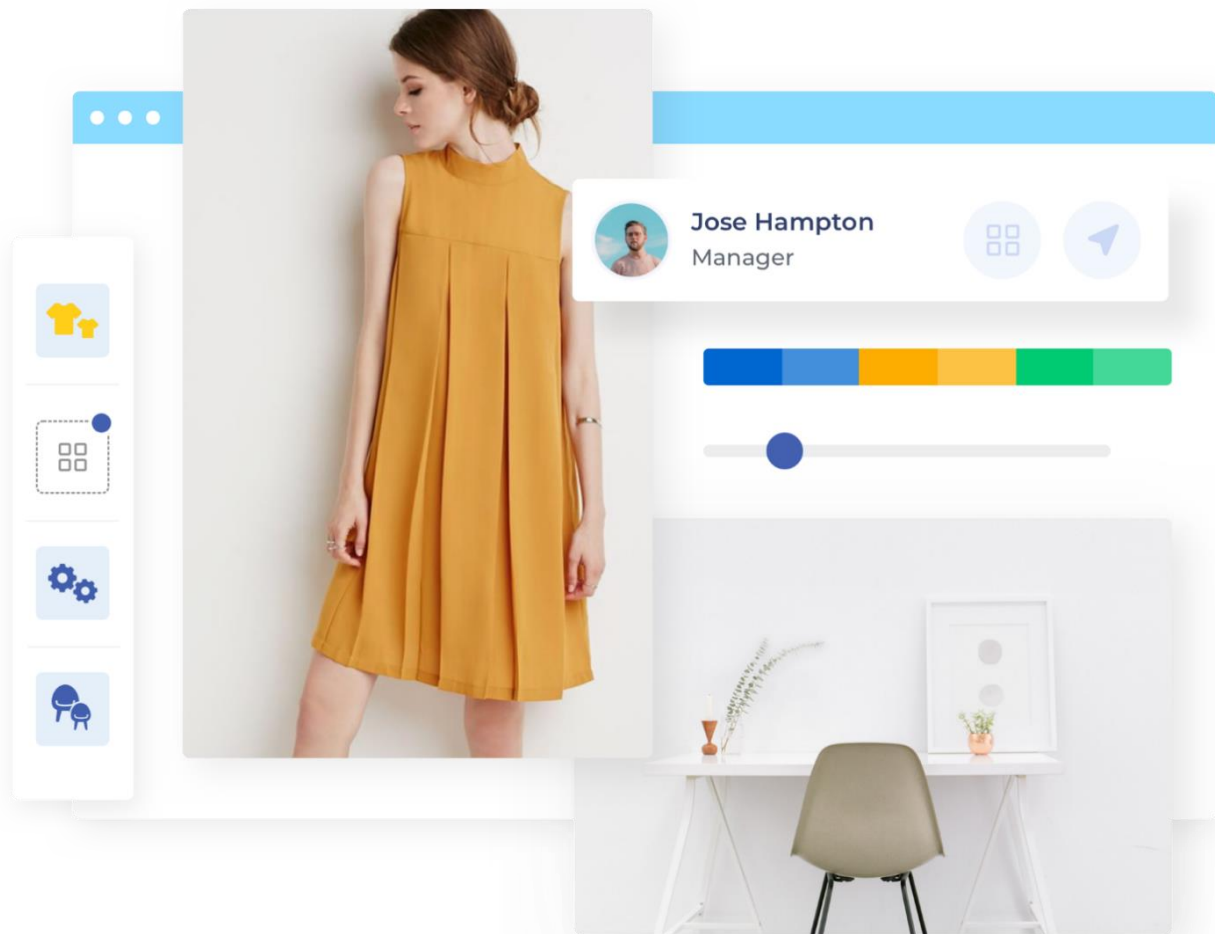


**WHITE PAPER**

VISUAL AND FULLTEXT SEARCH ON PORTALS

This document contains information about the full text search setup and functionality in TRIMIT for Business Central BC15.

