Web Database Programming

Created: 2011-01-21

Last update: 2014-01-14

Contents

Introduction

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Use EasyDataSet as Data Source

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Bind-data to single field

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Data Query

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Data Search

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Data Navigation

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Modify and Save Data

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

MySql Database Credential

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Bind Data to Html Table

See http://www.limnor.com/support/webDatabaseProgramming1.pdf

Web Page Security

See http://www.limnor.com/support/webDatabaseProgramming2.pdf

Field Editors

See http://www.limnor.com/support/webDatabaseProgramming3.pdf

Data Repeater

In chapter “Bind-data to single field”, we saw that a form can be designed with data-bound controls. Data from database are automatically displayed on the controls. The user may modify data on the controls and the data modifications can be saved back to database.

In such arrangement, one record is displayed on one web page.

Data Repeater allows you to design the form with data-bound controls, but the same design can be repeated on one web page. Thus many records can be displayed on one web page.

For details, see http://www.limnor.com/support/WebDataRepeater.pdf

Change Query Filters at Runtime

EasyDataSet has some methods allowing changing query filters at runtime so that you may build UI to let your users decide how to search database for data.

- `QueryWithWhere` – When you use this method to create an action, Limnor Studio analysis the filters you want to use and find the parameters in the filters. Each parameter becomes a property of the action. The advantage of using this method is simple to use. The disadvantage of this method is that you have to create an action for every situation your users may want to search the database. Suppose you give your users 5 different filters to use then there will be about 80 different combinations of those filters. You do not want to create so many actions.
- `QueryWithWhereDynamic` -- You may use this method to create an action which may search the database for all situations. This chapter demonstrates how this method can be used.

Arrange for dynamic query

EasyDataSet has a property, **DynamicParameters**. You use this property to define all parameters to be used in all dynamic query actions. Each parameter defined becomes a property of a `QueryWithWhereDynamic` action. Each time a `QueryWithWhereDynamic` action is executed, the query parameters have to be among the defined parameters.

Define Search Options

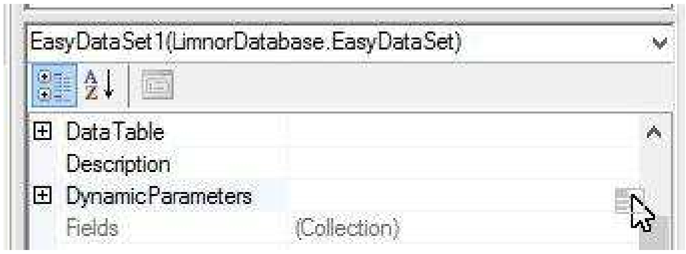
Suppose we want to search for employee records in a database, and give users following search options:

- Employee name contains a given text
- Notes field contains a given text
- Employee date is later than a given date

Define parameters for search options

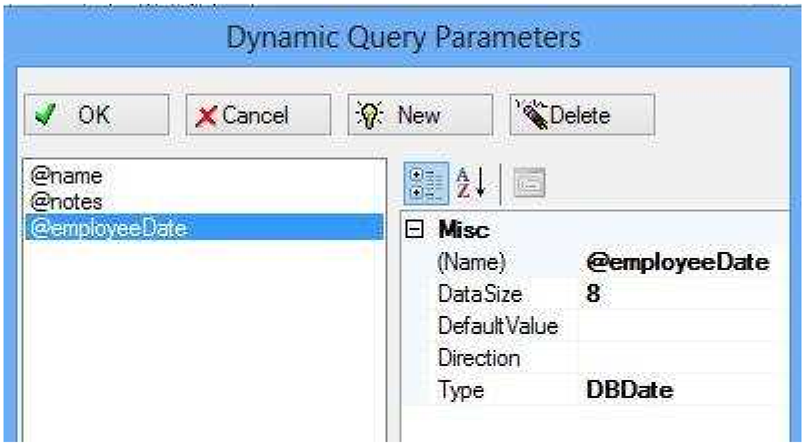
- @name – representing the given text for Employee Name search
- @notes – representing the given text for Notes search.
- @employeeDate – representing the given date for employee date search

Edit DynamicParameters property:



EasyDataSet1(LimnorDatabase.EasyDataSet)

⊞ DataTable	
Description	
⊞ DynamicParameters	
Fields	(Collection)



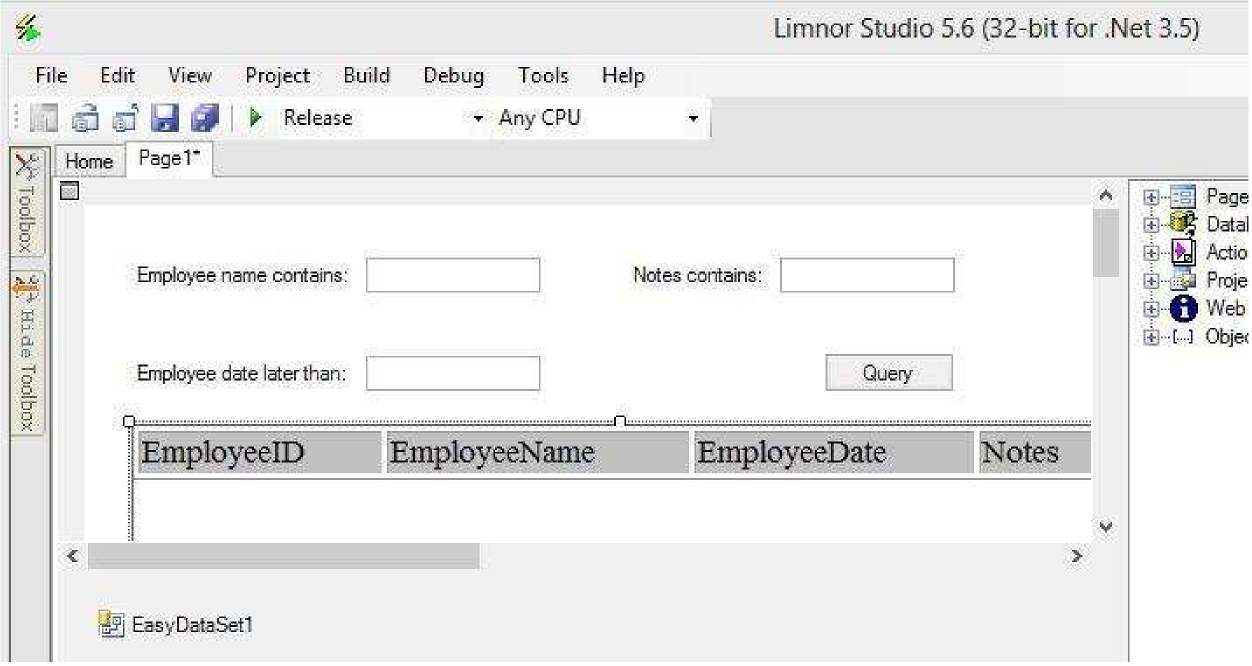
@name
@notes
@employeeDate

Misc

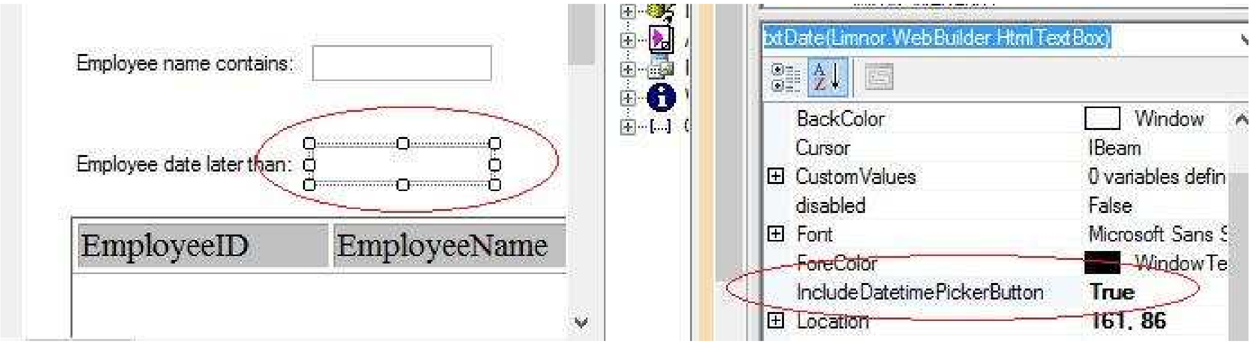
(Name)	@employeeDate
DataSize	8
DefaultValue	
Direction	
Type	DBDate

Create Search Options UI

Let’s build UI to allow the user to enter search values:



Note that the text box for entering date value has its IncludeDatetimePickerButton set to True:



xtDate(Limnor.WebBuilder.HtmlTextBox)

BackColor	Window
Cursor	IBeam
CustomValues	0 variables defin
disabled	False
Font	Microsoft Sans S
ForeColor	WindowTe
IncludeDatetimePickerButton	True
Location	161, 86

Make dynamic query filters

To construct query filters is to write SQL WHERE clause. If you do not know SQL language then you may use Limnor Studio Query Builder to create filters and see what is the corresponding WHERE clause the Query Builder generates. The Query Builder can be accessed by setting SQL property of EasyDataSet.

Suppose the SQL property for our EasyDataSet is as following:

```
SELECT
`employees`.`EmployeeID`,`employees`.`EmployeeName`,`employees`.`EmployStartDate`,`employees`.`
Notes` FROM `employees`
```

The filters for each search option are as following:

- Employee name contains a given text; parameter @name represents the given text for Employee Name search. The filter is
  ```
  EmployeeName LIKE CONCAT('%',@name,'%')
  ```
- Notes field contains a given text; parameter @notes represents the given text for Notes search. The filter is
  ```
  Notes LIKE CONCAT('%',@notes,'%')
  ```
- Employee date is later than a given date; parameter @employeeDate represents the given date for employee date search. The filter is
  ```
  EmployStartDate > @employeeDate
  ```

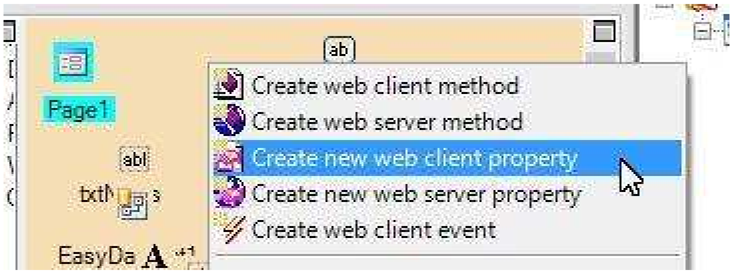
Suppose our UI logic is that if a text box is empty then the corresponding filter should not be used. All used filters are linked by AND logic. For example, if the user enters data for the name search and notes search text boxes, then the WHERE clause should be as following:

```
EmployeeName LIKE CONCAT('%',@name,'%') AND Notes LIKE CONCAT('%',@notes,'%')
```

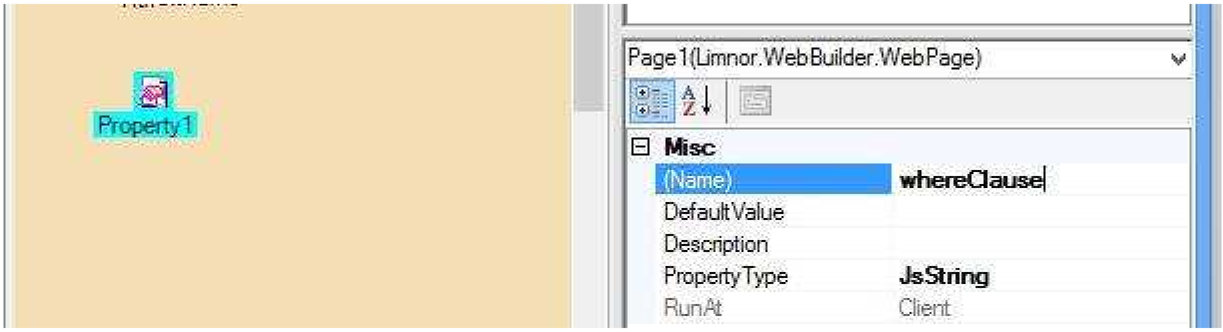
There can be many ways of generating WHERE clause following the UI logic. Here we describe one way of doing it.

Page property for WHERE clause

We create a property to hold the WHERE clause we are going to create:



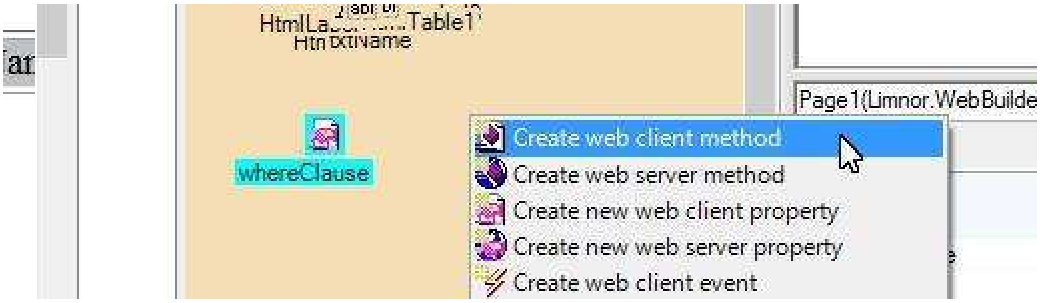
Rename the new property to whereClause:



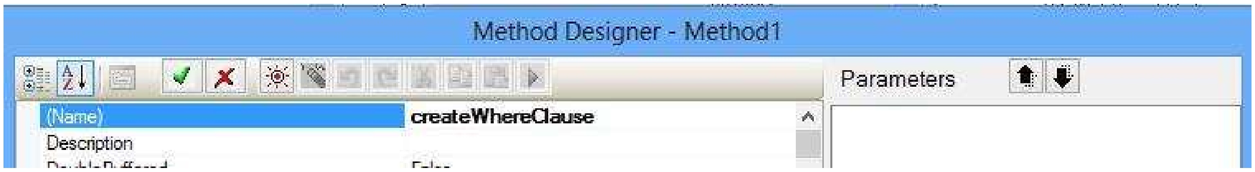
(Name) whereClause
DefaultValue
Description
PropertyType JsString
RunAt Client

Method for generating WHERE clause

Create a method to generate the WHERE clause:



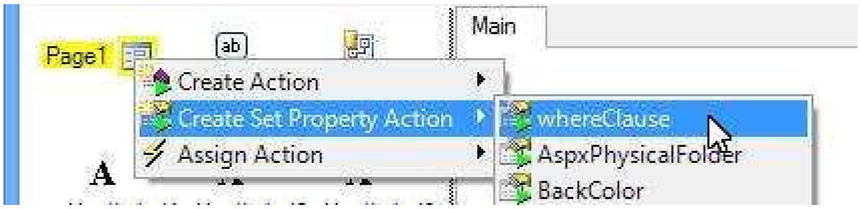
Rename the new method to createWhereClause:

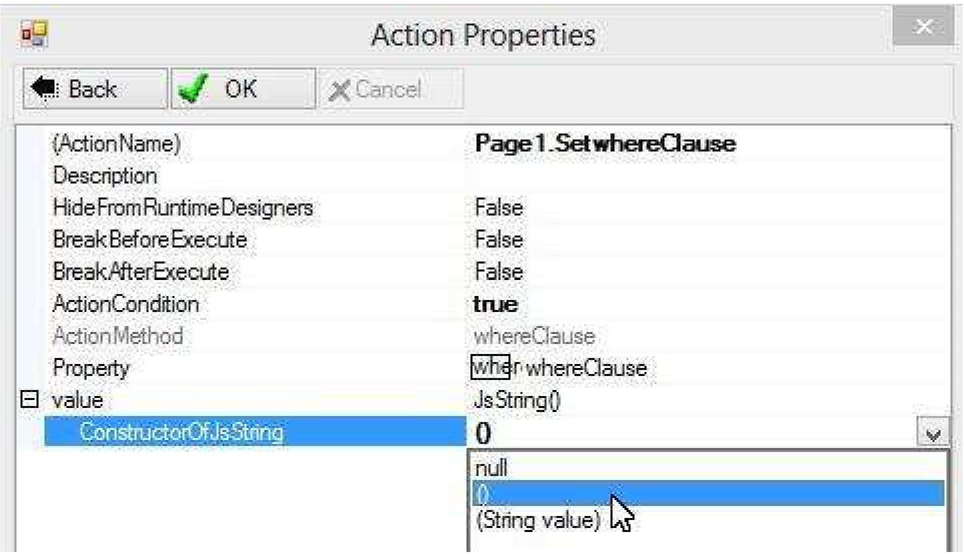


(Name) createWhereClause
Description

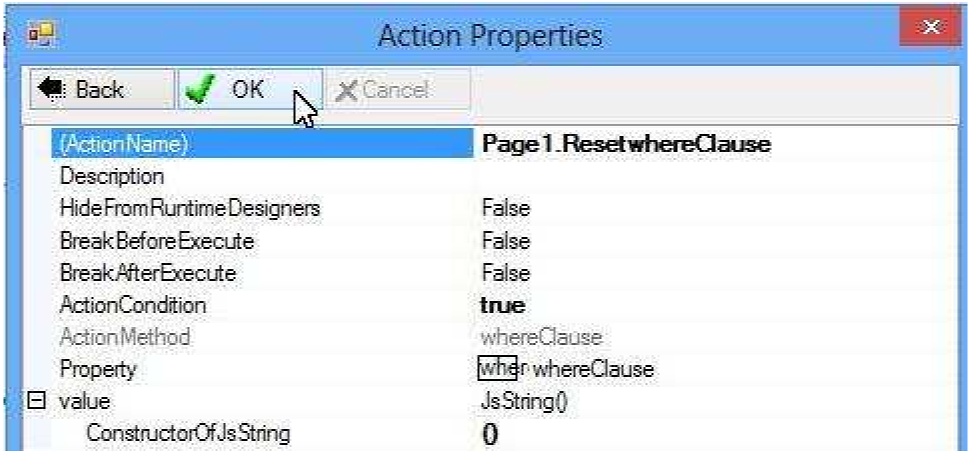
Reset whereClause

Set whereClause property to an empty string first:





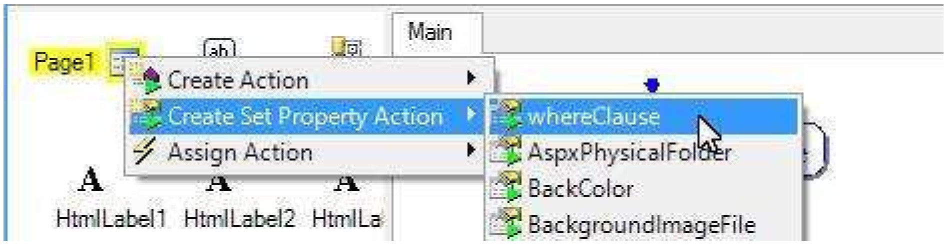
(ActionName)	Page1.SetwhereClause
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	whereClause
Property	wher whereClause
value	JsString()
ConstructorOfJsString	()



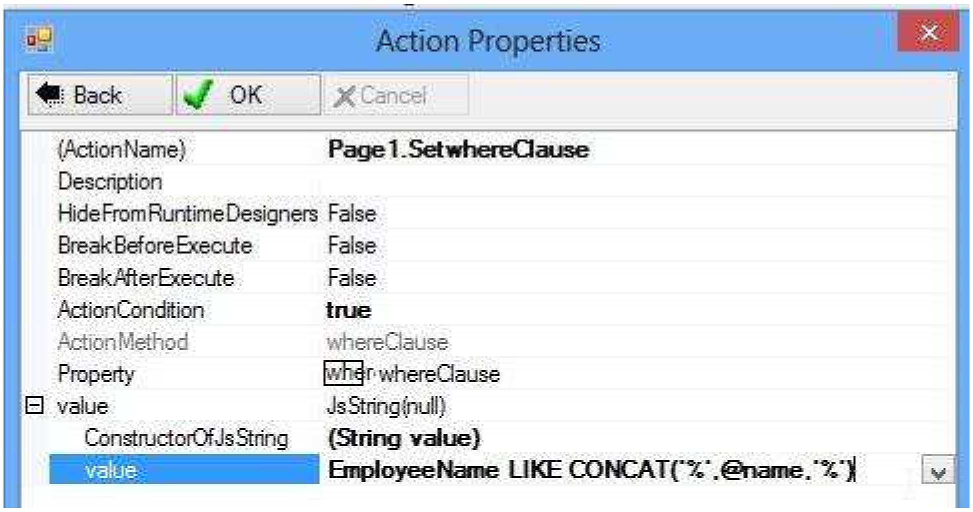
(ActionName)	Page1.ResetwhereClause
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	whereClause
Property	wher whereClause
value	JsString()
ConstructorOfJsString	()

Add name search filter

Set whereClause to EmployeeName LIKE CONCAT('%',@name,'%') if the text box is not empty:

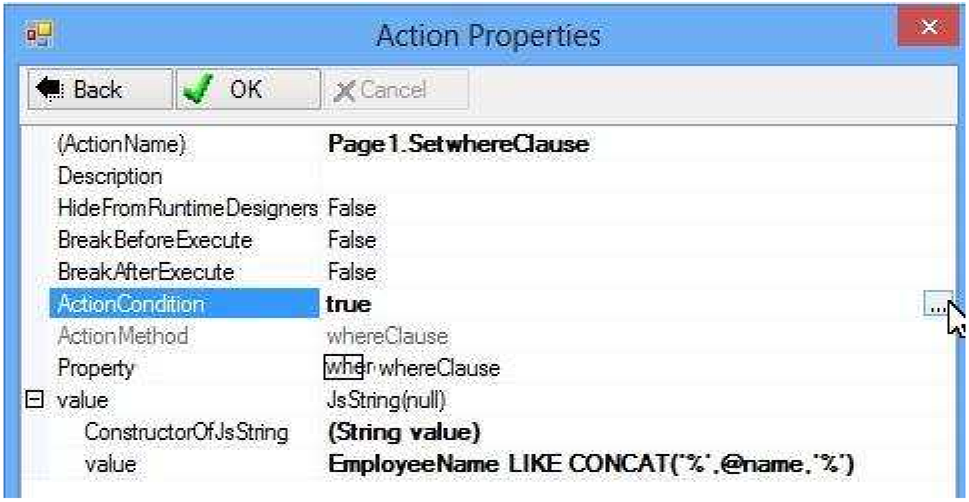


Set value to EmployeeName LIKE CONCAT('%',@name,'%'):

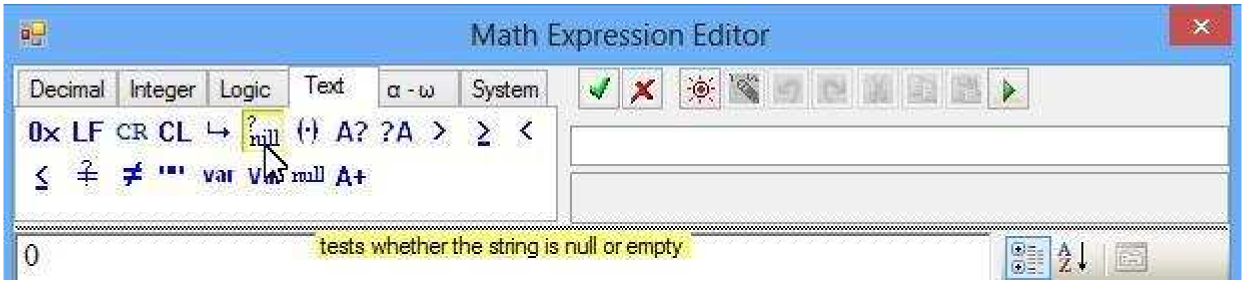


(ActionName)	**Page1.SetwhereClause**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	whereClause
Property	wher whereClause
⊟ value	JsString(null)
ConstructorOfJsString	**(String value)**
value	**EmployeeName LIKE CONCAT('%',@name,'%')**

Set ActionCondition to that the text box is not empty:

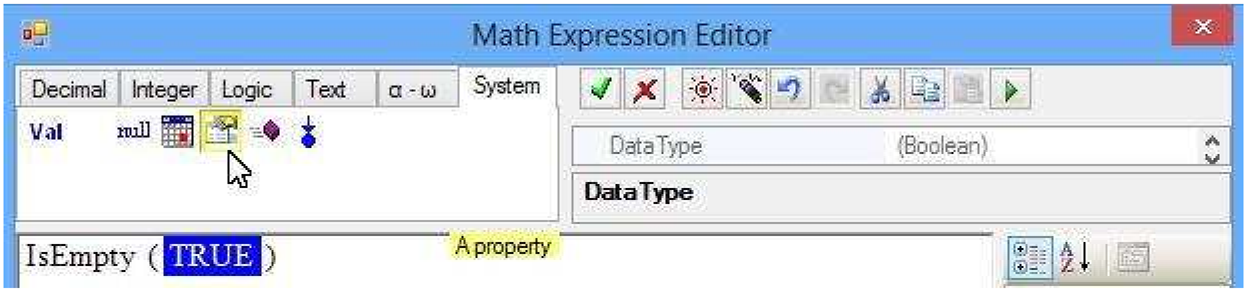


(ActionName)	**Page1.SetwhereClause**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	whereClause
Property	wher whereClause
⊟ value	JsString(null)
ConstructorOfJsString	**(String value)**
value	**EmployeeName LIKE CONCAT('%',@name,'%')**



0

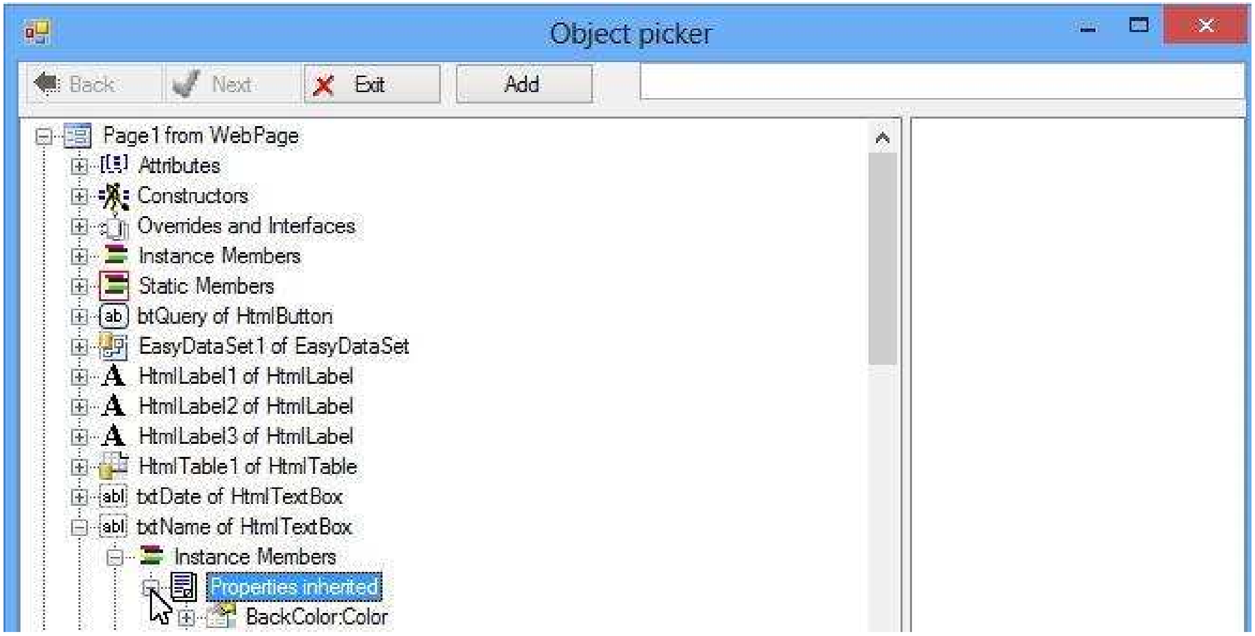
tests whether the string is null or empty



