VB.NET ASP.NET

Full VB.NET ASP.NET (object class) A Search for filtering 4 categories, filter, merging arrays, arraylist intersections

This object looks big but if you need a tool to filter down products, this will work well. The class has multiple methods that can return intersections or unions of arrays. You can easily bind the results to a repeater or a drop down lists...etc. The object takes in arraylists so just get your arraylist of available options via a selection from your asp.net page with any control and then query your DB for a list to build your arrays and send them to the object. Use getRIDS() function to return the results you want. You will need to alter the sql statements in the object to select what you need according to your columns and of course DBName.

```
Imports System.Data
Imports System.Configuration.ConfigurationManager
Imports System.Collections

Public Class SearchItems
 Inherits ExpandIT.Page
 'some arrays not initialized for methods to know whether they need to add items or see if items are contained
 Public ItemIDsIntersectMember As ArrayList = Nothing
 Private ItemIDsIntersectHoldMember As New ArrayList
 Private ItemIDsUniqueMember As New ArrayList
 Public ItemIDsUnionMember As ArrayList = Nothing
 Private ItemIDsUnionHoldMember As New ArrayList
 Private avarRIDSMember As New ArrayList
 Private bvarRIDSMember As New ArrayList
 Private dvarRIDSMember As New ArrayList
 Private cvarRIDSMember As New ArrayList
 Private arraysWithValues As New ArrayList
 Private ItemIDsToReturn As New ArrayList
 Private ItemIDsIntersectavarMember As ArrayList = Nothing
 Private ItemIDsIntersectavarHoldMember As New ArrayList
 Private ItemIDsUnionavarMember As ArrayList = Nothing
 Private ItemIDsUniondvarMember As ArrayList = Nothing
 Private ItemIDsIntersectcvarMember As ArrayList = Nothing
 Private ItemIDsIntersectcvarHoldMember As New ArrayList
 Private ItemIDsUnioncvarMember As ArrayList = Nothing
 Private ItemIDsUnionbvarMember As ArrayList = Nothing
 Private selectedavars As New ArrayList
 Private selecteddvars As New ArrayList
 Private selectedcvars As New ArrayList
 Private selectedbvars As New ArrayList
 Private allSelections As New ArrayList
 Dim userMessage As String = "Within "
 '2D array
 Private Rids2D As New ArrayList
```

```
Public Sub New(ByVal avars As ArrayList, ByVal bvars As ArrayList, ByVal dvars As ArrayList,
ByVal cvars As ArrayList) ', ByVal season As String

'set the selections
 selectedavars = avars
 selectedbvars = bvars
 selecteddvars = dvars
 selectedcvars = cvars
 'setting the selectable Item Ids for avar, Unions and Intersections
 For Each col As String In avars
 Dim sql As String = ""
 Dim dt As New DataTable
 Dim avarList As New ArrayList
 Dim sqlStart As String = "Select ItemID From DBNameHere Where DBNameHere.[Group] ="
 Dim sqlAppend As String = ""
 Dim sqlEnd As String = " AND DBNameHere.Category = N'COLOR' AND
DBNameHere.DateCreated <= GETDATE()"
 sqlAppend = sqlAppend & SafeString(col)
 sql = sqlStart & sqlAppend & sqlEnd
 dt = SQL2DataTable(sql)
 If dt IsNot Nothing AndAlso dt.Rows.Count > 0 Then
 For Each dr As DataRow In dt.Rows
 avarList.Add(dr.Item("ItemID").ToString())