Date: 15-04-2020

CONTENTS

Introduction.....	1
Setup.....	2
Full text search feature in SQL	2
Create full text search Database in Modelyn.....	2
Set theme value	3
SEARCH-DATA schedule	3
Search request in debug data	4
Search in Documents	6
Document search elements in Modelyn admin	7

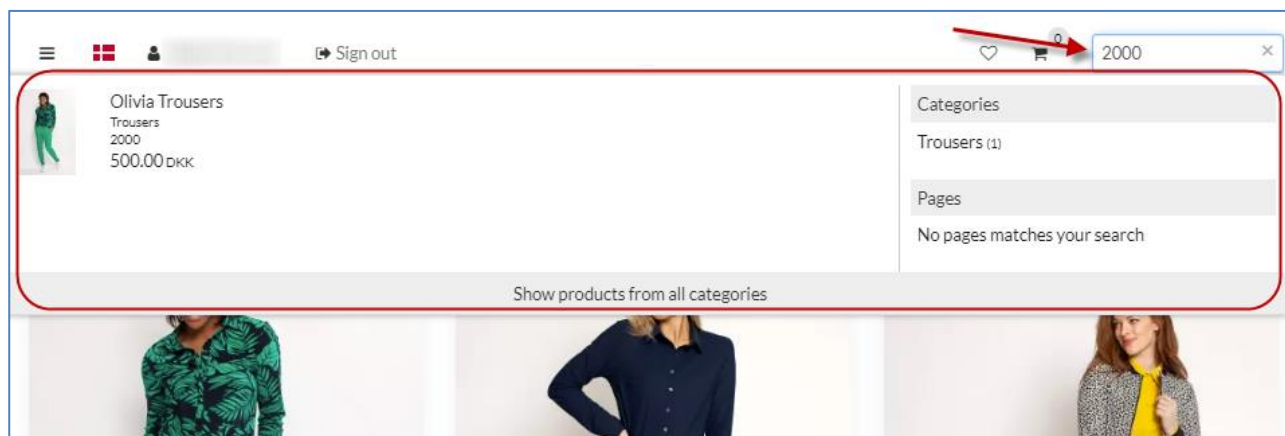
INTRODUCTION

This document describes a full text search functionality and the setup. The full text search refers to a behavior when user searched product in the search field on portal and along with the text being typed in the search field - there appears a drop-down menu with product occurrences which match with the search text. It is then possible to select a product, or category or a page in the drop down.

Full text search gives possibility to search products by typing either a product name, number (master or item number), un- / translated description (Modelyn and not BC description) or color (i.e. item attributes). Full text search in TRIMIT portals is based on indexes created and maintained in Modelyn database. It is available on TRIMIT B2C, B2B and SA portals.

The functionality described is the functionality in **TRIMIT for Business Central BC13 R2**.

The pictures in this White Paper are based on the demo database for **TRIMIT Business Central W1 BC13 R2**, which is based on Microsoft Dynamics 365 Business Central on premises 2019 release wave 2 (October release), in the demo company **CRONUS TRIMIT W1 Ltd.** (based on CRONUS International Ltd. of Microsoft Dynamics 365 Business Central).



SETUP

This chapter contains information about the setups that you need in order to start using full text visual drop-down search. This functionality is available from portals version BC13 R2. After upgrade from a previous version or after a brand-new installation it is required to accomplish the following tasks.

- Make sure full text search feature is enabled on web server SQL
- Create full text search database by running a script in Modelyn
- Set theme value *"UseFreeText" = TRUE*

Read more details below.

FULL TEXT SEARCH FEATURE IN SQL

Full text search is an optional component in SQL Server Database Engine. If full text search component was not installed, then rerun the SQL installation wizard and select it.