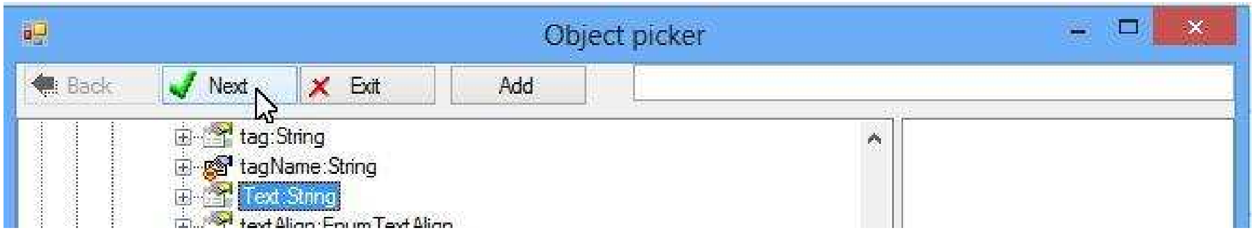
Data Type (Boolean)

Data Type

IsEmpty (TRUE)

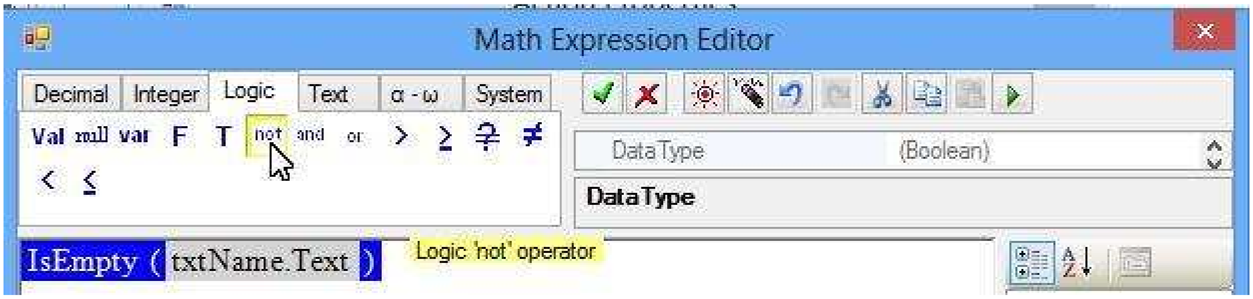


```
Page1 from WebPage
    Attributes
    Constructors
    Overrides and Interfaces
    Instance Members
    Static Members
    btQuery of HtmlButton
    EasyDataSet1 of EasyDataSet
    HtmlLabel1 of HtmlLabel
    HtmlLabel2 of HtmlLabel
    HtmlLabel3 of HtmlLabel
    HtmlTable1 of HtmlTable
    txtDate of HtmlTextBox
    txtName of HtmlTextBox
        Instance Members
            Properties inherited
                BackColor:Color
```



```
tag:String
tagName:String
Text:String
textAlign:EnumTextAlign
```

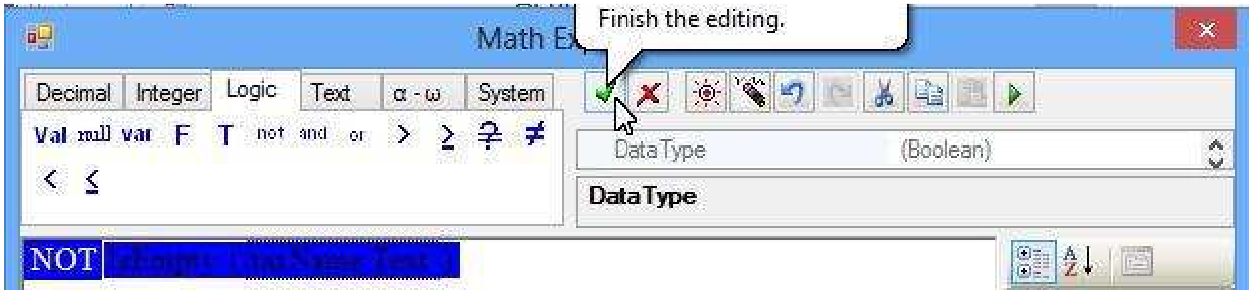
Apply logic “NOT”:



Data Type (Boolean)

Data Type

IsEmpty (txtName.Text)

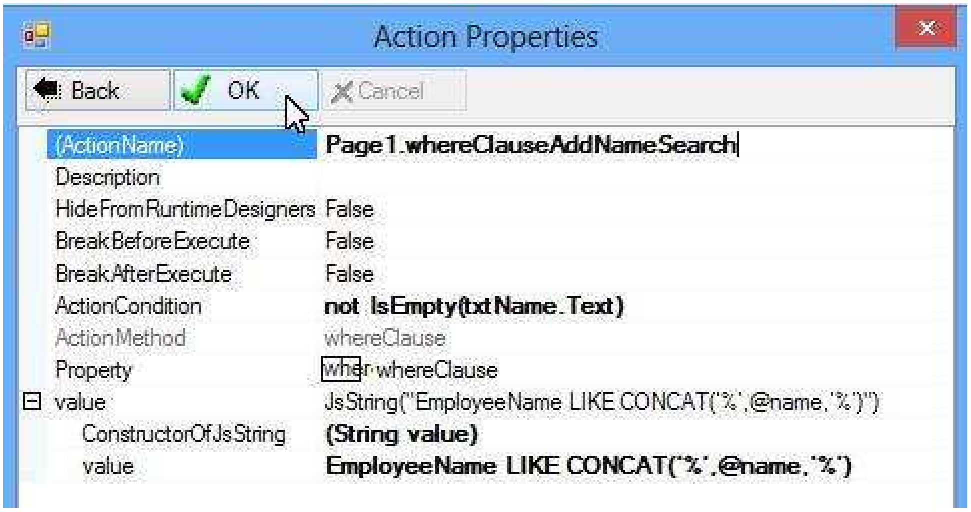


DataType	(Boolean)

DataType

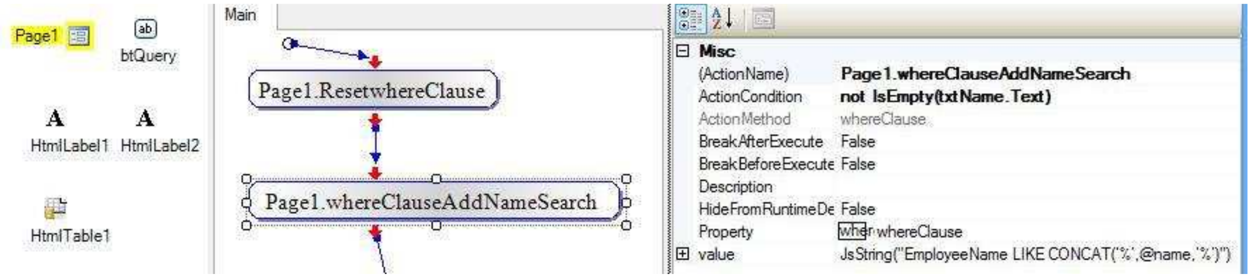
NOT IsEmpty(txtName.Text)

Rename the action and click OK:



(ActionName)	Page1.whereClauseAddNameSearch
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	not IsEmpty(txtName.Text)
ActionMethod	whereClause
Property	wher whereClause
⊟ value	JsString("EmployeeName LIKE CONCAT('%',@name,'%')")
ConstructorOfJsString	(String value)
value	EmployeeName LIKE CONCAT('%',@name,'%')

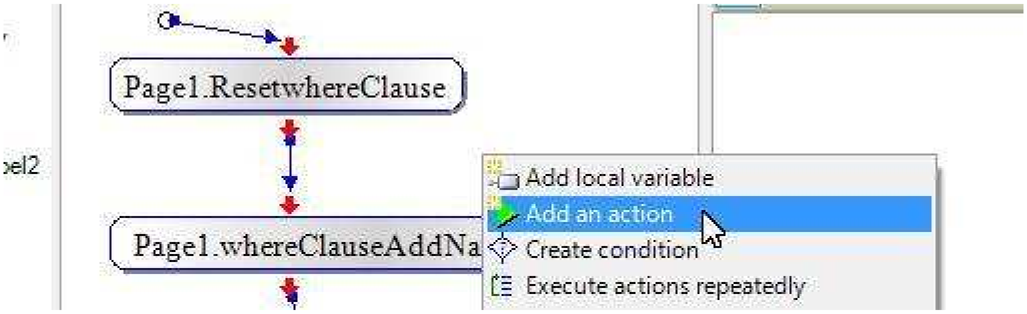
Link the actions:

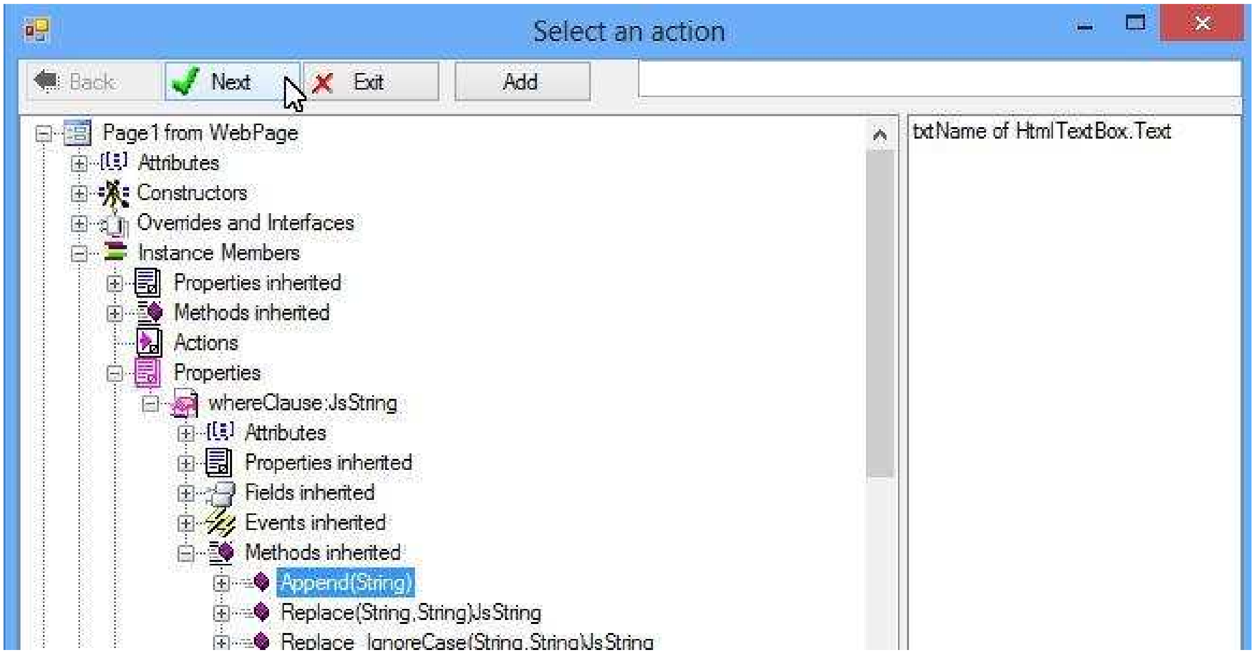


⊟ Misc	
(ActionName)	Page1.whereClauseAddNameSearch
ActionCondition	not IsEmpty(txtName.Text)
ActionMethod	whereClause
BreakAfterExecute	False
BreakBeforeExecute	False
Description	
HideFromRuntimeDe	False
Property	wher whereClause
⊞ value	JsString("EmployeeName LIKE CONCAT('%',@name,'%')")

Add notes search filter

Append Notes LIKE CONCAT('%',@notes,'%') to whereClause if the text box is not empty.

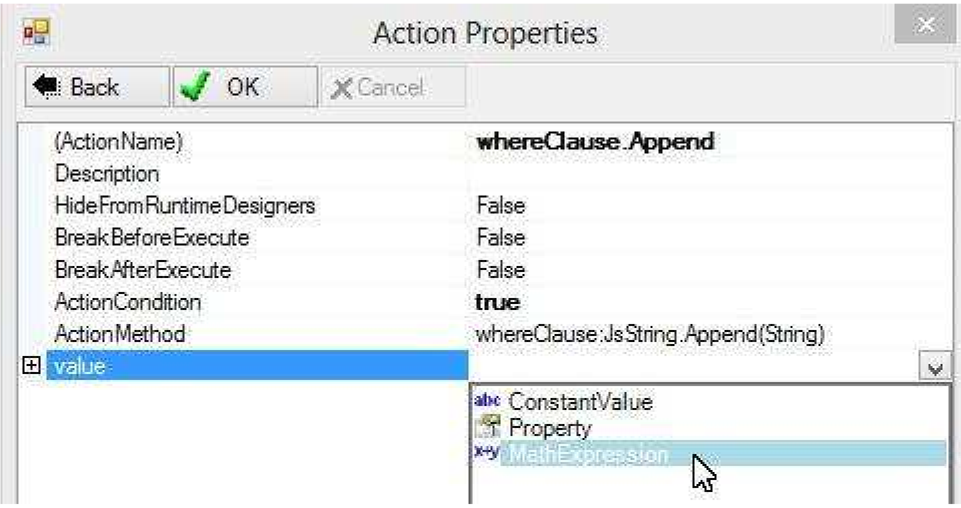




- Page1 from WebPage
  - Attributes
  - Constructors
  - Overrides and Interfaces
  - Instance Members
    - Properties inherited
    - Methods inherited
    - Actions
    - Properties
      - whereClause:JsString
        - Attributes
        - Properties inherited
        - Fields inherited
        - Events inherited
        - Methods inherited
          - Append(String)
          - Replace(String,String)JsString
          - Replace_IgnoreCase(String,String)JsString

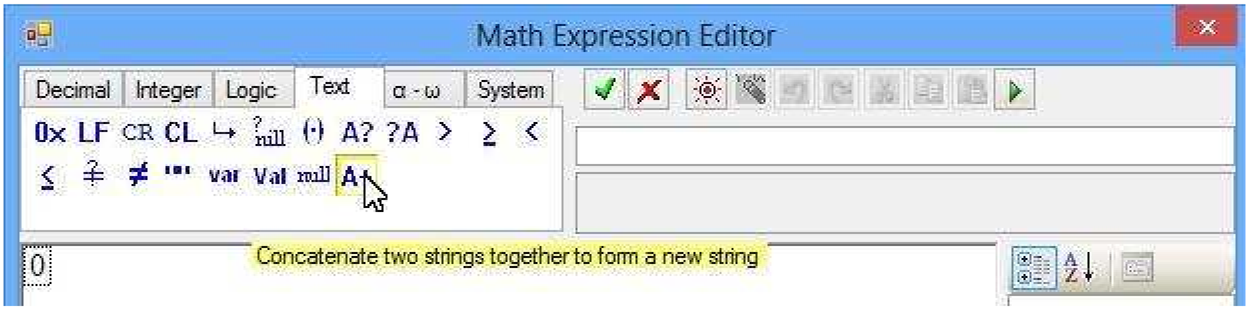
txtName of HtmlTextBox.Text

Use expression for “value” because we need to form the filter:



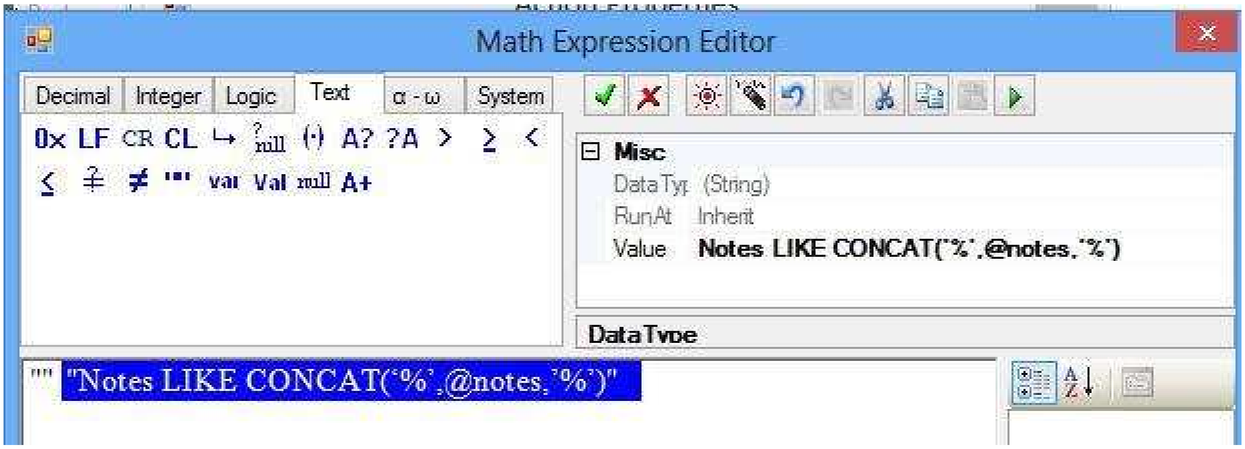
(ActionName)	**whereClause.Append**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	whereClause:JsString.Append(String)
value	

If whereClause is not empty then we need add “ AND “ before Notes LIKE CONCAT('%',@notes,'%'); if whereClause is empty then we do not add “ AND “. We use two parts for this expression:



Concatenate two strings together to form a new string

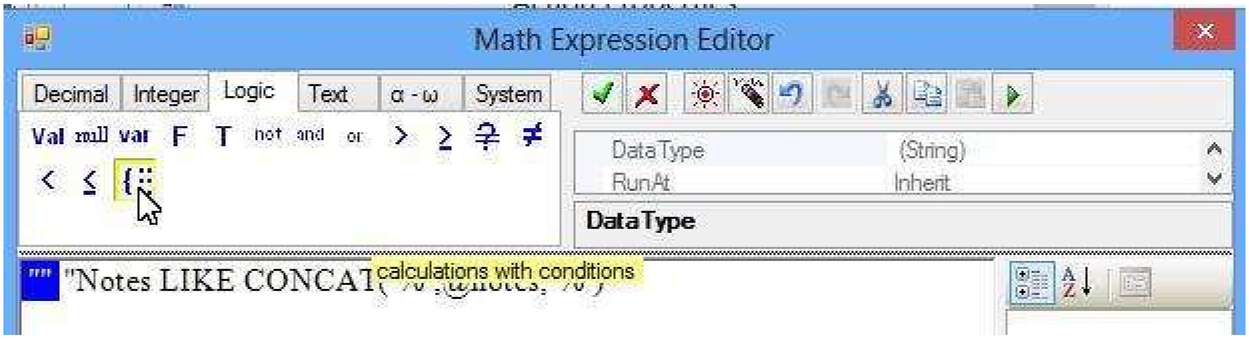
The second part is Notes LIKE CONCAT('%',@notes,'%'):



Data Typ (String)
RunAt Inherit
Value Notes LIKE CONCAT('%',@notes,'%')

"" "Notes LIKE CONCAT('%',@notes,'%')"

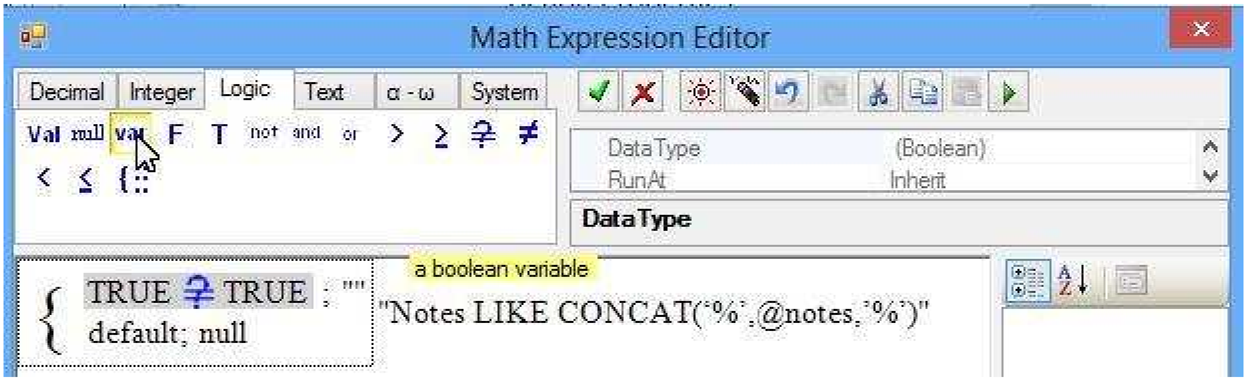
We use a switch for the first part:



Data Type (String)
RunAt Inherit

"" "Notes LIKE CONCAT('%',@notes,'%')"

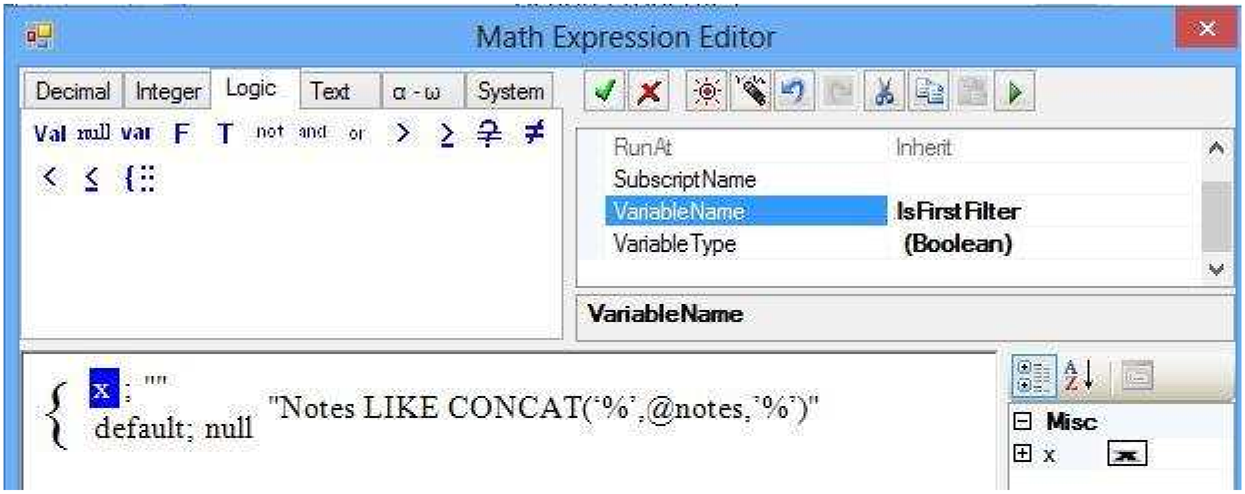
To check for text box emptiness, use a Boolean variable:



Data Type (Boolean)
RunAt Inherit

{ TRUE ≠ TRUE ; ""
default; null
"Notes LIKE CONCAT('%',@notes,'%')"

Rename the variable to make the expression more readable:

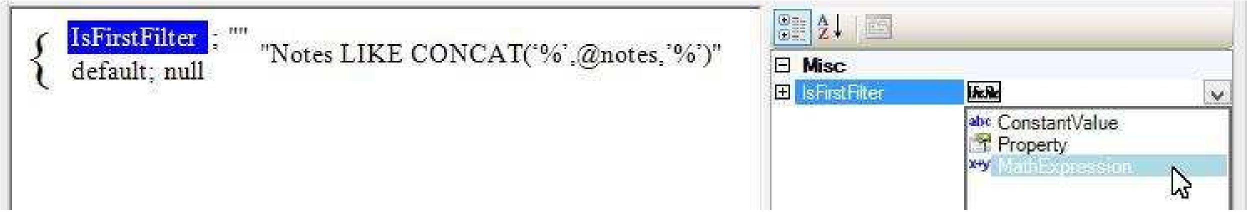


RunAt	Inherit
SubscriptName	
VariableName	IsFirstFilter
VariableType	(Boolean)

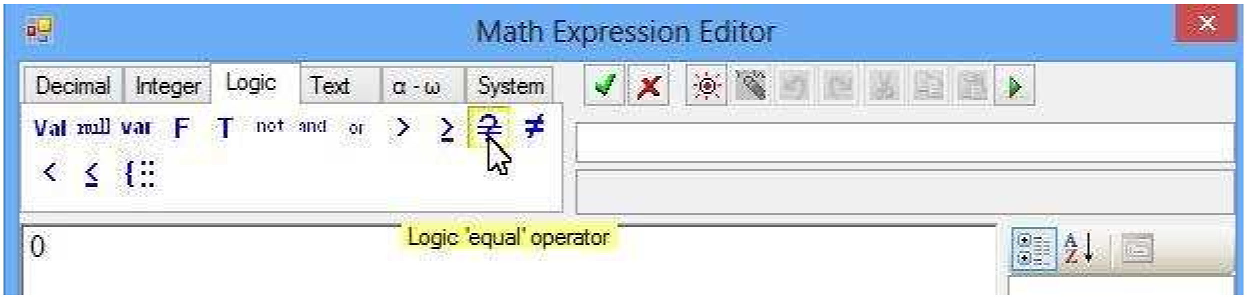
VariableName

{ x : ""
default; null "Notes LIKE CONCAT('%',@notes,'%')"

Use the variable to check the length of whereClause:

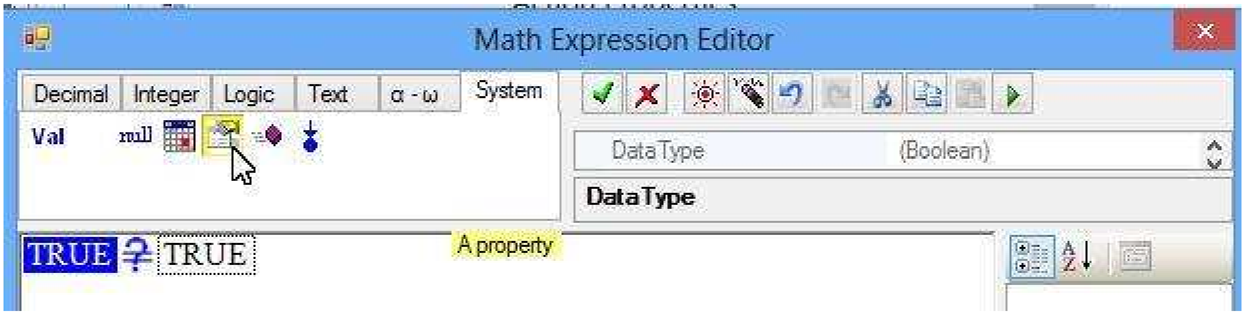


{ IsFirstFilter : ""
default; null "Notes LIKE CONCAT('%',@notes,'%')"