 Next
 End If
 'setIntersectItemIDs(avarList)
 setIntersectavarItemIDs(avarList)
 setUnionavarItemIDs(avarList)
 Next
 'setting the selectable Item Ids for bvars, Unions Only
 For Each occ As String In bvars
 Dim sql As String = ""
 Dim dt As New DataTable
 Dim bvarList As New ArrayList
 Dim sqlStart As String = "Select ItemID From DBNameHere Where DBNameHere.[Group] ="
 Dim sqlAppend As String = SafeString(occ)
 Dim sqlEnd As String = " AND DBNameHere.Category = N'OCCASION' AND
DBNameHere.DateCreated <= GETDATE()"
 sql = sqlStart & sqlAppend & sqlEnd
 dt = SQL2DataTable(sql)
 If dt IsNot Nothing AndAlso dt.Rows.Count > 0 Then
 For Each dr As DataRow In dt.Rows
 bvarList.Add(dr.Item("ItemID").ToString())
 Next
 End If
 'setIntersectItemIDs(bvarList)
 setUnionbvarItemIDs(bvarList)
```

```
Next
'setting the selectable Item Ids for dvars, Unions Only
For Each sz As String In dvars
Dim sql As String = ""
Dim dt As New DataTable
Dim dvarList As New ArrayList
Dim sqlStart As String = "Select ItemID From DBNameHere Where DBNameHere.[Group] ="
Dim sqlAppend As String = SafeString(sz)
Dim sqlEnd As String = " AND DBNameHere.Category = N'SIZE' AND
DBNameHere.DateCreated <= GETDATE()"
sql = sqlStart & sqlAppend & sqlEnd
dt = SQL2DataTable(sql)
If dt IsNot Nothing AndAlso dt.Rows.Count > 0 Then
For Each dr As DataRow In dt.Rows
dvarList.Add(dr.Item("ItemID").ToString())
Next
End If
'setIntersectItemIDs(dvarList)
setUniondvarItemIDs(dvarList)
Next
'setting the selectable Item Ids for cvars, Unions and Intersections
For Each flow As String In cvars
Dim sql As String = ""
Dim dt As New DataTable
Dim cvarList As New ArrayList
Dim sqlStart As String = "Select ItemID From DBNameHere Where DBNameHere.[Group] ="
Dim sqlAppend As String = ""
Dim sqlEnd As String = " AND DBNameHere.Category = N'FLOWERS' AND
DBNameHere.DateCreated <= GETDATE()"
sqlAppend = sqlAppend & SafeString(flow)
sql = sqlStart & sqlAppend & sqlEnd
dt = SQL2DataTable(sql)
If dt IsNot Nothing AndAlso dt.Rows.Count > 0 Then
For Each dr As DataRow In dt.Rows
cvarList.Add(dr.Item("ItemID").ToString())
Next
End If
'setIntersectItemIDs(cvarList)
setIntersectcvarItemIDs(cvarList)
setUnioncvarItemIDs(cvarList)
Next
End Sub
'Method for setting all intersectin arrays for the class final intersect member
Private Sub setIntersectItemIDs(ByVal arr As ArrayList)
Dim match As Boolean = False
If ItemIDsIntersectMember Is Nothing Then
ItemIDsIntersectMember = New ArrayList
For Each rid As String In arr
ItemIDsIntersectMember.Add(rid)
```

```
Next
Else
'check to see if at least one recipeID matches
For Each rid As String In arr
If ItemIDsIntersectMember.Contains(rid) Then
match = True
ItemIDsIntersectHoldMember.Add(rid)
End If
Next
If match Then
ItemIDsIntersectMember.Clear()
For Each Rid As String In ItemIDsIntersectHoldMember
ItemIDsIntersectMember.Add(Rid)
Next
ItemIDsIntersectHoldMember.Clear()
Else
'No matching recipeIDs
ItemIDsIntersectMember.Clear()
End If
End If
End Sub

'Method for setting final Union Class Member
Private Sub setUnionItemIDs(ByVal arr As ArrayList)
'cannot be sure the arr passed will be anything so initialize on entry
If arr Is Nothing Then
arr = New ArrayList
End If