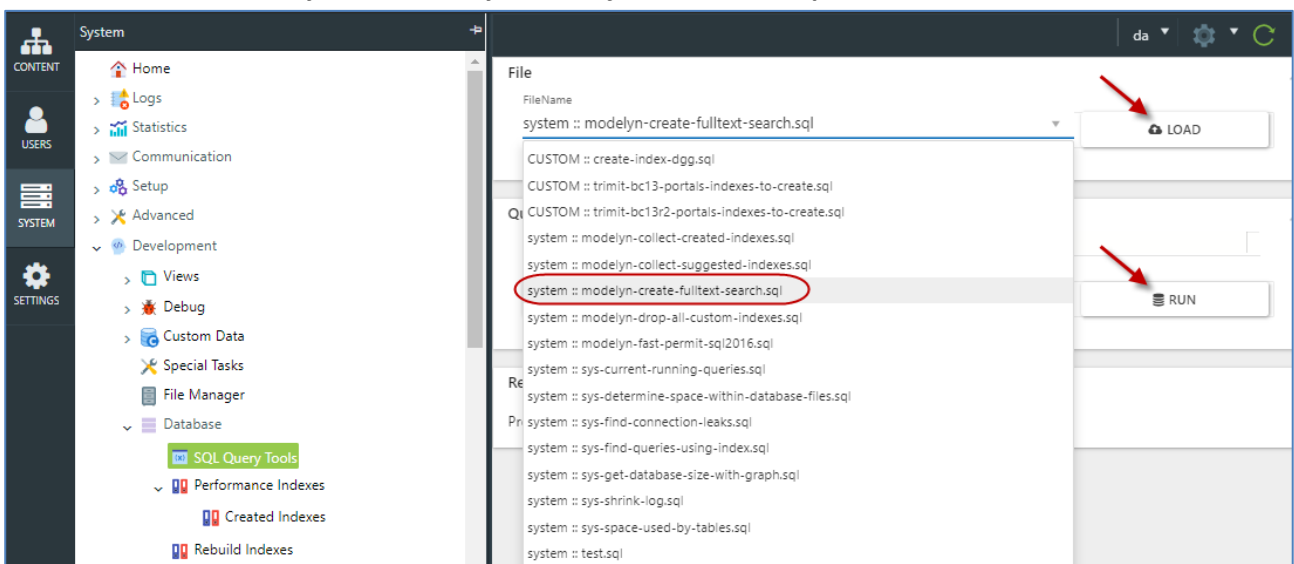
More info on full text search component is SQL here:

<https://docs.microsoft.com/en-us/sql/relational-databases/search/full-text-search?view=sql-server-ver15>

<https://codingsight.com/implementing-full-text-search-in-sql-server-2016-for-beginners/>

CREATE FULL TEXT SEARCH DATABASE IN MODELYN

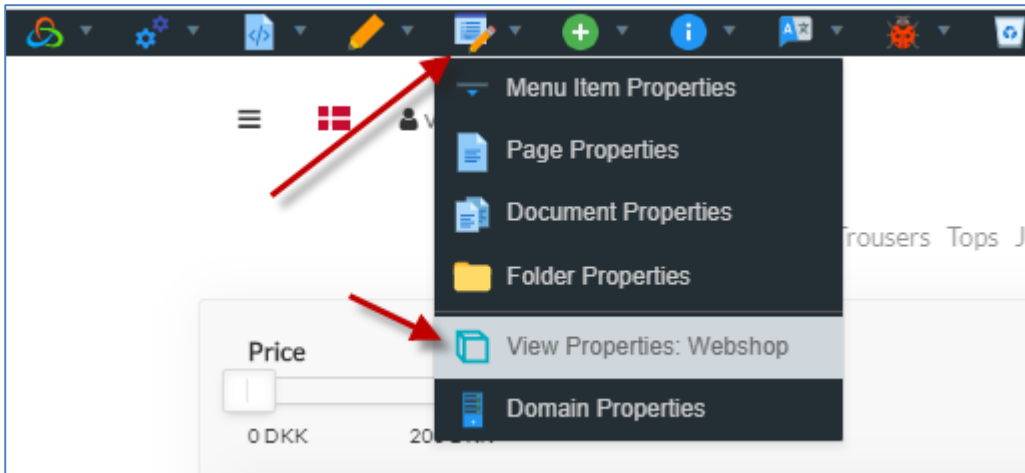
On a new installation or after upgrade from an older Modelyn version – it is required to create full text search database. To do this go to **Modelyn admin / System / Development / Database / SQL Query Tools**. Then in **FileName** select **system :: modelyn-create-fulltext-search.sql** and click **Load**. Then click **Run**.



The full text search database is created.