0

Logic 'equal' operator

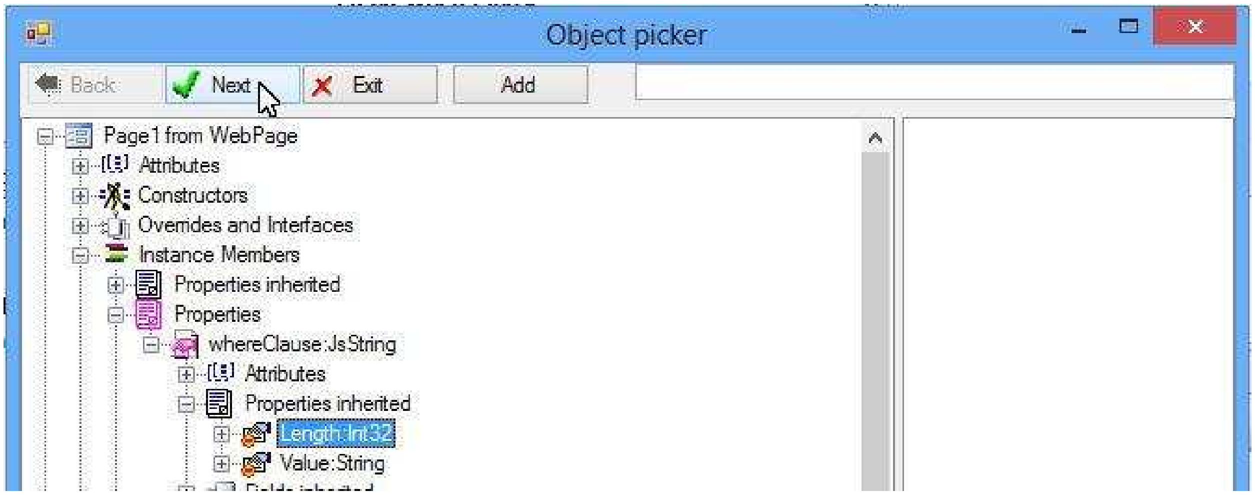


DataType	(Boolean)

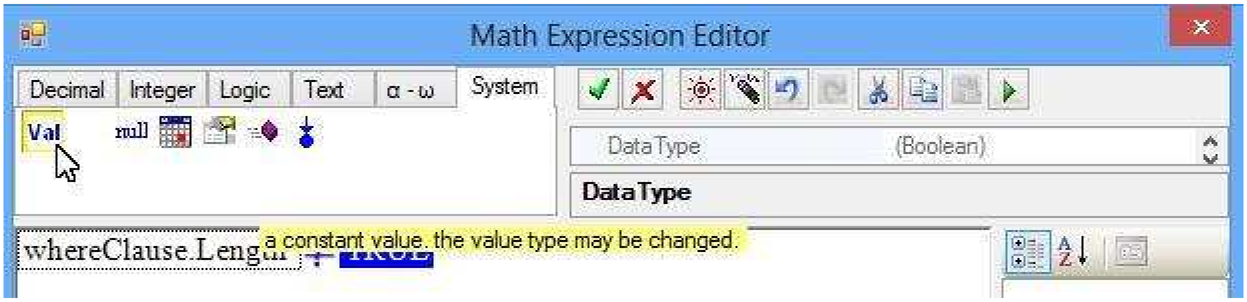
DataType

TRUE = TRUE

A property



- Page1 from WebPage
  - Attributes
  - Constructors
  - Overrides and Interfaces
  - Instance Members
    - Properties inherited
    - Properties
      - whereClause:JsString
        - Attributes
        - Properties inherited
          - Length:Int32
          - Value:String

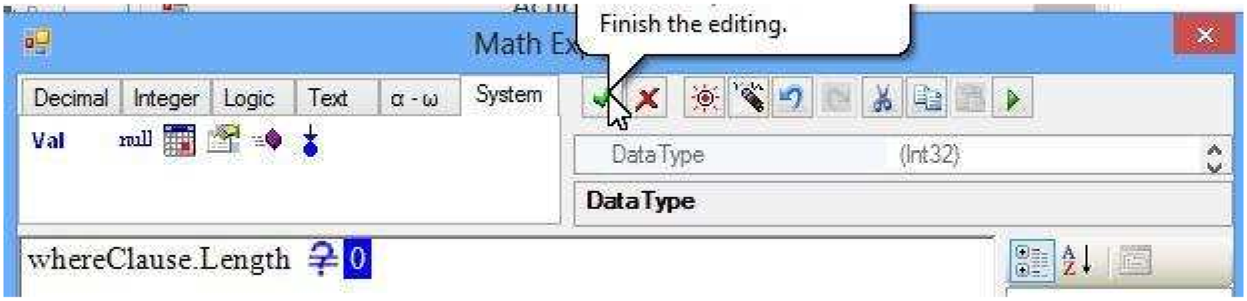


Data Type (Boolean)

Data Type

whereClause.Length ≠ TRUE

a constant value. the value type may be changed.

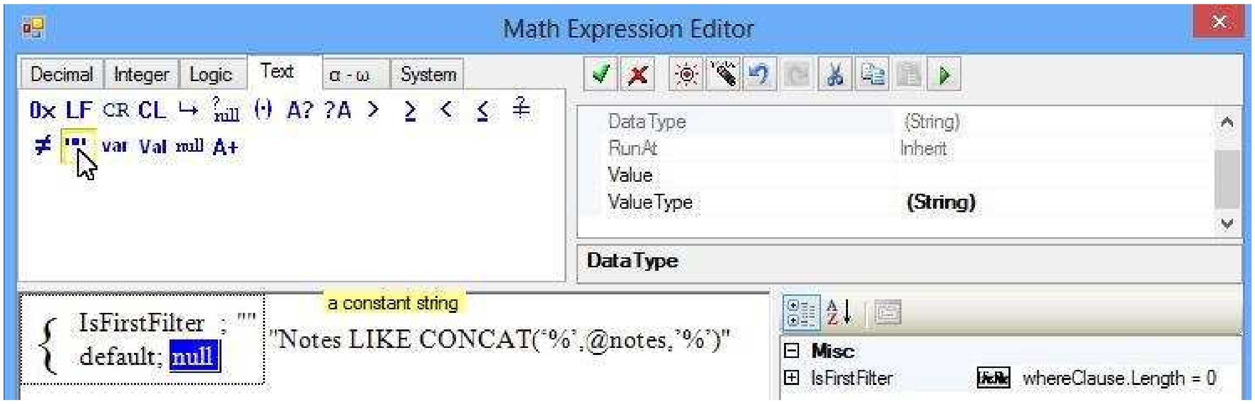


Data Type (Int32)

Data Type

whereClause.Length ≠ 0

Set “ AND “ for default value:



Data Type	(String)
RunAt	Inherit
Value	
ValueType	**(String)**

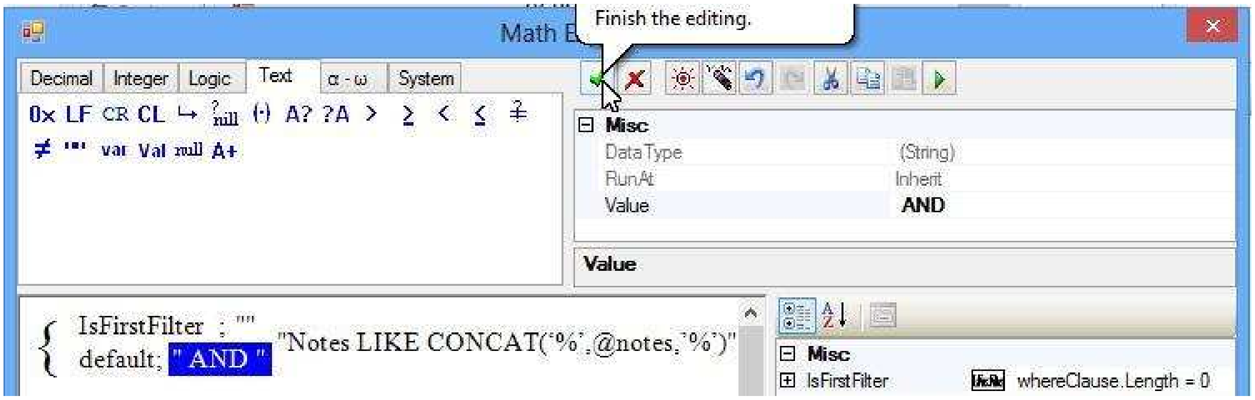
Data Type

a constant string

{ IsFirstFilter ; ""
default; null } "Notes LIKE CONCAT('%',@notes,'%')"

Misc

IsFirstFilter whereClause.Length = 0


Math E
Finish the editing.
Decimal Integer Logic Text α - ω System
0x LF CR CL ↳ ?null (·) A? ?A > ≥ < ≤ ≠ ≠ "" var Val null A+

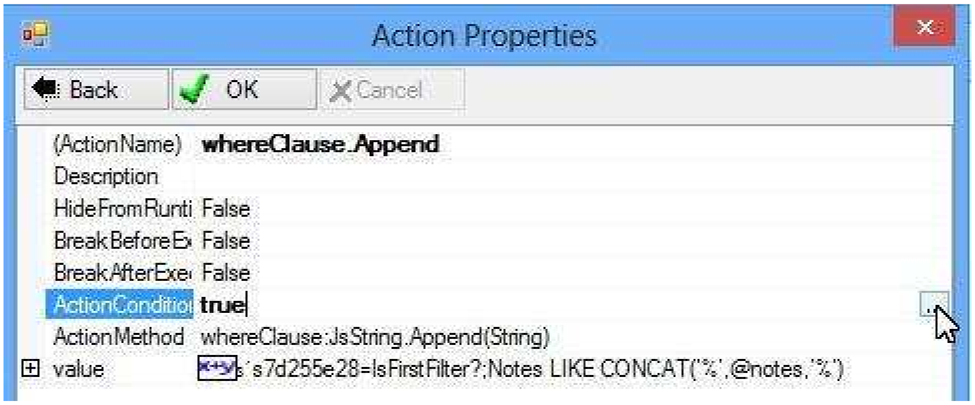
Misc
DataType (String)
RunAt Inherit
Value AND

Value

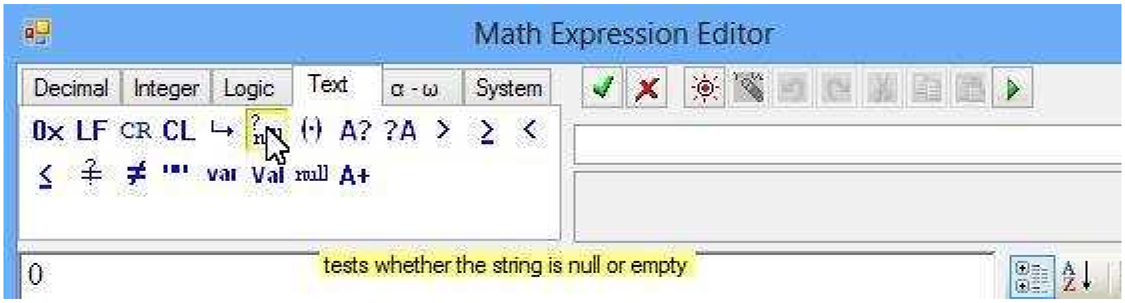
IsFirstFilter ; ""
default; " AND " "Notes LIKE CONCAT('%',@notes,'%')"

Misc
IsFirstFilter whereClause.Length = 0

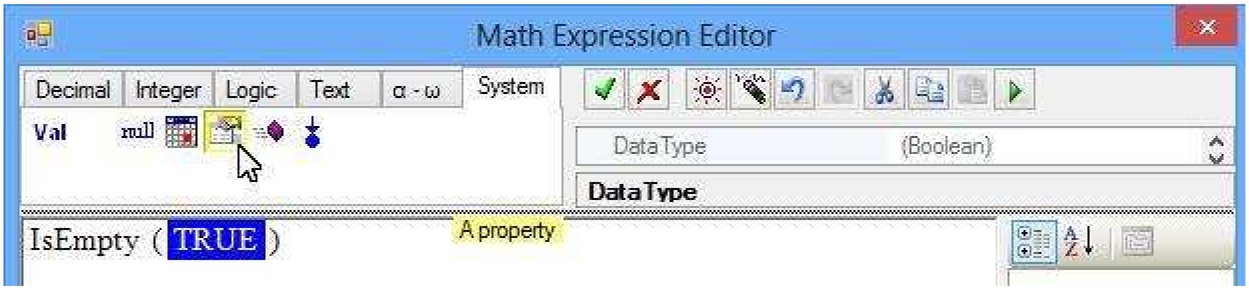
Set ActionCondition to check if the text box is empty:



(ActionName) whereClause.Append
Description
HideFromRunti False
BreakBeforeEx False
BreakAfterExe False
ActionConditio true
ActionMethod whereClause:JsString.Append(String)
value s7d255e28=IsFirstFilter?;Notes LIKE CONCAT('%',@notes,'%')


Math Expression Editor
Decimal Integer Logic Text α - ω System
0x LF CR CL ↳ ?null (·) A? ?A > ≥ < ≤ ≠ ≠ "" var Val null A+

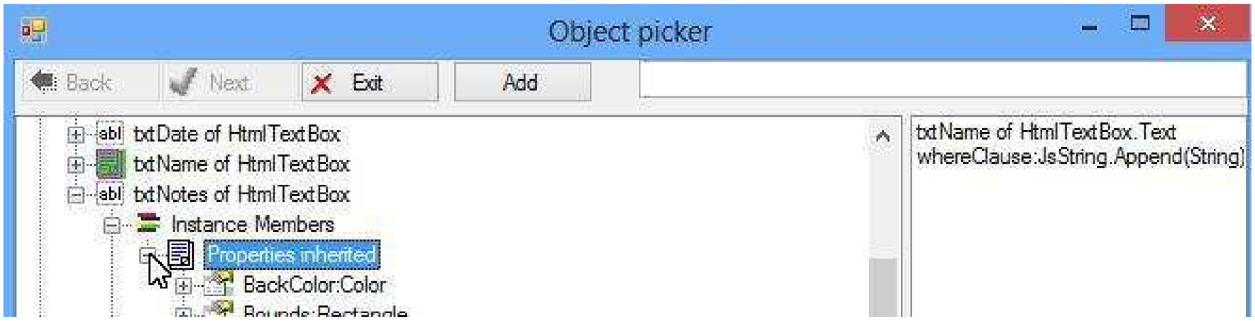
0 tests whether the string is null or empty



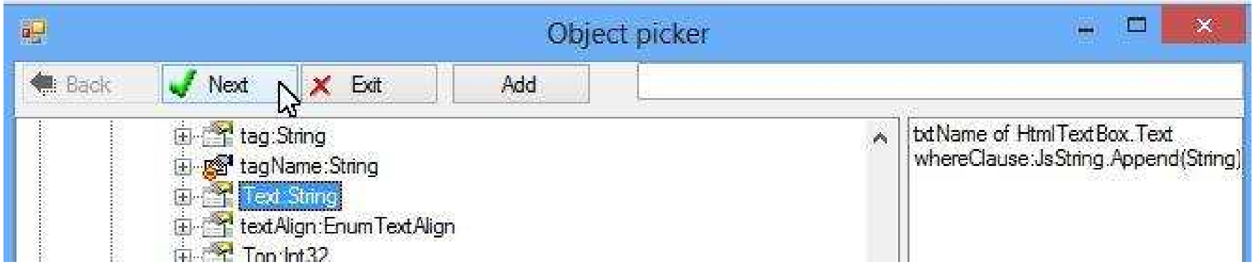
DataType (Boolean)

DataType

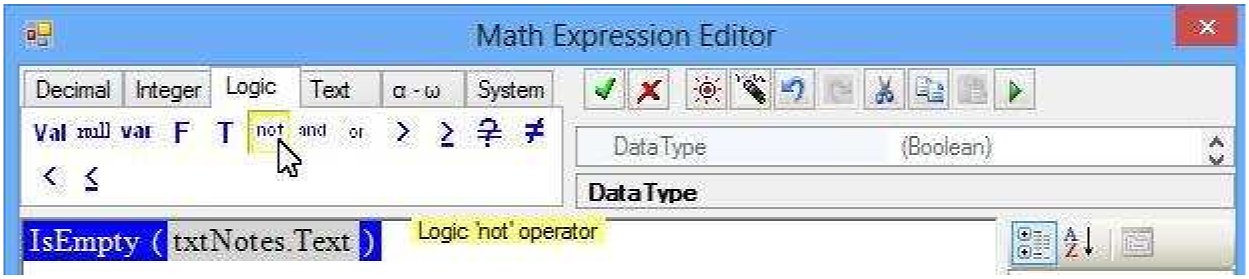
IsEmpty (TRUE) A property



txtName of HtmlTextBox.Text
whereClause:JsString.Append(String)



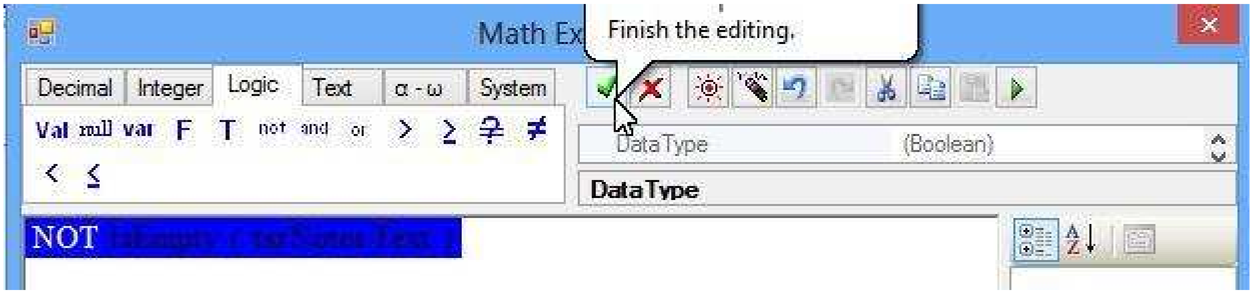
txtName of HtmlTextBox.Text
whereClause:JsString.Append(String)



Data Type (Boolean)

Data Type

IsEmpty (txtNotes.Text)

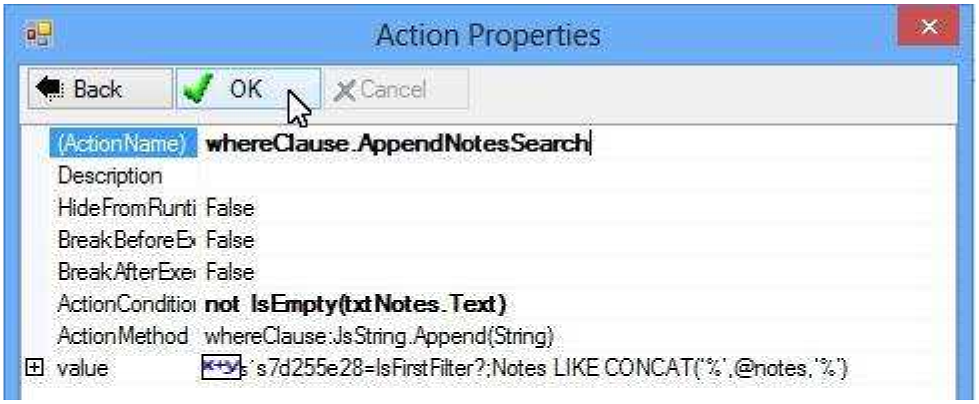


Data Type (Boolean)

Data Type

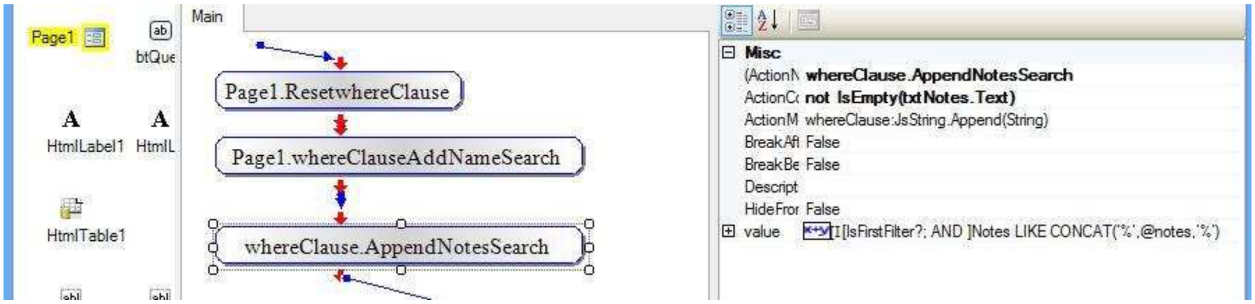
NOT IsEmpty (txtNotes.Text)

Rename the action and click OK:



(ActionName) whereClause.AppendNotesSearch
Description
HideFromRunti False
BreakBeforeEx False
BreakAfterExe False
ActionConditio not IsEmpty(txtNotes.Text)
ActionMethod whereClause:JsString.Append(String)
value s's7d255e28=IsFirstFilter?;Notes LIKE CONCAT('%',@notes,'%')

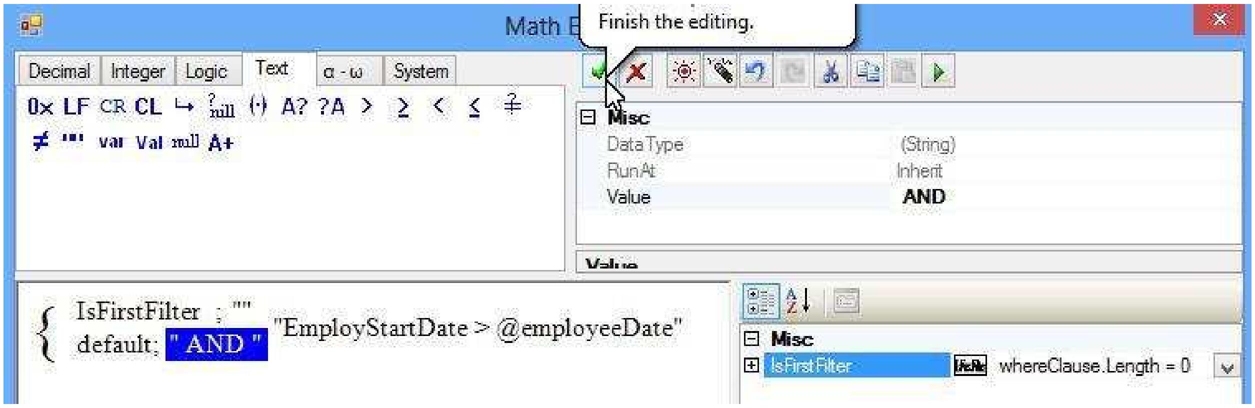
Link the actions:



(ActionN whereClause.AppendNotesSearch
ActionC not IsEmpty(txtNotes.Text)
ActionM whereClause:JsString.Append(String)
BreakAft False
BreakBe False
Descript
HideFror False
value I[IsFirstFilter?; AND]Notes LIKE CONCAT('%',@notes,'%')

Add date search filter

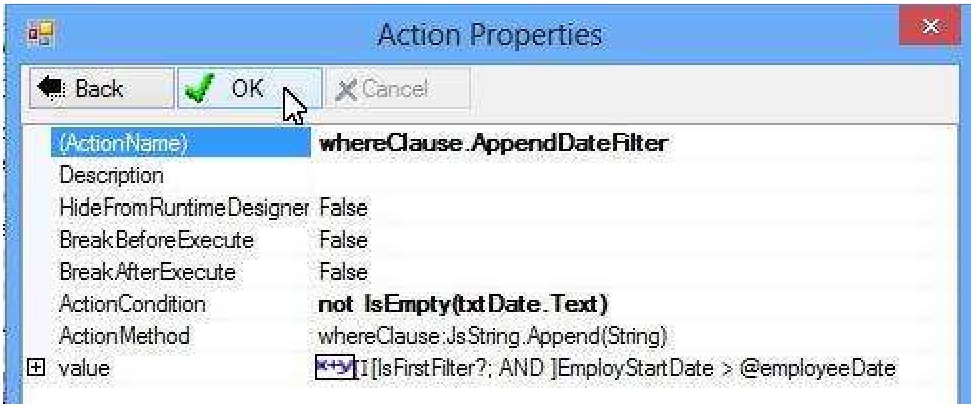
Append EmployStartDate > @employeeDate to whereClause if the text box is not empty. The programming is very much like we just did for adding notes search filter. So, we do not go step by step.