'if first array then initialize union array
If ItemIDsUnionMember Is Nothing Then
ItemIDsUnionMember = New ArrayList
For Each rid As String In arr
ItemIDsUnionMember.Add(rid)
Next
Else
'check to see if at least one recipeID matches
For Each rid As String In arr
If ItemIDsUnionMember.Contains(rid) Then
'we already have it
Else
'add it
ItemIDsUnionMember.Add(rid)
End If
Next
End If
End Sub
'Method for setting the intersecting avars
Private Sub setIntersectavarItemIDs(ByVal arr As ArrayList)
```

```
Dim match As Boolean = False
 If ItemIDsIntersectavarMember Is Nothing Then
 ItemIDsIntersectavarMember = New ArrayList
 For Each rid As String In arr
 ItemIDsIntersectavarMember.Add(rid)
 Next
 Else
 'check to see if at least one recipeID matches
 For Each rid As String In arr
 If ItemIDsIntersectavarMember.Contains(rid) Then
 match = True
 ItemIDsIntersectavarHoldMember.Add(rid)
 End If
 Next
 If match Then
 ItemIDsIntersectavarMember.Clear()
 For Each Rid As String In ItemIDsIntersectavarHoldMember
 ItemIDsIntersectavarMember.Add(Rid)
 Next
 ItemIDsIntersectavarHoldMember.Clear()
 Else
 'No matching recipeIDs
 ItemIDsIntersectavarMember.Clear()
 End If
 End If
 End Sub
 'Method for setting Union of selected avars
 Private Sub setUnionavarItemIDs(ByVal arr As ArrayList)
 'if first array then initialize union array
 If ItemIDsUnionavarMember Is Nothing Then
 ItemIDsUnionavarMember = New ArrayList
 For Each rid As String In arr
 ItemIDsUnionavarMember.Add(rid)
 Next
 Else
 'check to see if at least one recipeID matches
 For Each rid As String In arr
 If ItemIDsUnionavarMember.Contains(rid) Then
 'we already have it
 Else
 'add it
 ItemIDsUnionavarMember.Add(rid)
 End If
 Next
 End If
 End Sub

'Method for setting Intersecting cvars selections
 Private Sub setIntersectcvarItemIDs(ByVal arr As ArrayList)
```

```
Dim match As Boolean = False
 If ItemIDsIntersectcvarMember Is Nothing Then
 ItemIDsIntersectcvarMember = New ArrayList
 For Each rid As String In arr
 ItemIDsIntersectcvarMember.Add(rid)
 Next
 Else
 'check to see if at least one recipeID matches
 For Each rid As String In arr
 If ItemIDsIntersectcvarMember.Contains(rid) Then
 match = True
 ItemIDsIntersectcvarHoldMember.Add(rid)
 End If
 Next
 If match Then
 ItemIDsIntersectcvarMember.Clear()
 For Each Rid As String In ItemIDsIntersectcvarHoldMember
 ItemIDsIntersectcvarMember.Add(Rid)
 Next
 ItemIDsIntersectcvarHoldMember.Clear()
 Else
 'No matching recipeIDs
 ItemIDsIntersectcvarMember.Clear()
 End If
 End If
 End Sub

'Method for setting Union cvar selections
 Private Sub setUnioncvarItemIDs(ByVal arr As ArrayList)
 'if first array then initialize union array
 If ItemIDsUnioncvarMember Is Nothing Then
 ItemIDsUnioncvarMember = New ArrayList
 For Each rid As String In arr
 ItemIDsUnioncvarMember.Add(rid)
 Next
 Else
 'check to see if at least one recipeID matches
 For Each rid As String In arr
 If ItemIDsUnioncvarMember.Contains(rid) Then
 'we already have it
 Else
 'add it
 ItemIDsUnioncvarMember.Add(rid)
 End If
 Next
 End If
 End Sub

'Method for setting Union for selected dvars
```

```
Private Sub setUniondvarItemIDs(ByVal arr As ArrayList)
'if first array then initialize union array