SET THEME VALUE

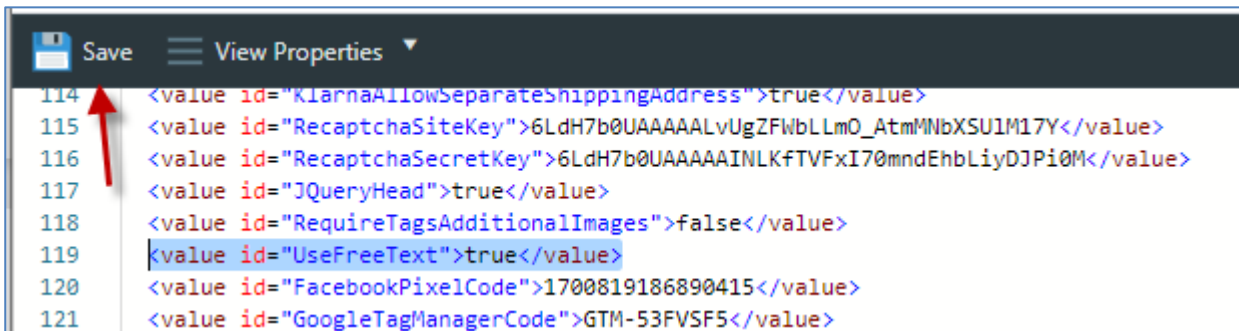
Set theme value **UseFreeText** to **TRUE**. To do this – log in as admin and select *View Properties* in admin helper menu:



Set the value as follows:

```
<value id="UseFreeText">true</value>
```

And click Save:



SEARCH-DATA SCHEDULE

Schedule **Products – Search Data (search-data-sync)** collects the data from other CustomData groups and adds the data into group TRM-SEARCH-DATA. Therefore, it is important to update the product data (including attributes) before running search data schedule. Or run all the schedules by clicking **Update shop**.

CustomData group TRM-SEARCH-DATA contains records per master / item and language.

Description of the group columns:

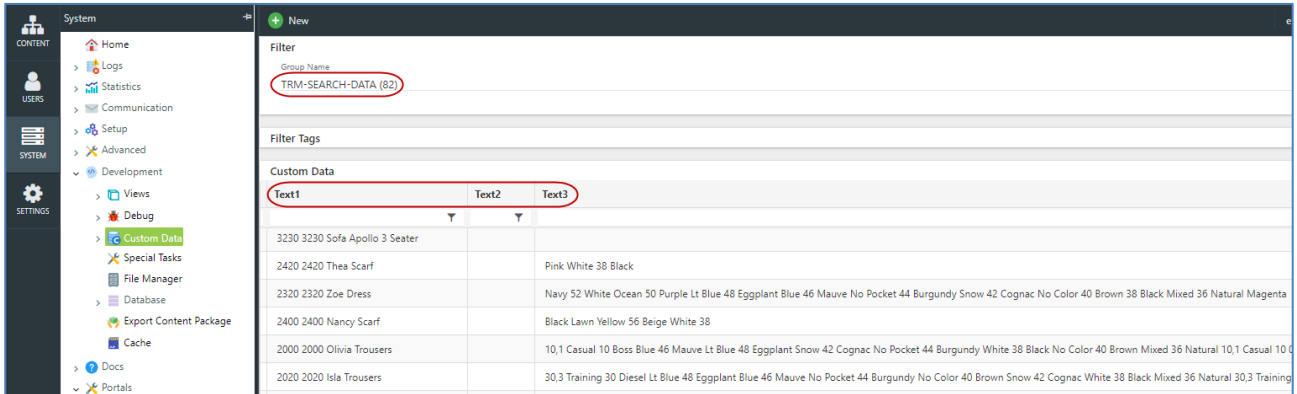
Text1 – populated with master number, item number, untranslated description / product name;

Text2 – populated with translated descriptions;

Text3 – populated with item attributes from BC (master attributes + item attributes);

(Text4) – not implemented, can be used for custom search words per master / item and language.

Hence, when searching for product on portals, the data is searched in text1, text2 and text3 columns and finds matching occurrences and returns the master / item in JSON format (plus image URL etc.).



Be aware, however, that other parameters can determine if you get what you expect in the search dropdown as products can be in different collections, order types etc. Therefore, even if you see the data in the TRM-SEARCH-DATA, but not in the search dropdown, it can be due to limitations on your order etc.