DataType (String)
RunAt Inherit
Value AND

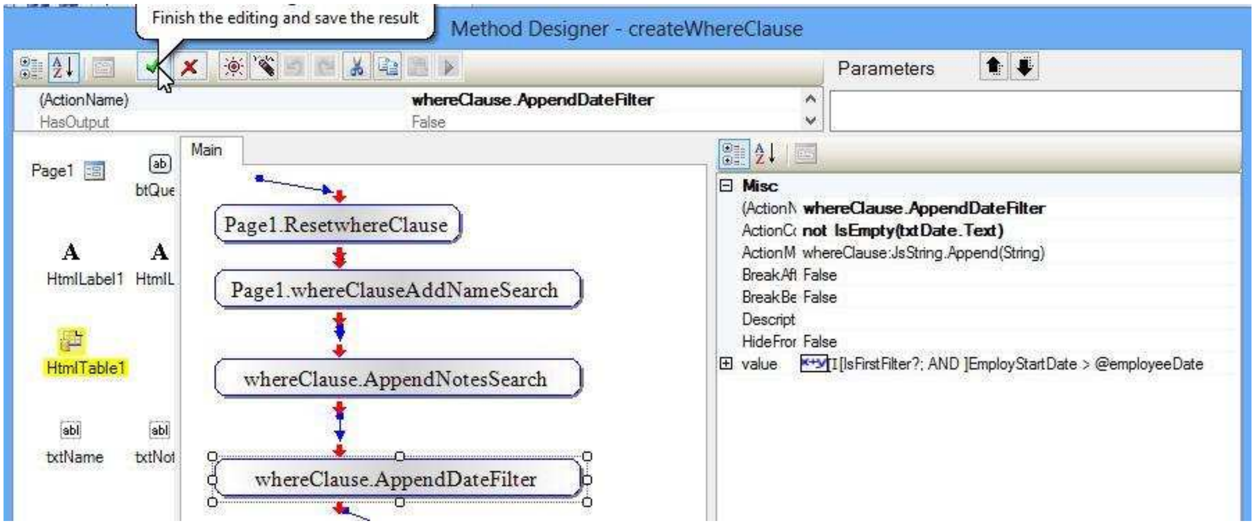
{ IsFirstFilter ; ""
default; " AND " "EmployStartDate > @employeeDate"

IsFirstFilter whereClause.Length = 0



(ActionName) whereClause.AppendDateFilter
Description
HideFromRuntimeDesigner False
BreakBeforeExecute False
BreakAfterExecute False
ActionCondition not IsEmpty(txtDate.Text)
ActionMethod whereClause:JsString.Append(String)
value I[IsFirstFilter?; AND]EmployStartDate > @employeeDate

That is all for this method:

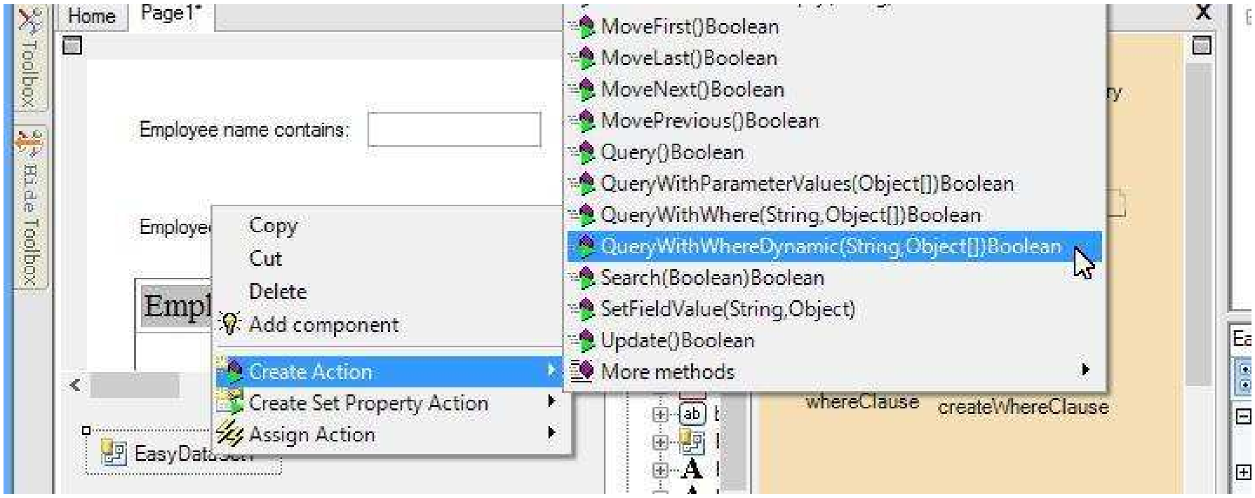

Finish the editing and save the result
Method Designer - createWhereClause
Parameters
Page1
btQue
HtmlLabel1
HtmlL
HtmlTable1
txtName
txtNot
Main
Page1.ResetwhereClause
Page1.whereClauseAddNameSearch
whereClause.AppendNotesSearch
whereClause.AppendDateFilter
Misc

(ActionName) whereClause.AppendDateFilter
HasOutput False

(ActionN whereClause.AppendDateFilter
ActionC not IsEmpty(txtDate.Text)
ActionM whereClause:JsString.Append(String)
BreakAft False
BreakBe False
Descript
HideFror False
value I[IsFirstFilter?; AND]EmployStartDate > @employeeDate

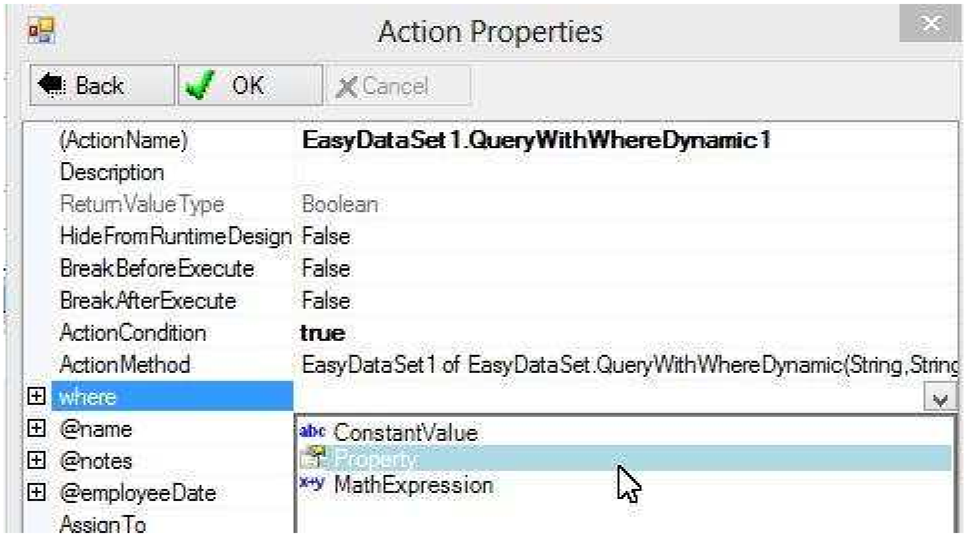
Create QueryWithWhereDynamic action

Right-click EasyDataSet1, choose “Create Action”, choose “QueryWithWhereDynamic”:

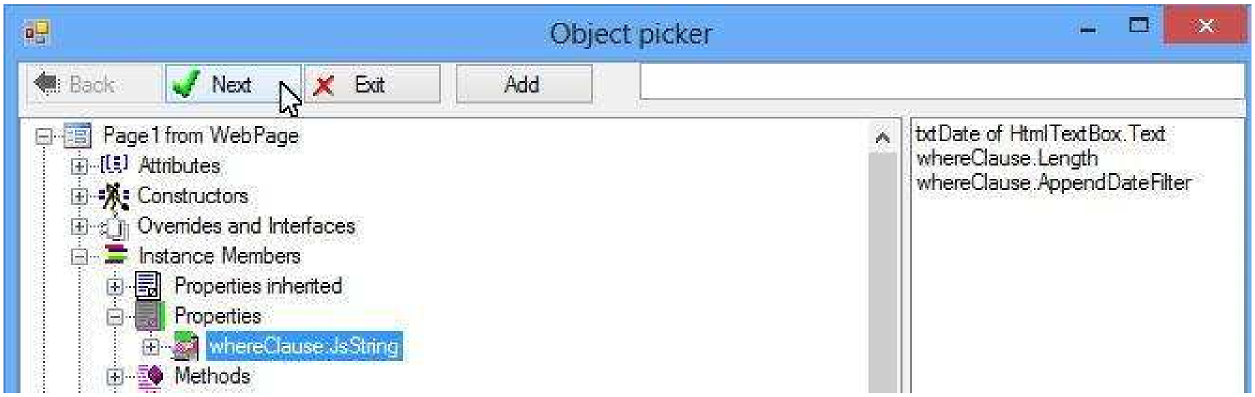


Use dynamic filter

Pass whereClause property to the action:



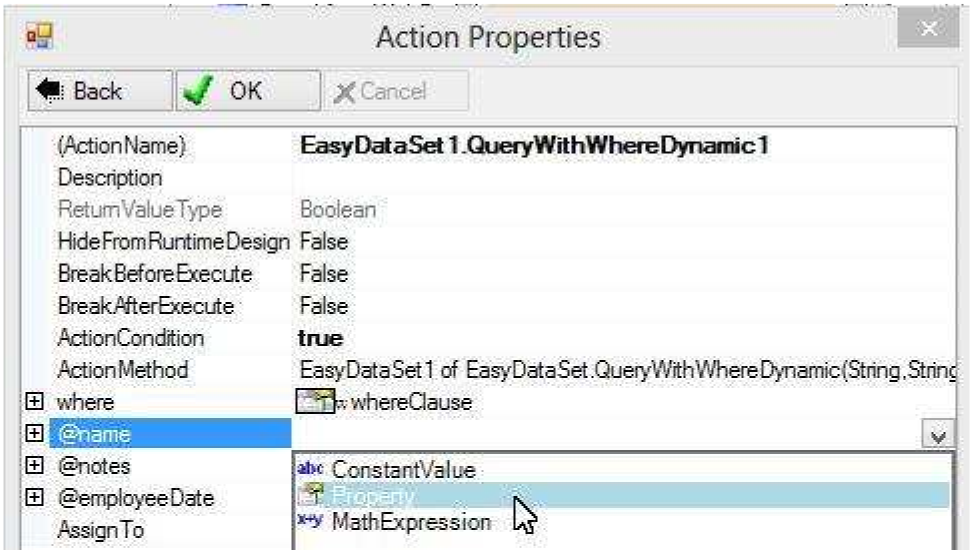
(ActionName)	EasyDataSet1.QueryWithWhereDynamic1
Description	
ReturnValueType	Boolean
HideFromRuntimeDesign	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	EasyDataSet1 of EasyDataSet.QueryWithWhereDynamic(String,String
⊞ where	
⊞ @name	
⊞ @notes	
⊞ @employeeDate	
AssignTo	



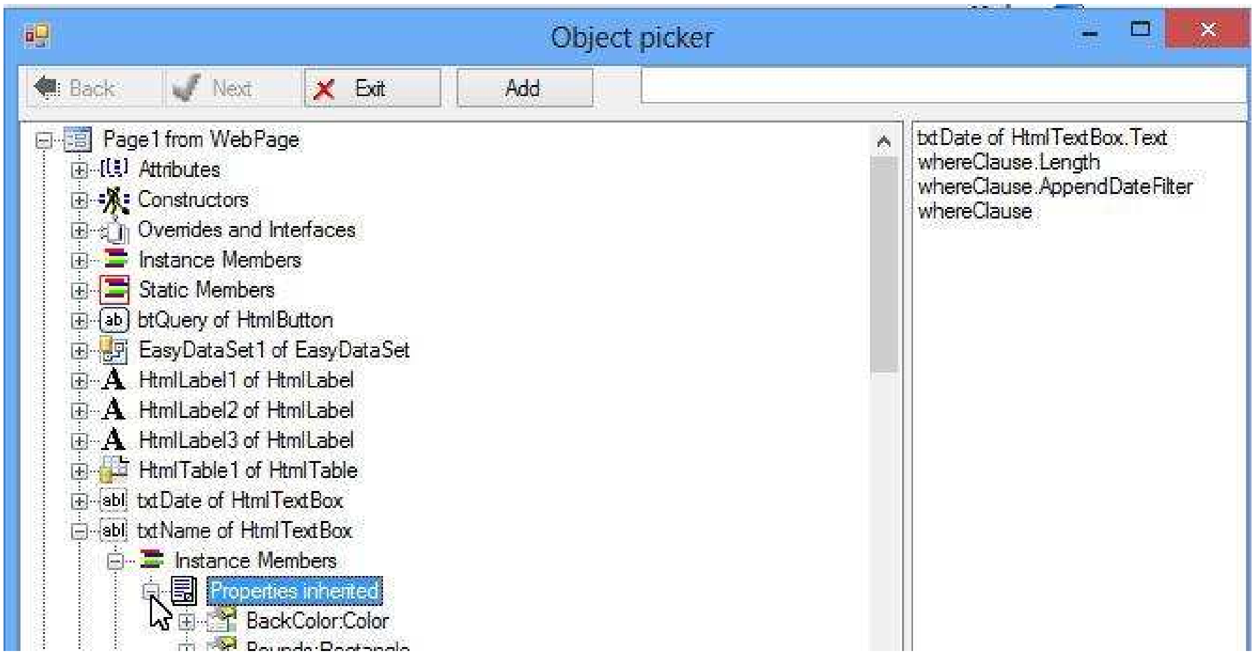
txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter

Use parameter values

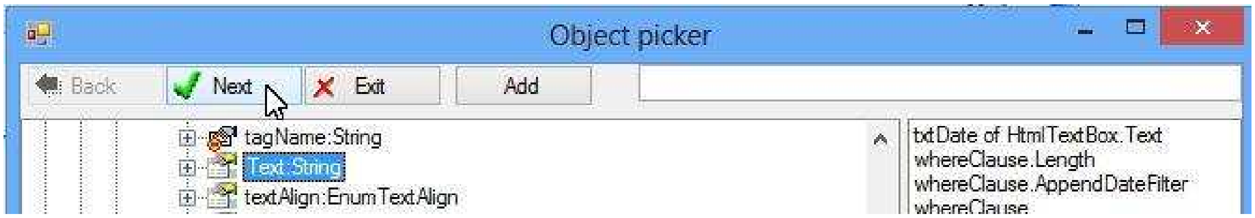
Pass text box values to the action:



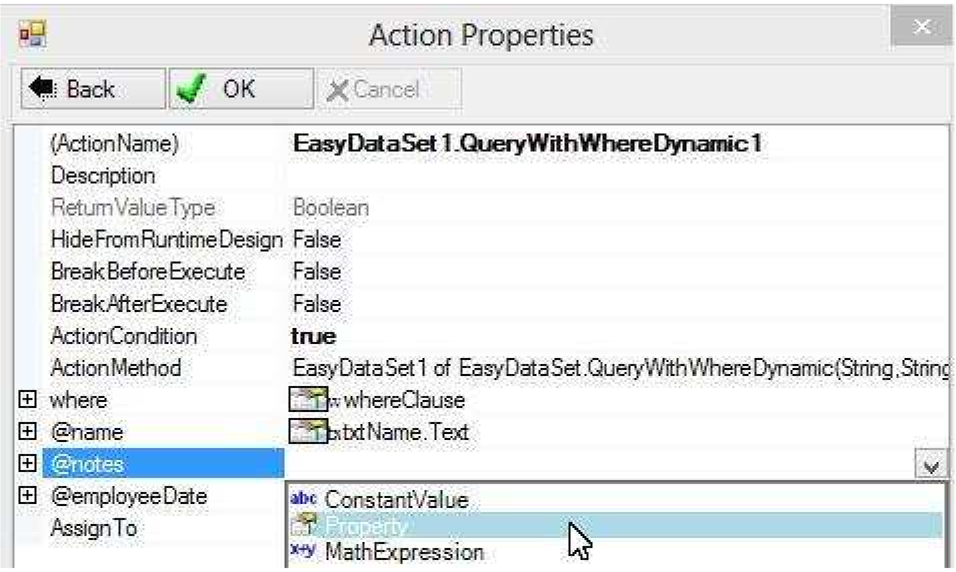
(ActionName)	EasyDataSet1.QueryWithWhereDynamic1
Description	
ReturnValueType	Boolean
HideFromRuntimeDesign	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	EasyDataSet1 of EasyDataSet.QueryWithWhereDynamic(String,String
where	whereClause
@name	
@notes	
@employeeDate	
AssignTo	



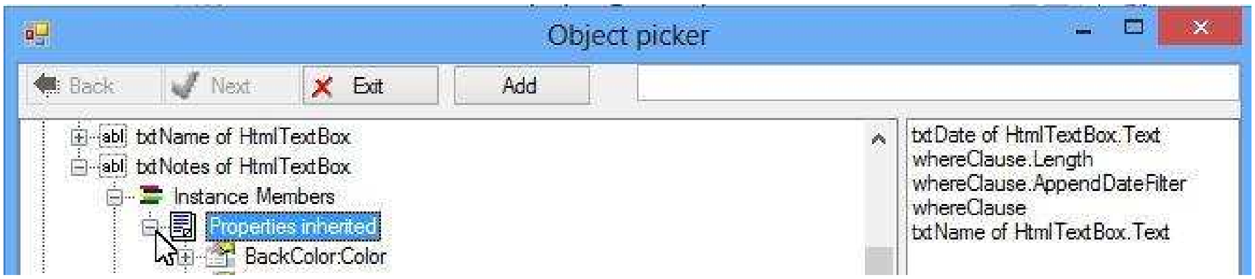
txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause



txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause

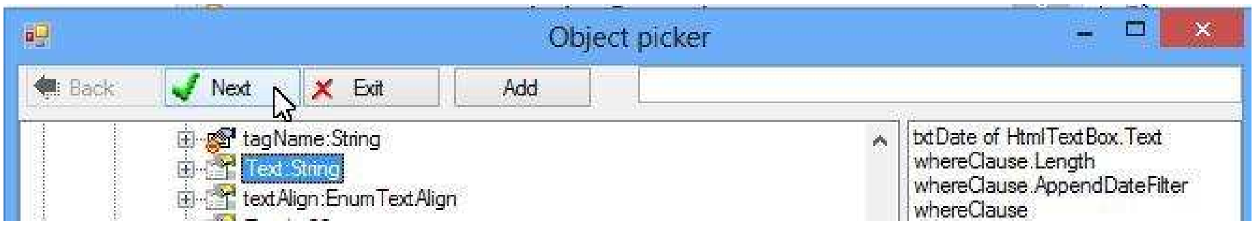


(ActionName)	EasyDataSet1.QueryWithWhereDynamic1
Description	
ReturnValueType	Boolean
HideFromRuntimeDesign	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	EasyDataSet1 of EasyDataSet.QueryWithWhereDynamic(String,String
⊞ where	whereClause
⊞ @name	txtName.Text
⊞ @notes	
⊞ @employeeDate	
AssignTo	



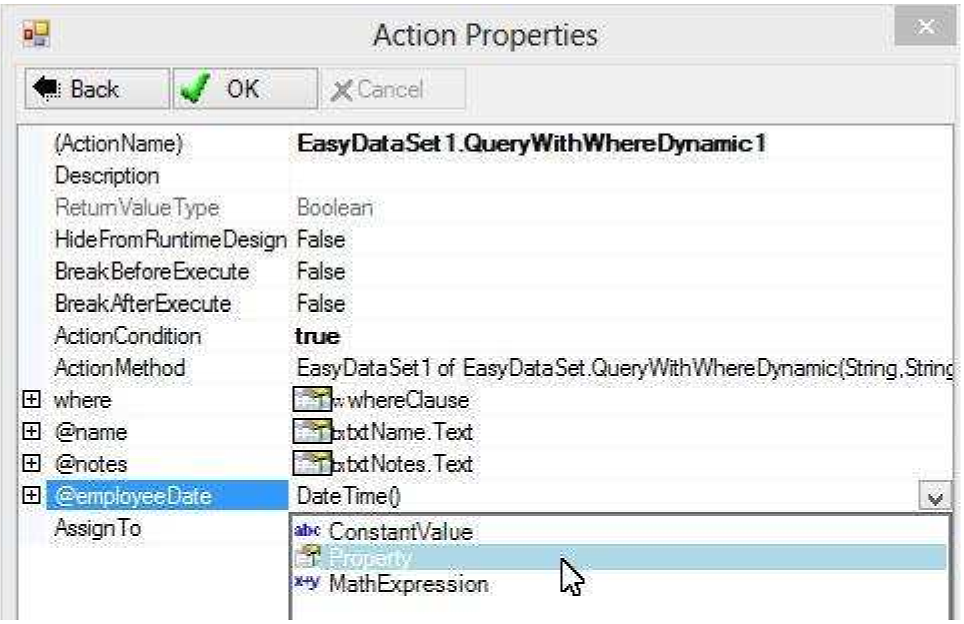
```
⊞ abl txtName of HtmlTextBox
⊟ abl txtNotes of HtmlTextBox
   ⊟ Instance Members
      ⊟ Properties inherited
         ⊞ BackColor:Color
```

txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause
txtName of HtmlTextBox.Text

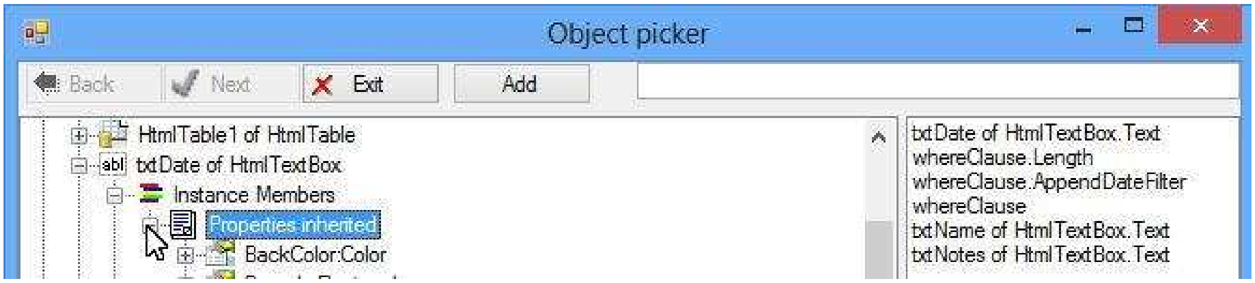


```
⊞ tagName:String
⊞ Text:String
⊞ textAlign:EnumTextAlign
```

txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause

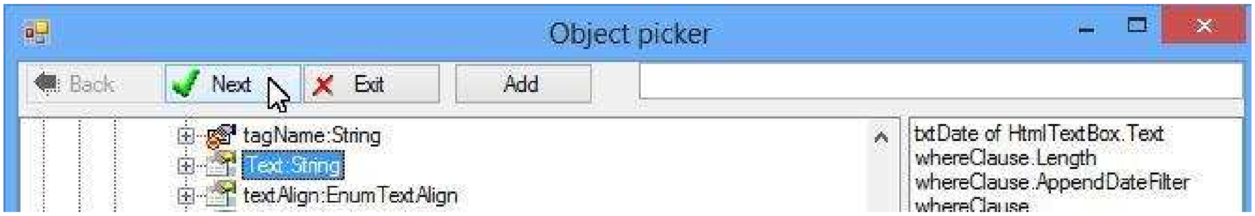


(ActionName)	**EasyDataSet1.QueryWithWhereDynamic1**
Description	
ReturnValueType	Boolean
HideFromRuntimeDesign	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	EasyDataSet1 of EasyDataSet.QueryWithWhereDynamic(String,String
⊞ where	whereClause
⊞ @name	txtName.Text
⊞ @notes	txtNotes.Text
⊞ @employeeDate	DateTime()
AssignTo	



HtmlTable1 of HtmlTable
txtDate of HtmlTextBox
Instance Members
Properties inherited
BackColor:Color

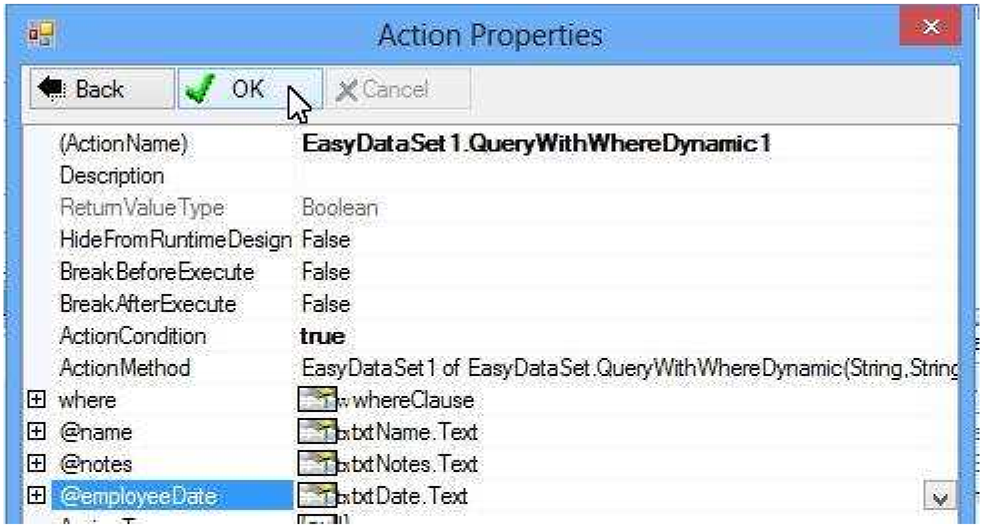
txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause
txtName of HtmlTextBox.Text
txtNotes of HtmlTextBox.Text



tagName:String
Text:String
textAlign:EnumTextAlign

txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause

Click OK, the action is created:

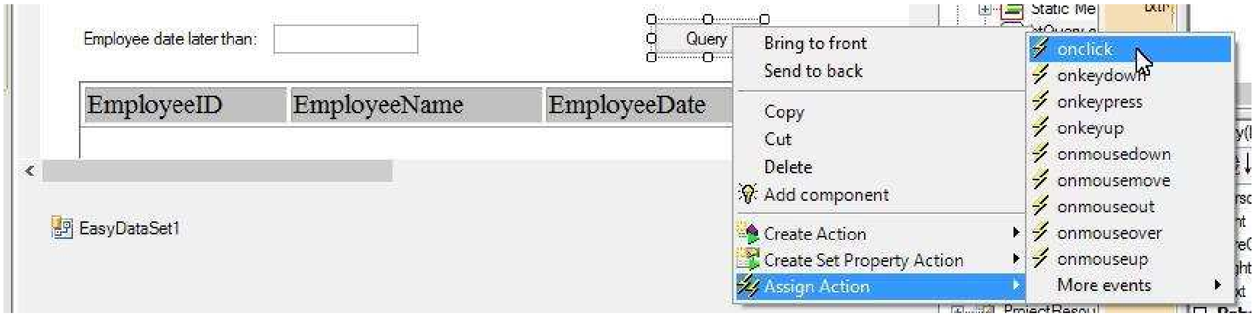


(ActionName)	EasyDataSet1.QueryWithWhereDynamic1
Description	
ReturnValueType	Boolean
HideFromRuntimeDesign	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	true
ActionMethod	EasyDataSet1 of EasyDataSet.QueryWithWhereDynamic(String,String
⊞ where	whereClause
⊞ @name	txtName.Text
⊞ @notes	txtNotes.Text
⊞ @employeeDate	txtDate.Text

