Use Date Time in Web Client

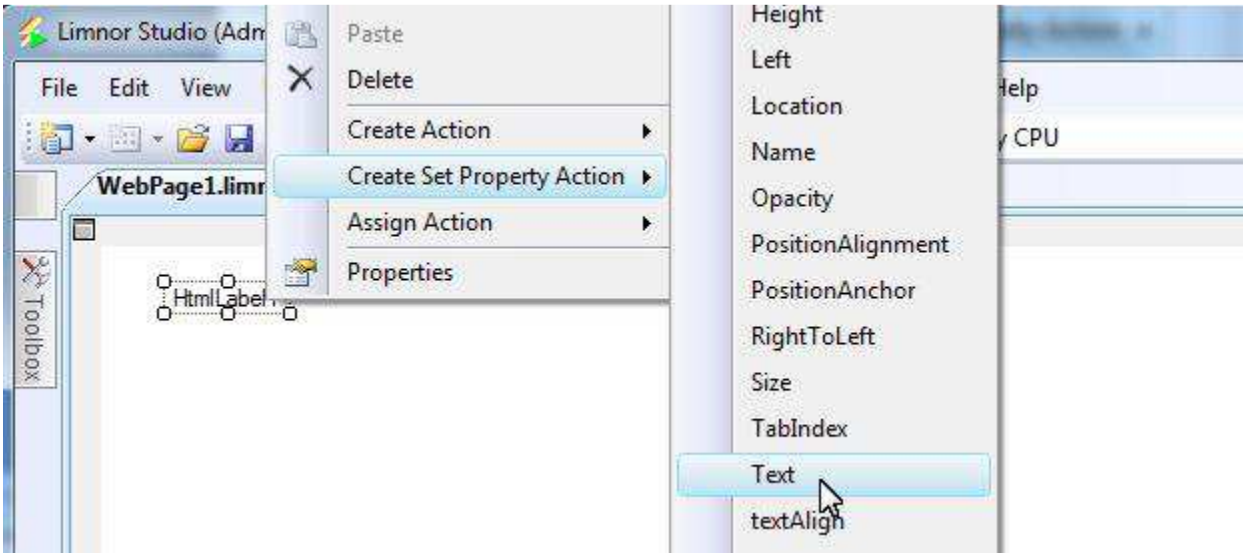
Contents

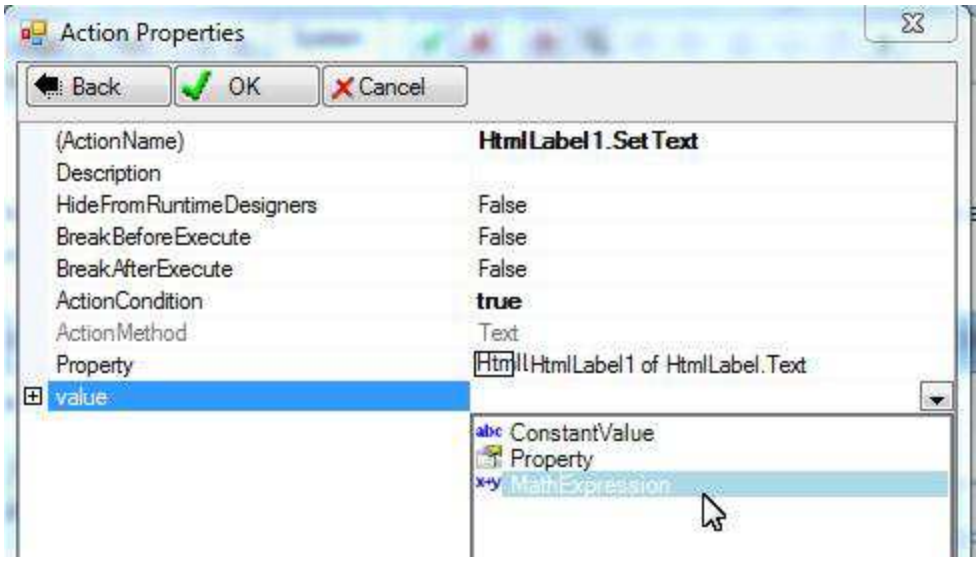
Use “Now”

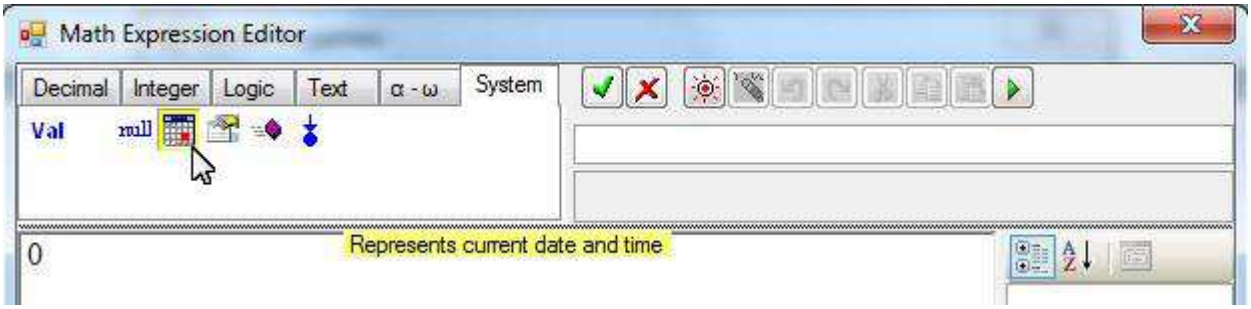
“Now” expression element

Under System tab, there is a “Now” element representing the current date and time.

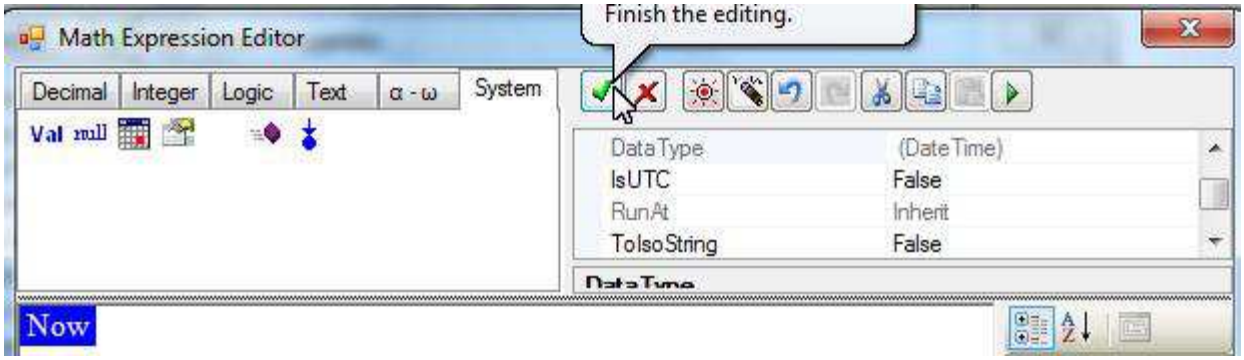
Let’s use it to display current date time on a web page. Add a label to the web page. Create an action to set the Text property of the label to “Now”.



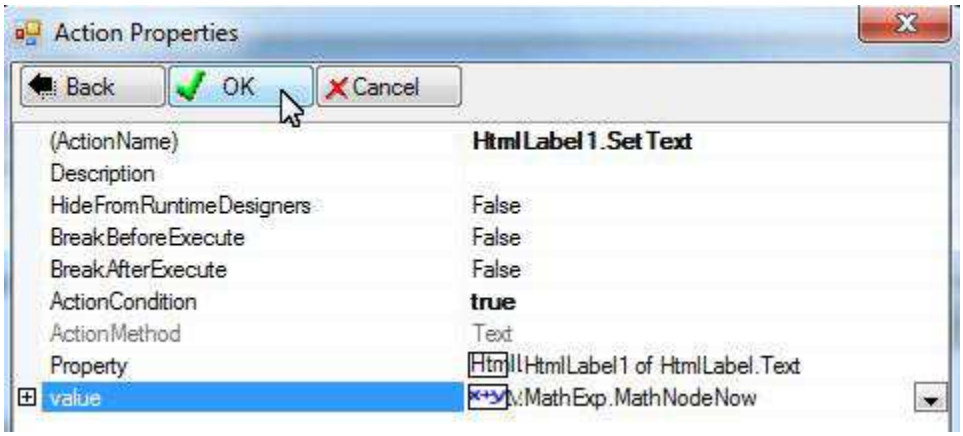




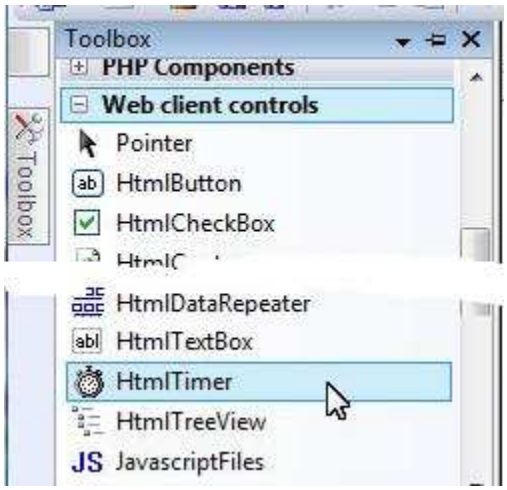
"IsUTC" specifies whether UTC should be used. If it is True then UTC time is used.



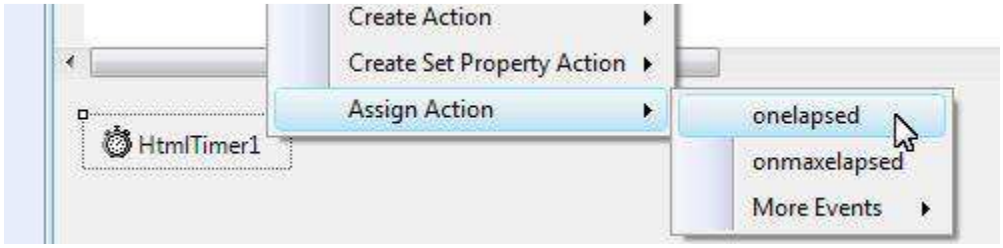
Click OK to finish creating this action.

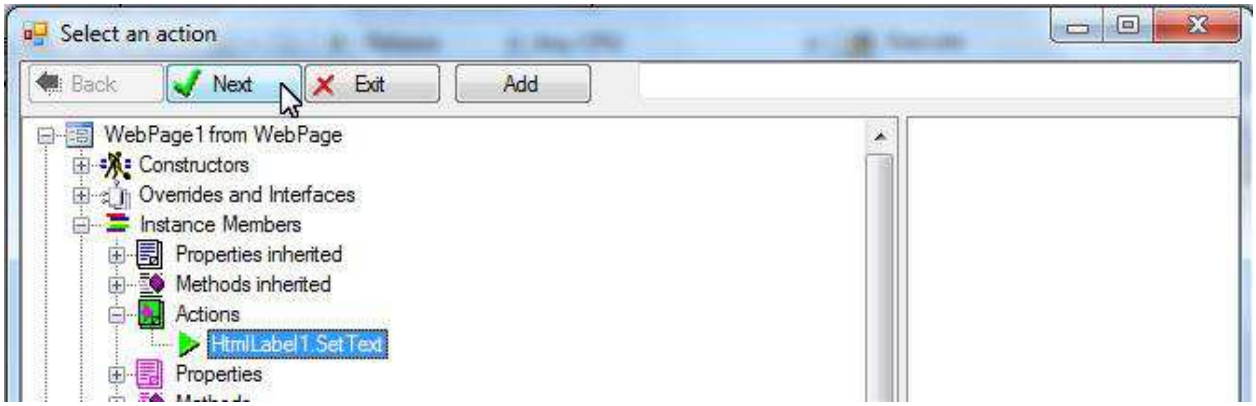


We may use a Timer to execute the above action in every second.



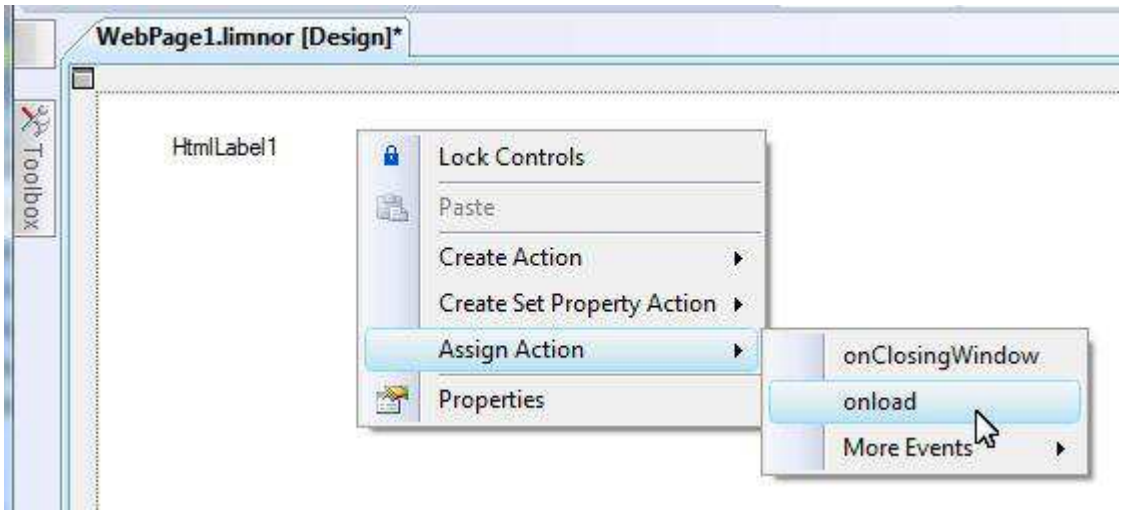
Assign the above action to the Timer:

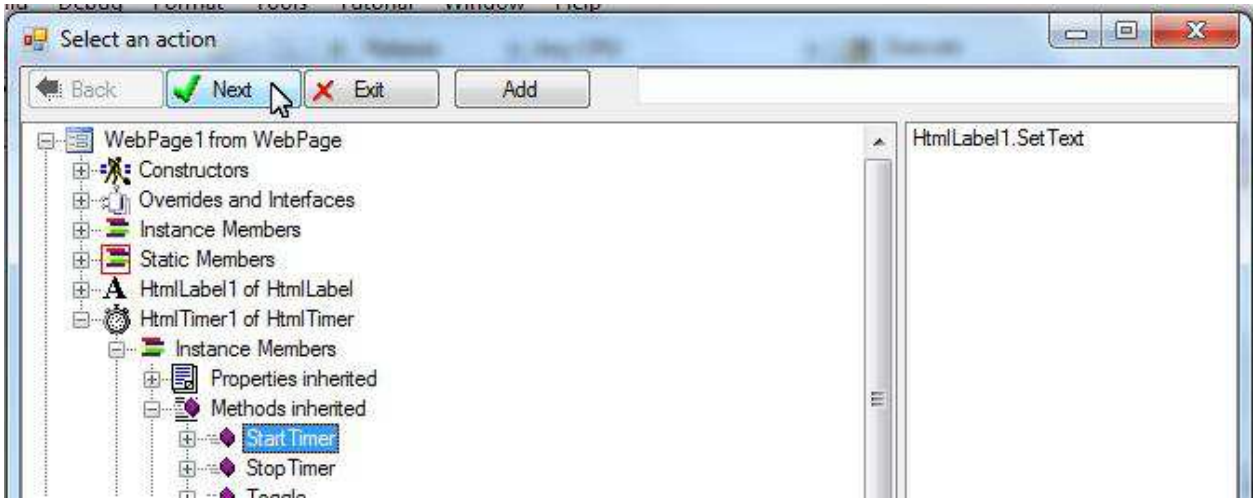




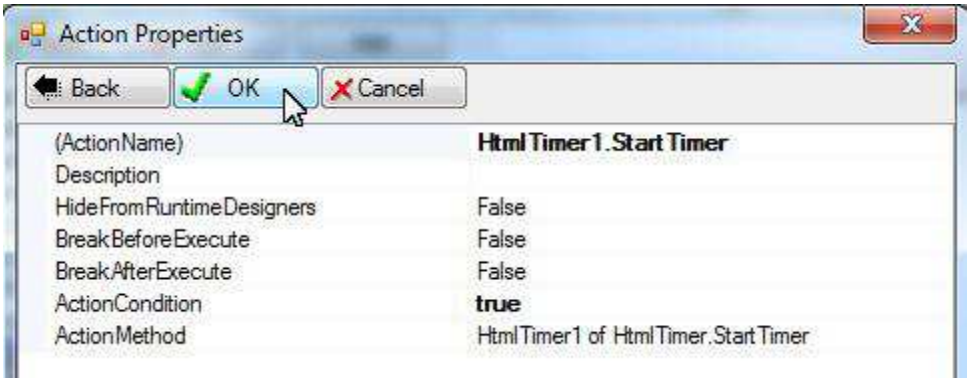
By default the timer interval is 1 second.

Start the Timer when the web page is loaded:

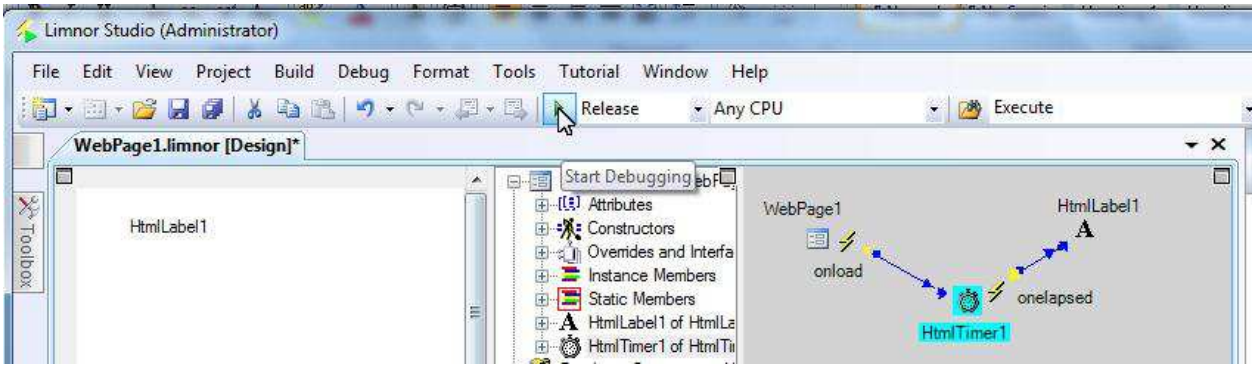




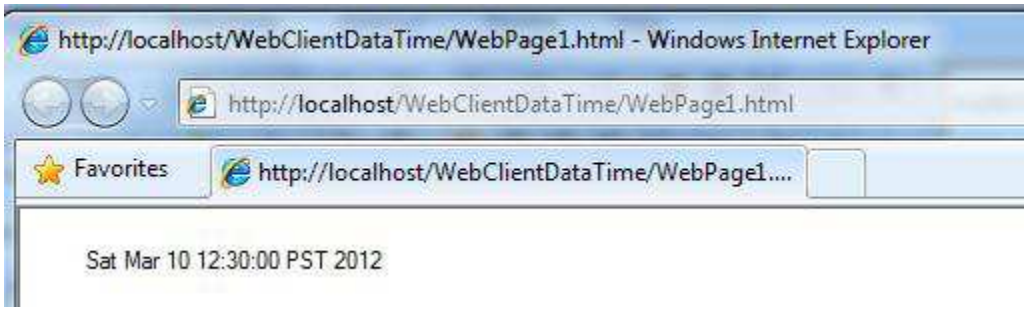
Click OK and the action is created and assigned to the onload event of the web page:



We may test the web page now:



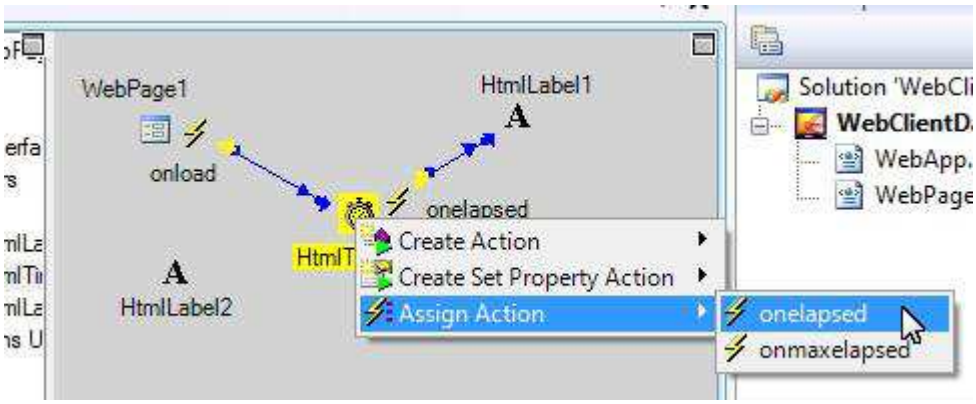
The web page appears. It shows current date time every second.



