Note: The information on this page applies to search results retrieved from locally stored indexes. To learn how to display search results from Azure Search indexes, see Integrating Azure Search into pages.

You can use the following default transformations to display search results using the **Smart search dialog with results** and **Smart search results** web parts:

- **CMS.Root.SmartSearchResults**
- **CMS.Root.SmartSearchResultsWithImages**

The system returns search results in a **search dataset**. No matter how the fields are named in the found objects, the search dataset always contains the following fields that are automatically mapped to the corresponding object fields:

Search result field	Description
score	Expresses the relevance of the search result item as a numeric value. Higher values indicate higher relevance.
title	Mapped to the field selected as the **Title field** on the **Search fields** tab for the given object type.
content	Mapped to the field selected as the **Content field** on the **Search fields** tab for the given object type.
created	Mapped to the field selected as the **Date field** on the **Search fields** tab for the given object type.
image	Mapped to the field selected as the **Image field** on the **Search fields** tab for the given object type.
index	The code name of the search index where the given result was found.
documentExtensions	Available for page search results. Contains the extension of pages that represent files (typically CMS.File pages). For example, you can use the field to display MIME type font icons for file pages in your search results – call the *GetFileIconClass* transformation method to get the appropriate CSS class for the extension: `<i class="<%# GetFileIconClass(Eval<string>("documentExtensions")) %>"></i>`

In the code of ASCX transformations, you can get the values from the dataset by using the **Eval("<field name>")** method: *Eval("score")*, *Eval("title")*, etc.

To learn how to configure search fields for different object types, see the following pages:

- Configuring search settings for page fields
- Configuring search settings for custom table fields
- Configuring search settings for form fields
- Configuring search settings for user fields
- Configuring search settings for other objects using general indexes

Example

The following example shows how to create a basic ASCX transformation for displaying smart search results and configure it for the search web part. For more information about creating transformations, see Creating transformations for pages.

1. In the **Pages** application, open web part properties of the smart search web part, i.e., **Smart search dialog with results** or **Smart search results**.
2. In the **Transformations** section, click **New** to create a transformation.
   a. Assign the new transformation to a specific page type or custom table.
   b. Fill in the transformation definition.

Smart search result ASCX transformation example

```
<div>
        <a href="<%# SearchResultUrl() %>"><%# Eval("Title") %></a> (<%#
Eval("Created") %>) <br>
        <%#
                // Applies keyword highlight to the text of the 'Content'
field stripped of HTML tags and limited to 200 characters
                SearchHighlight(LimitLength(HTMLHelper.StripTags
(Eval<string>("Content")), 200), "<strong>", "</strong>")
        %>
</div>
```

c. Click **Save**.

3. Click **Save & Close** to apply the web part setting.

The format of the smart search result items now corresponds to the selected transformation definition.

For more advanced examples of smart search transformations, see the default transformations for pages and products used on our sample sites:

- CMS.Root.SmartSearchResults
- CMS.Root.SmartSearchResultsWithImages
- DancingGoat.Transformations.SmartSearchProductList
- DancingGoat.Transformations.SmartSearchResults
- EcommerceSite.Transformations.SearchResults

Search transformation methods

You can also use the following methods in your ASCX transformations:

Method	Parameters	Description and examples

SearchResultUrl	• *bool absolute* • *bool addLangParameter*	Returns the URL of the page containing the details of the search result. • The optional 'absolute' parameter indicates if the returned URL is absolute. ○ False by default. • The optional 'addLangParameter' indicates if a culture specific query should be added to the URL. ○ True by default. Note that clicking a link leading to a different culture changes the visitor's site culture. <%# SearchResultUrl(true, false) %> **Note**: The **SearchResultUrl** method does not return valid URLs for search results produced by general indexes, since the indexed objects are not pages and there is no default page to display the object details. You need to write and use a custom transformation method to generate the correct URL of a custom page displaying the appropriate information. **Combine with default culture** When the Combine with default culture setting is enabled, visitors will see pages in the default culture if they aren't translated to their current culture. • If the *addLangParameter* is set to false, the site culture will NOT change once visitors click the link leading to the page in the default culture. That is, they keep on browsing the site in the culture they made the search in. • If the *addLangParameter* is set to true, the site culture will change once visitors click the link leading to the page in the default culture. That is, they keep on browsing the site in the default culture afterwards.
SearchHighlight	• string text • string startTag • string endTag	Wraps the text of the first parameter into the tags specified by the other two parameters. <%# SearchHighlight(SearchResultUrl(),"<strong>","</strong>")%>
GetSearchImageUrl	• string noImageUrl • int maxSideSize	Returns the URL of the current search result's image. The first parameter specifies the URL used if no image is found – enter either the full relative path starting from the application root (~/), or a partial path starting from the *~/App_Themes /<skin_folder>/Images/* directory. The second parameter specifies the maximum side size to which the method resizes the image. <%# GetSearchImageUrl("~/App_Themes/Default/Images/CMSModules/CMS_SmartSearch /no_image.gif", 90) %> <%# GetSearchImageUrl("/CMSModules/CMS_SmartSearch/no_image.gif", 90) %>
GetSearchValue	• string columnName	Returns the value of the specified field for the current search result. Allows you to access both the general fields of the given objects type (page, custom table etc.) and any other fields included in the search index. <%# GetSearchValue("DocumentName") %>

GetSearchedContent	• string content	Parses the searched content as XML if required, and removes dynamic controls and macro expressions.