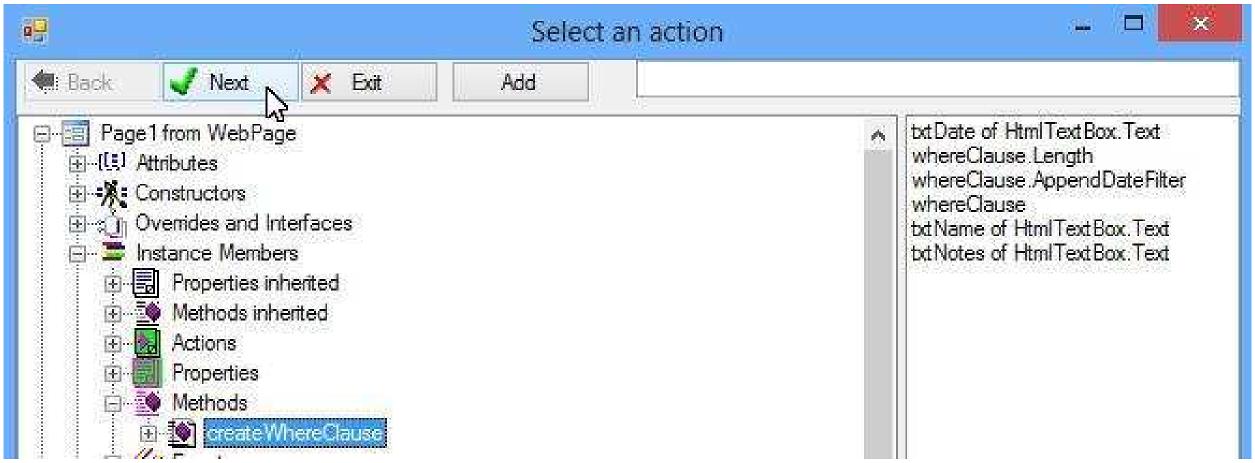
Execute dynamic query

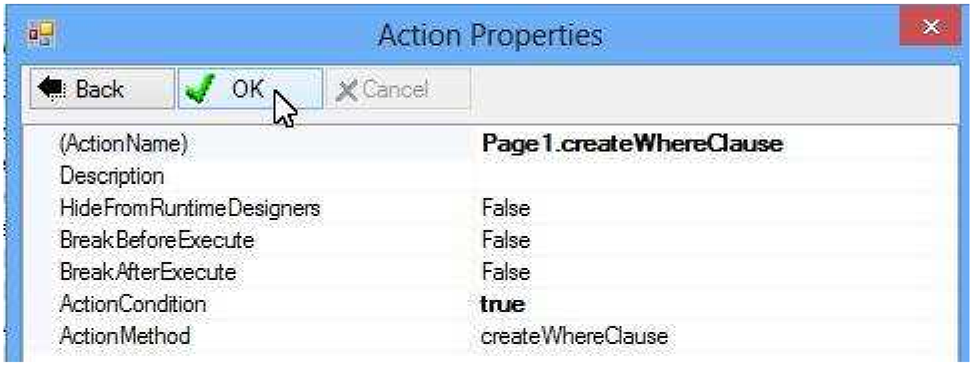
Generate WHERE clause

For this sample, we use a button to trigger executing dynamic query. Right-click the button; choose “Assign Actions”; choose “onclick” event:



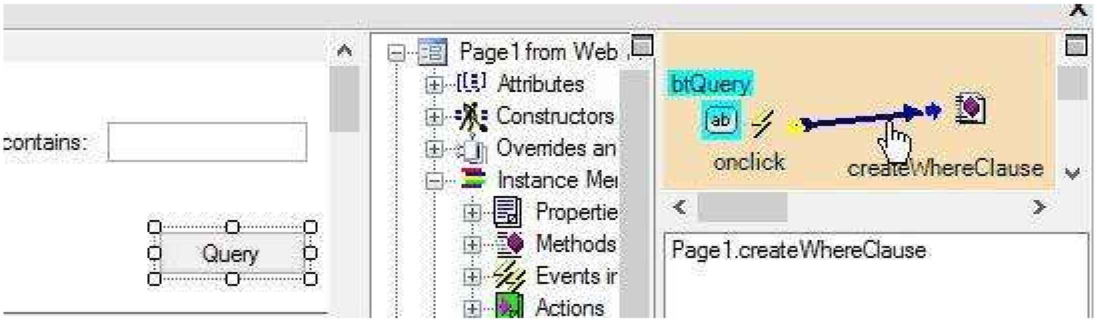
The first action we want to execute is the createWhereClause method:





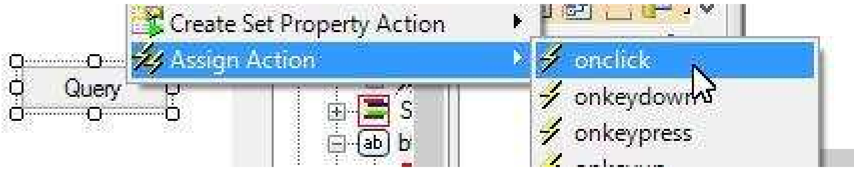
(ActionName)	**Page1.createWhereClause**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	createWhereClause

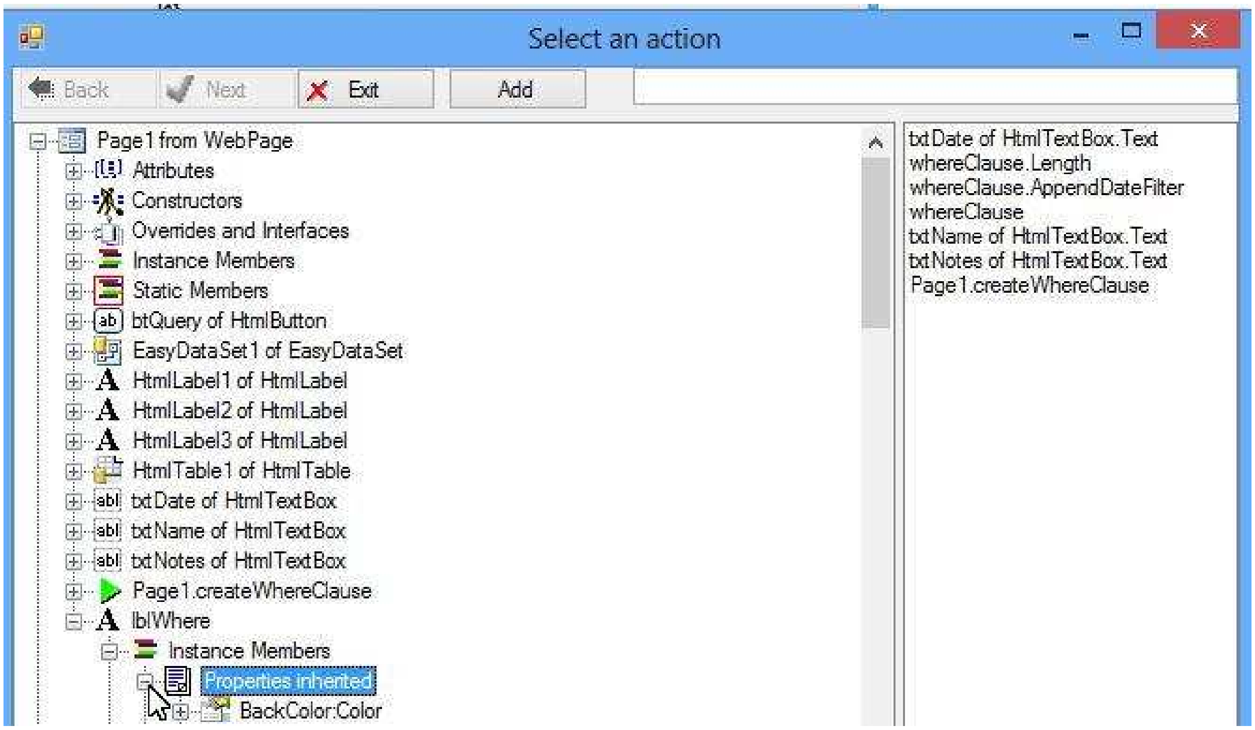
An action is created and assigned to the button:



Display WHERE clause for demo

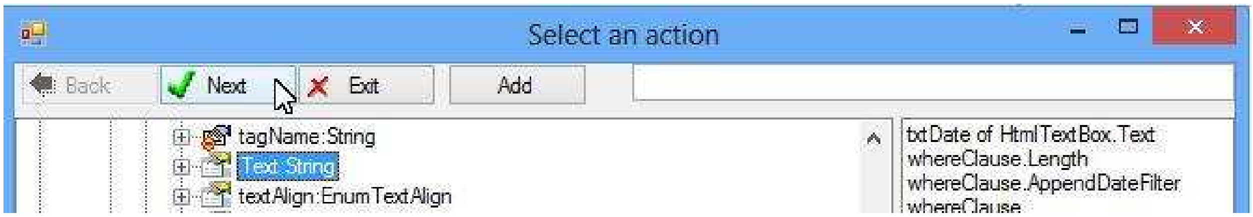
After executing the action, property whereClause will contain the WHERE clause to be used by database query. For demonstration purpose, we may show property whereClause on a label:





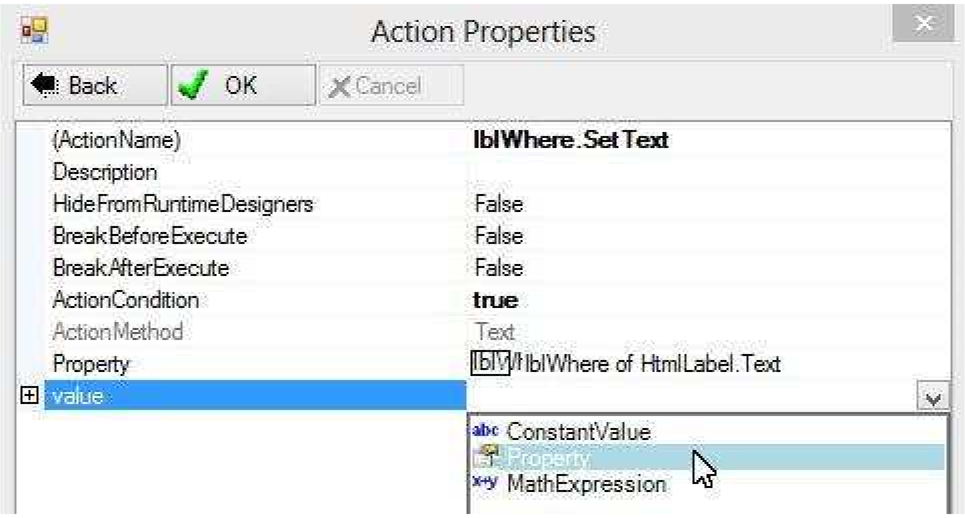
Page1 from WebPage
- Attributes
- Constructors
- Overrides and Interfaces
- Instance Members
- Static Members
- btQuery of HtmlButton
- EasyDataSet1 of EasyDataSet
- HtmlLabel1 of HtmlLabel
- HtmlLabel2 of HtmlLabel
- HtmlLabel3 of HtmlLabel
- HtmlTable1 of HtmlTable
- txtDate of HtmlTextBox
- txtName of HtmlTextBox
- txtNotes of HtmlTextBox
- Page1.createWhereClause
- lblWhere
  - Instance Members
    - Properties inherited
      - BackColor:Color

txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause
txtName of HtmlTextBox.Text
txtNotes of HtmlTextBox.Text
Page1.createWhereClause

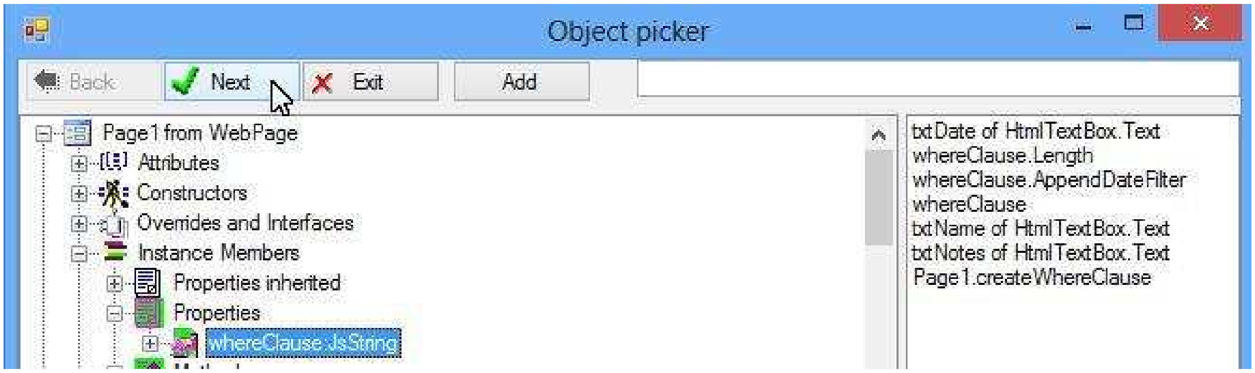


- tagName:String
- Text:String
- textAlign:EnumTextAlign

txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause

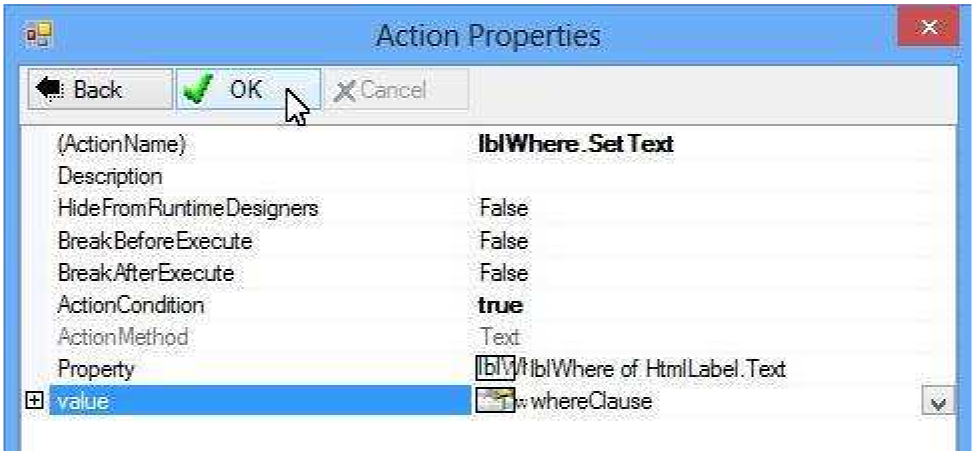


(ActionName)	**lblWhere.SetText**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	Text
Property	lblWhere of HtmlLabel.Text
value	



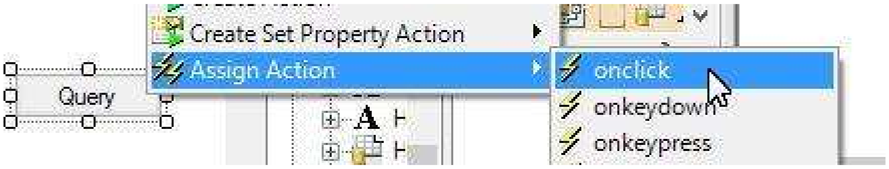
```
Page1 from WebPage
  Attributes
  Constructors
  Overrides and Interfaces
  Instance Members
    Properties inherited
    Properties
      whereClause:JsString
```

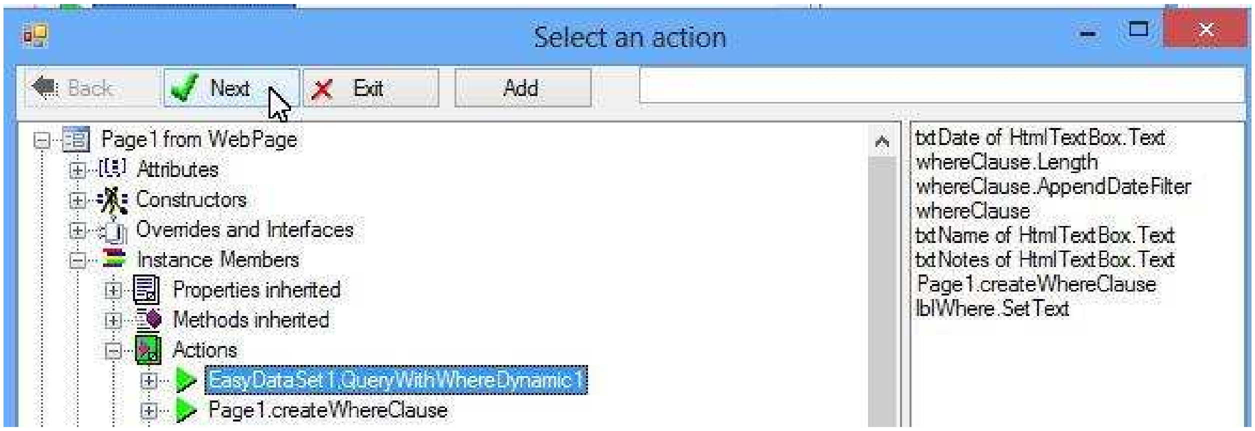
txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause
txtName of HtmlTextBox.Text
txtNotes of HtmlTextBox.Text
Page1.createWhereClause



(ActionName)	**lblWhere.SetText**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	Text
Property	lblWhere of HtmlLabel.Text
value	whereClause

Execute query

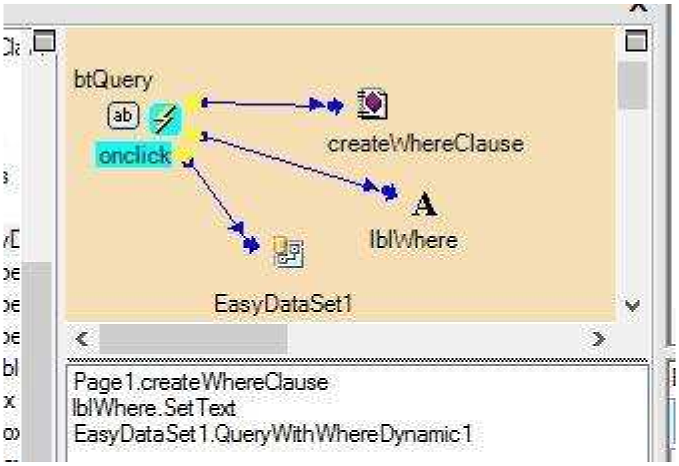




```
Page1 from WebPage
  Attributes
  Constructors
  Overrides and Interfaces
  Instance Members
    Properties inherited
    Methods inherited
    Actions
      EasyDataSet1.QueryWithWhereDynamic1
      Page1.createWhereClause
```

txtDate of HtmlTextBox.Text
whereClause.Length
whereClause.AppendDateFilter
whereClause
txtName of HtmlTextBox.Text
txtNotes of HtmlTextBox.Text
Page1.createWhereClause
lblWhere.SetText

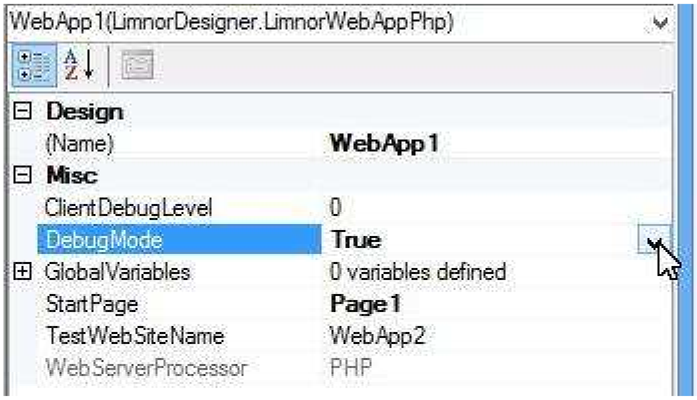
These are all the actions we want to execute:



Page1.createWhereClause
lblWhere.SetText
EasyDataSet1.QueryWithWhereDynamic1

Test

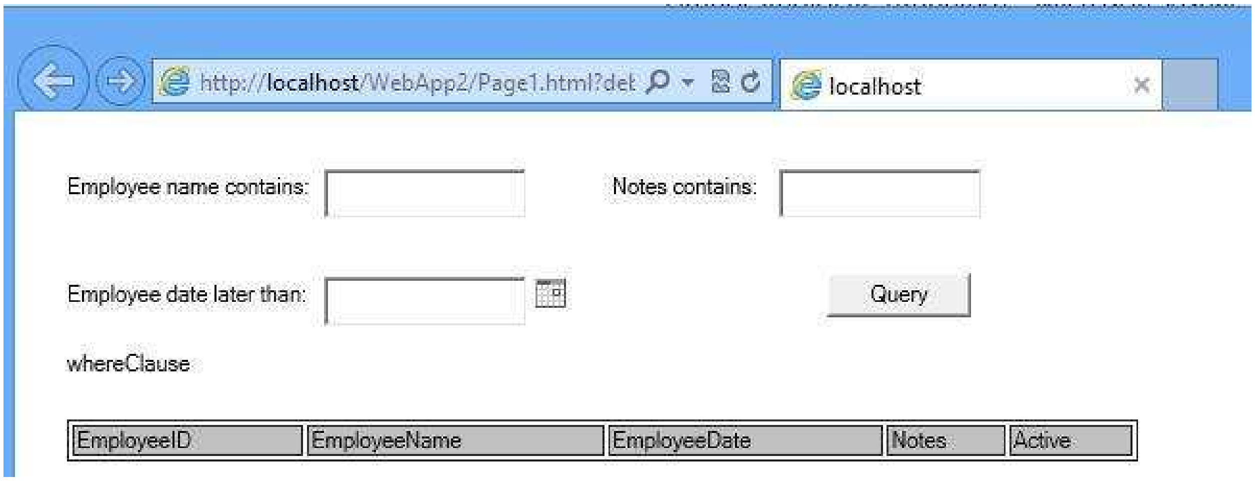
Turn on debug mode so that we may watch what data are going through the internet:



WebApp1(LimnorDesigner.LimnorWebAppPhp)

Design	
(Name)	**WebApp1**
Misc	
ClientDebugLevel	0
DebugMode	**True**
GlobalVariables	0 variables defined
StartPage	**Page1**
TestWebSiteName	WebApp2
WebServerProcessor	PHP

Run the project, the web page appears in the default browser:

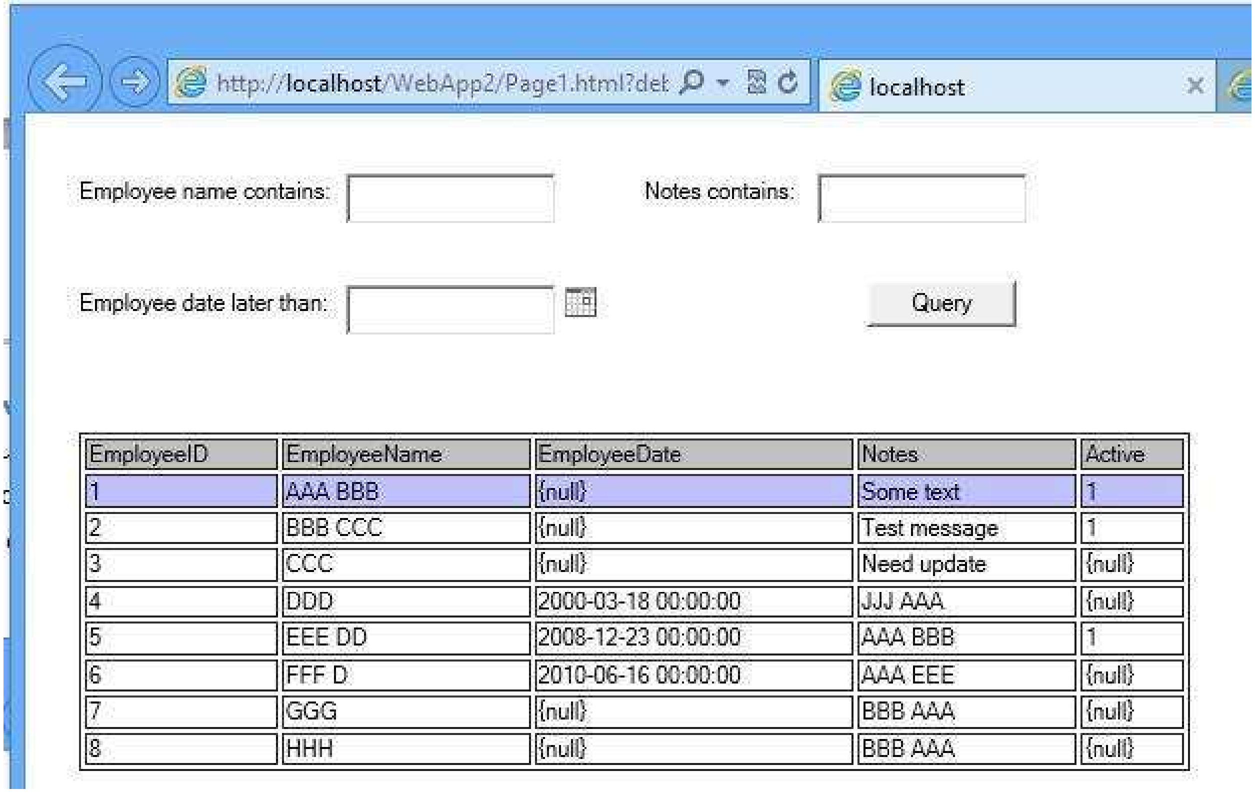


http://localhost/WebApp2/Page1.html?deb

EmployeeID	EmployeeName	EmployeeDate	Notes	Active

Query without filters

Click "Query" button while all text boxes are empty. No filters will be used, and all records will be displayed on the web page. This may not be desired in most situations. You may set action condition so that the query action will only be executed when property whereClause is not empty. For this sample, showing records let us know what are in our sample database:



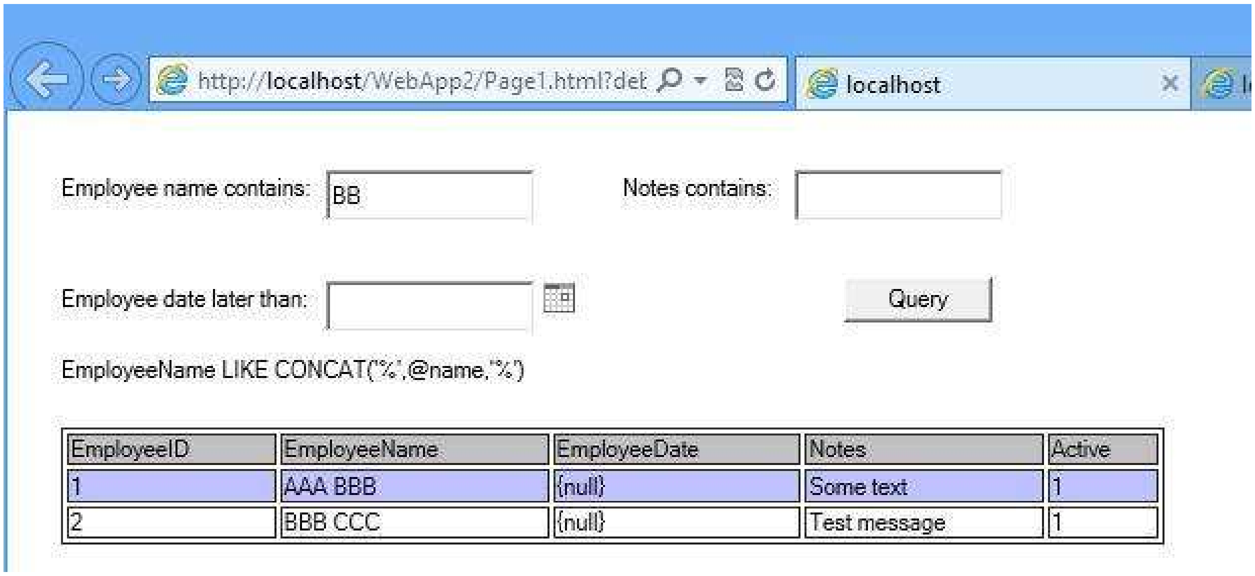
Employee name contains:

Notes contains:

Employee date later than:

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
1	AAA BBB	{null}	Some text	1
2	BBB CCC	{null}	Test message	1
3	CCC	{null}	Need update	{null}
4	DDD	2000-03-18 00:00:00	JJJ AAA	{null}
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1
6	FFF D	2010-06-16 00:00:00	AAA EEE	{null}
7	GGG	{null}	BBB AAA	{null}
8	HHH	{null}	BBB AAA	{null}