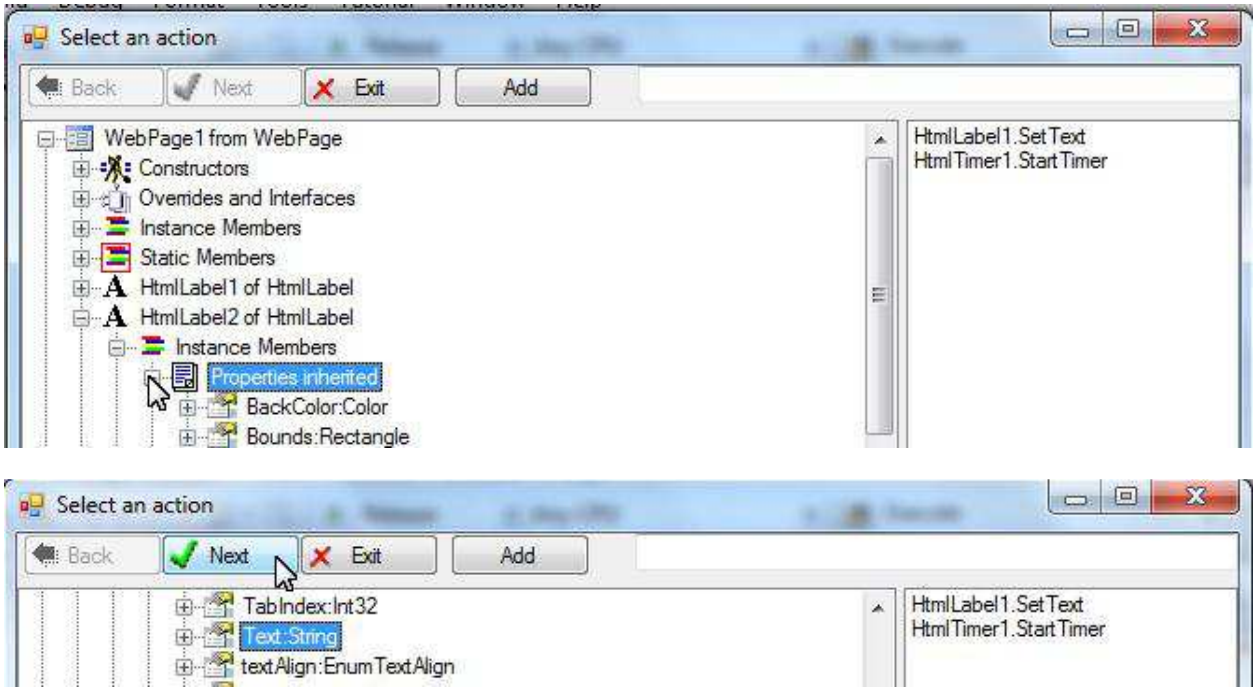
"Now" property of DateTime

Class DateTime has a static property "Now". We may also use it to get current date time. It also allows us to do date time calculations.

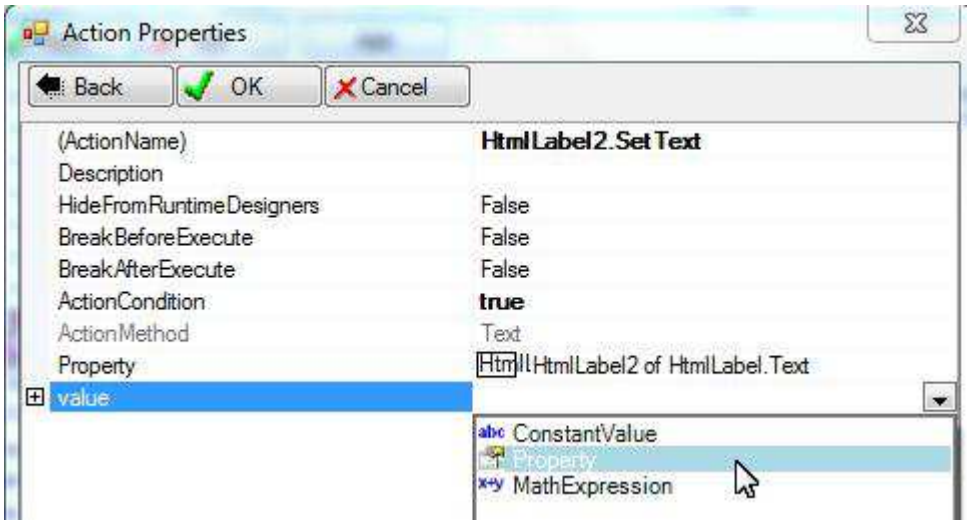
Let's use another label to show "Now" property by creating an action and assigning the action to the Timer:

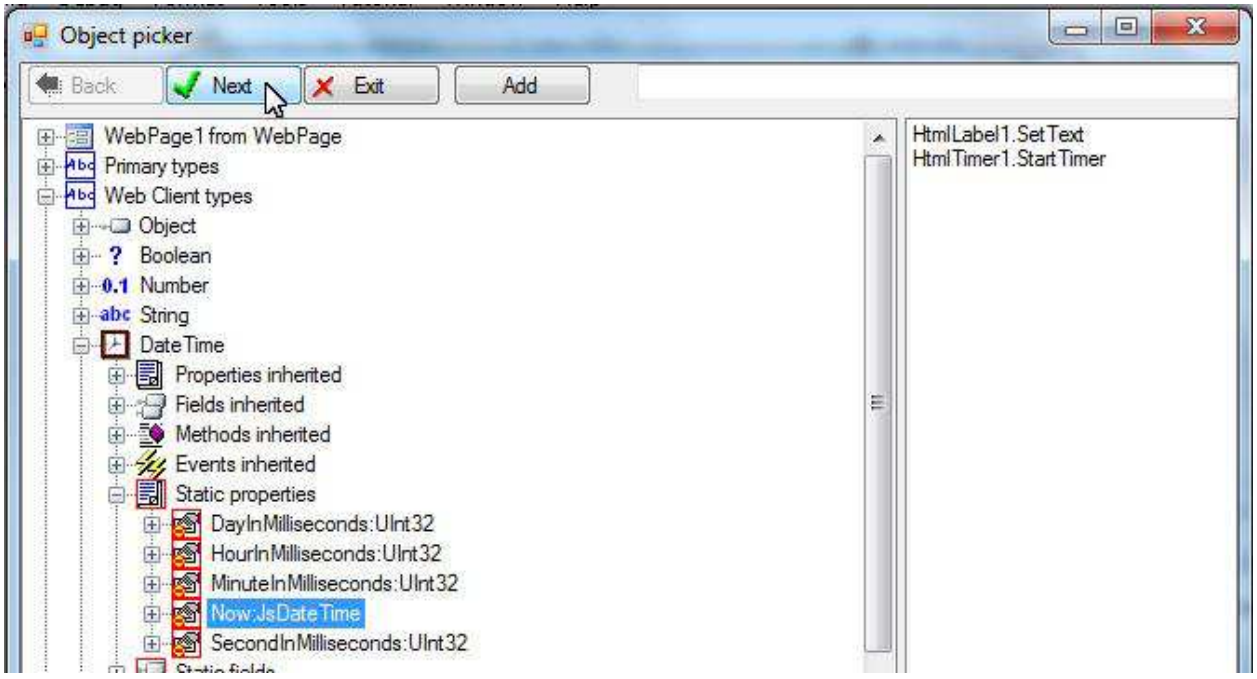


Select the Text property of the second label:

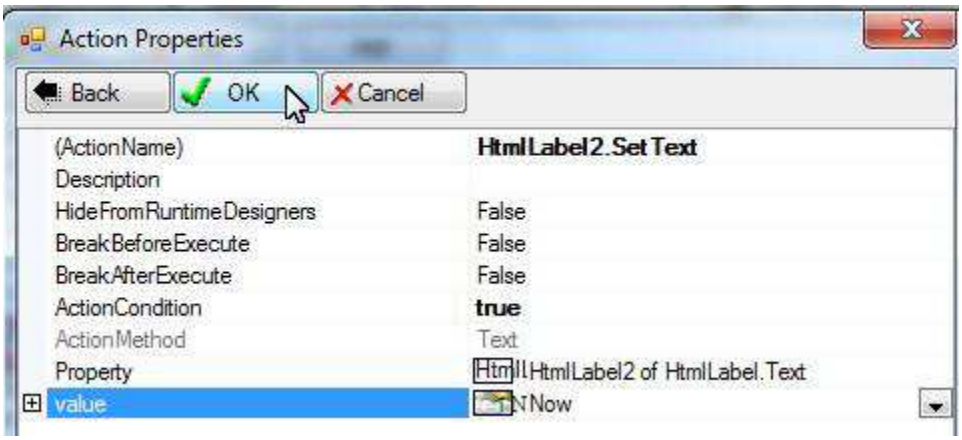


Select the “Now” property of the DateTime:

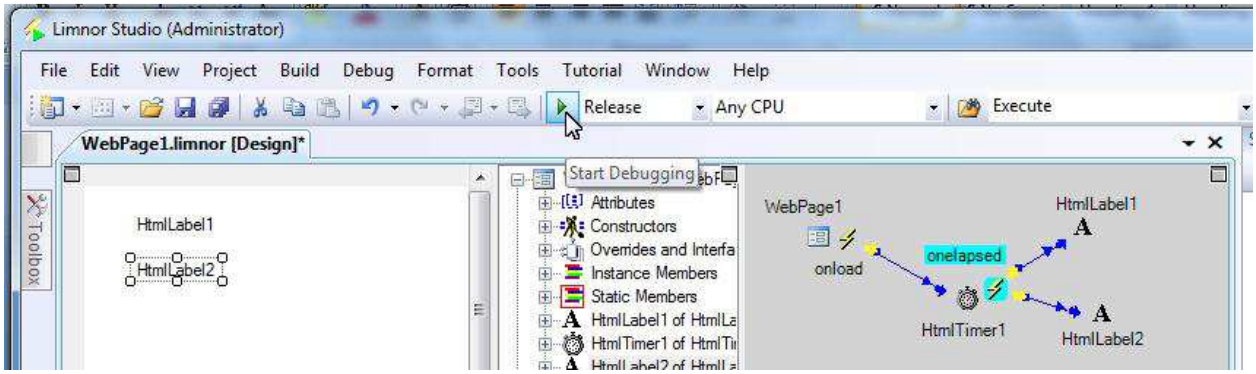




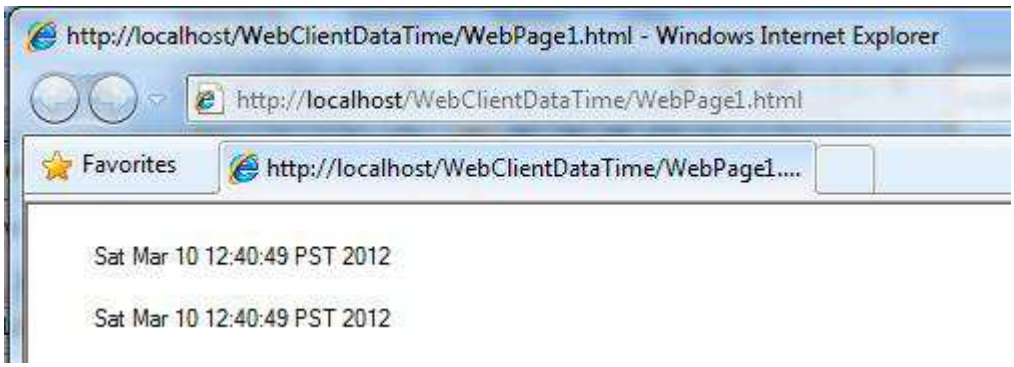
Click OK to finish creating the action and assigning it to the Timer:



We may test the web page:



The web page appears. Both labels show current date time and keep changing in every second.

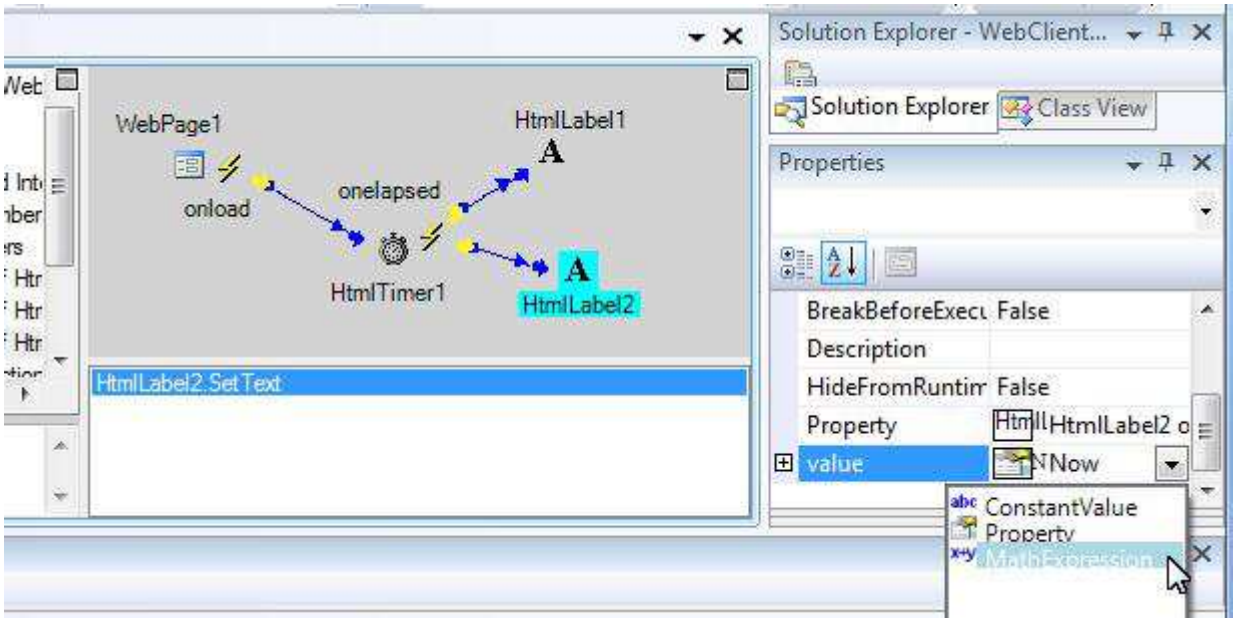


Date Time Calculations

Increase/Decrease

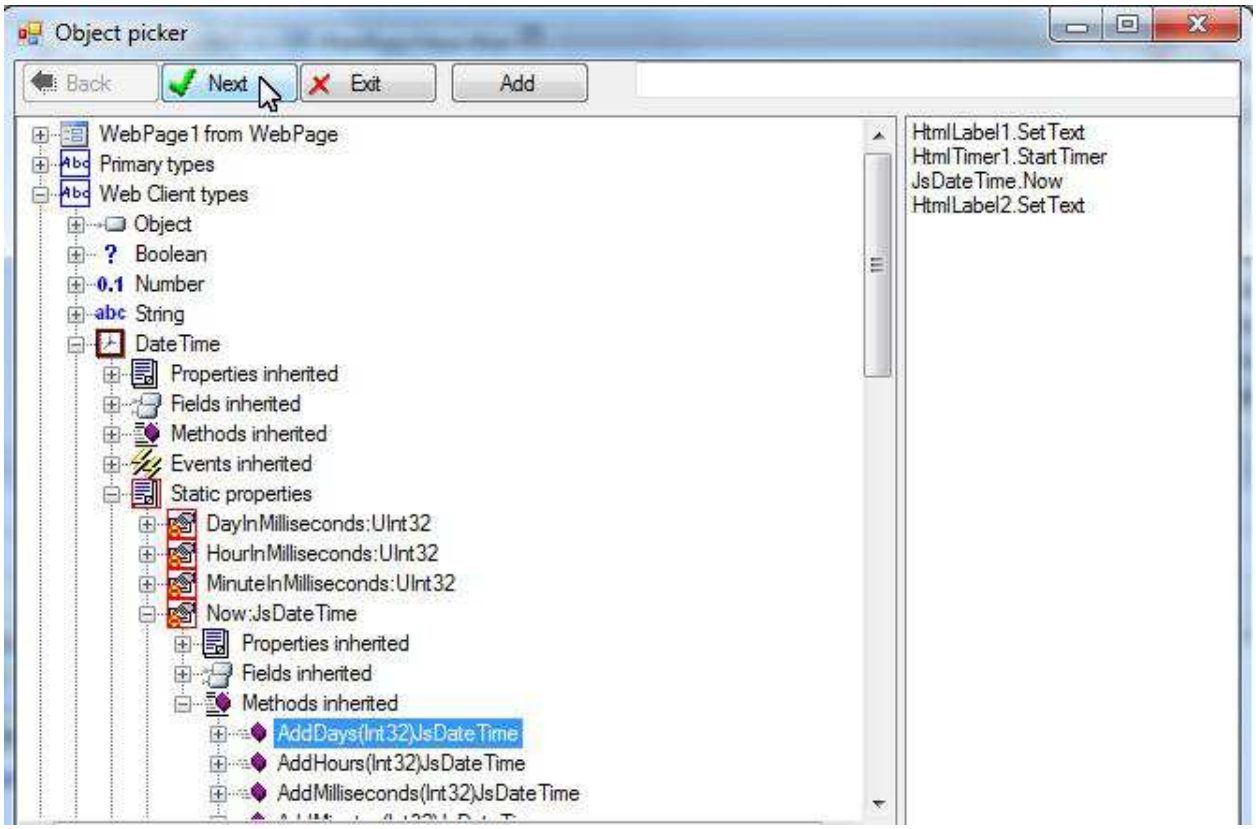
A date time value can be modified by adding a certain time span to it. The time span can be negative.

Let's use an example to show it. Suppose we want to change the previous action to show current date time plus 2 days.

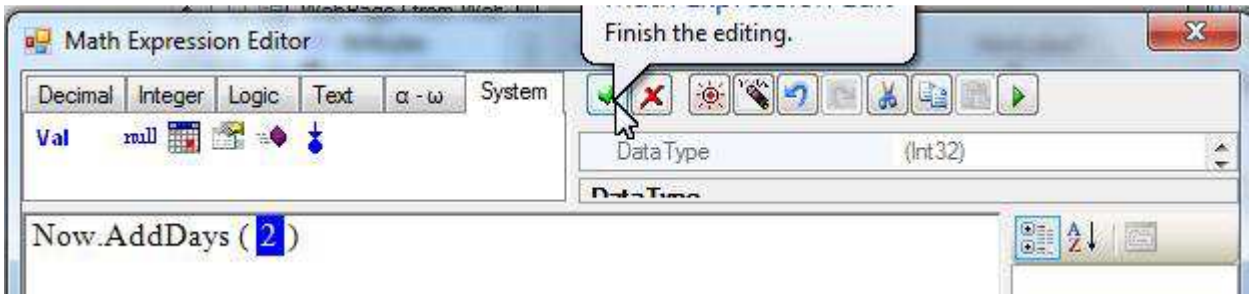


Select the Method icon to do the date time calculation:

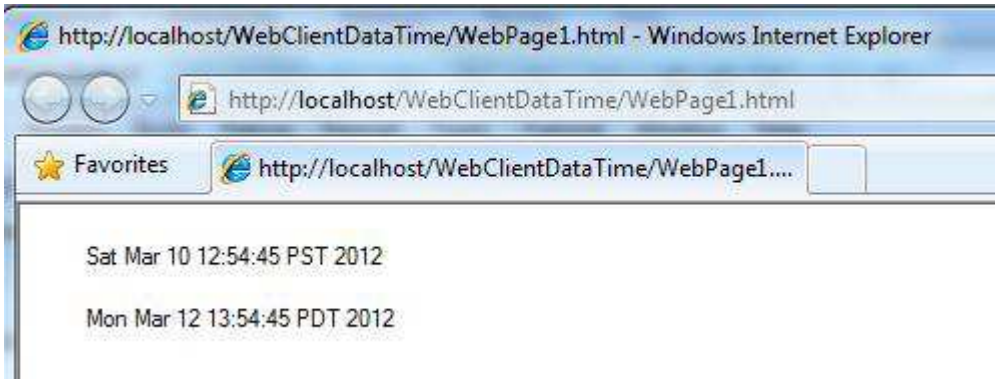
Select AddDays under “Now” property:



Set the parameter to 2 for adding 2 days:



Test the web page again. This time the date on the second label is 2 days ahead of the first label.