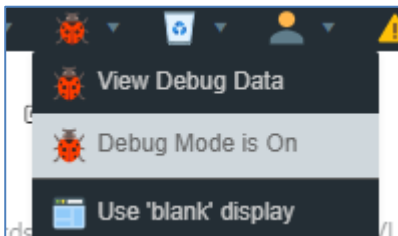
For example, if a collection is set on an order type and this order type is used on portal then product search will be done only in the context of this collection. Therefore, it is important that collection related products and portal category structure match.

But if order type does not have a related collection then product search will be done through the entire category structure which may also include collections (for example several collections under the same category). In this case a product related to a specific collection will be found and reflected in a search dropdown.

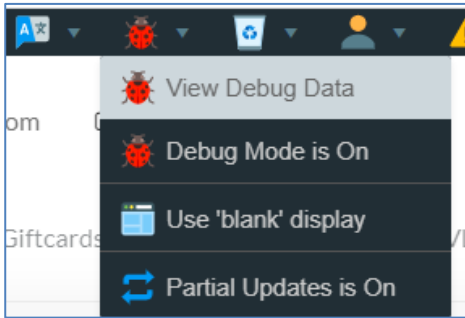
B2C does not use TRM-SEARCH-DATA as it uses directly C2 data, which has both master and item data. That is different from SA and B2B where TRM-SEARCH-DATA collects data from TRM-CATALOG, where only masters (without variants) and flat items are stored. Therefore, searching for a color on SA and B2B will not return master colors, but only colors for flat items in the category structure.

SEARCH REQUEST IN DEBUG DATA

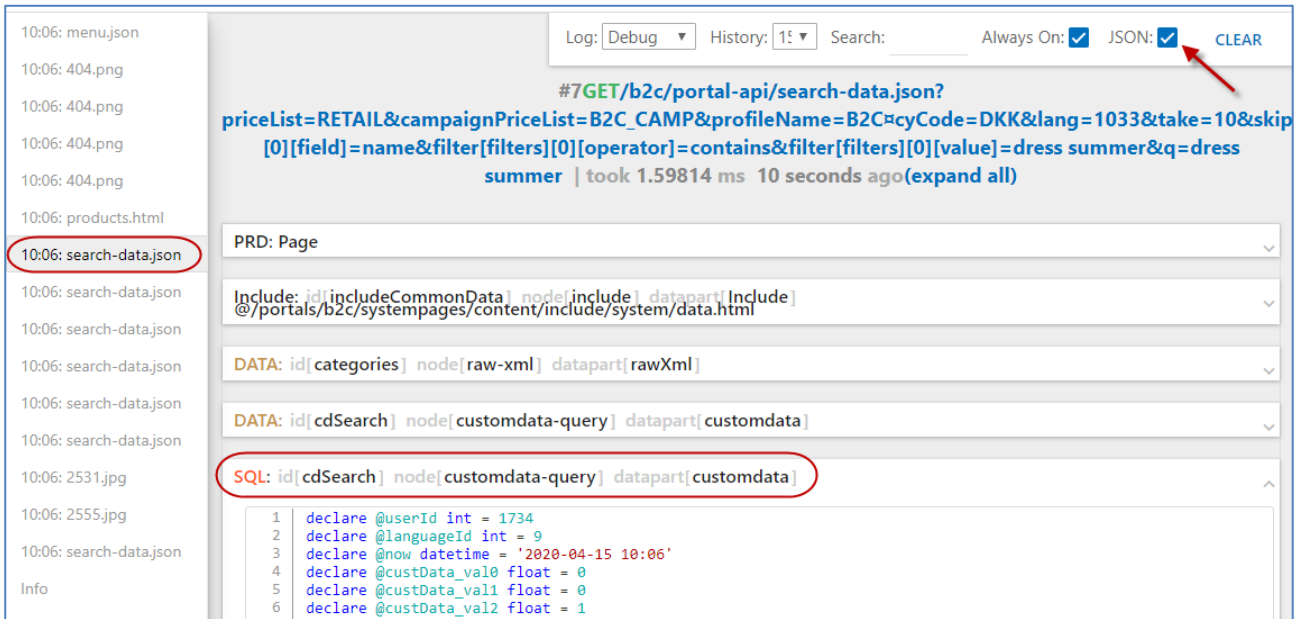
In order to see the actual request query when typing the search text do the following. Turn on the debug mode:



Type the search text. Now click View Debug Data:



Now click to show JSON data and select the last JSON request to the left and then expand **SQL: id[cdSearch]**



Log: Debug History: 1: Search: Always On: ☒ JSON: ☒ CLEAR

#7GET/b2c/portal-api/search-data.json?
 priceList=RETAIL&campaignPriceList=B2C_CAMP&profileName=B2C&cyCode=DKK&lang=1033&take=10&skip
 [0][field]=name&filter[filters][0][operator]=contains&filter[filters][0][value]=dress summer&q=dress
 summer | took 1.59814 ms 10 seconds ago(expand all)

PRD: Page

Include: id[includeCommonData] node[include] datapart[include]
 @/portals/b2c/systempages/content/include/system/data.html

DATA: id[categories] node[raw-xml] datapart[rawXml]

DATA: id[cdSearch] node[customdata-query] datapart[customdata]

SQL: id[cdSearch] node[customdata-query] datapart[customdata]

```

1 declare @userId int = 1734
2 declare @languageId int = 9
3 declare @now datetime = '2020-04-15 10:06'
4 declare @custData_val0 float = 0
5 declare @custData_val1 float = 0
6 declare @custData_val2 float = 1
  
```

To see the difference when **UseFreeText** theme value is set to TRUE or FALSE:

TRUE:



Log: Debug History: 1: Search:

WHERE ([_pricem].GroupName = N'PRICELIST'
 AND ((Key1 = custData.Key2)
 AND (Key8 = N'RETAIL')
 AND (Key9 = N'B2C_CAMP')
 AND (Key10 = N'DKK'))))
 , [_priceitem] =
 (SELECT TOP 1 _pricei.Float1 as [_itemprice]
 FROM CustomData as [_pricei]
 WHERE ([_pricei].GroupName = N'PRICELIST'
 AND ((Key3 = custData.Key3)
 AND (Key8 = N'RETAIL')
 AND (Key9 = N'B2C_CAMP')
 AND (Key10 = N'DKK'))))
 , [_categoryPath] =
 (SELECT TOP 1 _c2table.Text5 as [result]
 FROM CustomData as [_c2table]
 WHERE ([_c2table].GroupName = N'C2'
 AND (([_c2table].Int1 = N'1033')
 AND ([_c2table].String14 = CONCAT(LEFT(custData.String14, LEN(custData.String14) - CHARINDEX('/', REVERSE(custData.String14), 2)), '/'))))
 , (ISNULL(ft custData.[RANK], 0)) as [searchRank]
 FROM CustomData as [custData]
 LEFT JOIN CustomData as [_repItem] ON
 [_repItem].GroupName = N'TRM-REP-MASTERS'
 AND ((_repItem.Reckey = custData.Key10))
 LEFT JOIN CONTAINSTABLE([CustomData], ([String5],[String7],[String10],[Key1],[Text1],[Text4]), 'FORMSOF(INFLECTIONAL, "dress") AND FORMSOF(INFLECTIONAL, "summer")
 (custData.RecID = ft_custData.[KEY])
 WHERE ([custData].GroupName = N'C2'
 AND ((custData.Int1 = N'1033')
 AND (custData.Int2 = @custData_val0)
 AND (custData.Int3 <= @custData_val1)) AND ((ft_custData.[RANK] IS NOT NULL))
) as [custData]
 WHERE ([_RowNum] = @custData_val2)
 ORDER BY [_searchRank] DESC