Use one filter



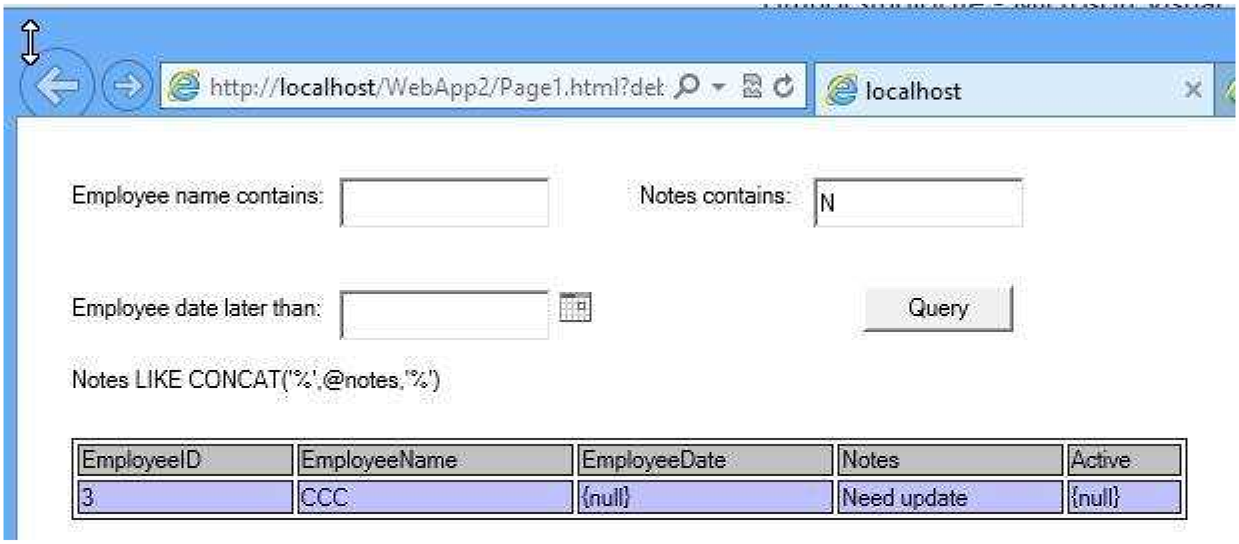
Employee name contains: BB

Notes contains:

Employee date later than:

EmployeeName LIKE CONCAT('%',@name,'%')

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
1	AAA BBB	{null}	Some text	1
2	BBB CCC	{null}	Test message	1



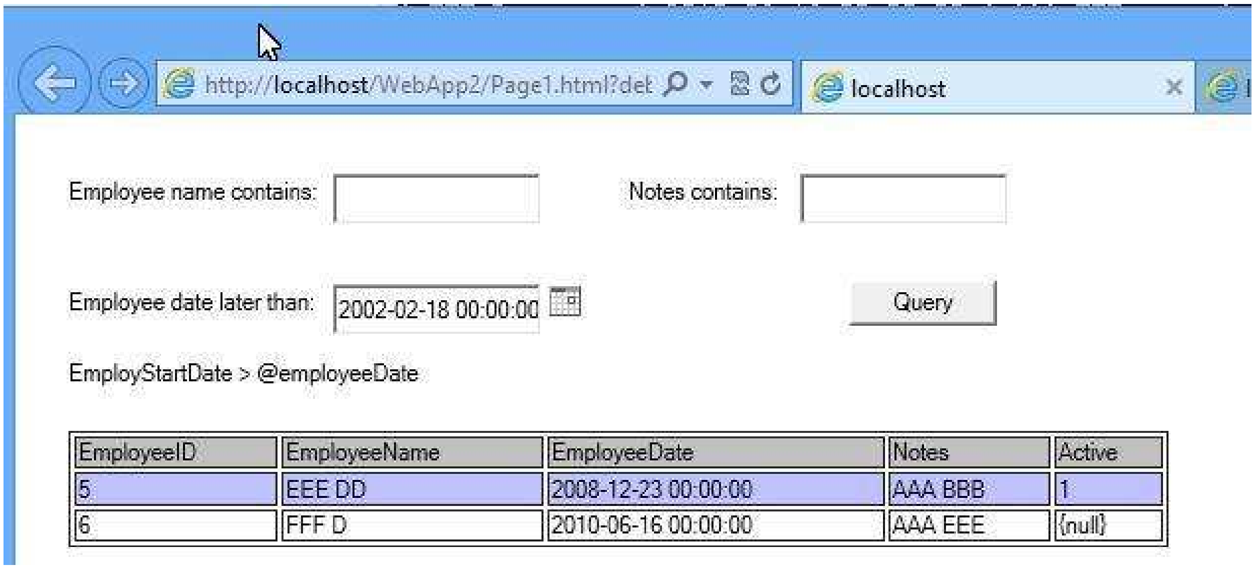
Employee name contains:

Notes contains: N

Employee date later than:

Notes LIKE CONCAT('%',@notes,'%')

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
3	CCC	{null}	Need update	{null}



Employee name contains:

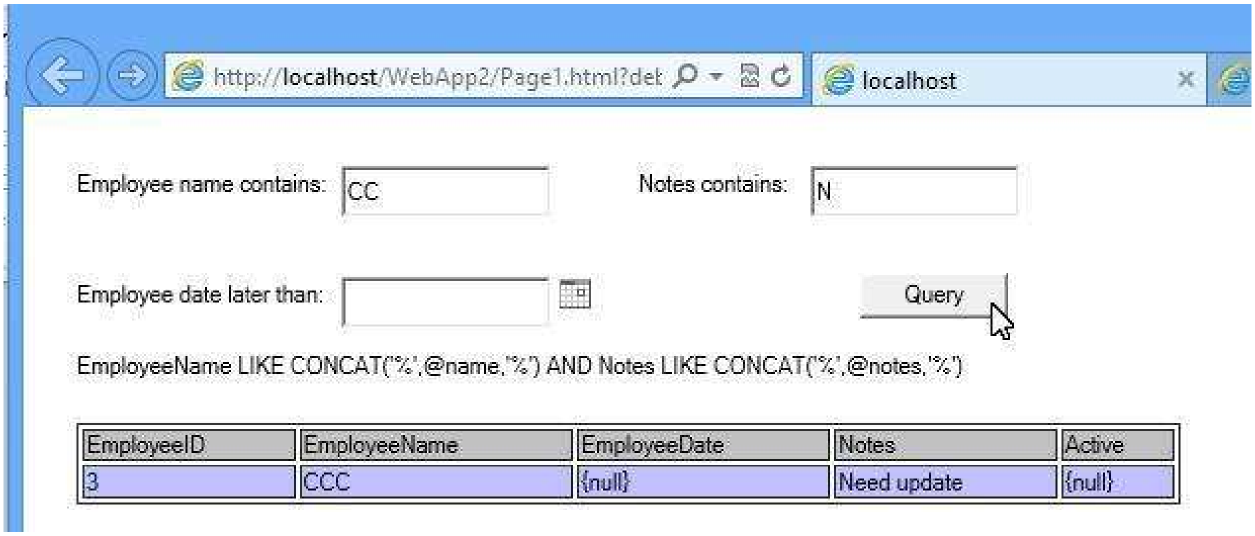
Notes contains:

Employee date later than: 2002-02-18 00:00:00

EmployStartDate > @employeeDate

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1
6	FFF D	2010-06-16 00:00:00	AAA EEE	{null}

Use two filters



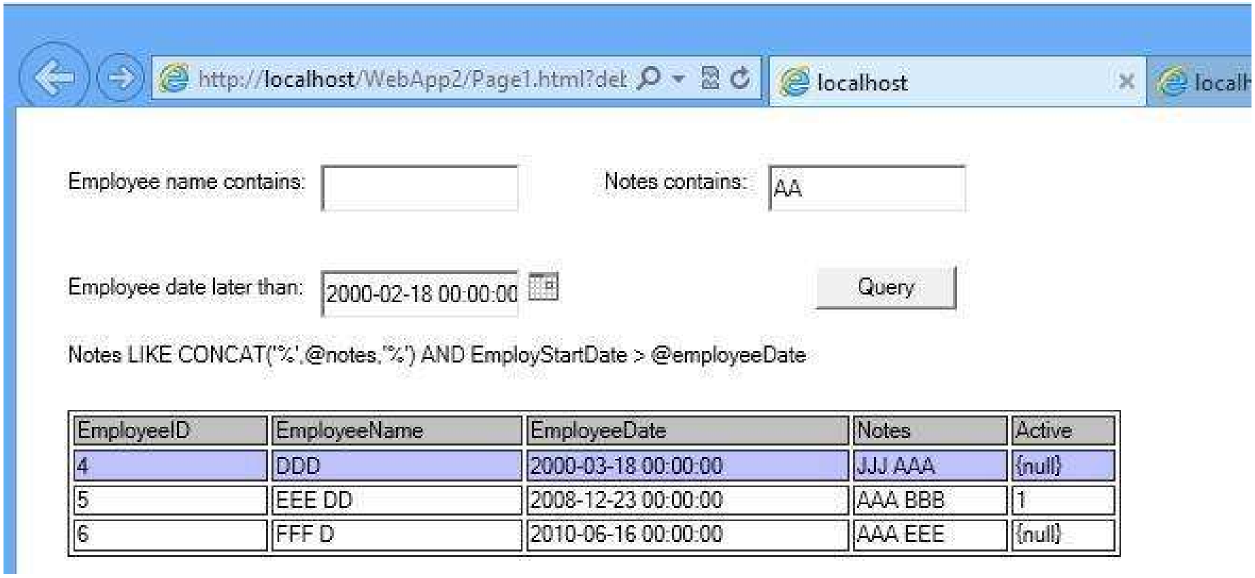
Employee name contains: CC

Notes contains: N

Employee date later than:

EmployeeName LIKE CONCAT('%',@name,'%') AND Notes LIKE CONCAT('%',@notes,'%')

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
3	CCC	{null}	Need update	{null}



Employee name contains:

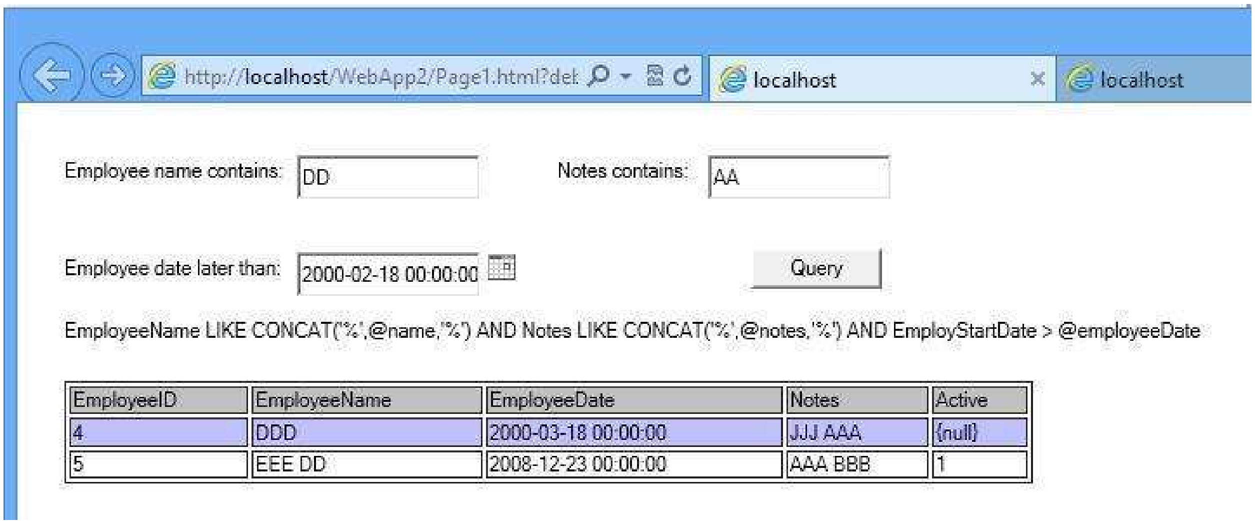
Notes contains: AA

Employee date later than: 2000-02-18 00:00:00

Notes LIKE CONCAT('%',@notes,'%') AND EmployStartDate > @employeeDate

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
4	DDD	2000-03-18 00:00:00	JJJ AAA	{null}
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1
6	FFF D	2010-06-16 00:00:00	AAA EEE	{null}

Use three filters



Employee name contains: DD

Notes contains: AA

Employee date later than: 2000-02-18 00:00:00

EmployeeName LIKE CONCAT('%',@name,'%') AND Notes LIKE CONCAT('%',@notes,'%') AND EmployStartDate > @employeeDate

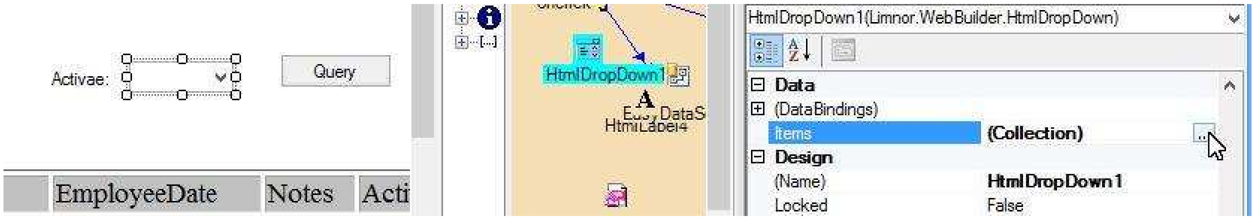
EmployeeID	EmployeeName	EmployeeDate	Notes	Active
4	DDD	2000-03-18 00:00:00	JJJ AAA	{null}
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1

Filters on Boolean Fields

Since a Boolean field may only have True/False value, you may choose not to use a parameter for filtering such a field.

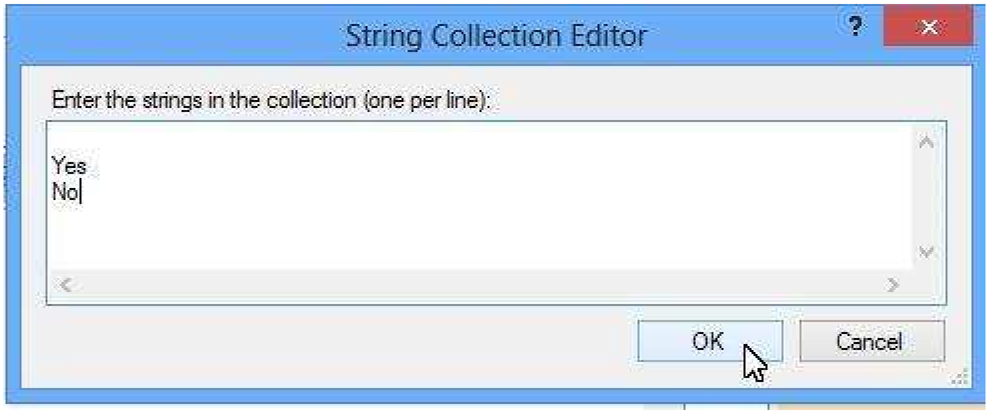
Use a drop down list box as UI

Suppose we use a drop down to give users an option to filter on Active field which is a BOOLEAN field. The drop down has 3 values: empty, Yes, and No:



HtmlDropDown1(Limnor.WebBuilder.HtmlDropDown)

Data	
(DataBindings)	
Items	(Collection)
Design	
(Name)	HtmlDropDown1
Locked	False



String Collection Editor

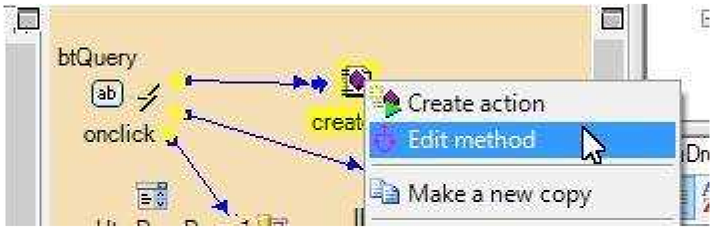
Enter the strings in the collection (one per line):

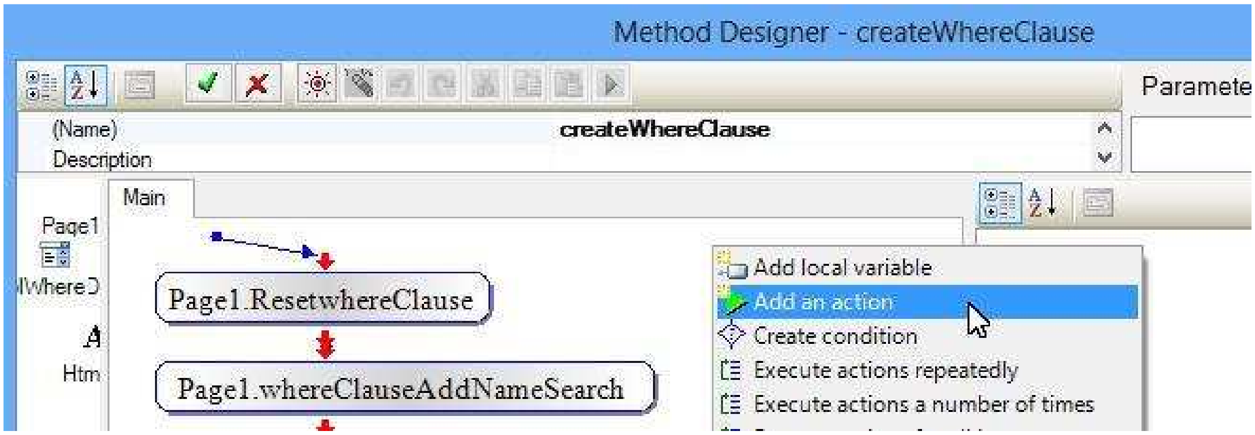
```

Yes
No
```

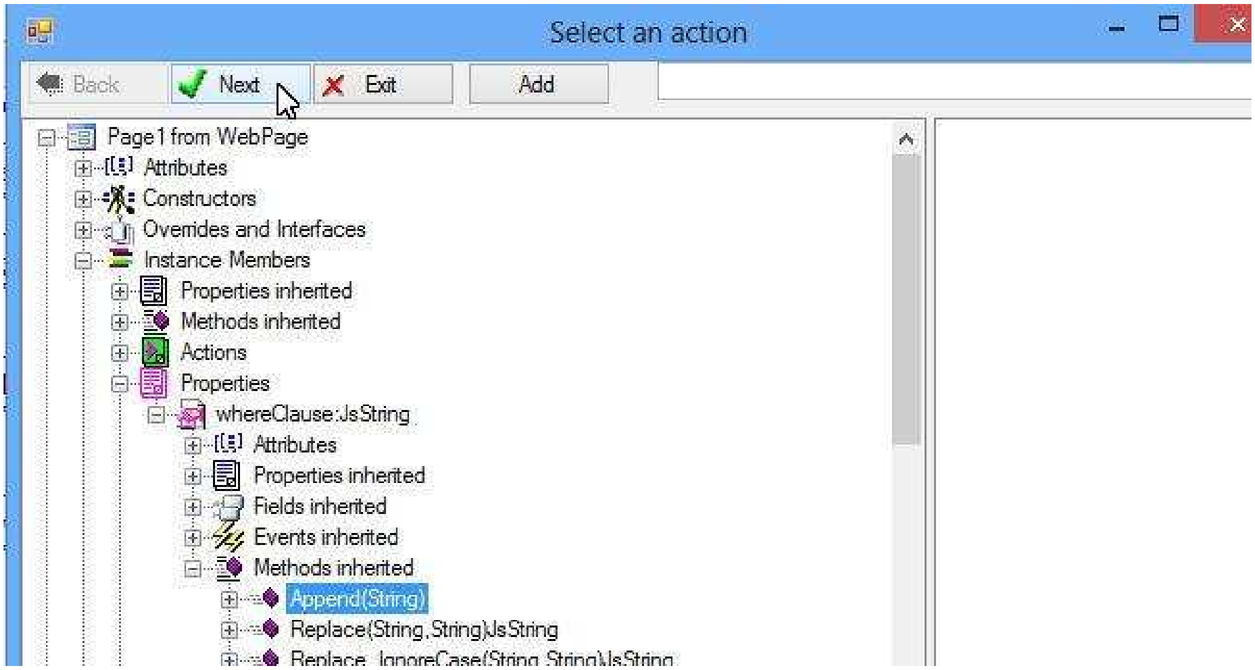
Add filtering

Modify method createWhereClause to append a filter on field Active:

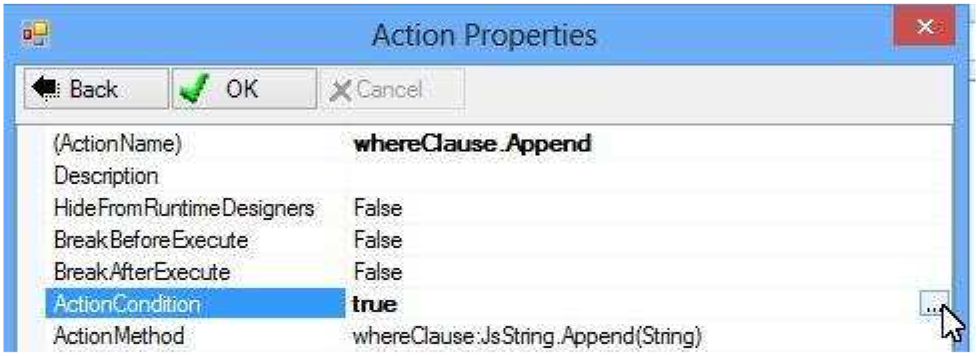



Method Designer - createWhereClause

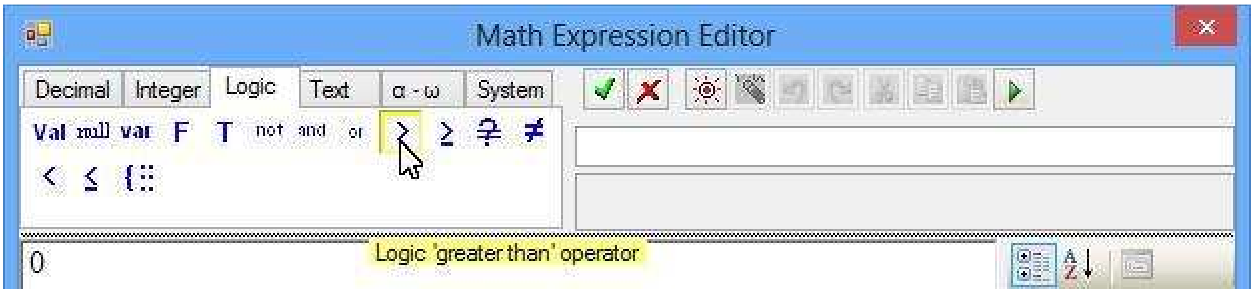
(Name) createWhereClause
Description


Select an action

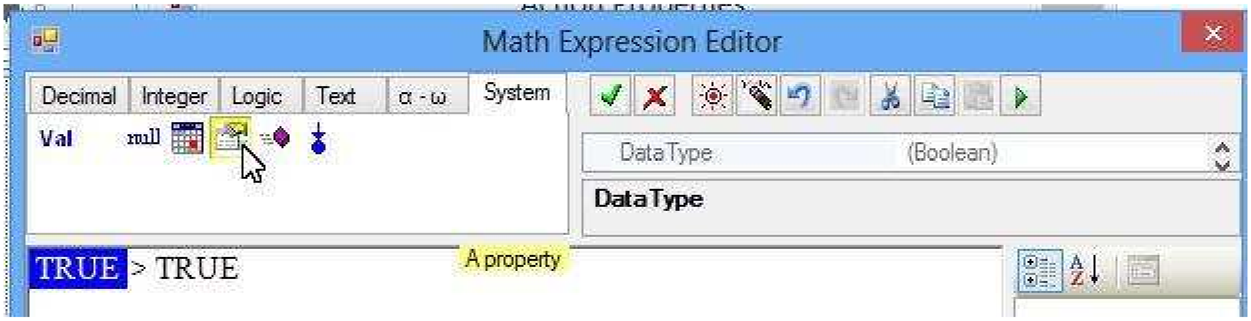
Set ActionCondition to that the selected index of the drop down is larger than 0 because the first item is empty, meaning not to filter on it:



(ActionName)	**whereClause.Append**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**true**
ActionMethod	whereClause:JsString.Append(String)



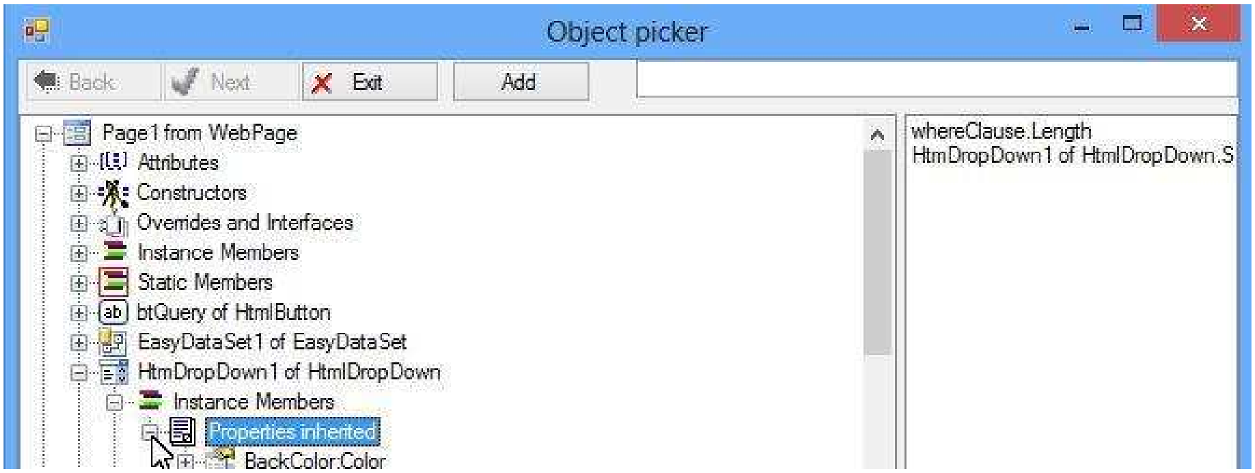
0



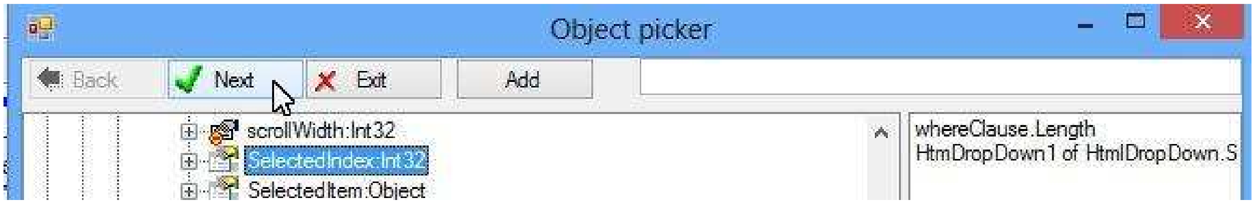
DataType	(Boolean)