Methods and Properties of a DateTime

The above example shows the use of AddDays. A DateTime has following methods and properties for the purpose of date time calculations.

Methods

Instance methods:

getTimespan – Returns a timespan object representing the time difference between this date and the specified start date. See chapter “Timespan” for using timespan.

SetToNow -- Update the value to current date time.

format(string mask) -- Returns a string representation of the date using the specified mask. See http://blog.stevenlevithan.com/archives/date-time-format for a description of the masks.

int **DifferenceInMilliseconds**(DateTime end) -- Returns milliseconds starts from this date-time to the date-time specified by parameter 'end'.

int **DifferenceInSeconds**(DateTime end) -- Returns seconds starts from this date-time to the date-time specified by parameter 'end'.

int **DifferenceInMinutes**(DateTime end) -- Returns minutes starts from this date-time to the date-time specified by parameter 'end'.

int **DifferenceInHours**(DateTime end) -- Returns hours starts from this date-time to the date-time specified by parameter 'end'.

int **DifferenceInDays**(DateTime end) -- Returns days starts from this date-time to the date-time specified by parameter 'end'.

The following date time modification methods modify the date time object and also return modified object. Thus these methods can be used in math expressions and be used to assign values to other objects.

DateTime **AddMilliseconds**(int milliseconds) -- Returns a new date-time object by adding milliseconds to this object.

DateTime **AddSeconds**(int seconds) -- Returns a new date-time object by adding seconds to this object.

DateTime **AddMinutes**(int minutes) -- Returns a new date-time object by adding minutes to this object.

DateTime **AddHours**(int hours) -- Returns a new date-time object by adding hours to this object.

DateTime **AddDays**(int days) -- Returns a new date-time object by adding days to this object.

DateTime **SetValue**(int year, int month, int day, int hours, int minutes, int seconds, int milliseconds) -- set this object to the specified date time.

DateTime **SetUTCValue**(int year, int month, int day, int hours, int minutes, int seconds, int milliseconds) -- set this object to the specified UTC date time.

DateTime **SetDayOfMonth**(int day)

DateTime **SetMonth**(int month)

DateTime **SetYear**(int year)

DateTime **SetHours**(int hours)

DateTime **SetMinutes**(int minutes)

DateTime **SetSeconds**(int seconds)

DateTime **SetMilliseconds**(int milliseconds)

DateTime **SetWholeTime**(int wholeTime) -- Sets the date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970.

DateTime **SetUTCDayOfMonth**(int day)

DateTime **SetUTCMonth**(int month)

DateTime **SetUTCYear**(int year)

DateTime **SetUTCHours**(int hours)

DateTime **SetUTCMinutes**(int minutes)

DateTime **SetUTCSeconds**(int seconds)

DateTime **SetUTCMilliseconds**(int milliseconds)

Static methods:

DateTime **ParseIso**(string datetime)

It converts a string in format 'yyyy-mm-dd hh:mm:ss.mmm' into a date time object, in the browser's locale.

DateTime **ParseIsoUTC**(string datetime)

It converts a string in format 'yyyy-mm-dd hh:mm:ss.mmm' into a date time object, in UTC.

Properties

Instance properties

int **DayOfMonth**

int **DayOfWeek**

int **Year**

int **Month** -- note that the first month is 0, not 1. The last month is 11, not 12.

int **Hours**

int **Minutes**

int **Seconds**

int **Milliseconds**

int **UTCDayOfMonth**

int **UTCDayOfWeek**

int **UTCYear**

int **UTCMonth** -- note that the first month is 0, not 1. The last month is 11, not 12.

int **UTCHours**

int **UTCMinutes**

int **UTCSeconds**

int **UTCMilliseconds**

int **TimezoneOffset**

int **WholeTime** -- Gets the number of milliseconds since midnight Jan 1, 1970

int **UTCWholeTime** -- Gets the number of milliseconds since midnight Jan 1, 1970, according to universal time.

string **AsDateString** -- Converts the date portion of a Date object into a readable string.

string **AsLocaleDateString** -- Converts the date portion of a Date object into a readable string, using locale conventions")]

string **AsTimeString** -- Converts the time portion of a Date object into a readable string.

string **AsLocaleTimeString** -- Converts the time portion of a Date object into a readable string, using locale conventions")]

string **AsString** -- Converts the Date object to a string")]

string **AsLocaleString** -- Converts the Date object to a string, using locale conventions")]

string **AsUTCString** -- Converts the Date object to a string, according to universal time")]

Static properties

SecondInMilliseconds -- Gets milliseconds in one second. It is a constant 1000.

MinuteInMilliseconds -- Gets milliseconds in one minute. It is a constant 60000.

HourInMilliseconds -- Gets milliseconds in one hour. It is a constant 3600000.

DayInMilliseconds -- Gets milliseconds in one day. It is a constant 86400000.

DateTime **Now** -- Gets the current date time

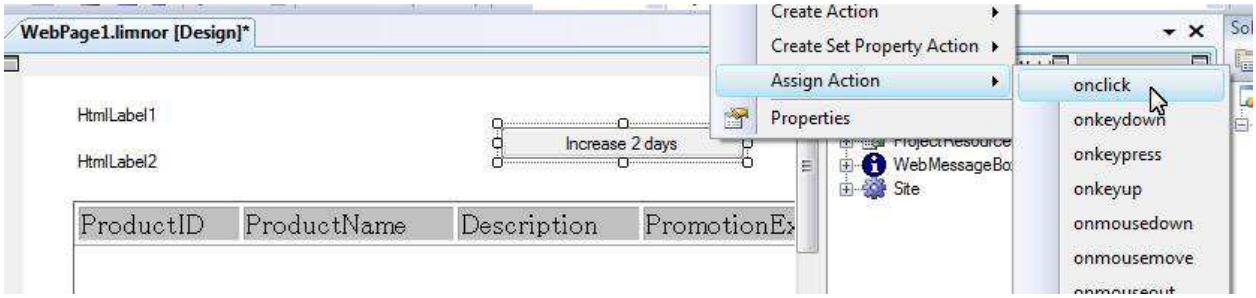
Date time calculations for database fields

If a database field is of type DateTime then we may use the methods and properties listed above to do calculations.

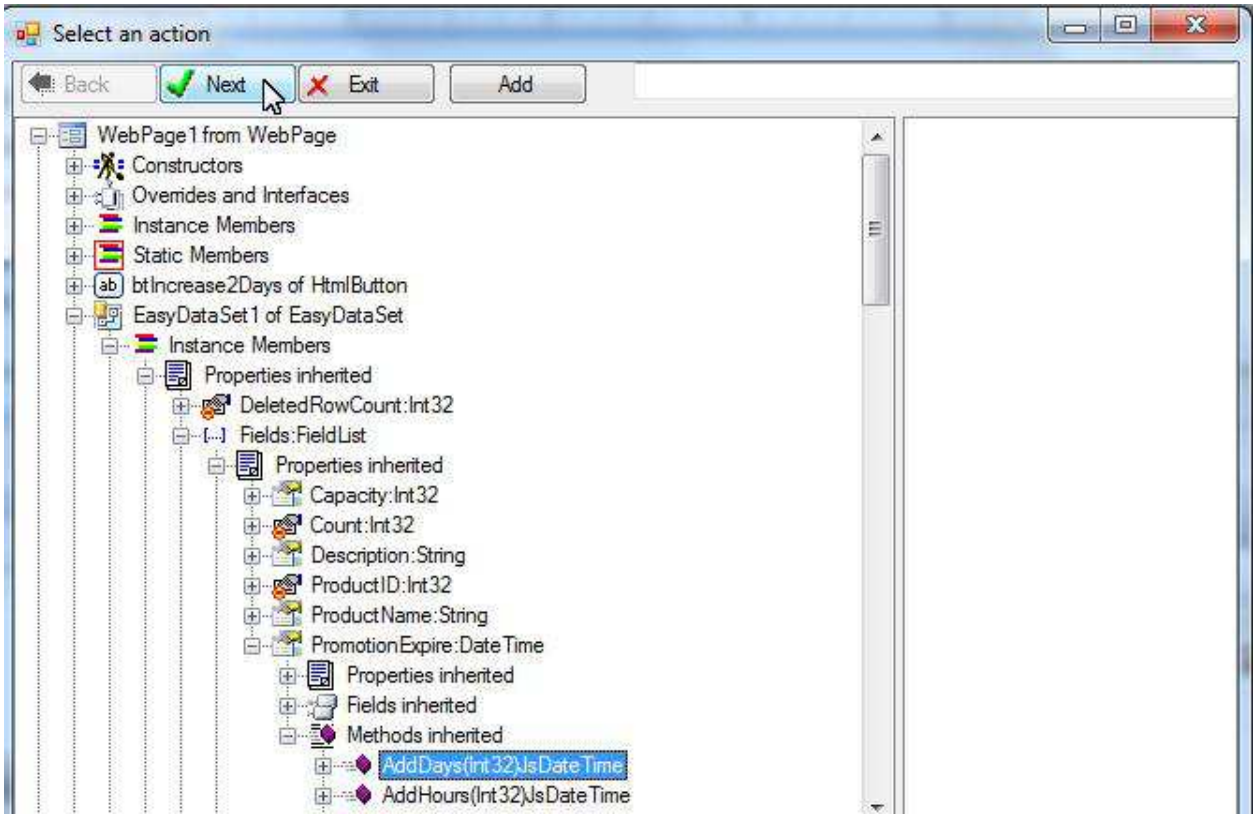
Suppose we want to set a date time field value by adding 2 days to the current field value. If the current field value is null then we set the field value by adding 2 days to the current date time.

We use an EasyDataSet to get data from database and use an HtmlTable to show the data. For web database programming, see http://www.limnor.com/support/webDatabaseProgramming.pdf

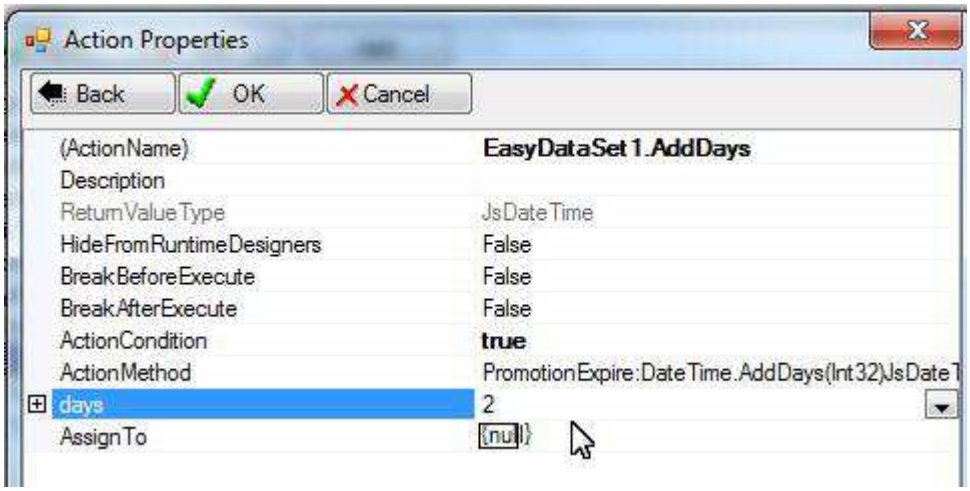
We use a button to carry out the field-value-setting actions.



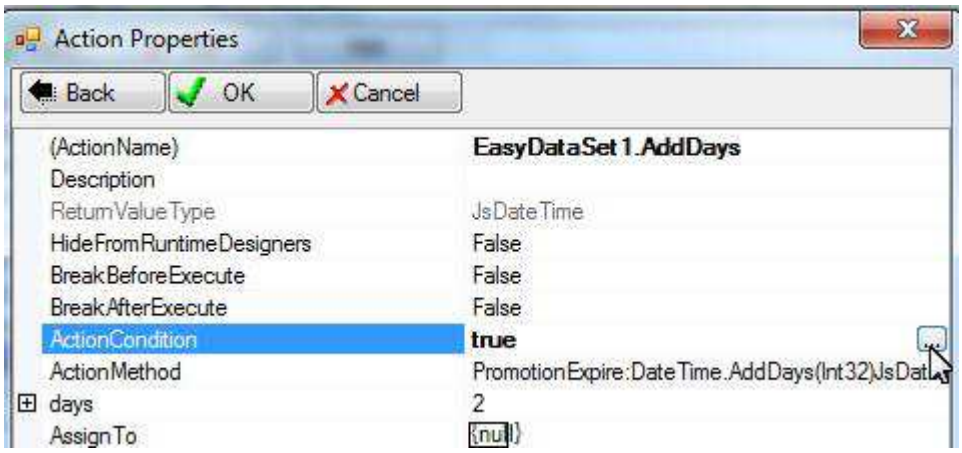
The field, PromotionExpire, is the field we want to set.

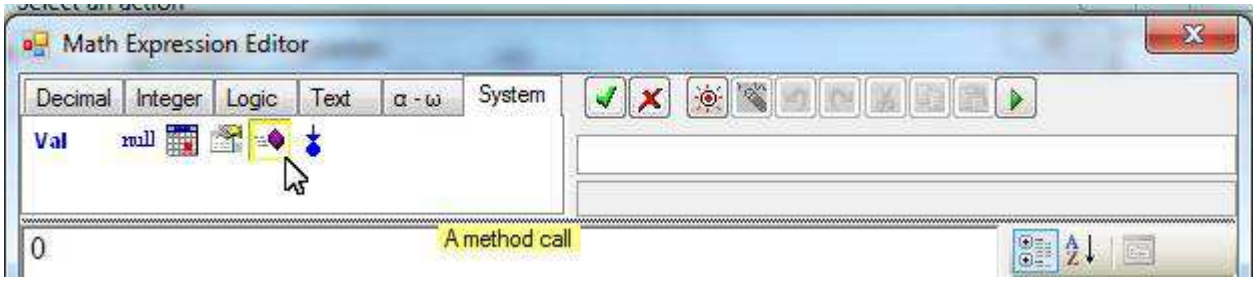


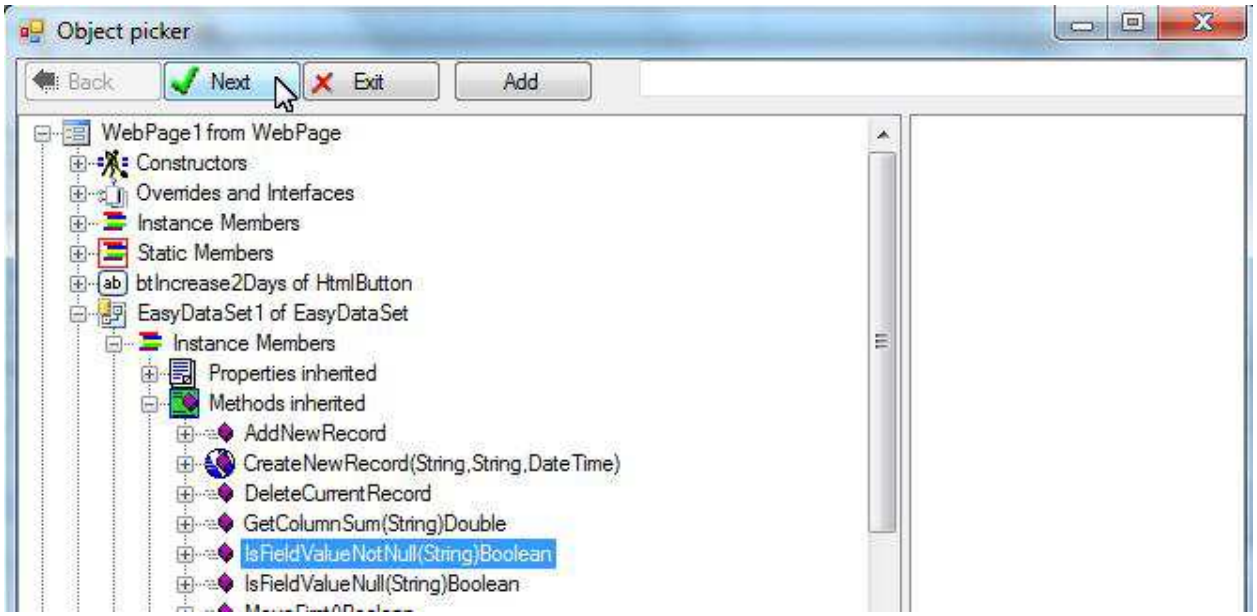
Set “days” to 2:



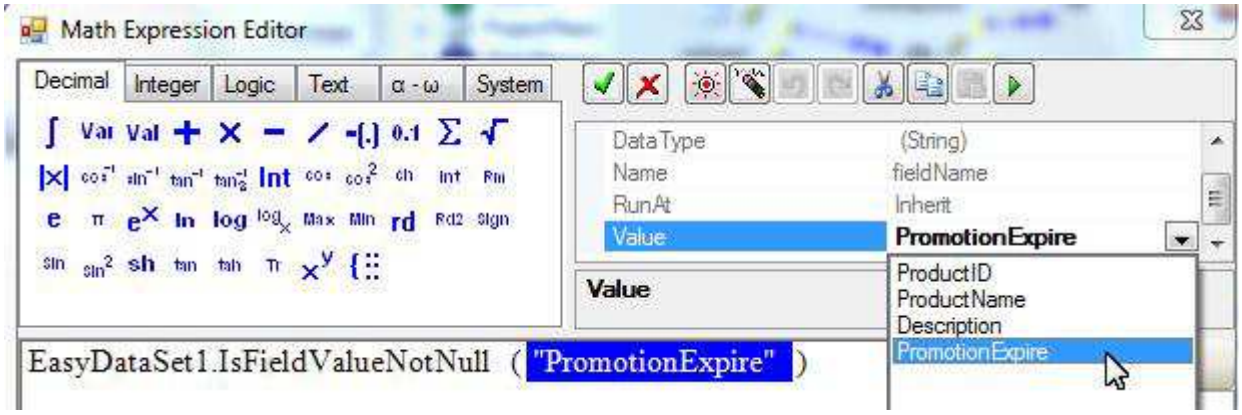
Set ActionCondition to that the field value is not null:



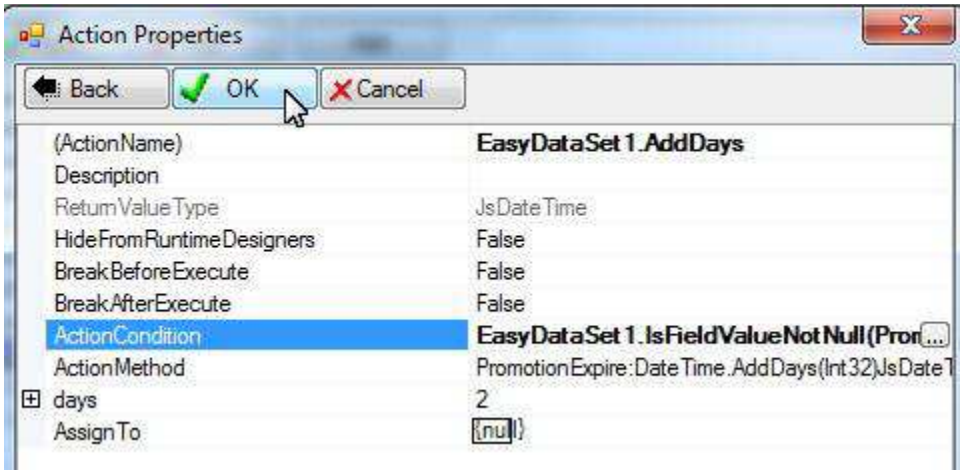




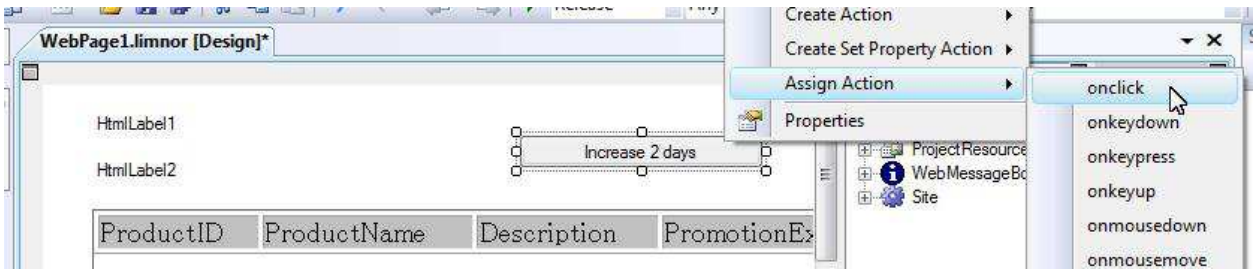
Enter the field name:



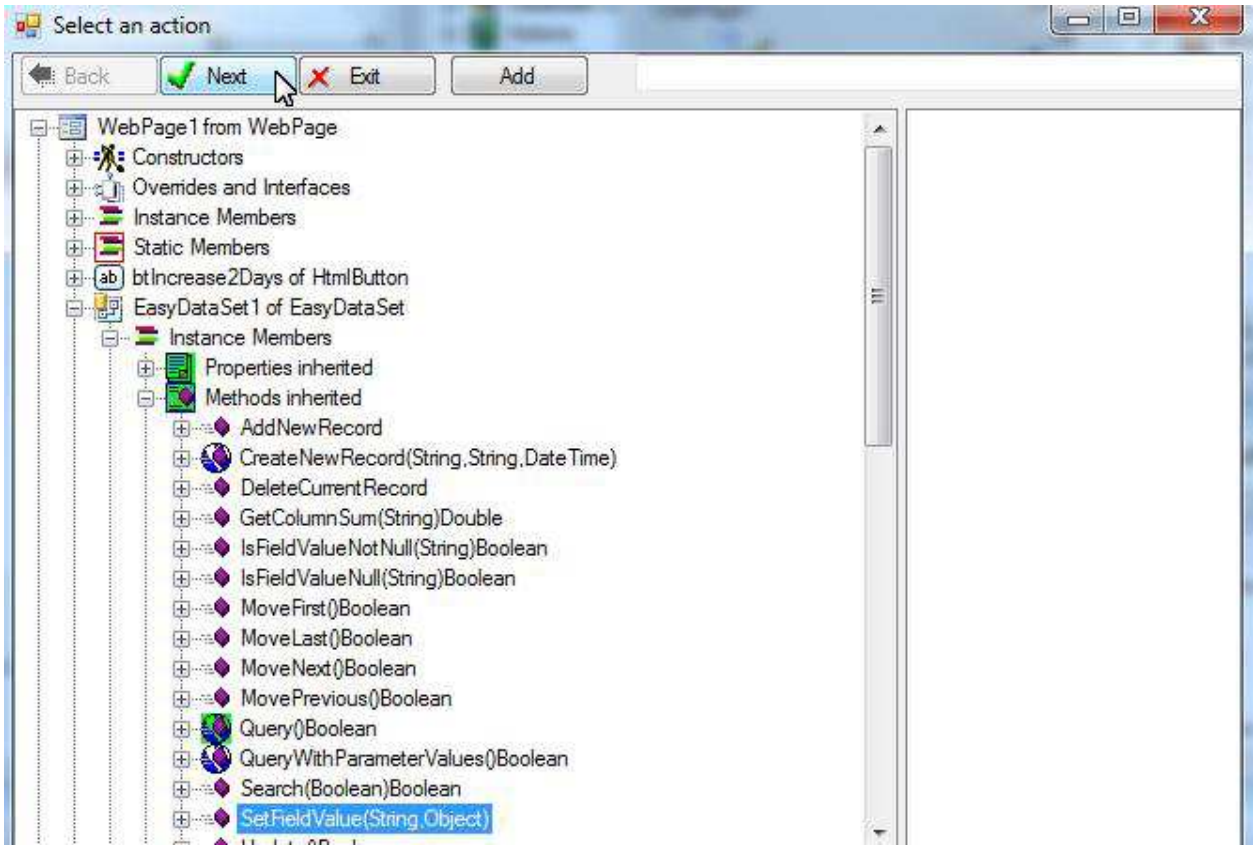
Click OK to finish creating this action and assign it to the button.



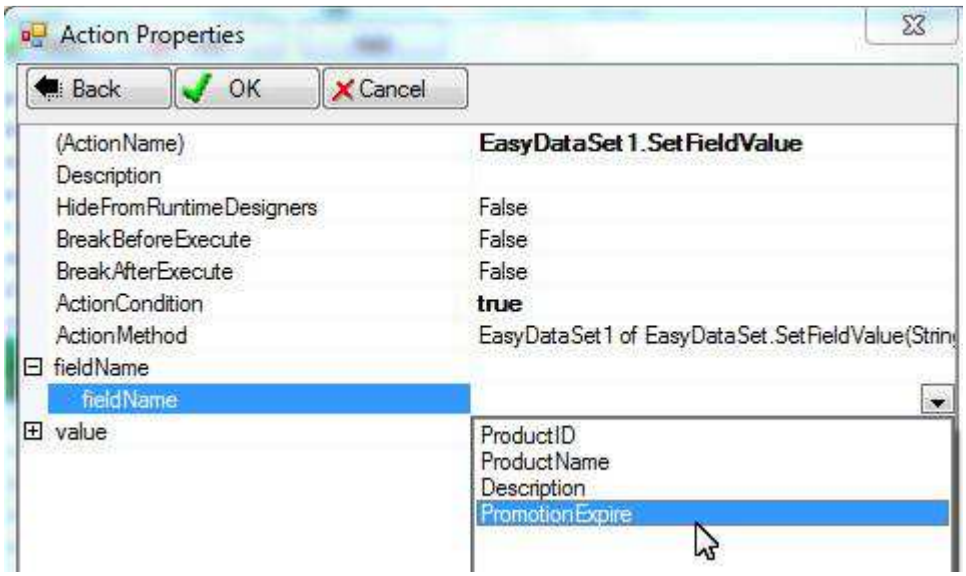
Add another action to the button to set the field value to the current date time plus 2 days if the field value is null.



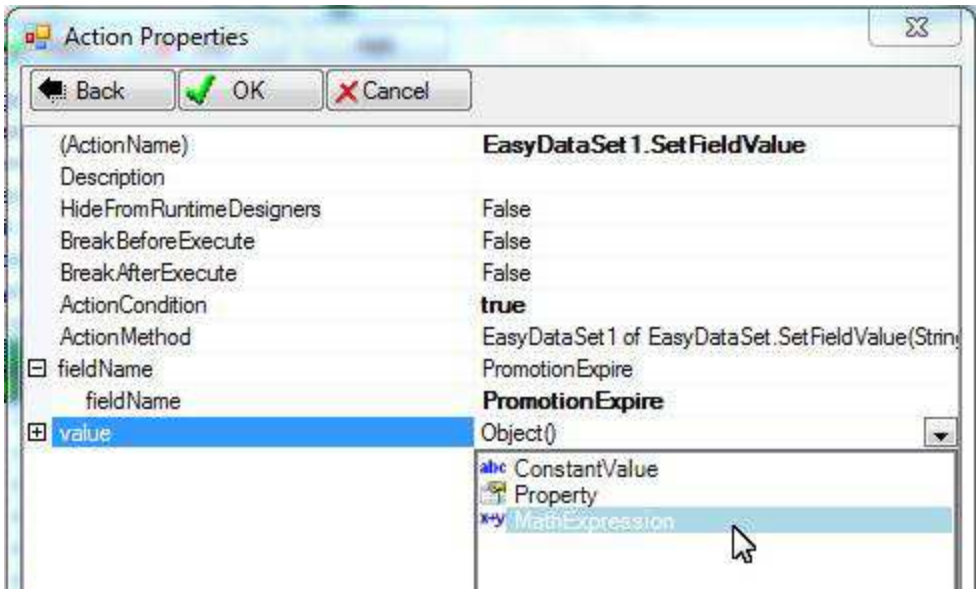
Select the SetFieldValue method of the EasyDataSet:

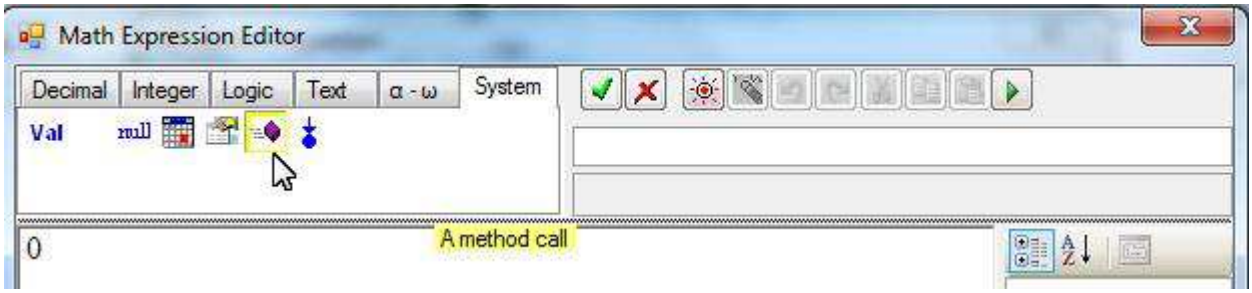


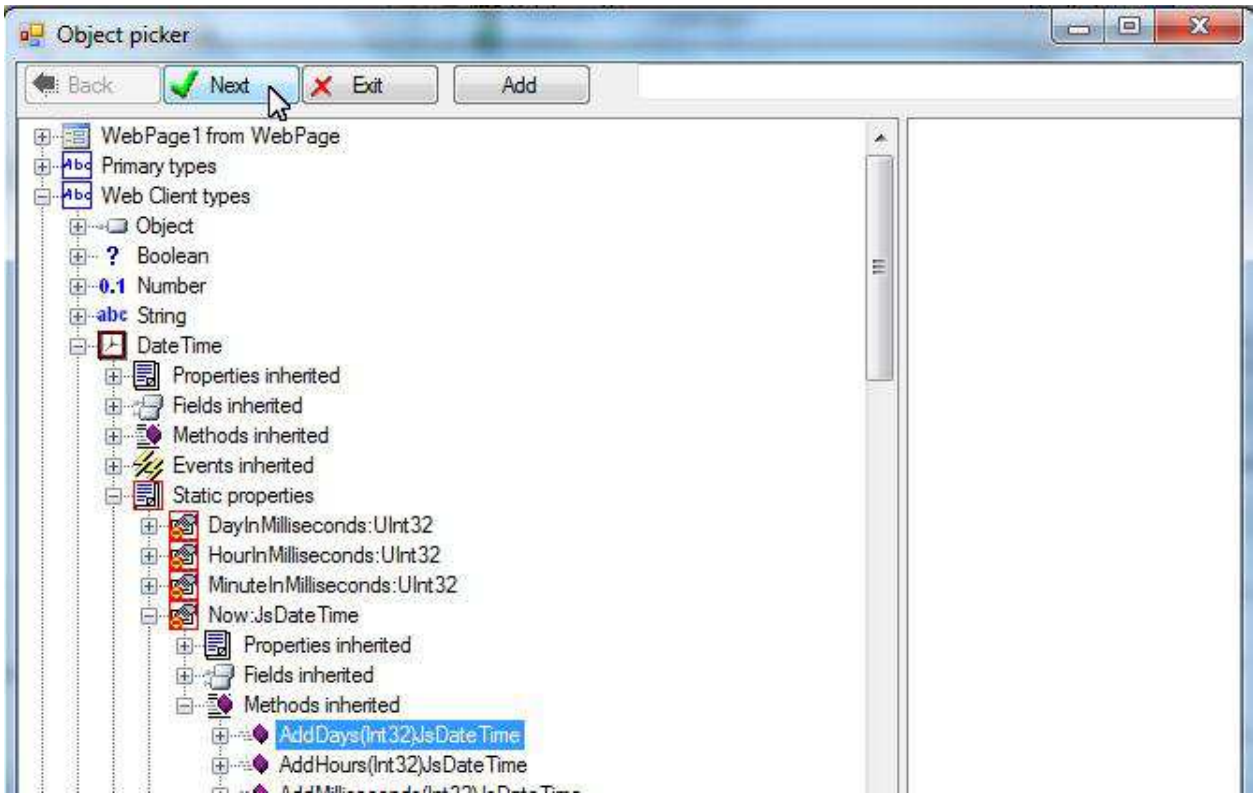
Select the field name:



For “value”, select AddDays method under “Now”:

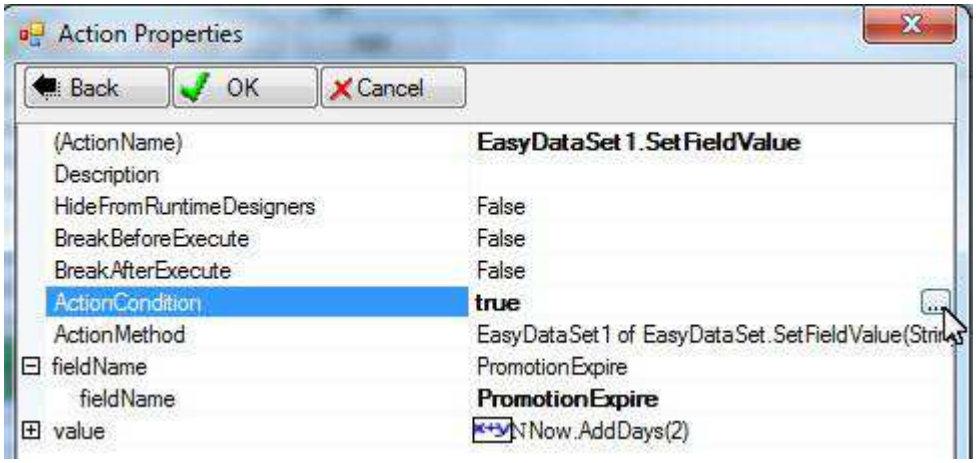

Action Properties
Back
OK
Cancel
(ActionName)
EasyDataSet1.SetFieldValue
Description
HideFromRuntimeDesigners
False
BreakBeforeExecute
False
BreakAfterExecute
False
ActionCondition
true
ActionMethod
EasyDataSet1 of EasyDataSet.SetFieldValue(Strin
fieldName
PromotionExpire
fieldName
PromotionExpire
value
Object()
ConstantValue
Property
MathExpression

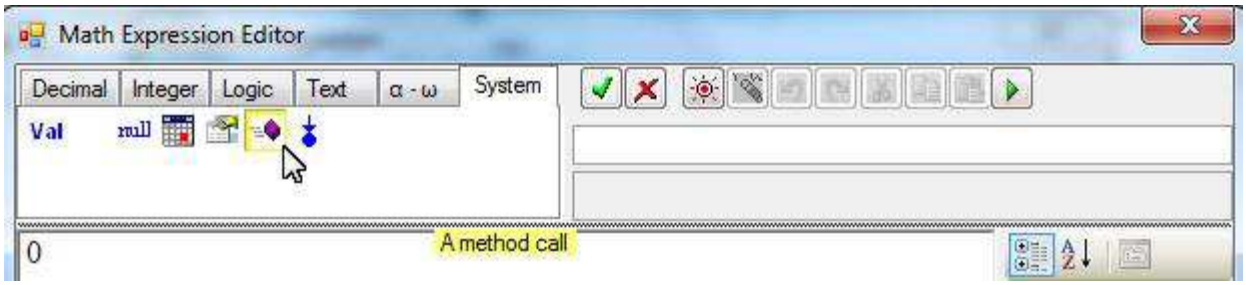

Math Expression Editor
Decimal
Integer
Logic
Text
α - ω
System
Val
null
A method call
0

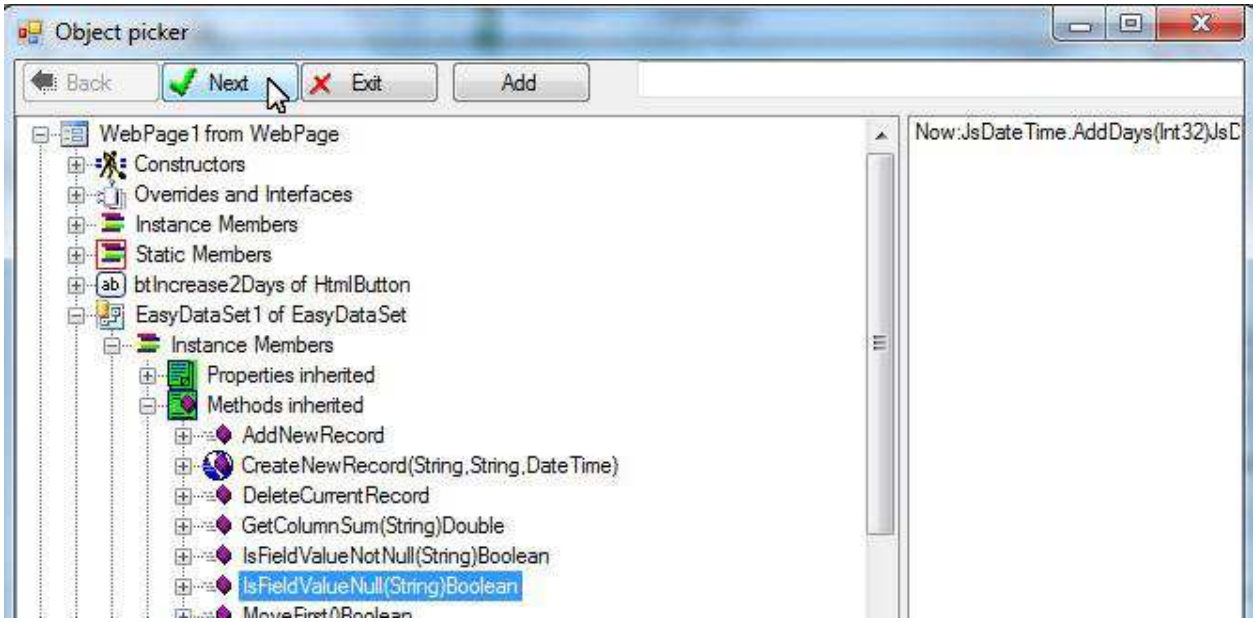

Object picker
Back
Next
Exit
Add
WebPage1 from WebPage
Primary types
Web Client types
Object
Boolean
Number
String
DateTime
Properties inherited
Fields inherited
Methods inherited
Events inherited
Static properties
DayInMilliseconds:UInt32
HourInMilliseconds:UInt32
MinuteInMilliseconds:UInt32
Now:JsDateTime
Properties inherited
Fields inherited
Methods inherited
AddDays(Int32)JsDateTime
AddHours(Int32)JsDateTime

Enter 2 for the method parameter:

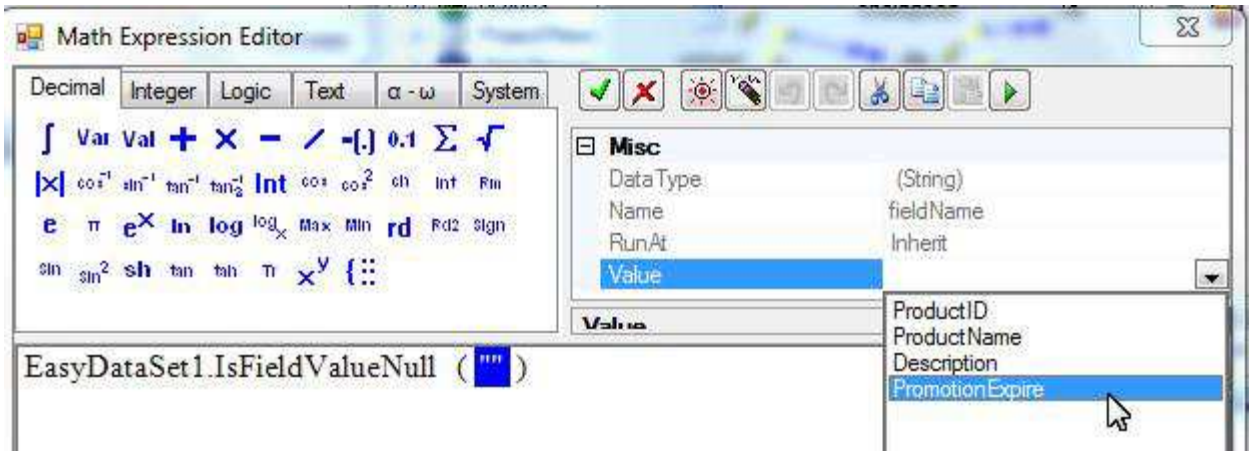
Set the ActionCondition to that the field value is null:



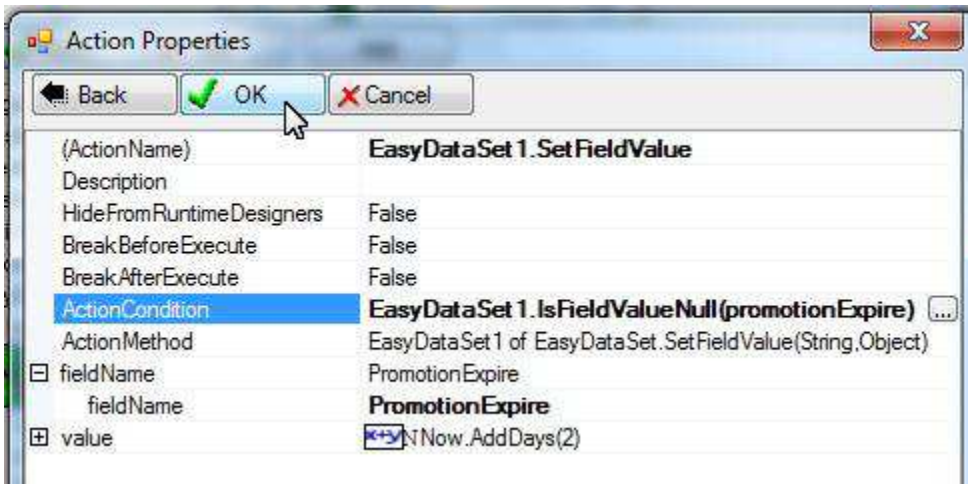




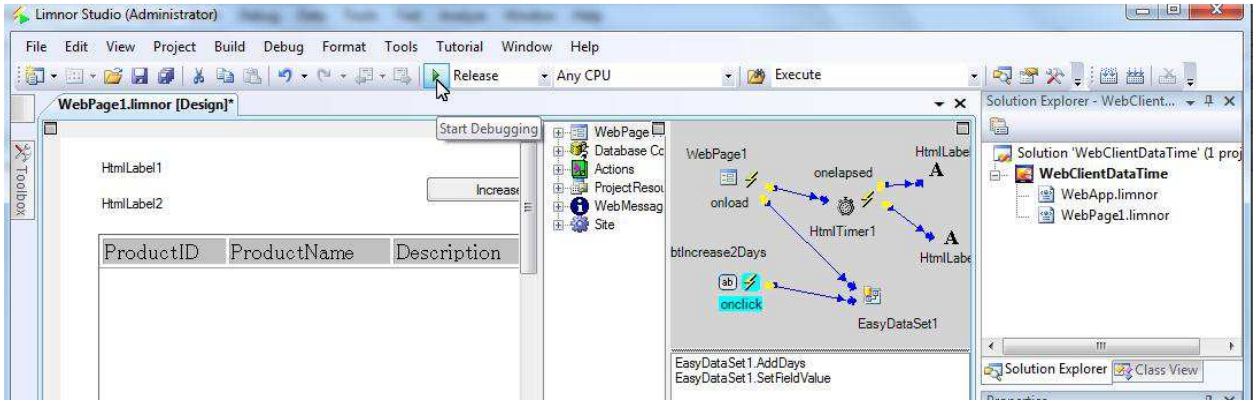
Enter field name:



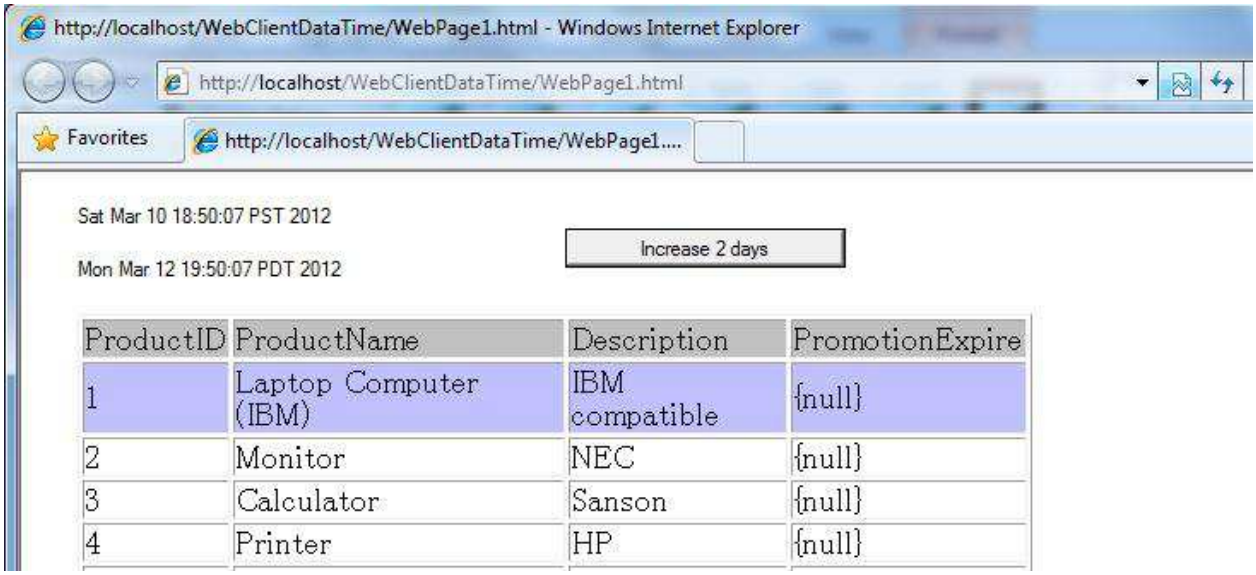
Click OK to finish creating this action and assign it to the button:



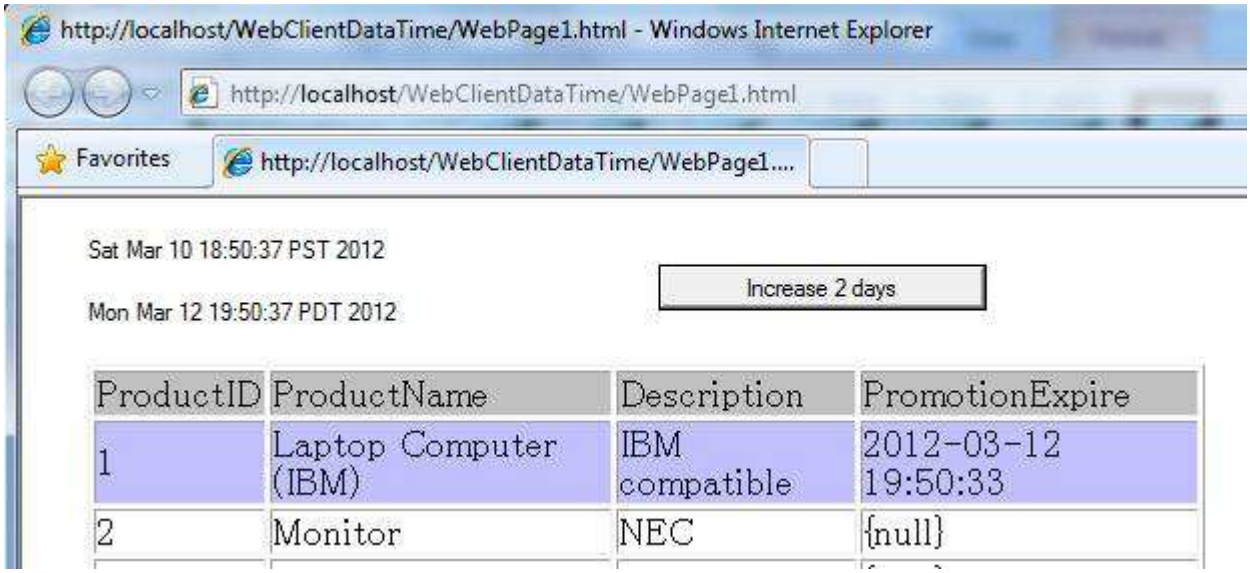
Let's test the web page:



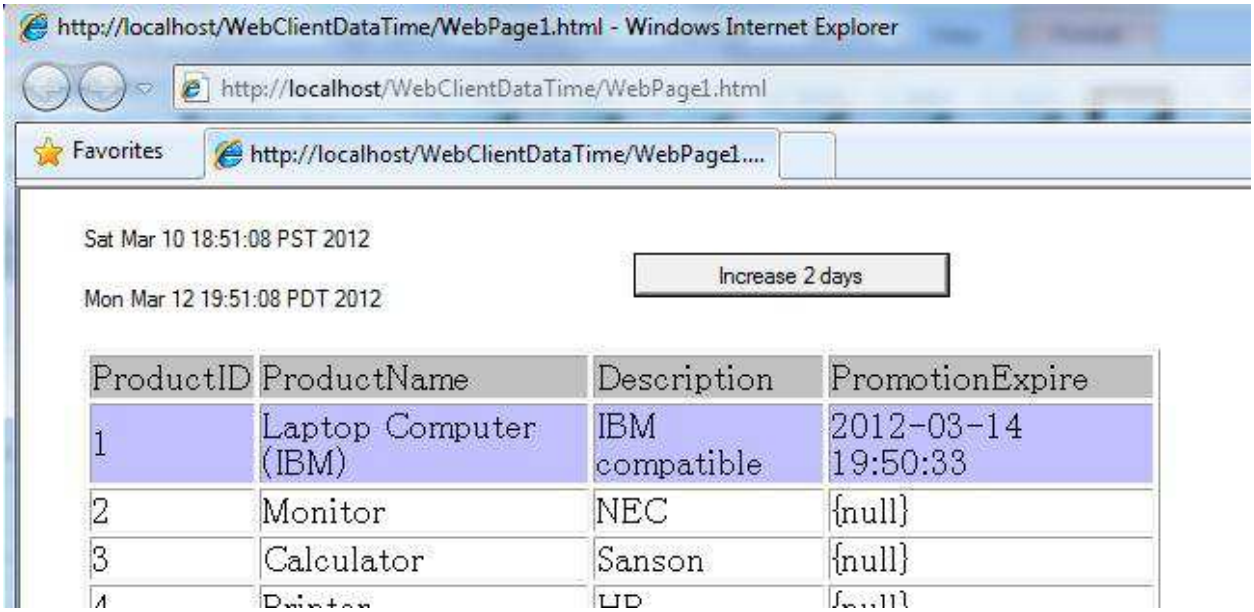
The web page appears. Wait for the data to appear:



Click button “Increase 2 days”. Because the field value is null, we can see that the field value is set to the current date time plus 2 days:



Click button “Increase 2 days” again. This time because the field value is not null anymore, the field value is increased by 2 days:

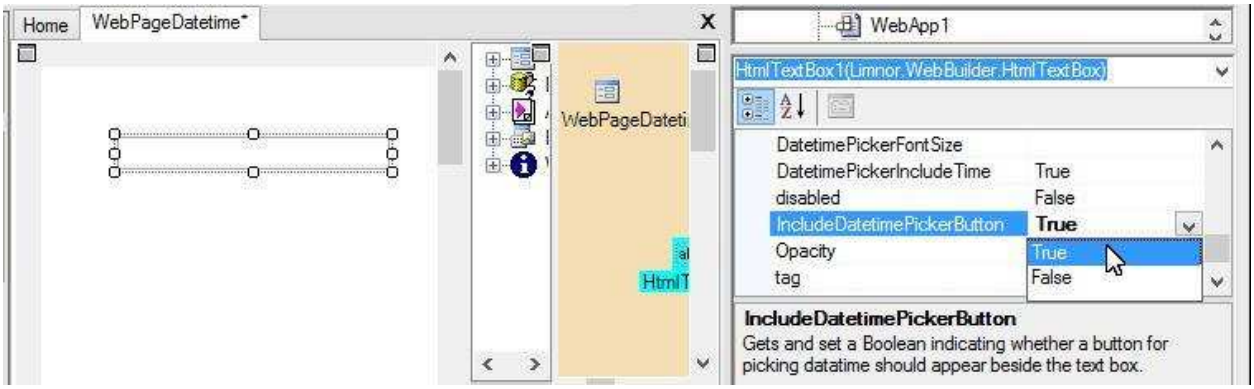


Date-Time Picker

Usually a date-time picker is used for date-time data input. Limnor Studio has some built-in support of date-time picker for your convenience.

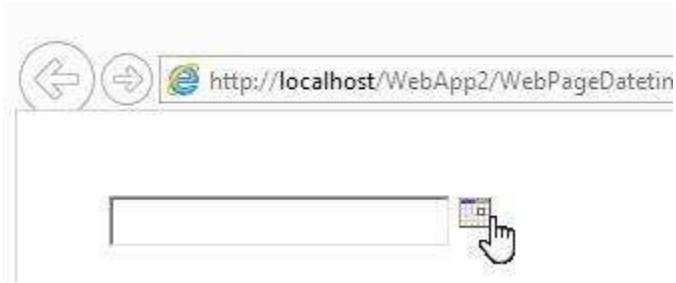
Use date-time picker with text box

When you use a text box for entering a date-time, you may set its IncludeDatetimePicker to True:

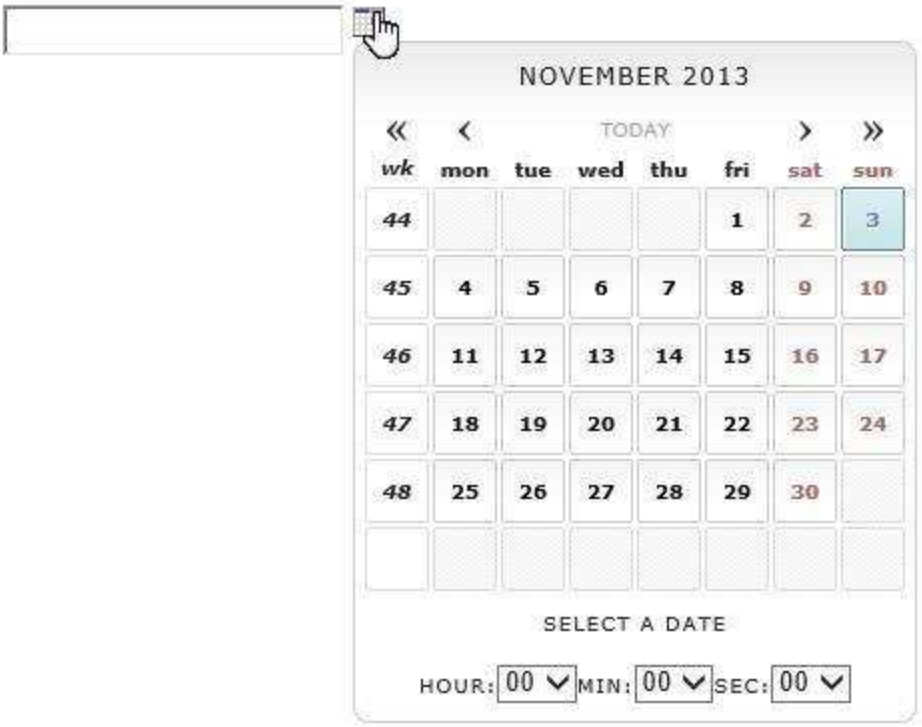


- IncludeDatetimePicker – It is a Boolean indicating whether a button will appear besides the text box for invoking a date-time picker.
- DatetimePickerFontSize – It is a string property indicating the font size. For example, 10px indicates that font size is 10 pixels. Usually 16px is a good size for touch screen usages. Default is 10px.
- DatetimePickerIncludeTime – It is a Boolean indicating whether time portion is needed.

If IncludeDatetimePicker is True then a button appears besides the text box:

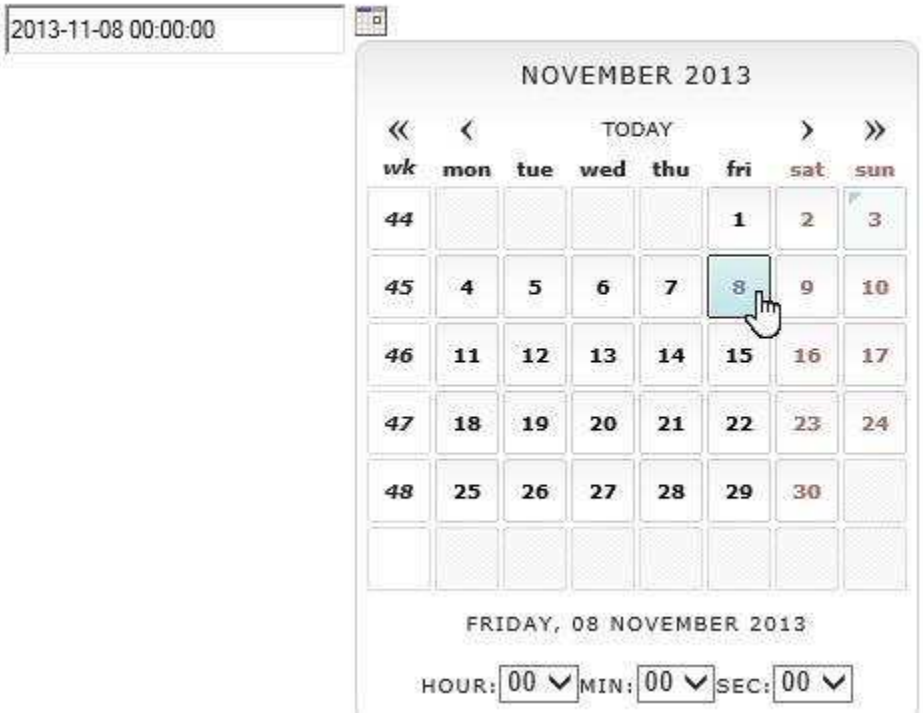


Click the button, a date-time picker appears:



Note that the size of the date-time picker is determined by DatetimePickerFontSize.

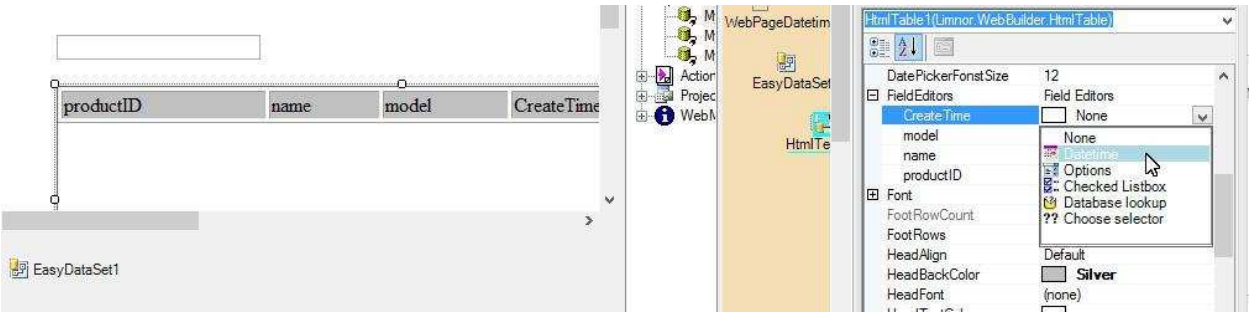
Click a date, the date-time appears in the text box:



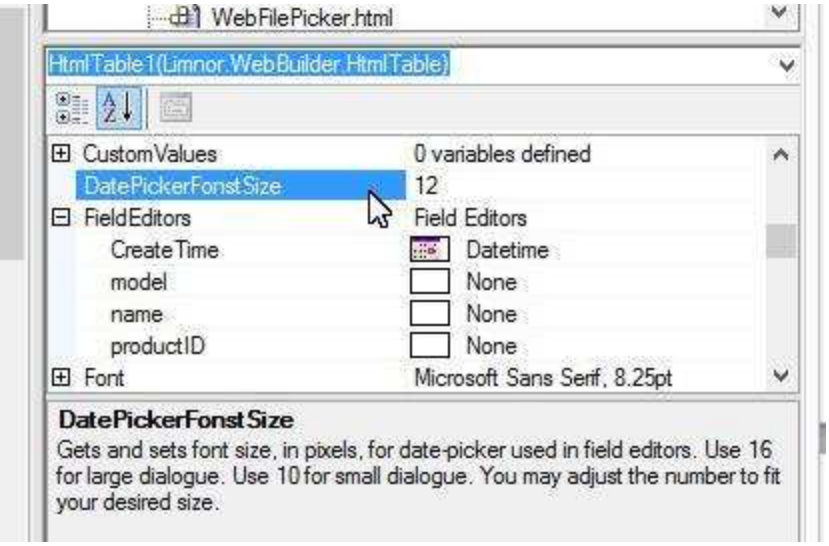
The date-time picker disappears if the user clicks a date or anywhere in the web page.