If ItemIDsUniondvarMember Is Nothing Then
ItemIDsUniondvarMember = New ArrayList
For Each rid As String In arr
ItemIDsUniondvarMember.Add(rid)
Next
Else
'check to see if at least one recipeID matches
For Each rid As String In arr
If ItemIDsUniondvarMember.Contains(rid) Then
'we already have it
Else
'add it
ItemIDsUniondvarMember.Add(rid)
End If
Next
End If
End Sub

'Method for setting Union of selected bvars
Private Sub setUnionbvarItemIDs(ByVal arr As ArrayList)
'if first array then initialize union array
If ItemIDsUnionbvarMember Is Nothing Then
ItemIDsUnionbvarMember = New ArrayList
For Each rid As String In arr
ItemIDsUnionbvarMember.Add(rid)
Next
Else
'check to see if at least one recipeID matches
For Each rid As String In arr
If ItemIDsUnionbvarMember.Contains(rid) Then
'we already have it
Else
'add it
ItemIDsUnionbvarMember.Add(rid)
End If
Next
End If
End Sub

'methods for getting selections
Public Function getSelectedavars() As ArrayList
Return selectedavars
End Function
Public Function getSelecteddvars() As ArrayList
Return selecteddvars
End Function
Public Function getSelectedbvars() As ArrayList
```

```
Return selectedbvars
End Function
Public Function getSelectedcvars() As ArrayList
Return selectedcvars
End Function
Public Function getAllSelections() As ArrayList
For Each i As String In selectedavars
allSelections.Add(i)
Next
For Each i As String In selecteddvars
allSelections.Add(i)
Next
For Each i As String In selectedbvars
allSelections.Add(i)
Next
For Each i As String In selectedcvars
allSelections.Add(i)
Next
Return allSelections
End Function

'Method for generating the counts on Postback
Public Function getCounts() As DataTable
Dim dt As New DataTable
If ItemIDsUniqueMember.Count <> 0 Then
'for each ItemID in cvars, select the cvar to display and the count of how many recipes it
appears in
Dim sql As String = ""
Dim sqlStart As String = ""
Dim sqlEnd As String = ""
Dim sqlAppend As String = ""
sqlStart = "SELECT COUNT(*) AS Count, Category, [Group] " & _
"FROM DBNameHere " & _
"WHERE (ItemID IN (SELECT ItemID FROM DBNameHere " & _
"WHERE ItemID IN ("
For i As Integer = 0 To ItemIDsUniqueMember.Count - 1
Dim val As String = ItemIDsUniqueMember(i).ToString()
'add comma
If i < cvarRIDSMember.Count - 1 Then
sqlAppend = sqlAppend & SafeString(val) & ","
Else
'end of string, no comma
sqlAppend = sqlAppend & SafeString(val)
End If
Next
sqlEnd = ") AND DBNameHere.DateCreated <= GETDATE() GROUP BY ItemID))" & _
"GROUP BY Category, [Group]" & _
"ORDER BY Category, [Group]"
sql = sql & sqlStart & sqlAppend & sqlEnd
```

```
dt = SQL2DataTable(sql)
 Else
 Dim sql As String = ""
 Dim sqlStart As String = ""
 Dim sqlEnd As String = ""
 Dim sqlAppend As String = ""
 sqlStart = "SELECT COUNT(*) AS Count, Category, [Group] " & _
 "FROM DBNameHere " & _
 "WHERE (ItemID IN (SELECT ItemID FROM DBNameHere " & _
 "WHERE ItemID IN ("
 sqlAppend = " Select ItemID From DBNameHere "
 sqlEnd = ") AND DBNameHere.DateCreated <= GETDATE() GROUP BY ItemID))" & _
 "GROUP BY Category, [Group]" & _
 "ORDER BY Category, [Group]"
 sql = sql & sqlStart & sqlAppend & sqlEnd

dt = SQL2DataTable(sql)
 End If
 Return dt
 End Function