DataType

TRUE > TRUE

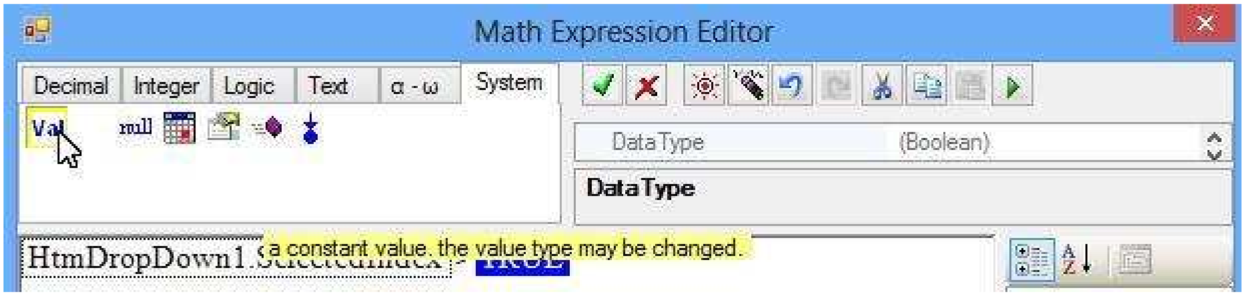


whereClause.Length
HtmDropDown1 of HtmlDropDown.S



scrollWidth:Int32
SelectedIndex:Int32
SelectedItem:Object

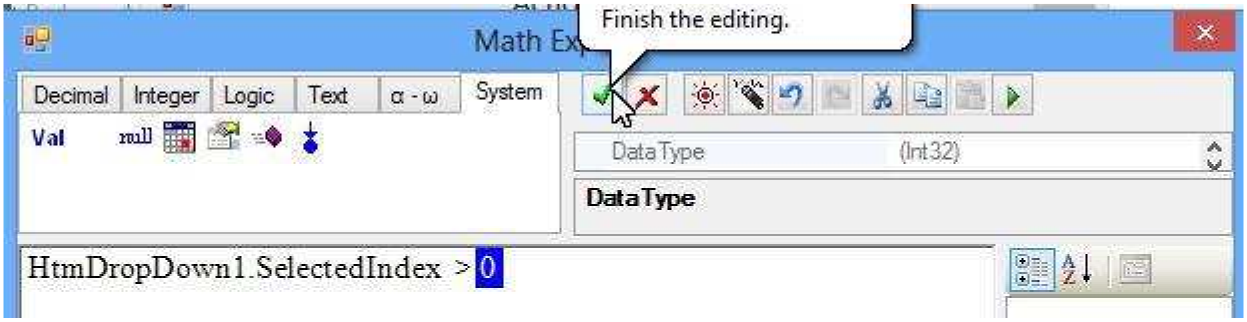
whereClause.Length
HtmDropDown1 of HtmlDropDown.S



DataType (Boolean)

HtmDropDown1.SelectedIndex > TRUE

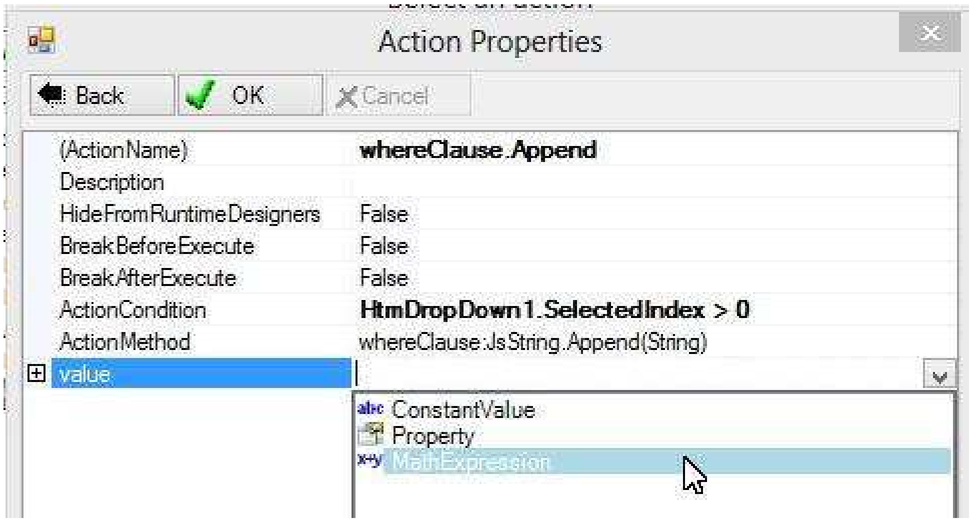
a constant value. the value type may be changed.



DataType (Int32)

HtmDropDown1.SelectedIndex > 0

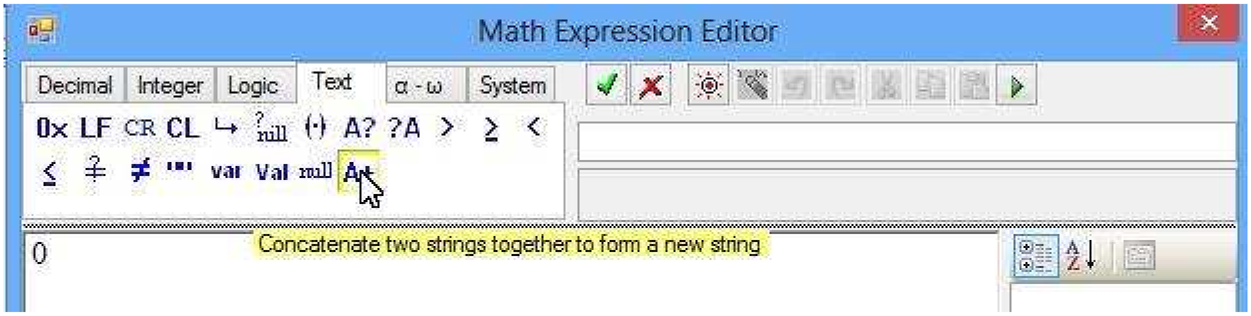
Build an expression for the filter:



(ActionName)	**whereClause.Append**
Description	
HideFromRuntimeDesigners	False
BreakBeforeExecute	False
BreakAfterExecute	False
ActionCondition	**HtmDropDown1.SelectedIndex > 0**
ActionMethod	whereClause:JsString.Append(String)
value	

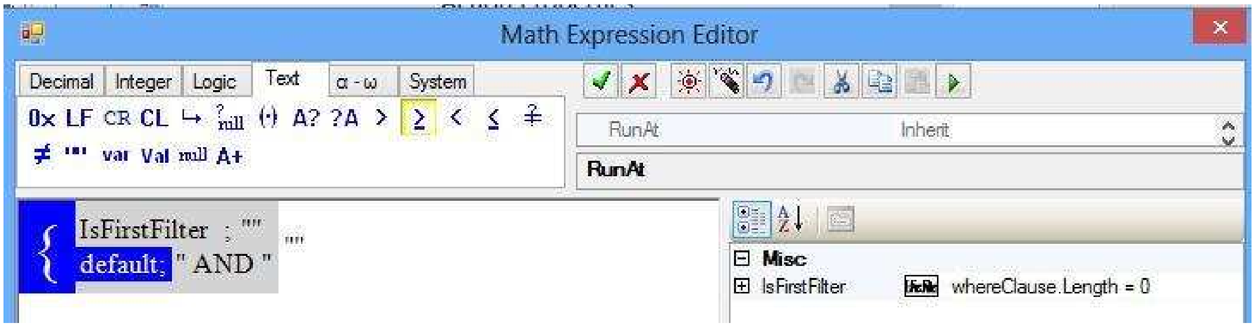
ConstantValue
Property
MathExpression

The expression has two parts and the first part is the same as for filtering on Notes and EmployStartDate:



0

Concatenate two strings together to form a new string



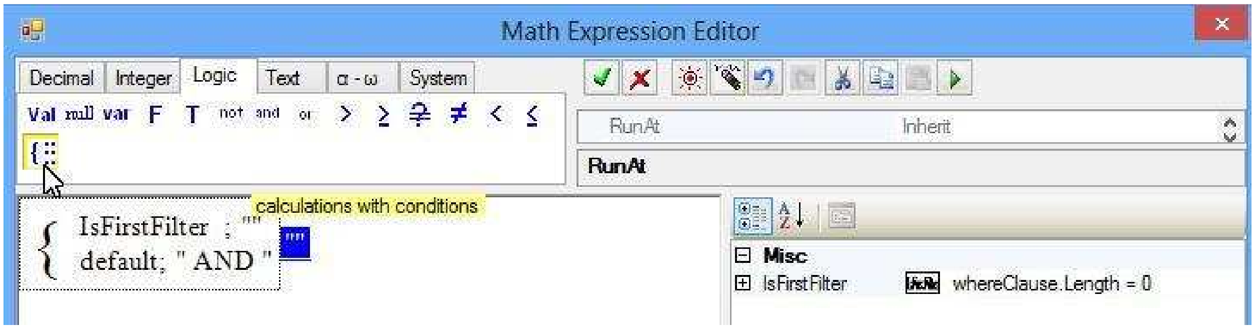
RunAt Inherit

RunAt

{ IsFirstFilter ; ""
default; " AND " ""

Misc
IsFirstFilter whereClause.Length = 0

We may also use a switch for the second part to form the filter:



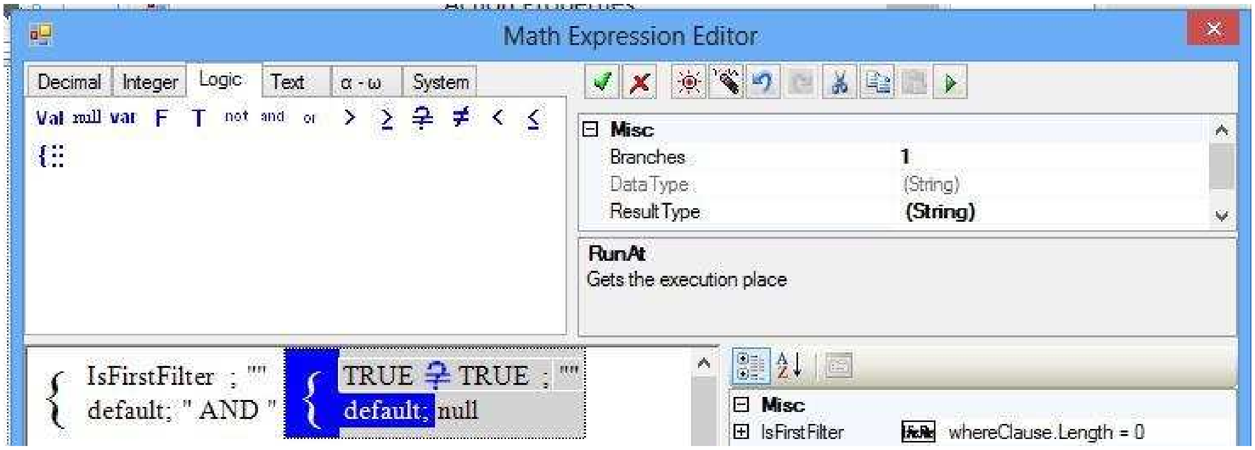
calculations with conditions

RunAt Inherit

RunAt

{ IsFirstFilter ; ""
default; " AND " ""

Misc
IsFirstFilter whereClause.Length = 0



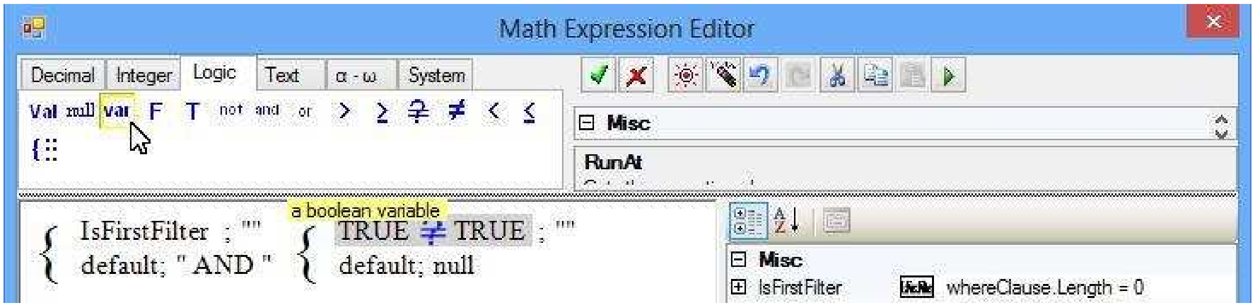
Misc
Branches 1
DataType (String)
ResultType (String)

RunAt
Gets the execution place

{ IsFirstFilter ; ""
default; " AND " { TRUE ≟ TRUE ; ""
default; null

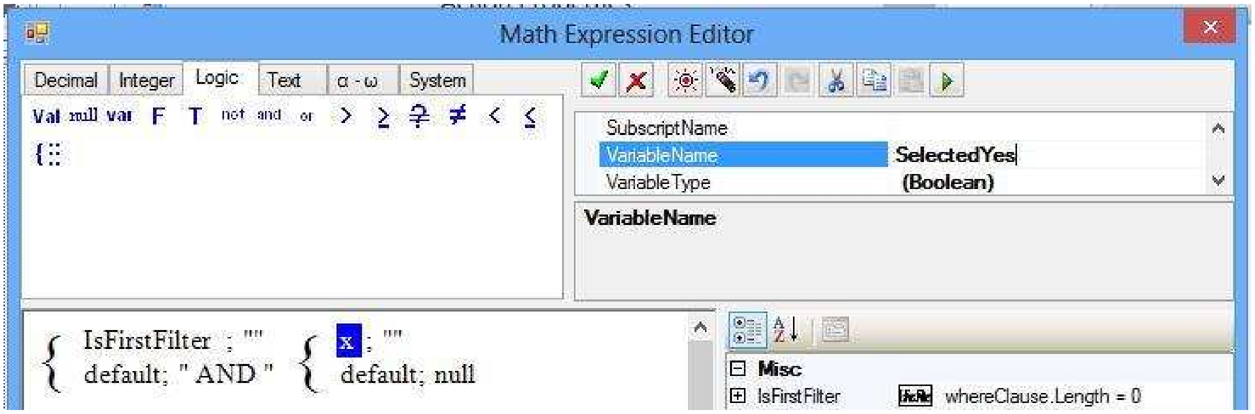
Misc
IsFirstFilter whereClause.Length = 0

If the selected index of the drop down is 1 then we use filter "Active = True"; else we use filter "(Active is NULL OR Active = False)":



```
{ IsFirstFilter ; ""      { TRUE ≑ TRUE ; ""
{ default; " AND "        { default; null
```

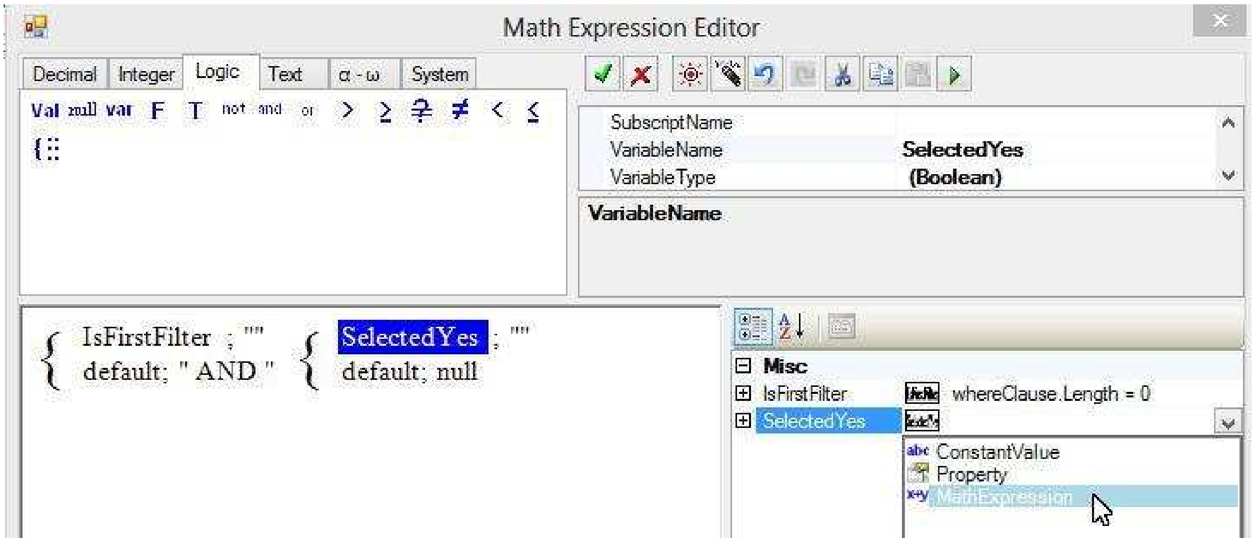
Misc
IsFirstFilter   whereClause.Length = 0



SubscriptName	
VariableName	SelectedYes
VariableType	(Boolean)

```
{ IsFirstFilter ; ""      { x ; ""
{ default; " AND "        { default; null
```

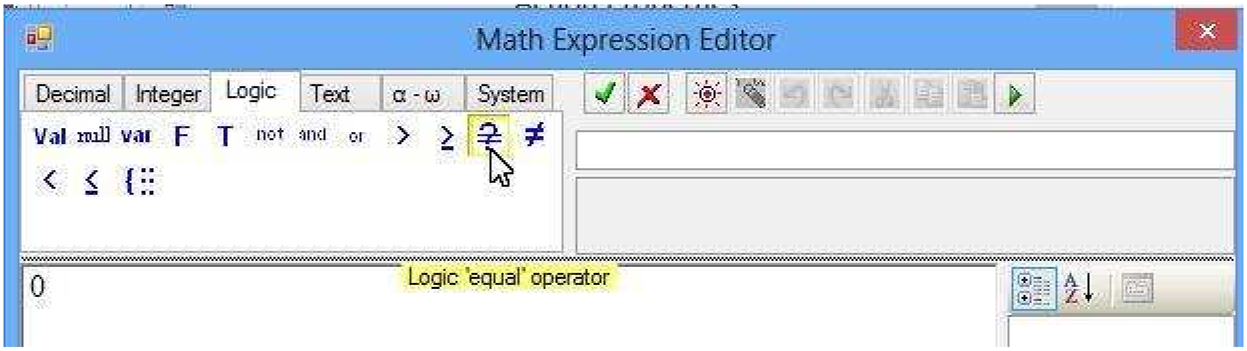
Misc
IsFirstFilter   whereClause.Length = 0



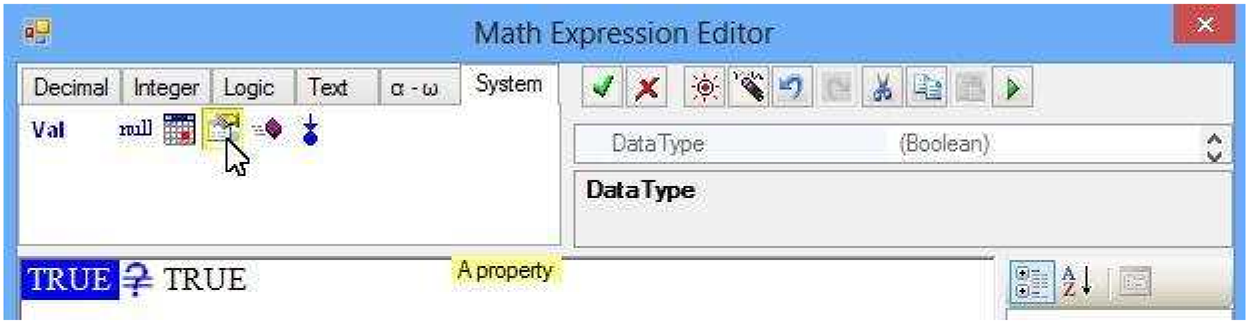
SubscriptName	
VariableName	SelectedYes
VariableType	(Boolean)

```
{ IsFirstFilter ; ""      { SelectedYes ; ""
{ default; " AND "        { default; null
```

Misc
IsFirstFilter   whereClause.Length = 0
SelectedYes



0

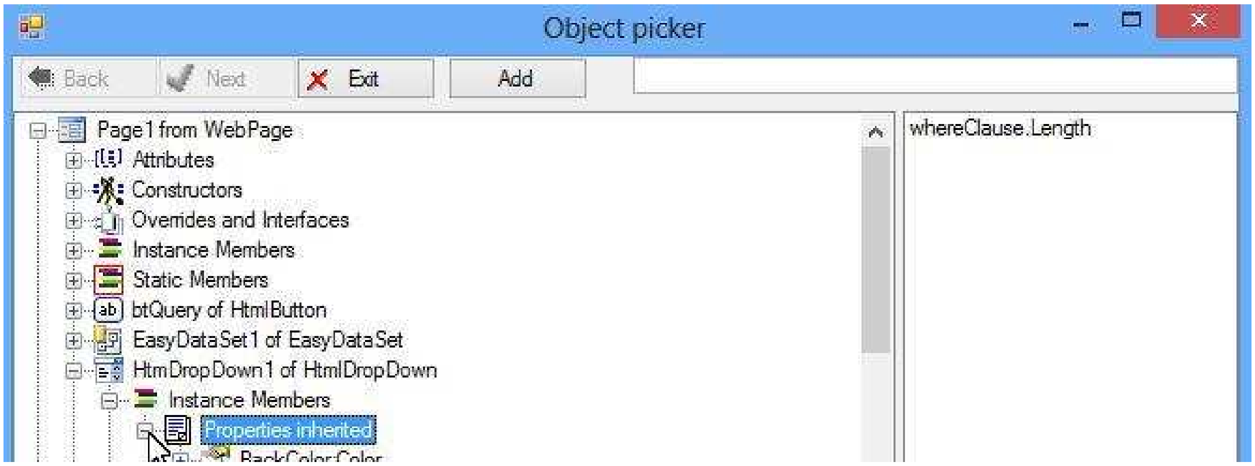


Data Type (Boolean)

Data Type

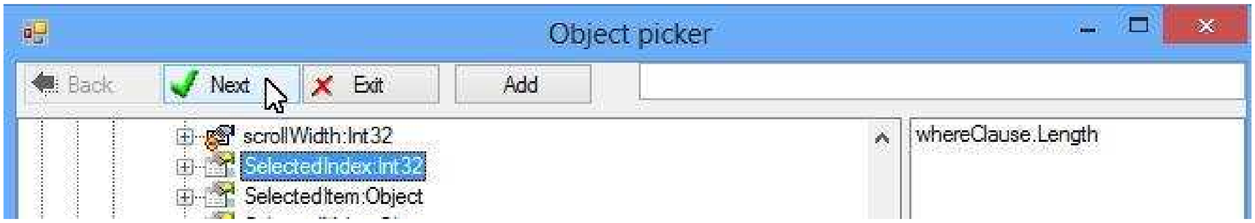
TRUE ≟ TRUE

A property



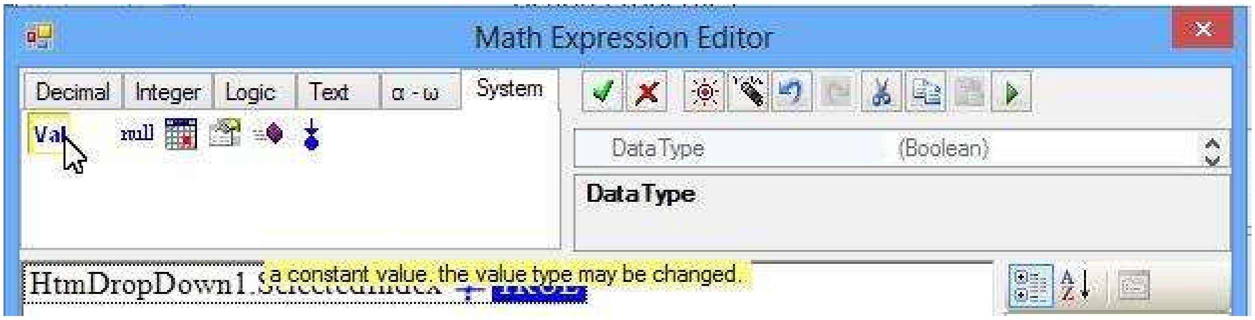
Page1 from WebPage
Attributes
Constructors
Overrides and Interfaces
Instance Members
Static Members
btQuery of HtmlButton
EasyDataSet1 of EasyDataSet
HtmDropDown1 of HtmlDropDown
Instance Members
Properties inherited
BackColor:Color

whereClause.Length



scrollWidth:Int32
SelectedIndex:Int32
SelectedItem:Object

whereClause.Length

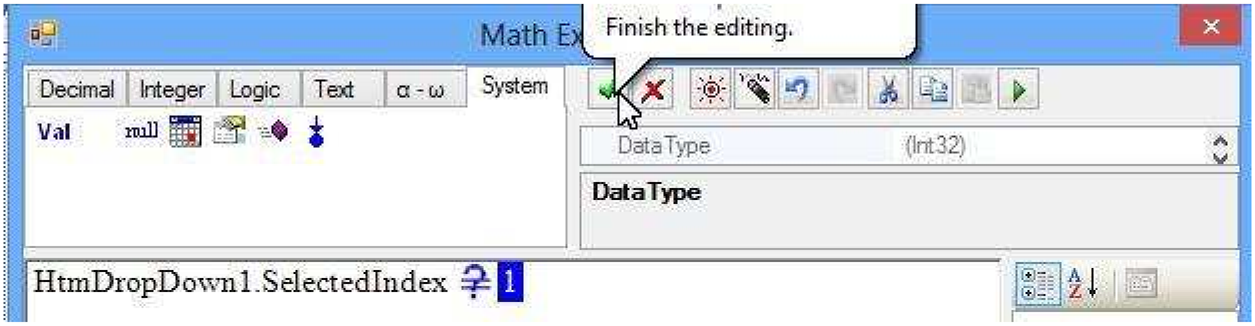


Data Type (Boolean)

Data Type

HtmDropDown1.SelectedIndex ≟ TRUE

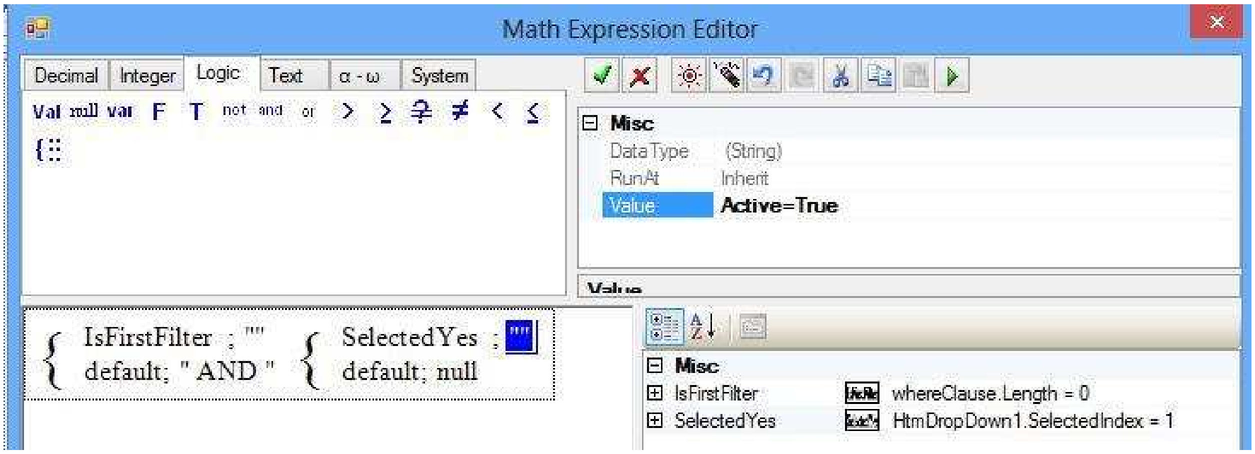
a constant value. the value type may be changed.