UseFreeText = TRUE

FALSE:



```

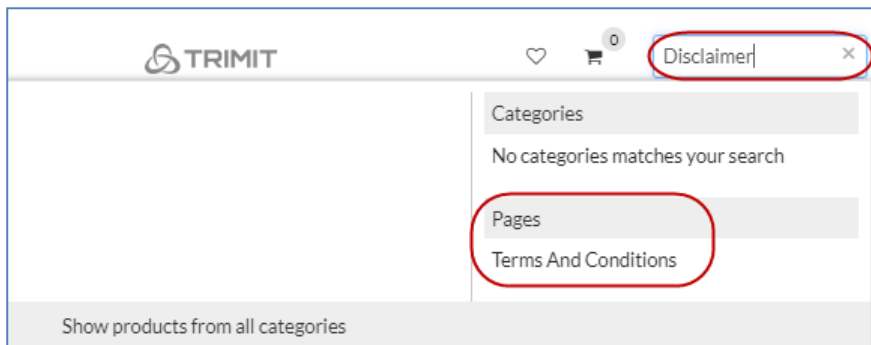
46 WHERE ([_pricem].GroupName = N'PRICELIST'
47 AND ( (Key1 = custData.Key2)
48 AND (Key8 = N'RETAIL')
49 AND (Key9 = N'B2C_CAMP')
50 AND (Key10 = N'DKK'))))
51
52 , [_priceitem] =
53 (SELECT TOP 1 _pricei.Float1 as [_itemprice]
54 FROM CustomData as [_pricei]
55 WHERE ([_pricei].GroupName = N'PRICELIST'
56 AND ( (Key3 = custData.Key3)
57 AND (Key8 = N'RETAIL')
58 AND (Key9 = N'B2C_CAMP')
59 AND (Key10 = N'DKK'))))
60
61 , [_categoryPath] =
62 (SELECT TOP 1 _c2table.Text5 as [result]
63 FROM CustomData as [_c2table]
64 WHERE ([_c2table].GroupName = N'C2'
65 AND ( ([_c2table].Int1 = N'1033')
66 AND ([_c2table].String14 = CONCAT(LEFT(custData.String14, LEN(custData.String14) - CHARINDEX('/', REVERSE(custData.String14), 2)), '/'))))
67
68 , custData.SortOrder as [_sort]
69 , custData.sysCreatedDt as [_createdDt]
70
71 FROM CustomData as [custData]
72 LEFT JOIN CustomData as [_repItem] ON
73 [_repItem].GroupName = N'TRM-REP-MASTERS'
74 AND ( [_repItem].RecKey = custData.Key10))
75
76 WHERE ([custData].GroupName = N'C2'
77 AND ( (custData.Int1 = N'1033')
78 AND (custData.Int2 = @custData_val0)
79 AND (custData.Int3 <= @custData_val1)
80 AND ((ISNULL(custData.String5, '') LIKE N'%white hat%')
81 OR (ISNULL(custData.String7, '') LIKE N'%white hat%')
82 OR (ISNULL(custData.String10, '') LIKE N'%white hat%')
83 OR (ISNULL(custData.Key1, '') LIKE N'%white hat%')
84 OR (ISNULL(custData.Text1, '') LIKE N'%white hat%')
85 OR (ISNULL(custData.Text4, '') LIKE N'%white hat%'))))
86 ) as [custData]
87 WHERE ([_RowNum] = @custData_val2)
88 ORDER BY [_sort] ASC, [_createdDt] DESC

```

UseFreeText = FALSE

SEARCH IN DOCUMENTS

it is also possible to search for word occurrences in website documents (non system pages).



Default non system pages (website documents) are the documents under Us section (in the website footer) – Terms, About, Customer Service, Size Guide, Legal Notice, Store Locator, Contact Form and Cookie Policy. To be able to search in more documents it is necessary to log in as admin and navigate to Soft admin / General in B2C or navigate to Soft Admin / Advanced in B2B or SA portal, scroll to the bottom of the page and locate section **Content included in search (Folders included in search** in SA and B2B).

