

String Search in SAS® Visual Analytics Records

Robbert Rahamat, Accenture Netherlands;

ABSTRACT

It is possible to create an in-memory record search in SAS Visual Analytics (SAS VA), on an existing or uploaded table, allowing you or other users to quickly search for records containing certain string values somewhere in that table. The fields that you want to search, need to be selected and constructed into the solution, but can be changed. It is possible to search with multiple string values. Each additional string to search for, will narrow the search.

INTRODUCTION

Using the In-Memory speed of SAS® LASR Tables, searching through a table can be done fast. This could be used for example on an uploaded ABT, an existing personnel table, a table with inventory details, or any other table with text fields that you want to be able to search fast.

This paper will show how to construct this search functionality with SAS Visual Analytics technology, for use by yourself or for other users that have access to the Dashboard.

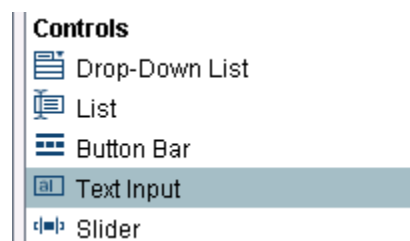
The solution will allow record search on partial string values in selected character fields of a SAS VA Table. Search on multiple string values can be done: each additional string value to search for, narrows the resulting records returned.

The same filtering can also be applied to other SAS VA Report Objects.

SAS VA COMPONENTS

Starting from an available SAS LASR Table that has been added as the Data Source in SAS VA, the solution needs the following SAS VA Components for its purpose:

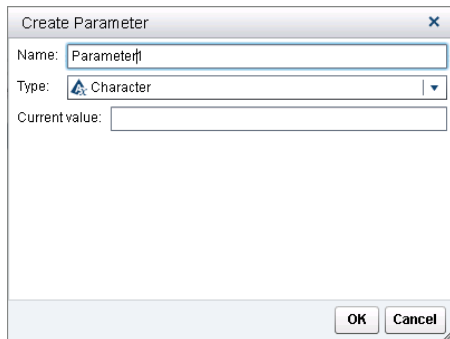
- A Text Input Control: for the user to enter the search strings into – as show in the display below.



Display 1. Text Input Control Definition

- A Parameter of type Character, e.g. "Parameter1": to hold the values the user enters – as shown in

the display below



Display 2. Parameter Definition

The Text Input Control can be added to:

- The Report Prompt area
- The Section Prompt area
- The Report objects area

In the last case Interactions need to be added to filter the Report Objects that are intended to be filtered, with the user input from the Text Input Control.

For each SAS VA Report Object that you intend to be filtered by the search strings, one or more filters need to be defined for that object.

FILTER FORMULA

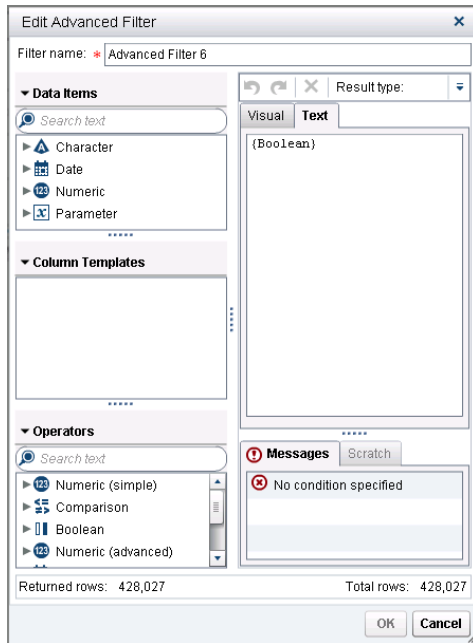
In each SAS VA Report Object that you want to be filtered with the string search functionality, one or more filters need to be defined in the filter section. The formula for and number of these filters depends on:

1. The fields that you have selected as fields that you want to search for these strings.
2. The number of strings that you want to search for and which string-number it is that you are implementing to search for.

Suppose you are searching six character fields of the table: Field1, Field 2, Field3, Field4, Field5 and Field6.

Suppose you are searching with at most four (partial) string values in these six character fields: String1, String2, String3 and String4.

String1 is entered first. String2, the second string entered, will narrow the results returned for string1, returning the records containing both String1 and String2. String3, the third string entered, will narrow the results further, returning the records containing all three strings. String4, the fourth and last string entered, will narrow the search further returning only records containing all four strings. The strings that you enter need to be separated by one or more spaces.



Display 3. Edit Advanced Filter Text window

In the Text window of the Edit Advanced Filter popup window – shown in the display above - write the following formula to enable string search for String1 in fields: Field1, Field2, Field3, Field4, Field5 and Field6:

```

IF ( GetLength(GetWord(UpCase(RemoveBlanks('Parameter1'p,
_LeadingAndTrailing_)), 1)) > 1 )
RETURN ( ( UpCase(RemoveBlanks('Field1'n, _All_)) Contains
GetWord(UpCase(RemoveBlanks('Parameter1'p, _LeadingAndTrailing_)), 1) ) OR
( ( UpCase(RemoveBlanks('Field2'n, _All_)) Contains
GetWord(UpCase(RemoveBlanks('Parameter1'p, _LeadingAndTrailing_)), 1) ) OR
( ( UpCase(RemoveBlanks('Field3'n, _All_)) Contains
GetWord(UpCase(RemoveBlanks('Parameter1'p, _LeadingAndTrailing_)), 1) ) OR
( ( UpCase(RemoveBlanks('Field4'n, _All_)) Contains
GetWord(UpCase(RemoveBlanks('Parameter1'p, _LeadingAndTrailing_)), 1) ) OR
( ( UpCase(RemoveBlanks('Field5'n, _All_)) Contains
GetWord(UpCase(RemoveBlanks('Parameter1'p, _LeadingAndTrailing_)), 1) ) OR
( UpCase(RemoveBlanks('Field6'n, _All_)) Contains
GetWord(UpCase(RemoveBlanks('Parameter1'p, _LeadingAndTrailing_)), 1) ) ) ) )
) ) )
ELSE ( 1 = 1 )

```

“Parameter1” is the parameter that is connected to the Text Input Control where the search strings are entered.

Continue to define three more advanced filters, each for another one of the search strings String2, String3 and String4. The search string number that you are implementing determines the value of the yellow marked numbers in the formula.

Save the report and refresh or open it for use of the string search functionality.

CONCLUSION

Using the In-Memory speed of SAS LASR Tables, fast string searching a table is possible with SAS VA. The solution presented in this paper offers you the basic tools for building your in-memory table search. It is a straightforward partial string search, not a regular expression search nor a phonetic search.

There might be limits to the number of fields that you can search and the number of search terms that you can search for: these limits have not been explored.

Also open for further investigation is the use of this filtering technique on other Report Objects in SAS VA.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Robbert Rahamat
Accenture Netherlands
+31646607226
robbert.rahamat@accenture.com

Copyright © 2016 Accenture

All rights reserved.

Accenture, its logo and High Performance Delivered are trademarks of Accenture.