Data Type (Int32)

Data Type

HtmDropDown1.SelectedIndex ≑ 1



Misc

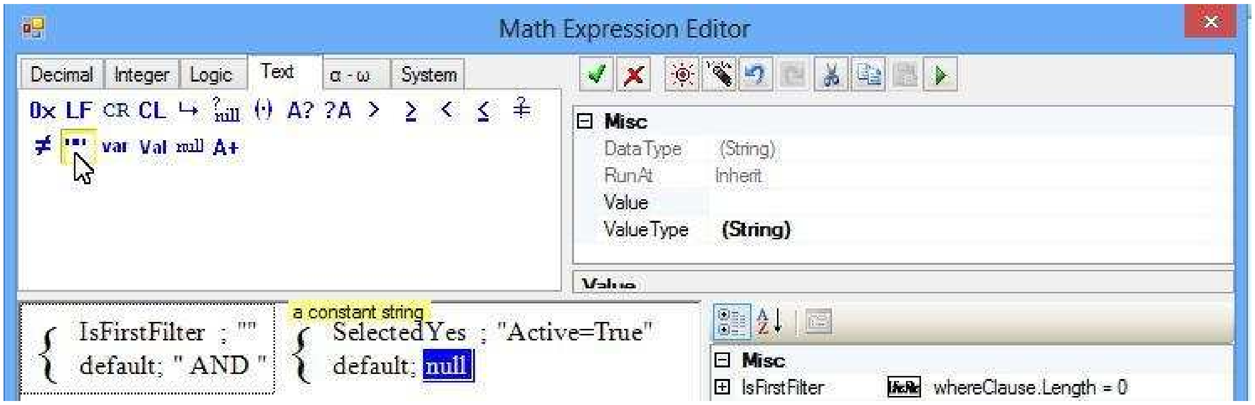
Data Type	(String)
RunAt	Inherit
Value	Active=True

IsFirstFilter ; ""
default; " AND "

SelectedYes ; ""
default; null

Misc

IsFirstFilter	whereClause.Length = 0
SelectedYes	HtmDropDown1.SelectedIndex = 1



Misc

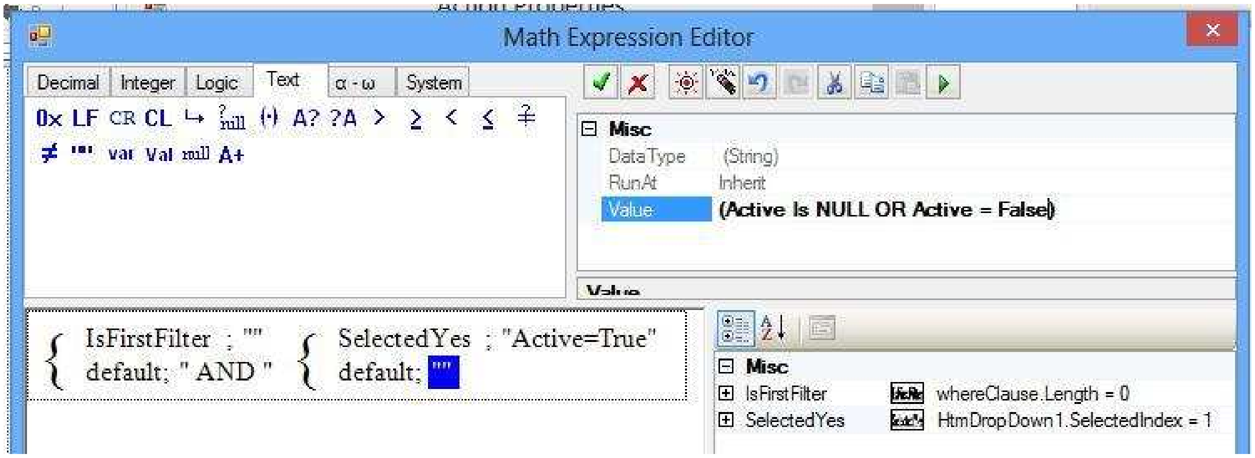
Data Type	(String)
RunAt	Inherit
Value	
Value Type	(String)

IsFirstFilter ; ""
default; " AND "

SelectedYes ; "Active=True"
default; null

Misc

IsFirstFilter	whereClause.Length = 0



Misc

Data Type	(String)
RunAt	Inherit
Value	(Active Is NULL OR Active = False)

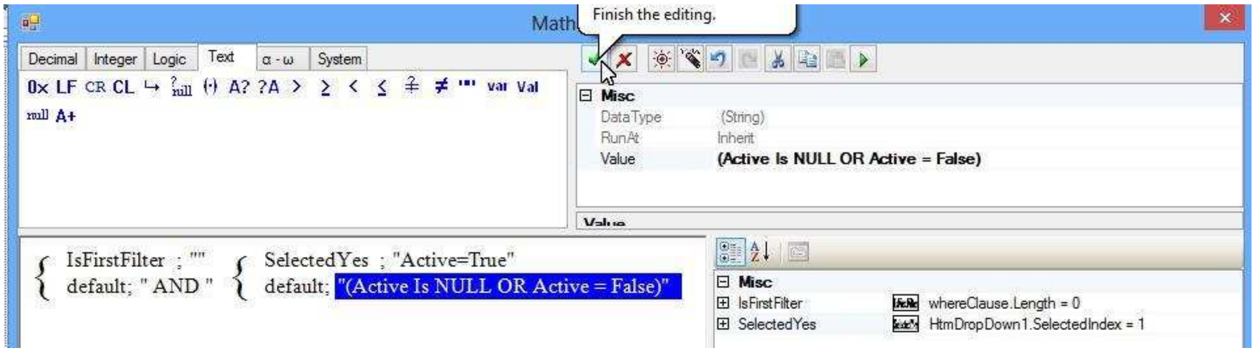
IsFirstFilter ; ""
default; " AND "

SelectedYes ; "Active=True"
default; ""

Misc

IsFirstFilter	whereClause.Length = 0
SelectedYes	HtmDropDown1.SelectedIndex = 1

This is our expression for filtering on field Active:

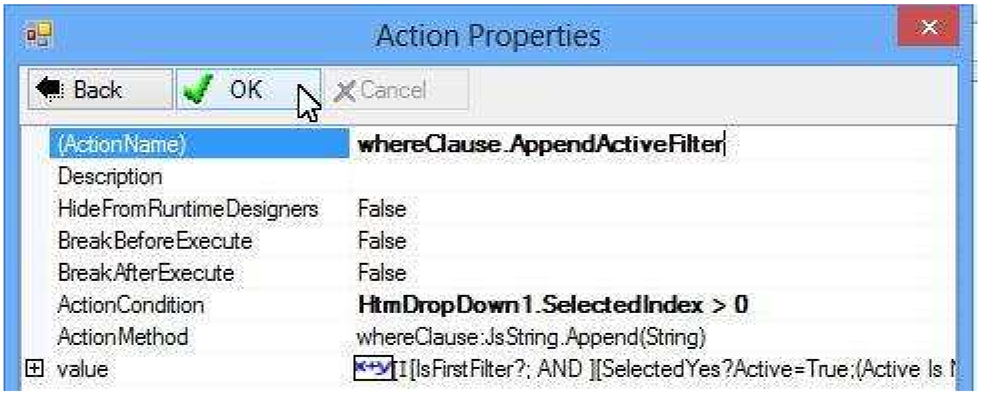


Misc
DataType (String)
RunAt Inherit
Value (Active Is NULL OR Active = False)

IsFirstFilter ; ""
default; " AND "
SelectedYes ; "Active=True"
default; "(Active Is NULL OR Active = False)"

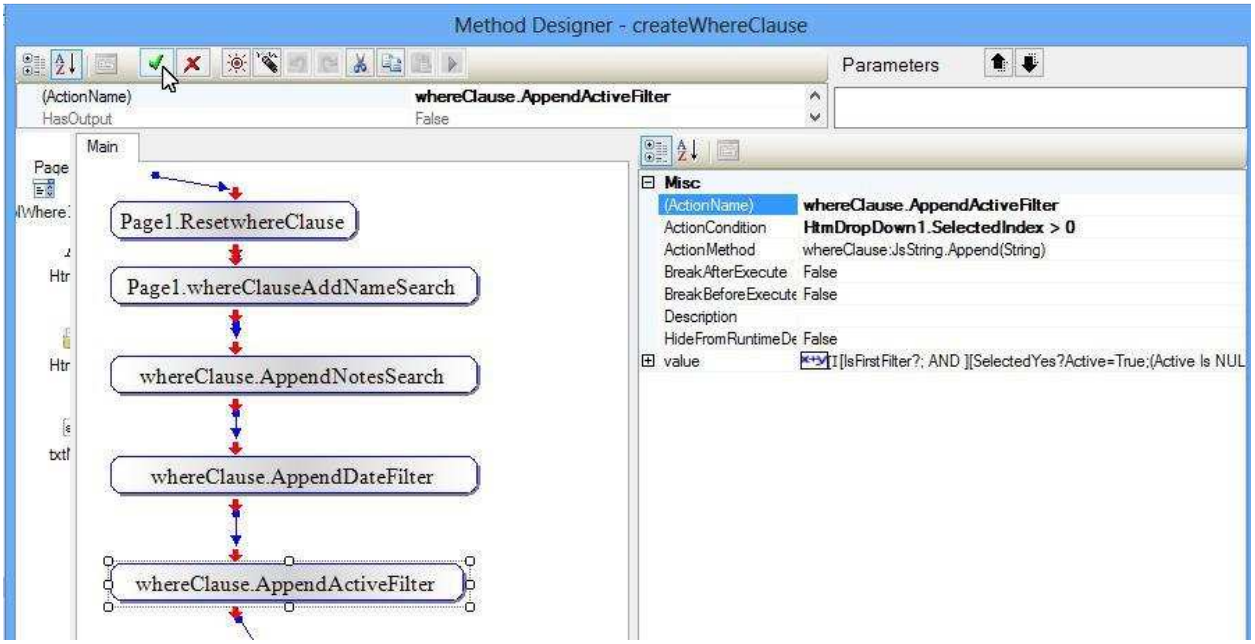
Misc
IsFirstFilter whereClause.Length = 0
SelectedYes HtmDropDown1.SelectedIndex = 1

Rename the action and click OK:



(ActionName) whereClause.AppendActiveFilter
Description
HideFromRuntimeDesigners False
BreakBeforeExecute False
BreakAfterExecute False
ActionCondition HtmDropDown1.SelectedIndex > 0
ActionMethod whereClause:JsString.Append(String)
value [I[IsFirstFilter?; AND][SelectedYes?Active=True;(Active Is N

Link the action:

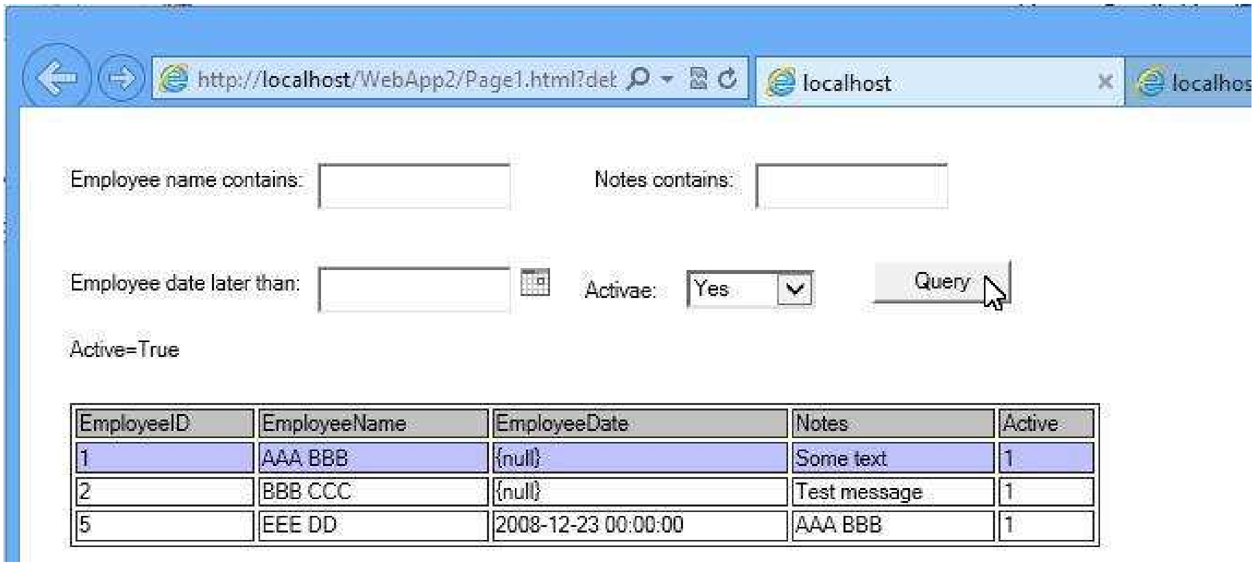

Method Designer - createWhereClause
Parameters
Main
Page1.ResetwhereClause
Page1.whereClauseAddNameSearch
whereClause.AppendNotesSearch
whereClause.AppendDateFilter
whereClause.AppendActiveFilter

(ActionName) whereClause.AppendActiveFilter
HasOutput False

Misc
(ActionName) whereClause.AppendActiveFilter
ActionCondition HtmDropDown1.SelectedIndex > 0
ActionMethod whereClause:JsString.Append(String)
BreakAfterExecute False
BreakBeforeExecute False
Description
HideFromRuntimeDe False
value [I[IsFirstFilter?; AND][SelectedYes?Active=True;(Active Is NUL

Demo

Run project and choose different drop items to see how the filtering is applied:



Employee name contains:

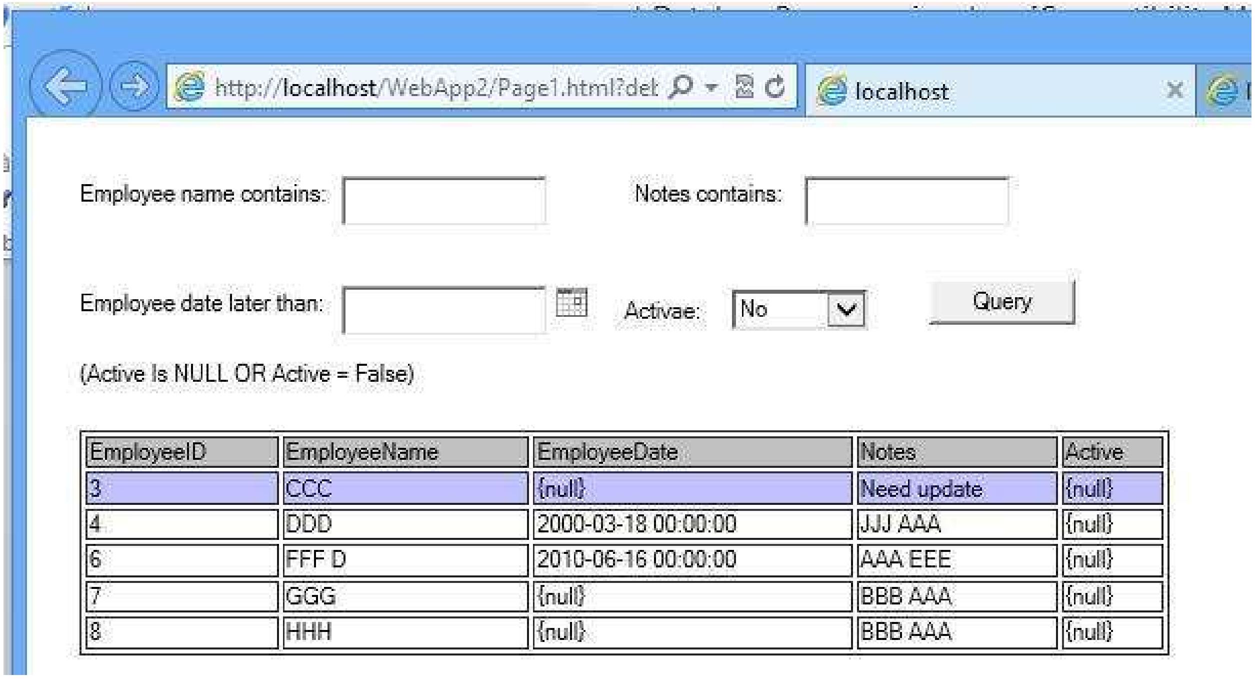
Notes contains:

Employee date later than:

Activae: Yes

Active=True

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
1	AAA BBB	{null}	Some text	1
2	BBB CCC	{null}	Test message	1
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1



Employee name contains:

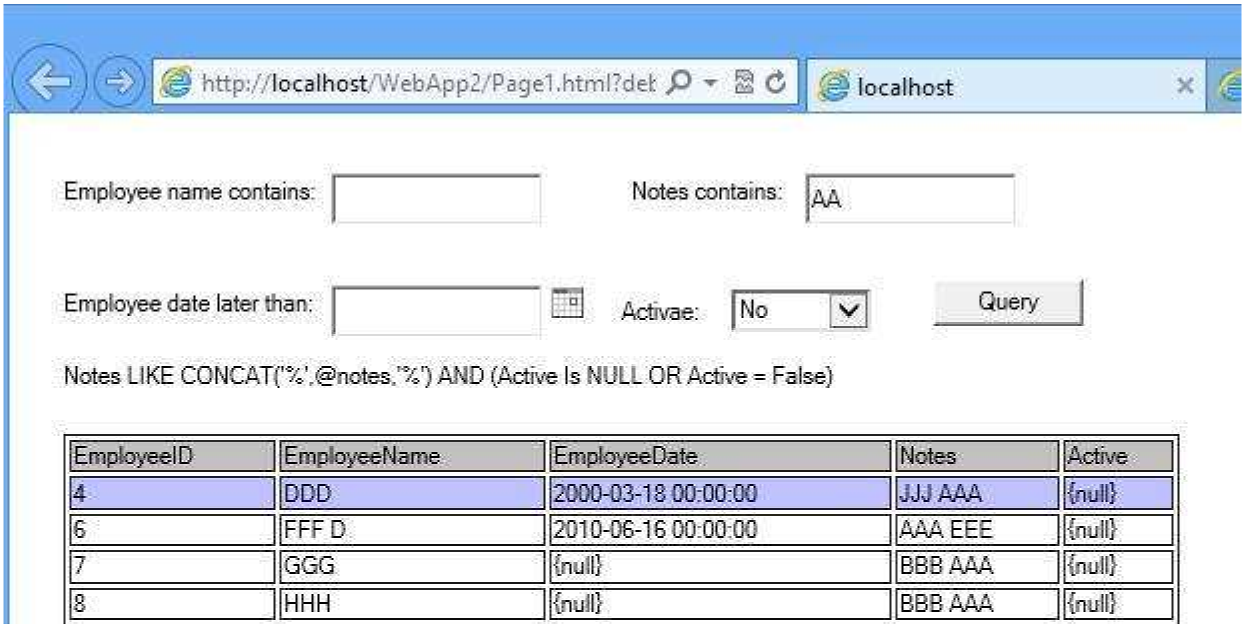
Notes contains:

Employee date later than:

Activae: No

(Active Is NULL OR Active = False)

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
3	CCC	{null}	Need update	{null}
4	DDD	2000-03-18 00:00:00	JJJ AAA	{null}
6	FFF D	2010-06-16 00:00:00	AAA EEE	{null}
7	GGG	{null}	BBB AAA	{null}
8	HHH	{null}	BBB AAA	{null}

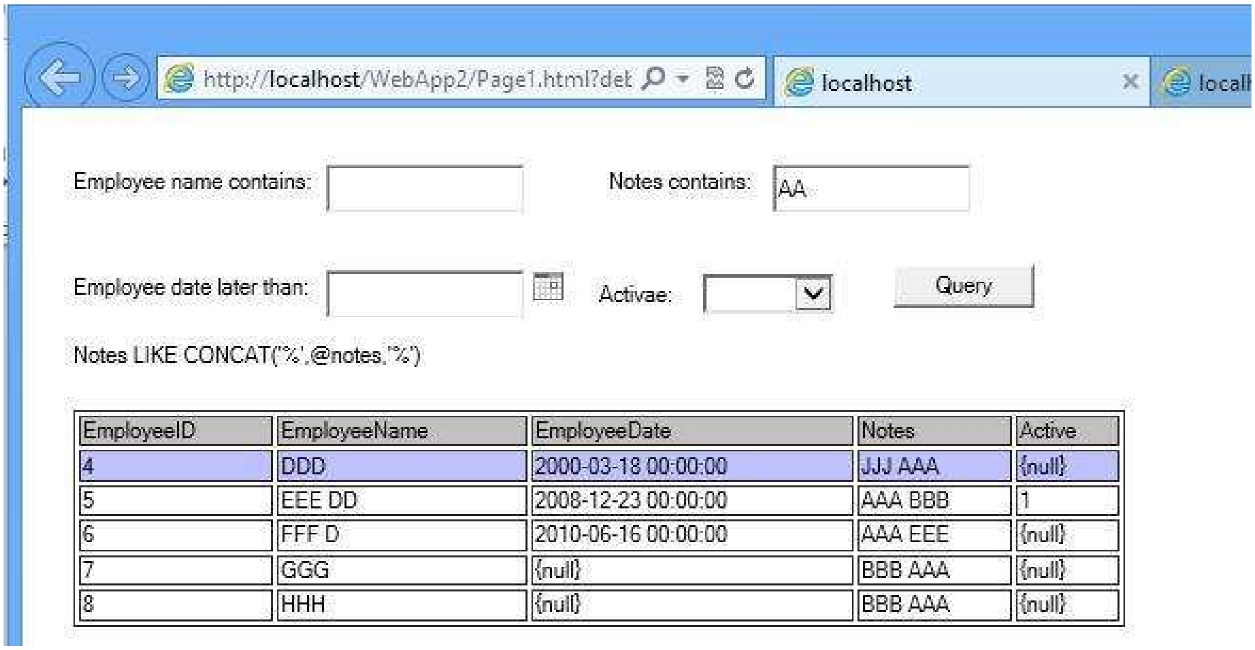


Employee name contains: Notes contains: AA

Employee date later than: Activae: No Query

Notes LIKE CONCAT('%',@notes,'%') AND (Active Is NULL OR Active = False)

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
4	DDD	2000-03-18 00:00:00	JJJ AAA	{null}
6	FFF D	2010-06-16 00:00:00	AAA EEE	{null}
7	GGG	{null}	BBB AAA	{null}
8	HHH	{null}	BBB AAA	{null}

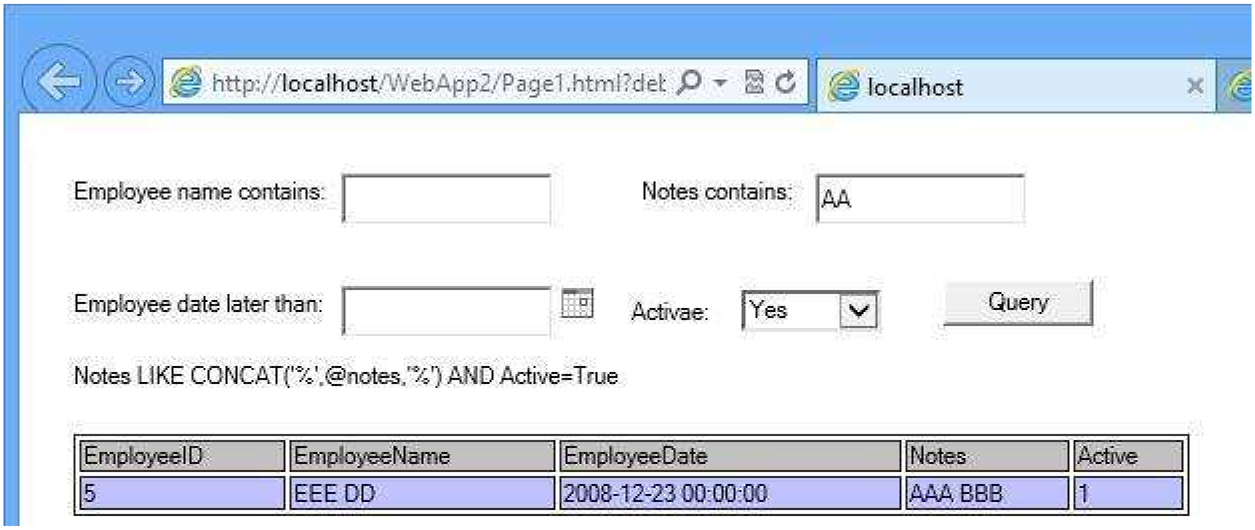


Employee name contains: Notes contains: AA

Employee date later than: Activae: Query

Notes LIKE CONCAT('%',@notes,'%')

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
4	DDD	2000-03-18 00:00:00	JJJ AAA	{null}
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1
6	FFF D	2010-06-16 00:00:00	AAA EEE	{null}
7	GGG	{null}	BBB AAA	{null}
8	HHH	{null}	BBB AAA	{null}



Employee name contains:

Notes contains: AA

Employee date later than:

Activae: Yes

Notes LIKE CONCAT('%',@notes,'%') AND Active=True

EmployeeID	EmployeeName	EmployeeDate	Notes	Active
5	EEE DD	2008-12-23 00:00:00	AAA BBB	1

Security notes

We see that we do not use a parameter for constructing filtering on a Boolean field. It looks simpler than using a parameter as we did for name search. But do not use it for filtering on text fields because it makes your web page vulnerable to SQL-injection attacks. It is safe to use it for Boolean field filtering.

Create New Records

See http://www.limnor.com/support/webDatabaseProgramming5.pdf

Data Streaming

See http://www.limnor.com/support/webDatabaseProgramming5.pdf

Fetch Data of One-to-Many Relation

See http://www.limnor.com/support/webDatabaseProgramming5.pdf

Change Data-binding

See http://www.limnor.com/support/webDatabaseProgramming5.pdf

Feedbacks

Please send your feedbacks to support@limnor.com, thanks!