 'Method that returns all the Available ItemIDs
 Public Function getRIDS() As ArrayList
 'We want the dvar Union (U_Sz1) that Intersects with the Occassion Union (U_Oc1) Group As
Group (I_Sz1_Oc1)---
 'Next get the intersection of avars as I_Col and Union of avars as U_Col---
 'then get the intersection of cvars selected as I_Flwr along with the Union of the selected cvars
as U_Flwr---
 'The dvars or bvars should expand the pool of ItemIDs available
 'The avar and cvar selections should drain the pool if the pool has water
 'First check the pool for water by checking if we have any selected dvars or bvars
 Dim PoolIsEmpty As Boolean = False
 Dim needIntersect As Boolean = False
 Dim I_Sz1_Oc1 As ArrayList = Nothing
 Dim getAll As Boolean = False
 If ItemIDsUnionbvarMember Is Nothing And ItemIDsUniondvarMember Is Nothing Then
 PoolIsEmpty = True
 End If
 If Not ItemIDsUnionbvarMember Is Nothing And Not ItemIDsUniondvarMember Is Nothing
Then
 needIntersect = True
 End If
 If needIntersect Then
 I_Sz1_Oc1 = New ArrayList
 I_Sz1_Oc1.Add(ItemIDsUnionbvarMember)
 I_Sz1_Oc1.Add(ItemIDsUniondvarMember)
 For Each arl As ArrayList In I_Sz1_Oc1
 setIntersectItemIDs(arl)
 Next
 End If
```

```
'Now check if either dvar or bvar only
Dim sz_or_occ As ArrayList = Nothing
'now check to see if either one or the other was selected
If Not PoolIsEmpty And Not needIntersect Then
'then either dvar or bvar has been selected but not both so no intersection is needed
If Not ItemIDsUnionbvarMember Is Nothing Then
sz_or_occ = New ArrayList
'bvar has values
sz_or_occ = ItemIDsUnionbvarMember
End If
If Not ItemIDsUniondvarMember Is Nothing Then
sz_or_occ = New ArrayList
'dvar has values
sz_or_occ = ItemIDsUniondvarMember
End If
End If

'if both were then the class member ItemIDsIntersectMember is holding values
If Not ItemIDsIntersectMember Is Nothing Then
sz_or_occ = New ArrayList
'set values equal to our function variable then clear the class member
For Each Str As String In ItemIDsIntersectMember
sz_or_occ.Add(Str)
Next
ItemIDsIntersectMember = Nothing
End If

'From here, can use sz_or_occ value for the pool if there is one
Dim unionIsSet As Boolean = False
Dim intersectIsSet As Boolean = False
Dim nothingSelected As Boolean = True
If sz_or_occ Is Nothing Then
'we don't have a pool
'subtract the avars and bvars from the pool if they are selected
'the class member for ItemIds will be nothing and so is sz_or_occ here
'class members are already set for the intersect and union for both avars and cvars if they were
selected
If Not ItemIDsIntersectavarMember Is Nothing Then
setIntersectItemIDs(ItemIDsIntersectavarMember)
intersectIsSet = True
End If
If Not ItemIDsUnionavarMember Is Nothing Then
setUnionItemIDs(ItemIDsUnionavarMember)
unionIsSet = True
End If
If Not ItemIDsIntersectcvarMember Is Nothing Then
setIntersectItemIDs(ItemIDsIntersectcvarMember)
intersectIsSet = True
End If
```

```
 If Not ItemIDsUnioncvarMember Is Nothing Then
 setUnionItemIDs(ItemIDsUnioncvarMember)
 unionIsSet = True
 End If
 nothingSelected = False
 End If

If Not sz_or_occ Is Nothing And Not ItemIDsIntersectavarMember Is Nothing Then
 'send the pool and avars selected
 setIntersectItemIDs(sz_or_occ)
