**Implement Multiple keyword Search Functionality in Spotfire**

Spotfire by default does not have any search functionality, but it does provide a highly configurable Text area. We can use Iron Python scripting as an effective tool to implement desired functionality.

Similar is the case with Search option. Using Iron Python and text area we were able to create a search functionality that can search for multiple keywords within the given column.

Search Functionality:-

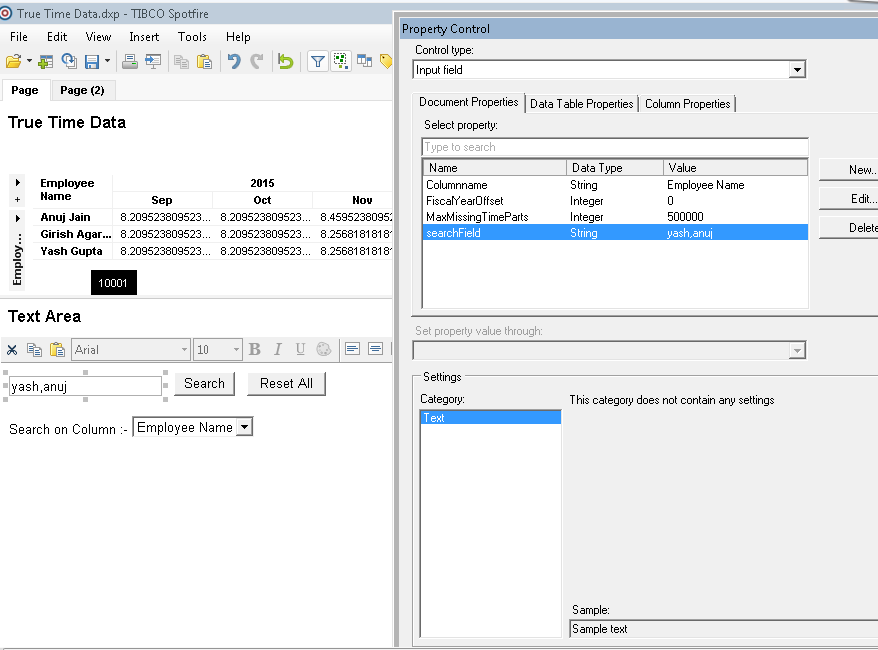
1. Comma separated multiple values can be used for search.
2. Any String case is applicable. Not case sensitive.
3. “SQL – Like” search not applicable in this solution

Let’s assume you have data which contains swipe in/out information about various employees in your company. Below is what it looks like.

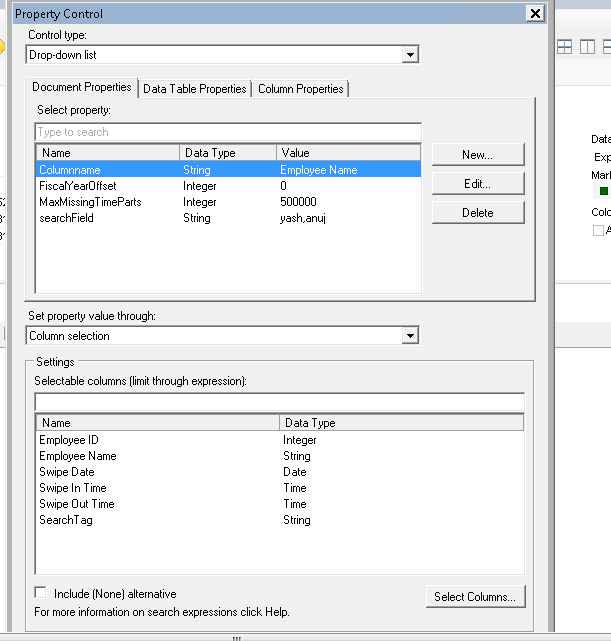
   


Add this table to a Spotfire dxp and follow the below steps to create search functionality for the same:-

1. Create a Text area with Input Field Property :-

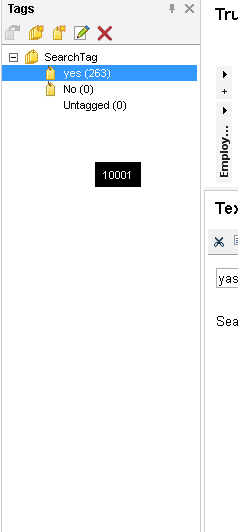


1. Create another property in same text area, which is a Drop down list and lists all available columns in existing table. This will help the user select the column within which the search is to be implemented



1. Create a Tag Collection with two tags :-

* Yes
* No



1. Create a script in same text area. Use Button for display and name the action control as “Search”. The code is as below :-

