alternatively, you can add the following item templates between the tags of the CMSSearchResults control:

Item template	Description	Sample value
FooterTemplate	Template rendered above the search results.	<hr />
HeaderTemplate	Template rendered below the search results.	<hr />
ItemTemplate	Template applied to search result items.	



The screenshot illustrates the visual output of the CMSSearchResults control. At the top, a **Search dialog** contains a text input with 'news' and a 'Go' button. Below this is the **Header template**, represented by a horizontal line. The main area displays **Search result items**, each consisting of a link (e.g., 'My blog', 'Nokia 6120', 'News') and its corresponding path (e.g., 'Path: MyFunctions.ReplaceSlash(/Blogs/My blog)'). This is followed by the **Footer template**, another horizontal line. At the bottom, a **Pager** shows 'Displaying results 1-10 (of 20)' and navigation controls like '< > 1 - 2 > >'. Colored lines and labels on the right side of the screenshot identify these components: blue for the header/footer templates, green for the search result items, and red for the search dialog and pager.



- **SearchDialog** - the [CMSSearchDialog](#) control specified by the *CMSSearchDialogID* property.
- **HeaderTemplate** - this area is defined by the code between the *<HeaderTemplate>* tags.
- **Search Result Items** - this area displays the search results. Defined by the transformation specified by the *TransformationName* property (*cms.root.searchresults* by default) or by the code between the *<ItemTemplate>* tags if the *IgnoreTransformations* property is enabled.
- **FooterTemplate** - this area is defined by the code between the *<FooterTemplate>* tags.
- **Pager** - the built-in [DataPager](#) control, which is used for the paging of search results unless you set the *EnablePaging* property to *false*. You can access the pager control through the *PagerControl* property.