Use Date-Time picker in Table Cell

If a column is for date-time then you may set FieldEditors property of an HtmlTable to use date-time picker. For more information, see “Field Editors” and “Date Time Selector” in http://www.limnor.com/support/webDatabaseProgramming.pdf

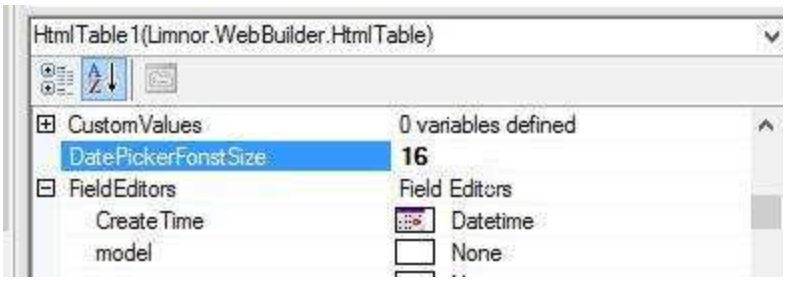


DatePickerFontSize property is an integer indicating the font size, in pixels, of the date-time picker:

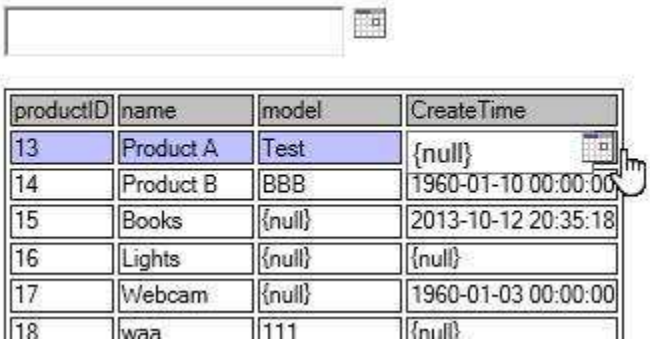


DatePickerFontSize property determines the size of date-time picker. If your web app is for touchscreen then you may try to use 16 for DatePickerFontSize to see if the size of the date-time picker is a good for you.

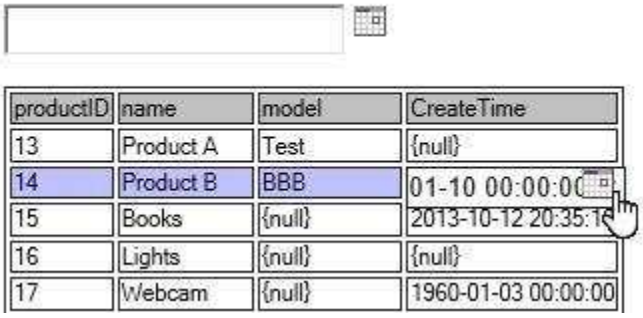
Let’s try 16 for DatePickerFontSize:



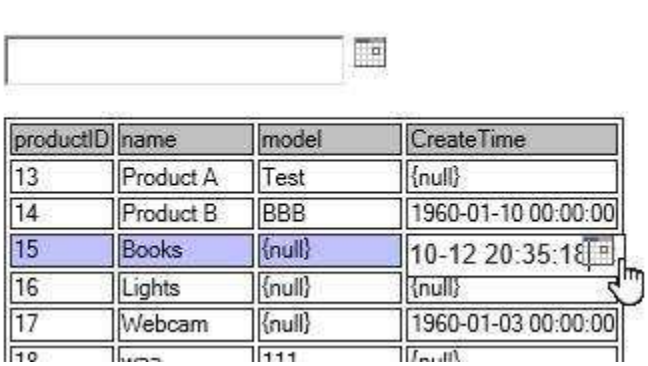
When the web page appears, a button will appear in the selected date-time cell:



productID	name	model	CreateTime
13	Product A	Test	{null}
14	Product B	BBB	1960-01-10 00:00:00
15	Books	{null}	2013-10-12 20:35:18
16	Lights	{null}	{null}
17	Webcam	{null}	1960-01-03 00:00:00
18	waa	111	{null}



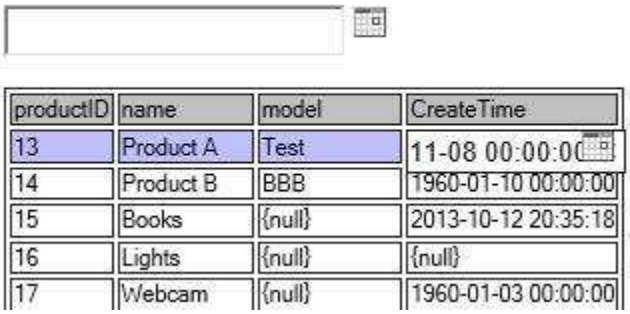
productID	name	model	CreateTime
13	Product A	Test	{null}
14	Product B	BBB	01-10 00:00:00
15	Books	{null}	2013-10-12 20:35:18
16	Lights	{null}	{null}
17	Webcam	{null}	1960-01-03 00:00:00



productID	name	model	CreateTime
13	Product A	Test	{null}
14	Product B	BBB	1960-01-10 00:00:00
15	Books	{null}	10-12 20:35:18
16	Lights	{null}	{null}
17	Webcam	{null}	1960-01-03 00:00:00
18	waa	111	{null}

Click the button, a date-time picker appears:

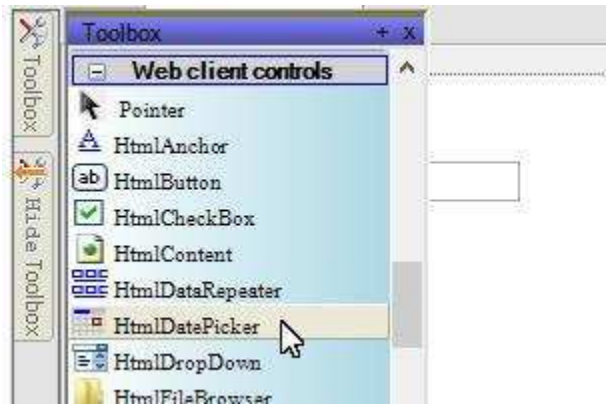
Click a date, the date-time appears in the cell:



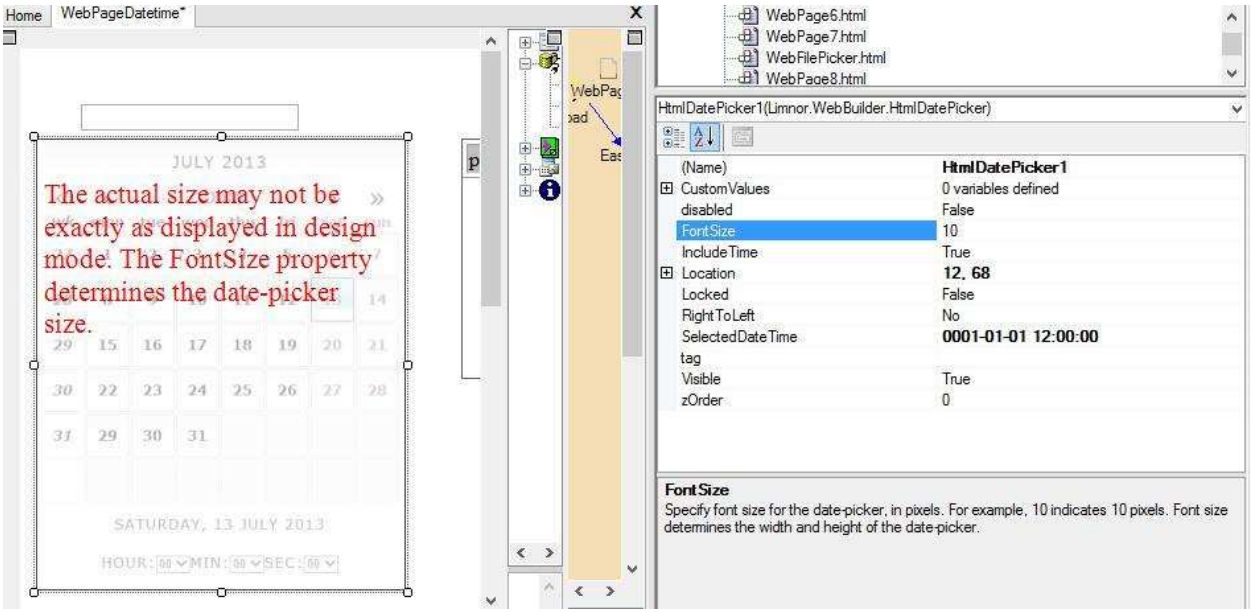
The date-time picker disappears on clicking a date on anywhere on the web page.

Use Date-time Picker Control

HtmlDatePicker can be added to a web page as a calendar.



Properties



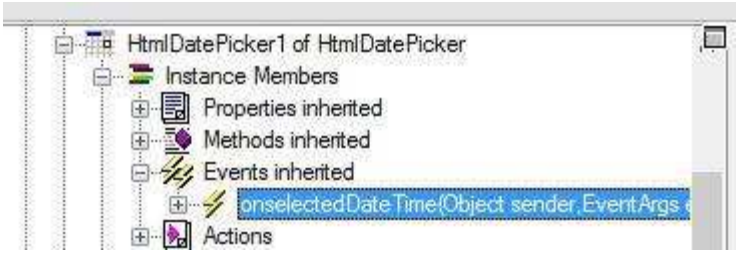
- FontSize – It is an integer indicating the font size, in pixels. It determines the size of the control.
- IncludeTime – It is a Boolean indicating whether time selection is enabled.
- SelectedDateTime – It is a date-time value indicating selected date-time. Setting this property will refresh the control display to show the selected date-time.
- disabled – it is a Boolean indicating whether the date-time picker is disabled, that is, the mouse and keyboard are not responsive on the control.
- Visible – It is a Boolean indicating whether the control is visible or hidden
- Left and Top – they indicate the location of the control

Methods

This control does not have methods specific to this control besides methods common to all web controls.

Events

onSelectedDateTime – it occurs when a date is selected.

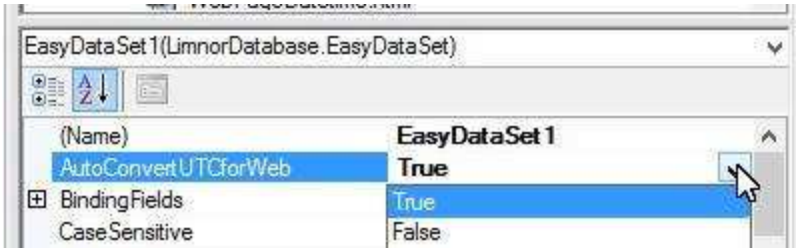


Automatic Time Zone Handling

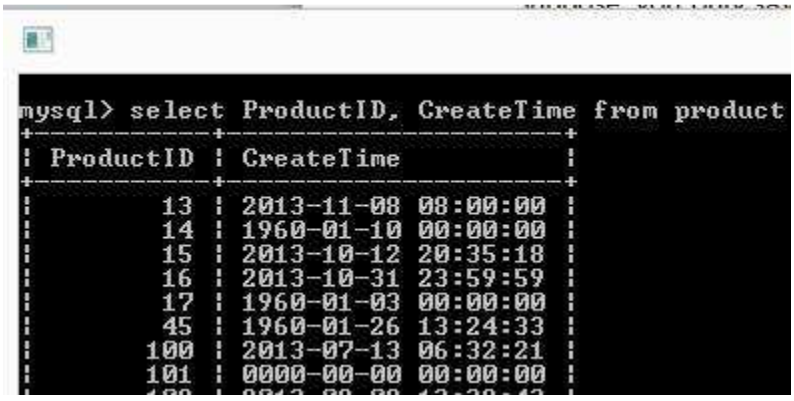
When you accept date-time inputs from web page visitors and save date-times to databases, the visitors may from different time zones. It is convenient for visitors to enter date-times in local time zones.

Suppose you only save UTC date-times in databases, then you do not have to keep tracking visitors' time zone information.

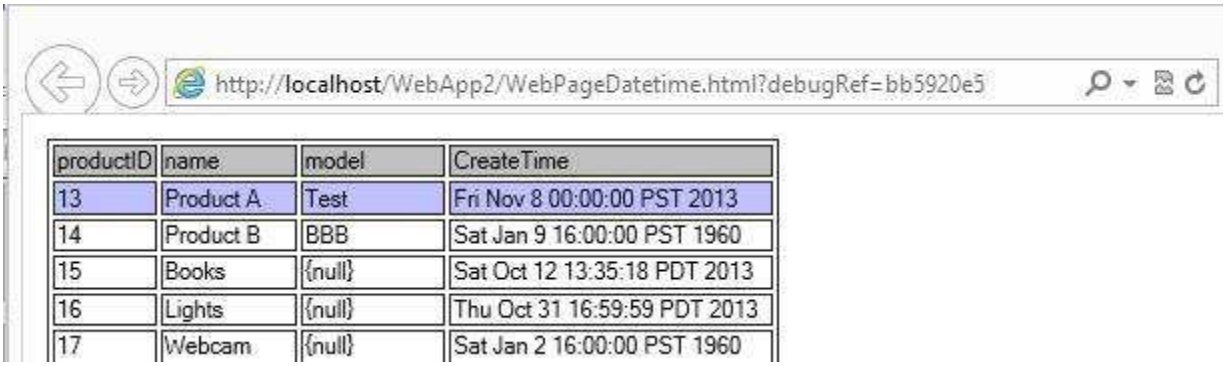
Suppose when displaying date-times in web pages, you want show date-times in local time; when sending local date-times to web server, you want to convert date-times to UTC values. You may let EasyDataSet do such conversions automatically, by setting AutoConvertUTCforWeb property to True:



Suppose we have following data in a database:

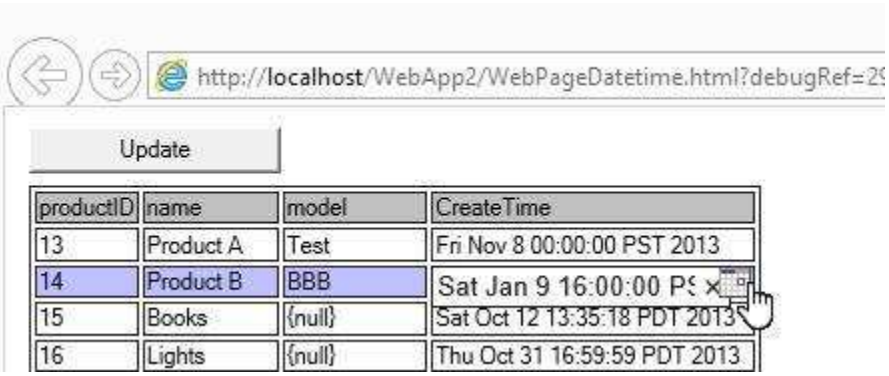


In a web page, such an EasyDataSet will show date-times in local time. For example, if a web browser runs in (UTC-08:00) Pacific Time (US & Canada) then the date-times are displayed in Pacific Time zone:

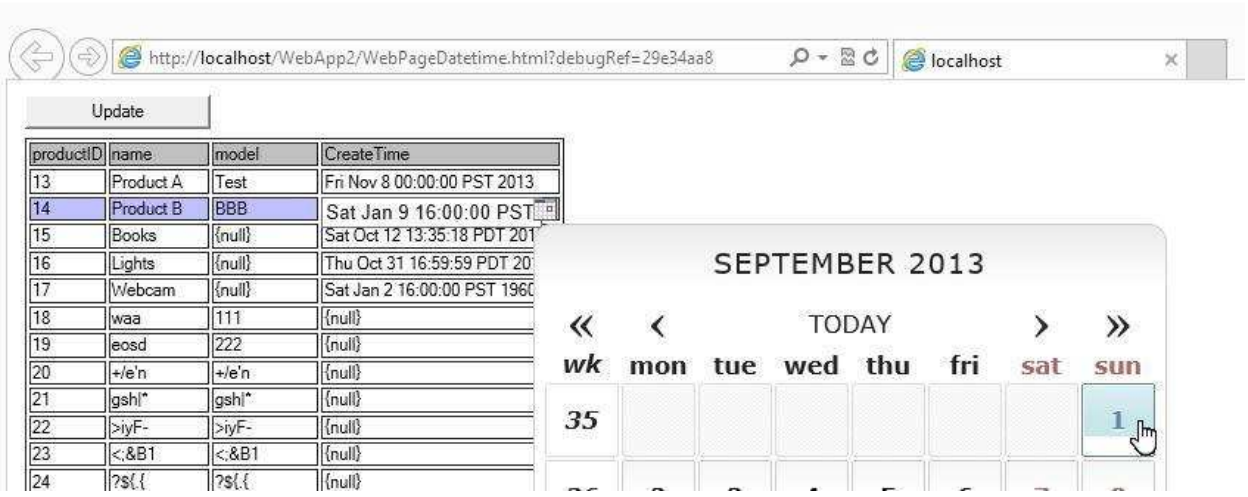


productID	name	model	CreateTime
13	Product A	Test	Fri Nov 8 00:00:00 PST 2013
14	Product B	BBB	Sat Jan 9 16:00:00 PST 1960
15	Books	{null}	Sat Oct 12 13:35:18 PDT 2013
16	Lights	{null}	Thu Oct 31 16:59:59 PDT 2013
17	Webcam	{null}	Sat Jan 2 16:00:00 PST 1960

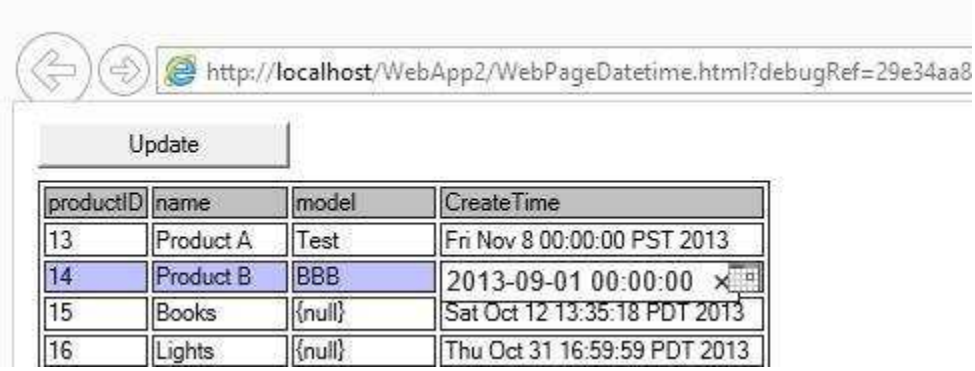
Let's enter a date-time via the web page:



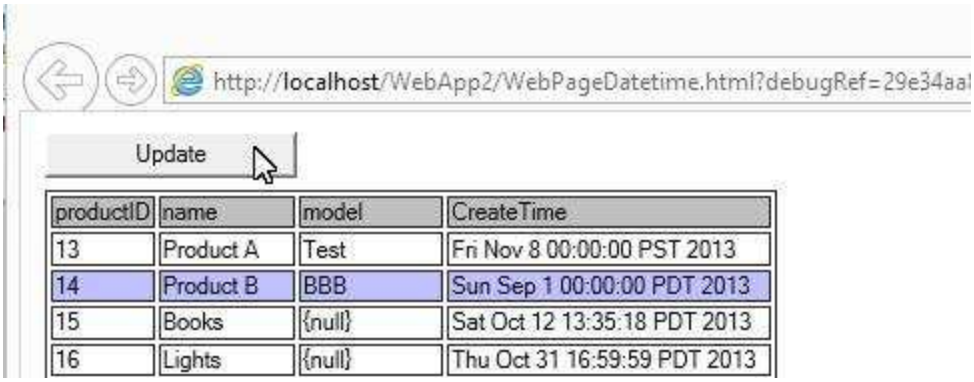
productID	name	model	CreateTime
13	Product A	Test	Fri Nov 8 00:00:00 PST 2013
14	Product B	BBB	Sat Jan 9 16:00:00 PS
15	Books	{null}	Sat Oct 12 13:35:18 PDT 2013
16	Lights	{null}	Thu Oct 31 16:59:59 PDT 2013



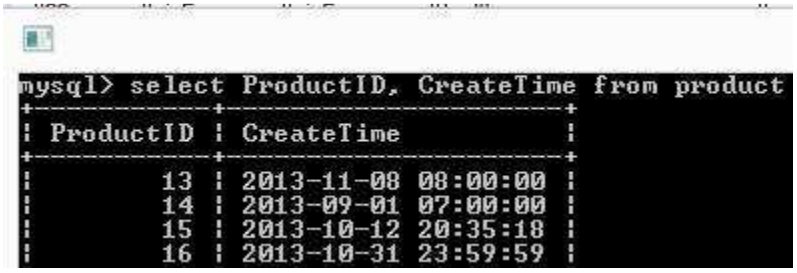
productID	name	model	CreateTime
13	Product A	Test	Fri Nov 8 00:00:00 PST 2013
14	Product B	BBB	Sat Jan 9 16:00:00 PST
15	Books	{null}	Sat Oct 12 13:35:18 PDT 201
16	Lights	{null}	Thu Oct 31 16:59:59 PDT 20
17	Webcam	{null}	Sat Jan 2 16:00:00 PST 196
18	waa	111	{null}
19	eosd	222	{null}
20	+/e'n	+/e'n	{null}
21	gshl*	gshl*	{null}
22	>iyF-	>iyF-	{null}
23	<;&B1	<;&B1	{null}
24	?s{.{	?s{.{	{null}



Click Update button to execute an Update action of the EasyDataSet. The date-time will be sent to the web server, in UTC value:



The new date-time is saved to the database, in UTC and taking effect of Day Light Saving:



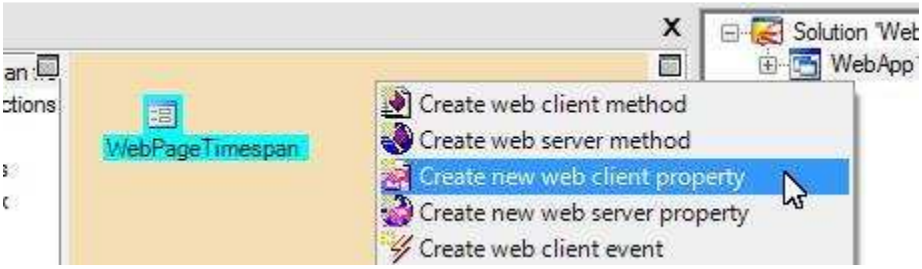
Note that if you use EasyUpdator to update database, or use CreateNewRecord of EasyDataSet to create a new record, then you need to convert client side date-time values to UTC by using AsUTCIsoString property of a date-time value.

If you have a UTC date-time in a web page then you manually convert it to a local time by accessing its toLocalDatetime property.

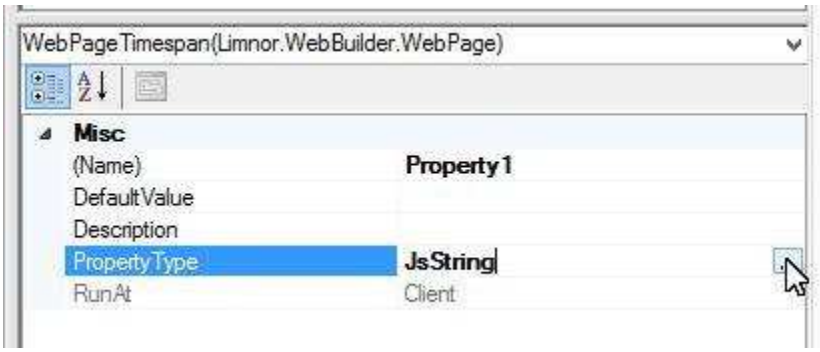
Timespan

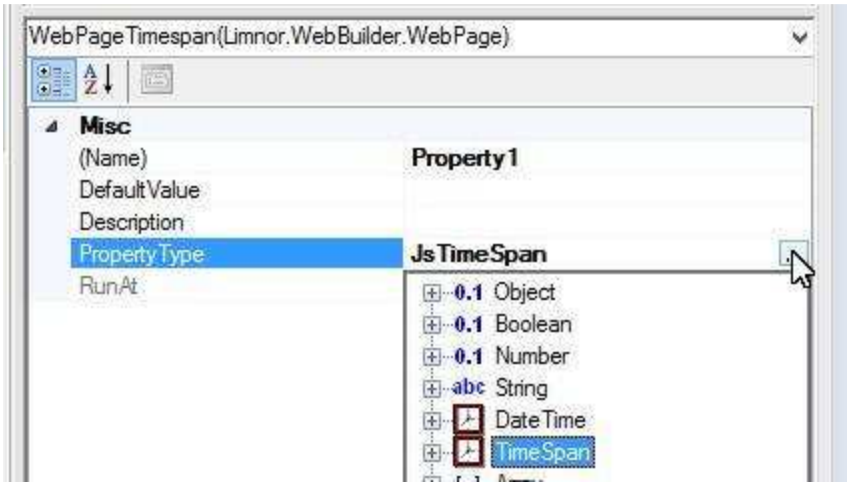
A timespan object represents a span of time. In Microsoft .Net Framework, there is a Timespan class. For web client Limnor Studio provides a JsTimespan class. This chapter shows the use of timespan in web applications and databases.

Let's create a timespan property in a web page to demonstrate the use of timespan:

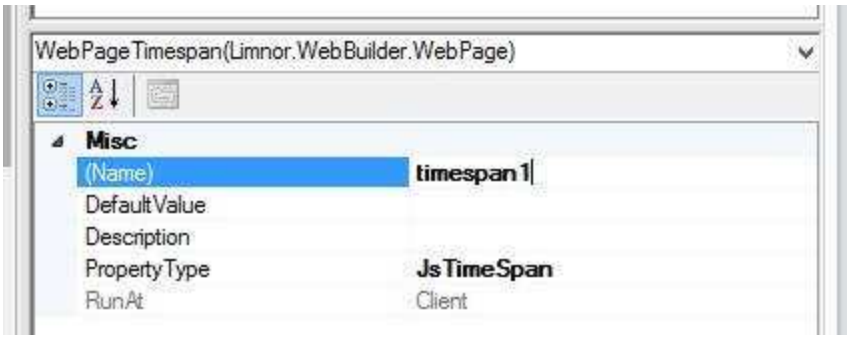


Change the property type to timespan:



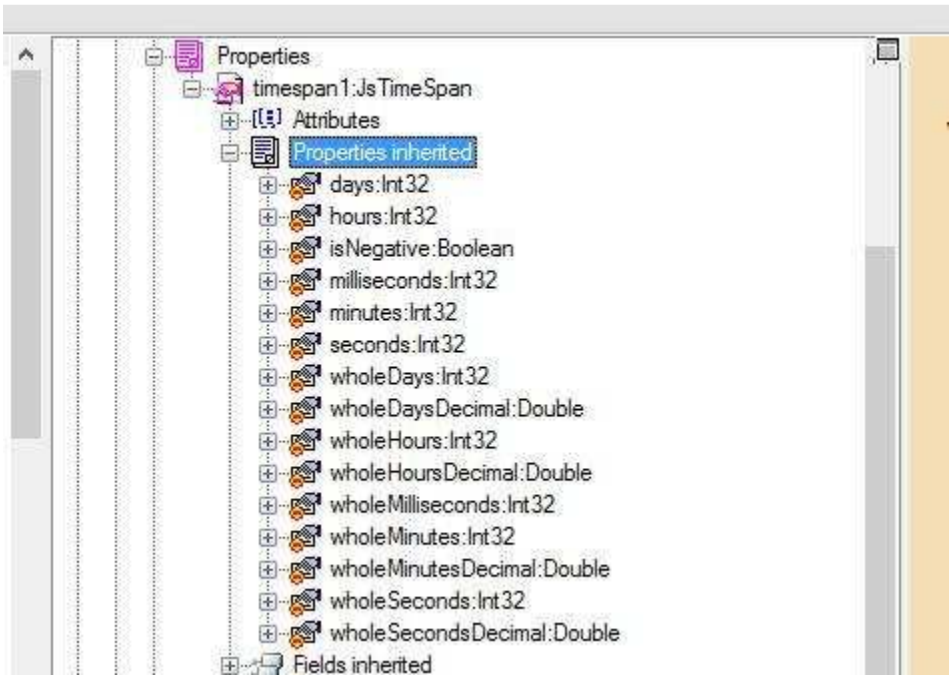


Rename the property:



Properties

A timespan object provides many properties to represent the span of time in different units and components.

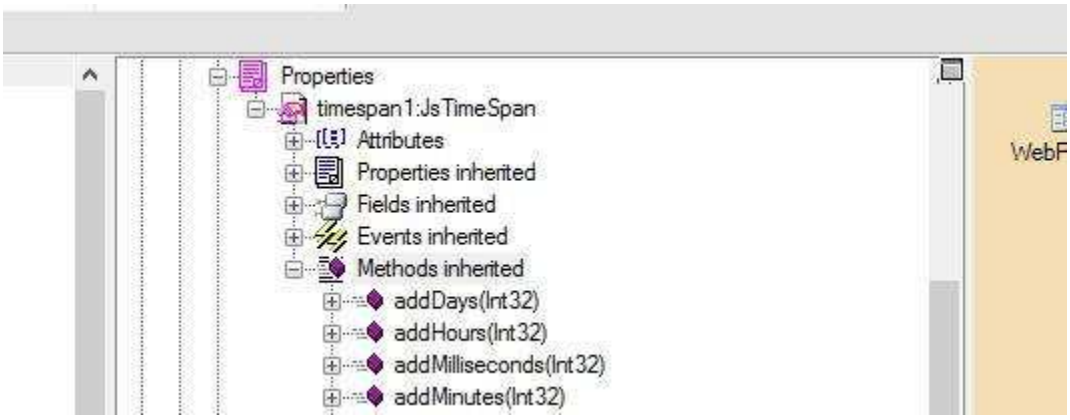


- **days**, **hours**, **minutes**, **seconds**, **milliseconds** – These 5 properties represent the 5 components of a timespan. All the 5 components adding together form the whole span of time, ignoring the fraction less than a millisecond.
- **wholeDays** – represents the span of time in days, in an integer value, ignoring the parts less than a day.
- **wholeDaysDecimal** – represents the span of time in days, in a decimal value. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeDaysDecimal = 1.0423726967592592
- **wholeHours** – represents the span of time in hours, in an integer value, ignoring parts less than an hour. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeHours = 25

- **wholeHoursDecimal** – represents the span of time in hours, in a decimal value. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeHoursDecimal = 25.016944722222224
- **wholeMinutes** – represents the span of time in minutes, in an integer value, ignoring parts less than a minute. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeMinutes = 1501
- **wholeMinutesDecimal** – represents the span of time in minutes, in a decimal value. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeMinutesDecimal = 1501.0166833333332
- **wholeSeconds** – represents the span of time in second, in an integer, ignoring parts less than a second. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeSeconds = 90061
- **wholeSecondsDecimal** – represents the span of time in seconds, in a decimal value. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeSecondsDecimal = 90061.001
- **wholeMilliseconds** – represents the span of time in milliseconds, in an integer, ignoring parts less than a millisecond. For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then wholeMilliseconds = 90061001.
- **wholeMillisecondsDecimal** – represents the span of time in milliseconds, in a decimal value. For example, if days=1, hours=1, minutes=1, seconds=1, milliseconds=1 and 0.23 milliseconds then wholeMillisecondsDecimal = 90061001.23.
- **isNegative** – the value of timespan can be negative or positive. For a negative timespan, all the above properties are negative values or 0. For a positive timespan, all the above properties are positive values or 0. This property is true if the timespan is negative. This property is false if the timespan is 0 or positive.

Methods

A timespan object provides following methods for manipulating of span of time.



- **parseIsoString** – this method uses a string to set its value of timespan. The string must in the format of “hours:minutes:seconds.milliseconds”. A minus sign in front is allowed.

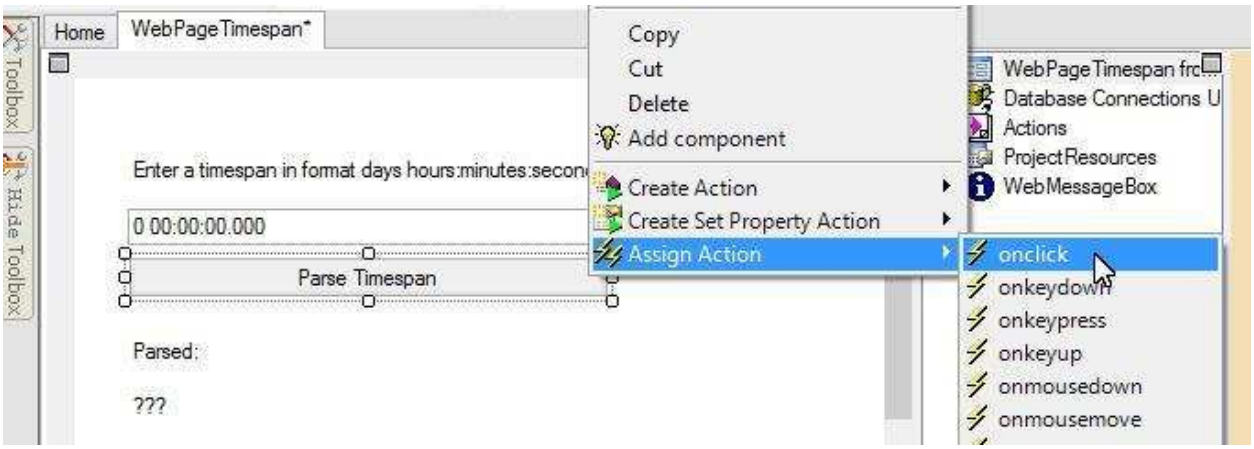
- **parseTimeSpan** – this method uses a string to set its value of timespan. The string must in the format of "days hours:minutes:seconds.milliseconds". "days" part is optional. A minus sign in front is allowed.
- **toWholeString** – this method returns a string representing the timespan value in format "days hours:minutes:seconds.milliseconds". For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then this method returns a string "1 01:01:01.1".
- **toTimeString** this method returns a string representing the timespan value in format "hours:minutes:seconds.milliseconds". For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then this method returns a string "25:01:01.1".
- **toShortTimeString** this method returns a string representing the timespan value in format "hours:minutes:seconds". For example, if days=1, hours=1, minutes=1, seconds=1, and milliseconds=1 then this method returns a string "25:01:01".
- **addDays**, **addHours**, **addMinutes**, **addSeconds**, **addMilliseconds** – these methods add a value, in a corresponding unit, to the timespan. The value can be negative or positive. The value can be an integer or a decimal.
- **addTimeSpan** – this method adds a span of time, represented by a timespan object, to the timespan.
- **setDays, setHours, setMinutes, setSeconds, setMilliseconds** – these methods replace the corresponding component of the timespan. For example, for "02:13:01", setMinutes(61) will make it into "03:01:01".
- **setTimeSpan** – this method sets the timespan value from a timespan object.
- **setValues** – this method sets the timespan value by specifying days, hours, minutes, seconds and milliseconds.
- **setWholeTimeByDays, setWholeTimeByHours, setWholeTimeByMinutes, setWholeTimeBySeconds, setWholeTimeByMilliseconds** – these methods set the timespan value by a decimal value in the corresponding units.
- **setWholeTimeByDates** – this method sets the timespan value by the time difference between a start date and an end date.

Timespan data entry

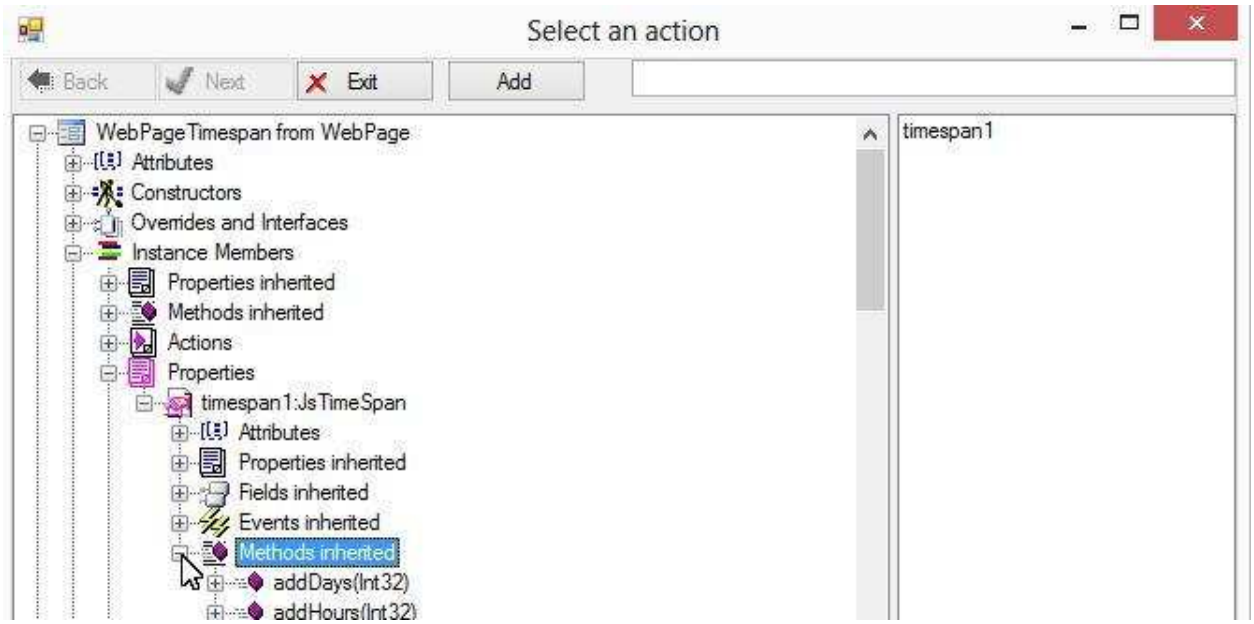
Suppose we want to let the user enter a timespan value. We may use a text box to allow the user to enter a timespan value. We have many options to interpret the value in the text box. For example, we may take the value as days by using a setWholeTimeByDays action; or as hours by using a setWholeTimeByHours action; or as minutes by using a setWholeTimeByMinutes action; or as seconds by using a setWholeTimeBySeconds; or as milliseconds by using a setWholeTimeByMilliseconds.

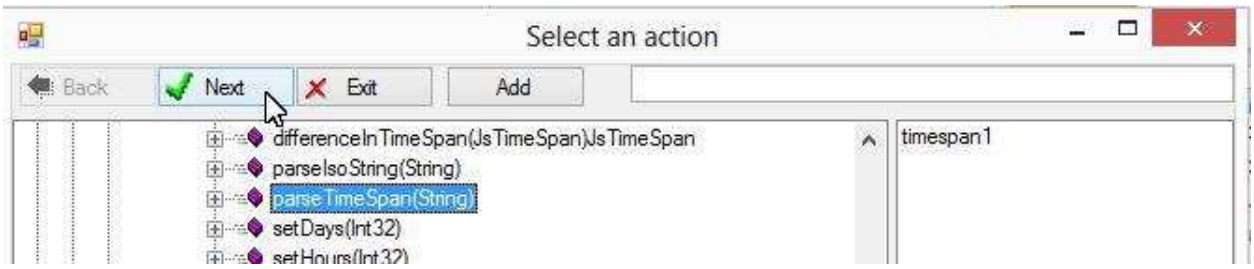
We may also allow the user to enter all parts of time in a format of "days hh:mm:ss.mmm", by using a parseTimeSpan action; or in a format of "hh:mm:ss.mmm" by using a parseIsoString action.