B2C:

Content included in search

Additional search path 1

Additional search path 2

Additional search path 3

Additional search path 4

Additional search path 5

SA and B2B:

Folders included in searches

Folder no. 1

Folder no. 2

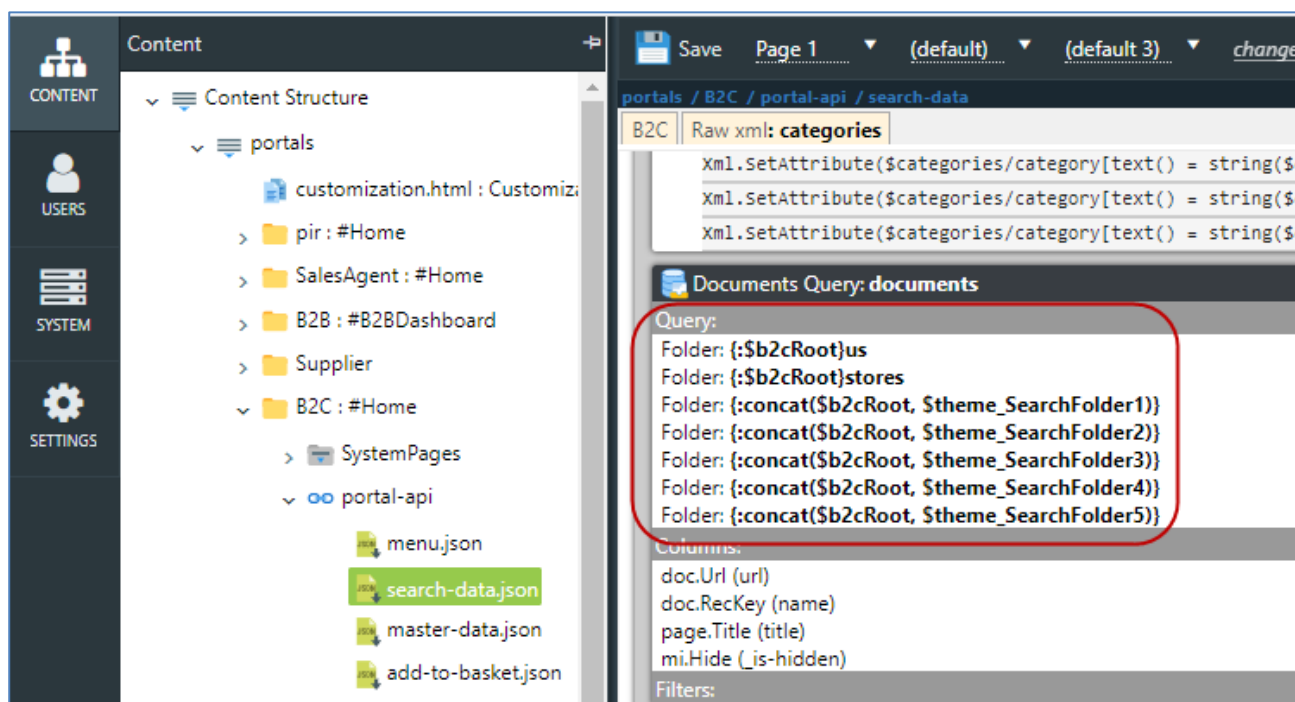
Folder no. 3

Folder no. 4

Folder no. 5

By default, B2C has the only option in drop down – **look-books**. If needed, more documents can be added to the content structure and then selected for search in this section. B2B and SA have more documents to select from in addition to default non system ones. The additional pages which can be added to search results are as follows: **statistics, services, look-books, News, Marketing, History, Orders, Home**.

DOCUMENT SEARCH ELEMENTS IN MODELYN ADMIN



The screenshot displays the Modelyn Admin interface. On the left, the 'CONTENT' sidebar shows the 'Content Structure' tree. Under 'portals', the 'B2C' folder is expanded, showing 'SystemPages' and 'portal-api'. The 'portal-api' folder contains several JSON files, including 'search-data.json' which is highlighted. On the right, the 'Documents Query' configuration is shown. The 'Query' section is circled in red and contains the following XML structure:

```
Folder: ({b2cRoot})us
Folder: ({b2cRoot})stores
Folder: ({concat($b2cRoot, $theme_SearchFolder1)})
Folder: ({concat($b2cRoot, $theme_SearchFolder2)})
Folder: ({concat($b2cRoot, $theme_SearchFolder3)})
Folder: ({concat($b2cRoot, $theme_SearchFolder4)})
Folder: ({concat($b2cRoot, $theme_SearchFolder5)})
```

The 'Columns' section lists the following fields: doc.Url (url), doc.RecKey (name), page.Title (title), mi.Hide (_is-hidden). The 'Filters' section is currently empty.