**from Spotfire.Dxp.Data import TagsColumn, RowSelection, DataValueCursor, IndexSet**

**import clr**

**import re**

**reasonList = Document.Properties["searchField"].split(',')**

**tagsColumn = myTable.Columns["SearchTag"].As[TagsColumn]()**

**filteringRowSelection = Document.ActiveFilteringSelectionReference.GetSelection(myTable)**

**\_reason = DataValueCursor.CreateFormatted(myTable.Columns[Document.Properties["Columnname"]])**

**tagsColumn.Tag("No", filteringRowSelection)**

**filteredSet = IndexSet(filteringRowSelection.AsIndexSet())**

**rowsToInclude = IndexSet(myTable.RowCount,True)**

**rowsToFilter = IndexSet(myTable.RowCount,False)**

**if (len(Document.Properties["searchField"]) == 0):**

**tagsColumn.Tag("yes", RowSelection(IndexSet.And(filteredSet,filteringRowSelection.AsIndexSet())))**

**elif (len(Document.Properties["searchField"]) > 0):**

**for row in myTable.GetRows(filteringRowSelection.AsIndexSet(),\_reason):**

**reason = \_reason.CurrentValue**

**reasonFound = 0**

**for item in reasonList:**

**result = re.findall('\\b'+item.strip()+'\\b', reason, flags=re.IGNORECASE)**

**if len(result)>0:**

**reasonFound = 1**

**break**

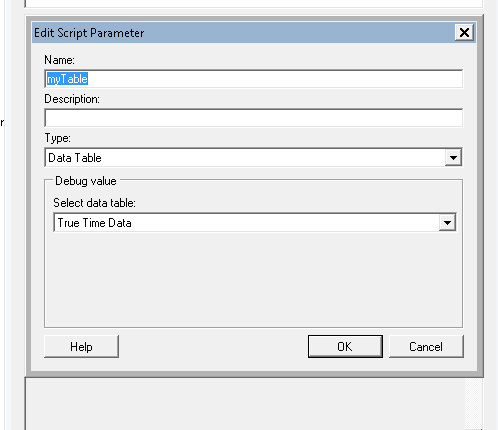
**if (reasonFound == 0):**

**filteredSet.RemoveIndex(int(row.Index))**

**tagsColumn.Tag("yes", RowSelection(IndexSet.And(filteredSet,filteringRowSelection.AsIndexSet())))**

Variables in Script:-

* SearchTag= Tag collection created earlier.
* Columnname= Drop down list property listing all columns in table.
* myTable = Created variable that represents data table



* searchField = Input Field property Name

1. Create another action control button named “Reset” and enter the below script :-

**from Spotfire.Dxp.Data import TagsColumn, RowSelection, DataValueCursor, IndexSet**

**import clr**

**import re**

**frs= Document.ActiveFilteringSelectionReference.GetSelection(myTable)**

**filteredSet = IndexSet(frs.AsIndexSet())**

**tagsColumn = myTable.Columns["SearchTag"].As[TagsColumn]()**

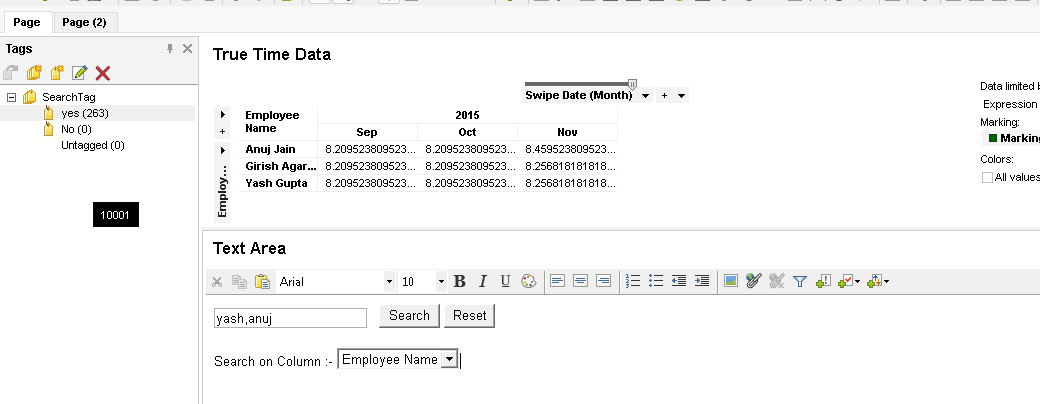
**tagsColumn.Tag("yes", RowSelection(IndexSet.And(filteredSet,frs.AsIndexSet())))**

**Document.Properties["searchField"]=''**

Variables in Script:-

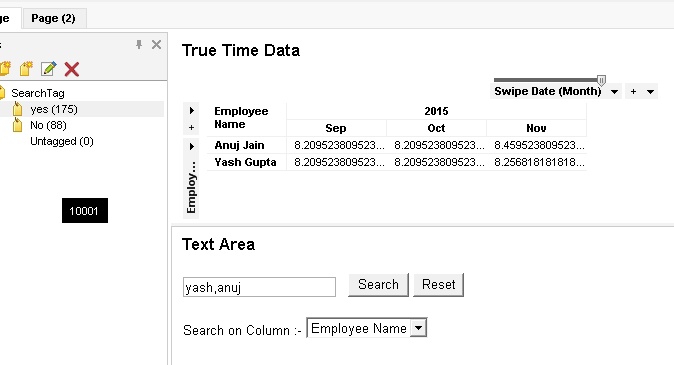
* SearchTag= Tag collection created earlier.
* myTable = Created variable that represents data table

Result should look like below:-



For example, we selected “Employee Name” as search Column and searched for two employees “yash,anuj”. We can put search keyword separated by comma and in any case.

On clicking search the result is as below:-



Click Reset button to clear all searches