 'sent the avars next
 setIntersectItemIDs(ItemIDsIntersectavarMember)
 nothingSelected = False

End If

If Not sz_or_occ Is Nothing And Not ItemIDsIntersectcvarMember Is Nothing Then
 'send the pool and cvars selected
 setIntersectItemIDs(sz_or_occ)
 setIntersectItemIDs(ItemIDsIntersectcvarMember)
 nothingSelected = False

End If

If nothingSelected Then
 'we have pool but have not set the ItemIDsIntersectMember because we cleared it above and
want to reuse it
 'and no avars or cvars were selected
 ItemIDsIntersectMember = New ArrayList
 For Each st As String In sz_or_occ
 ItemIDsIntersectMember.Add(st)
 Next
 End If

'Now check to see if we have anything set
 If unionIsSet Or intersectIsSet Then
 'we didn't have a pool but we had cvar or avar selections
 If intersectIsSet Then
 'set the unique members to return with our intersection then clear it for reuse
 For Each st As String In ItemIDsIntersectMember
 ItemIDsUniqueMember.Add(st)
 Next
 ItemIDsIntersectMember.Clear()
 End If
 If unionIsSet Then
 Dim temp As New ArrayList
 'hold current Union so we can reuse the setUnionItemIDs method
 For Each st As String In ItemIDsUnionMember
 temp.Add(st)
```

```
 Next
 'clear the union class member
 ItemIDsUnionMember.Clear()
 If intersectIsSet Then
 'intersection first
 setUnionItemIDs(ItemIDsUniqueMember)
 'clear ItemIDsUniqueMember because the union now contains the ids needed
 ItemIDsUniqueMember.Clear()
 End If
 'then add the union on top of the intersects that were just added before
 setUnionItemIDs(temp)
 'now ItemIDsUnionMember contains what we want, rebuild ItemIDsUniqueMember
 For Each st As String In ItemIDsUnionMember
 ItemIDsUniqueMember.Add(st)
 Next
 'clear the union for reuse
 ItemIDsUnionMember.Clear()
 End If

ElseIf Not ItemIDsIntersectMember Is Nothing Then
 'we had a pool and cvar or avar intersections
 ItemIDsUniqueMember = ItemIDsIntersectMember
 Else
 'we had no selections
 getAll = True
 End If

'gets all recipeIds when user arrives or nothing selected
 If getAll Then
 Dim sql As String = ""
 Dim dt As DataTable

sql = "SELECT sub.ItemID, rn, sub.DateCreated From (SELECT DBNameHere.ItemID,
DBNameHere.ItemName, " & _
 "DBNameHere.ItemDescription, DBNameHere.FeatureImage, DBNameHere.[Group],
DBNameHere.Category, row_number() over (partition by DBNameHere.ItemID " & _
 "Order By DBNameHere.ItemID) as rn, DBNameHere.DateCreated FROM DBNameHere
GROUP BY DBNameHere.ItemID, DBNameHere.ItemName, DBNameHere.ItemDescription, "
& _
 "DBNameHere.FeatureImage, DBNameHere.DateCreated, DBNameHere.[Group],
DBNameHere.Category HAVING " & _
 "DBNameHere.DateCreated <= GETDATE()) sub Where rn < 2 Order by DateCreated desc"
 dt = SQL2DataTable(sql)
 'Debug.DebugValue(sql, "sql123")
 If dt IsNot Nothing AndAlso dt.Rows.Count > 0 Then
 For Each dr As DataRow In dt.Rows
 ItemIDsUniqueMember.Add(dr.Item("ItemID").ToString())
 Next
 End If
```

```
 Return ItemIDsUniqueMember
 End If
 'class member returned with or without values
 Return ItemIDsUniqueMember
 End Function

Public Function getUserMessage() As String
 Return userMessage
 End Function
End Class
```

Unique solution ID: #1005
Author: Mark
Last update: 2014-08-15 22:19