We use a parseTimeSpan action for the demonstration. We use a label to show the result. We use a button to execute the action. Right-click the button; choose "Assign Action"; choose "Click" event:

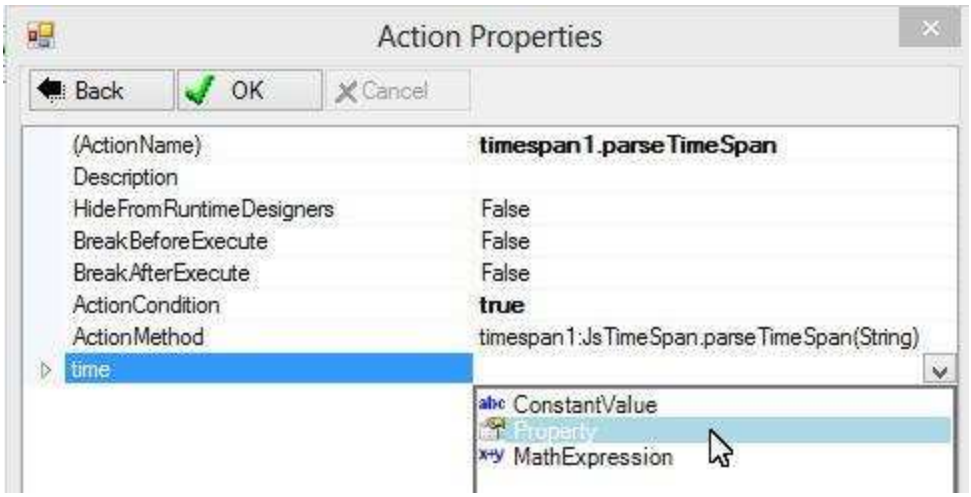


Select the parseTimeSpan method of the timespan property to create an action to execute the method:

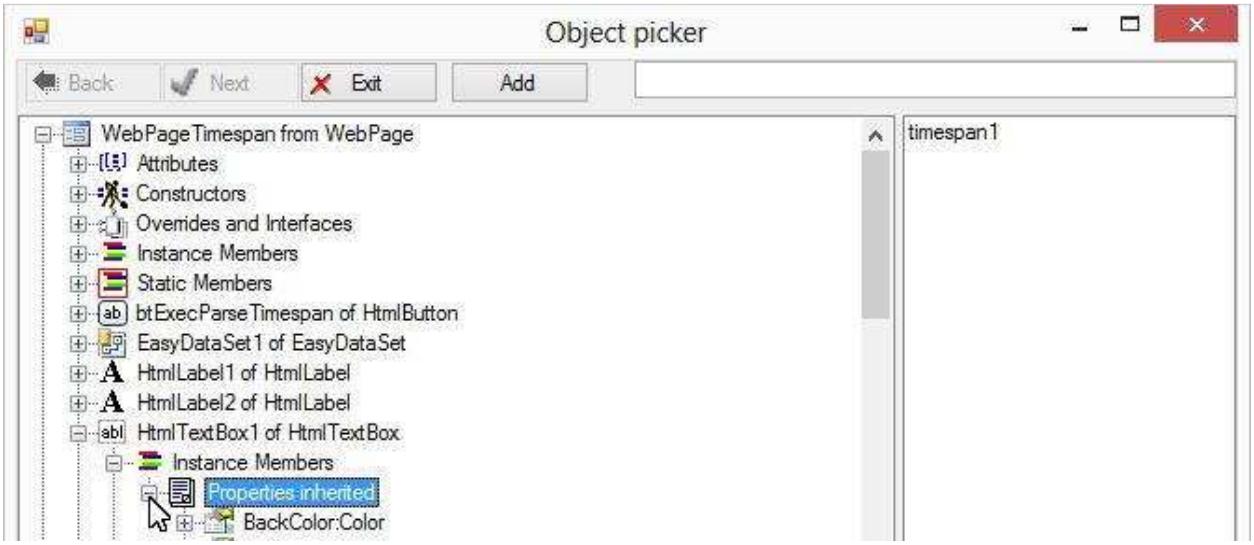




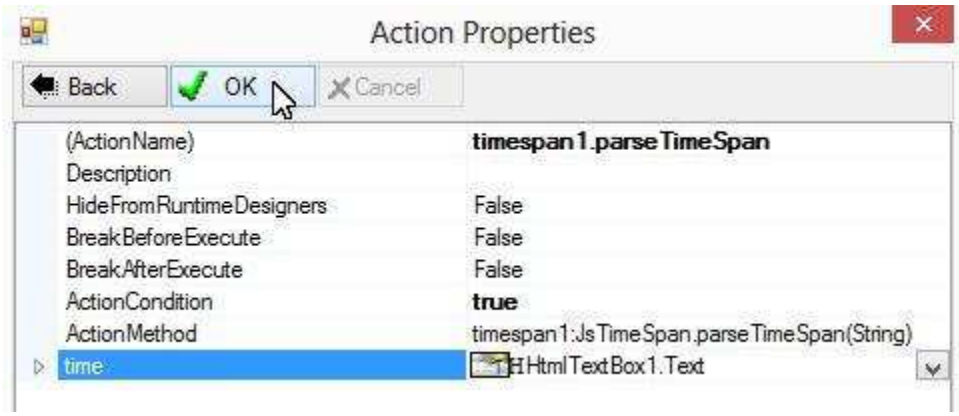
For the “time” parameter of the action, select “Property” to use the Text property of the text box:



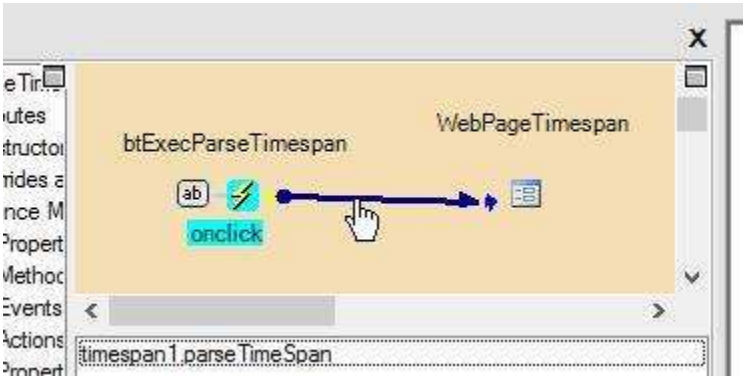
Select the Text property of the text box:



Click OK to create the action:



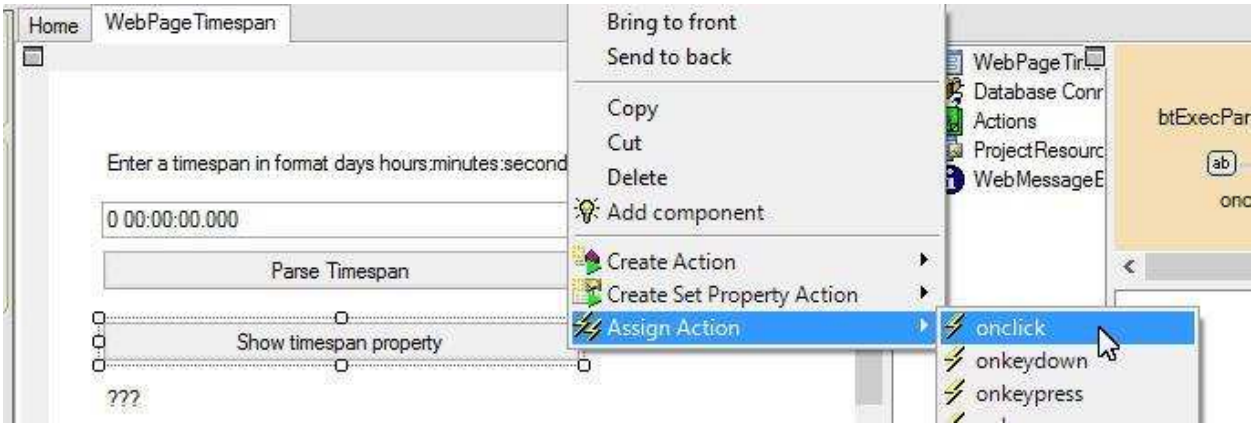
The action is created and assigned to the button:



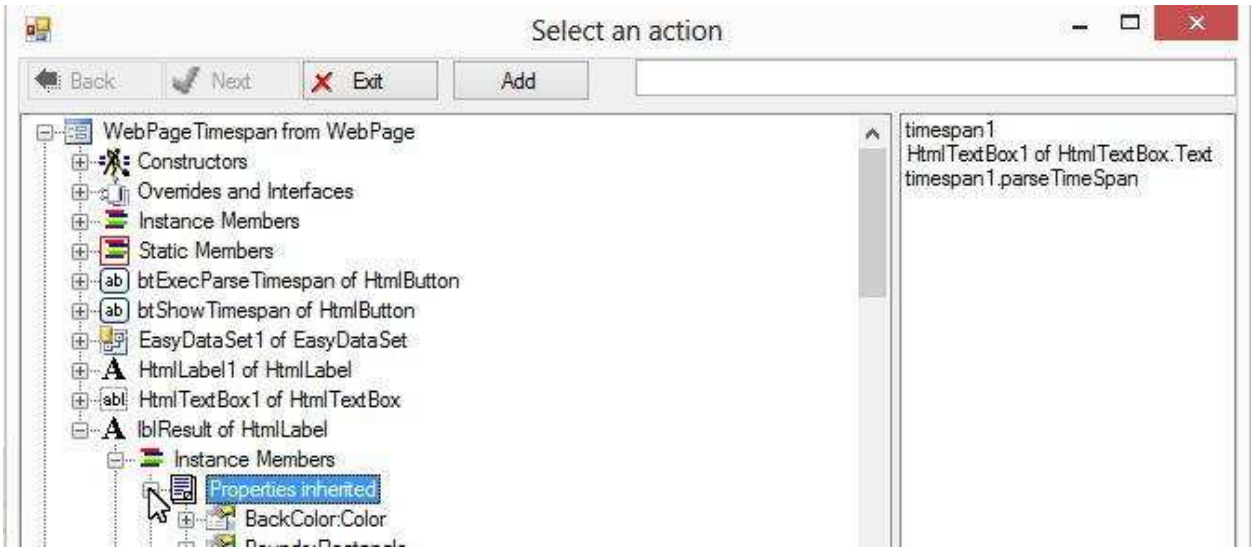
On executing parseTimeSpan, the property timespan1 will have the value entered in the text box.

Show time span as a string

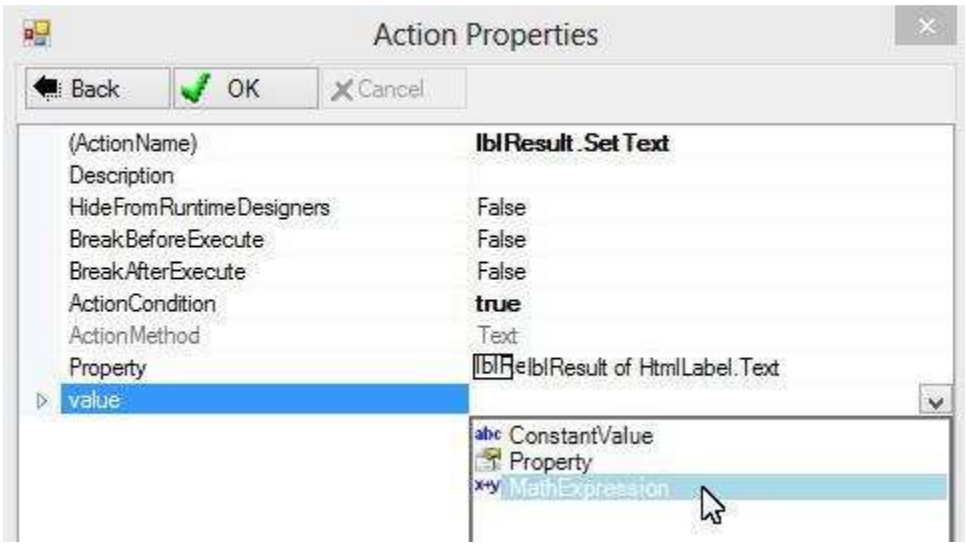
The method toWholeString of a timespan object returns a string in format “days hh:mm:ss.mmm”. We use a label to show the result of a toWholeString action. We use a button to execute such an action.



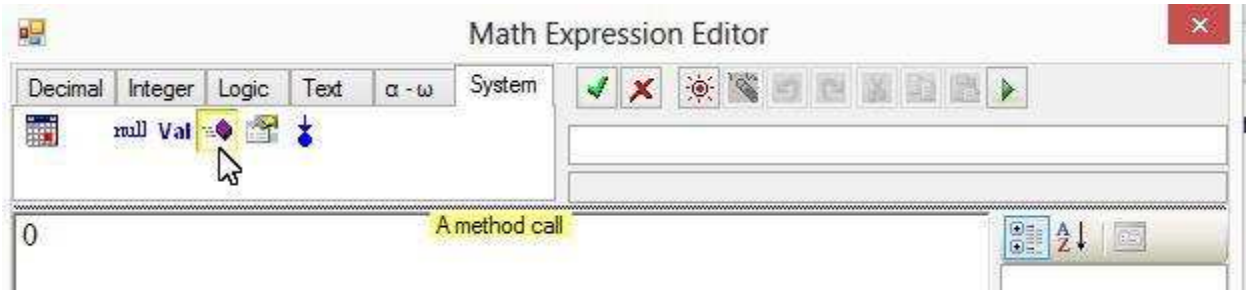
Select the Text property of the label:



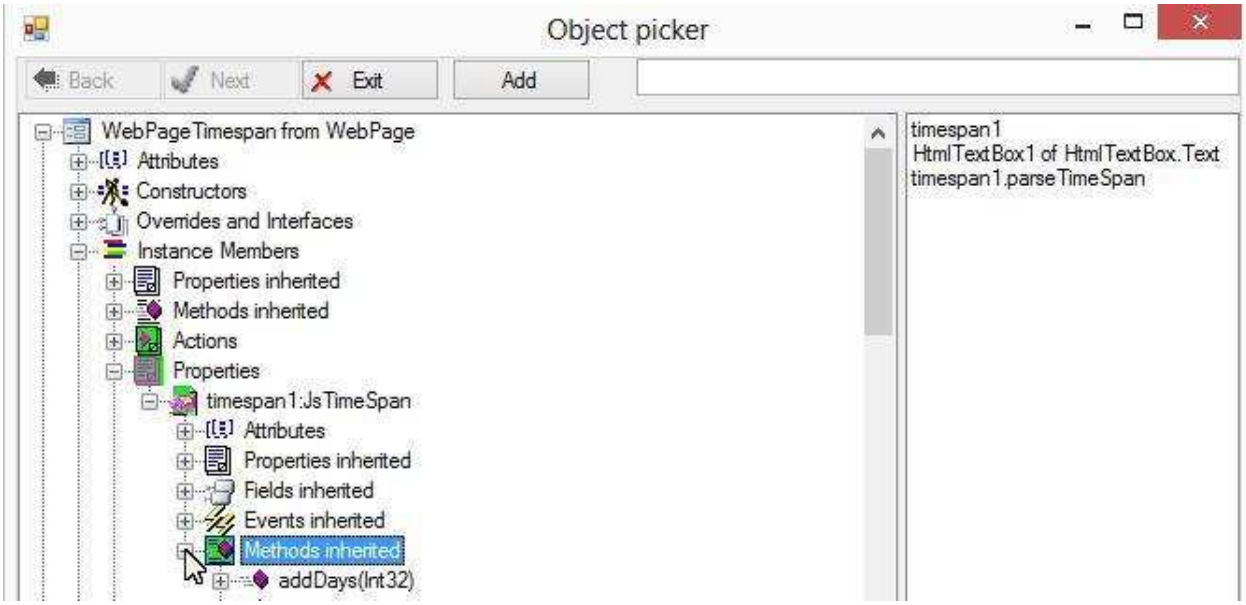
For the “value” of the action, select “Math Expression” so that we may execute a method:

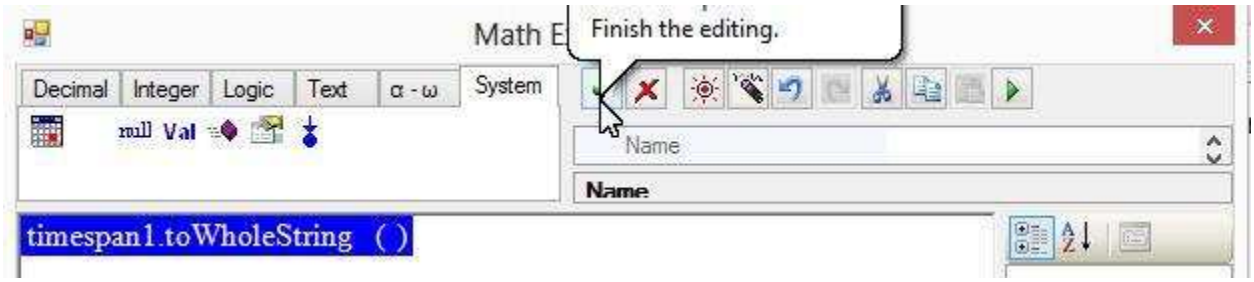


Click the method icon to select a method:

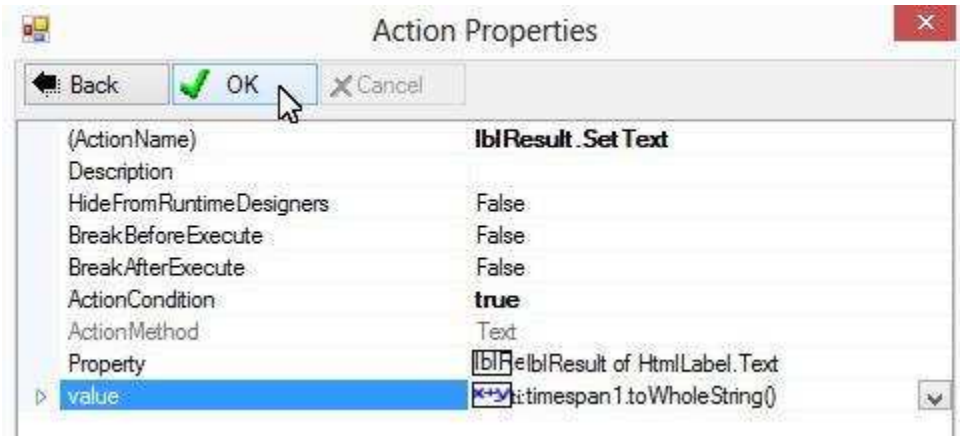


Select the toWholeString method of the property timespan1:

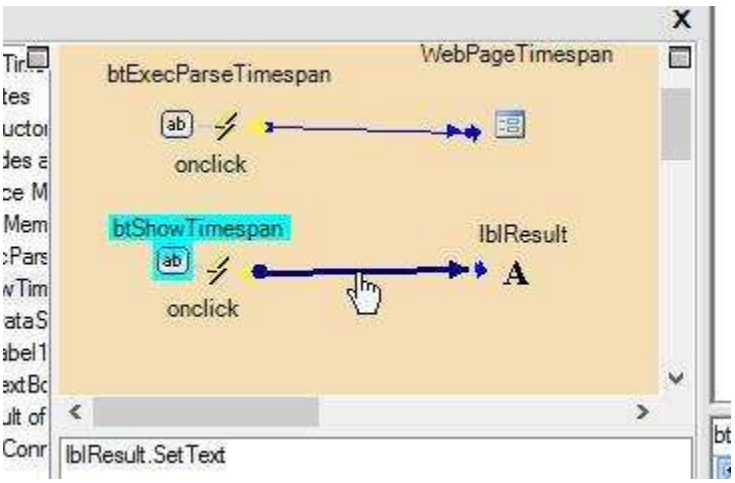




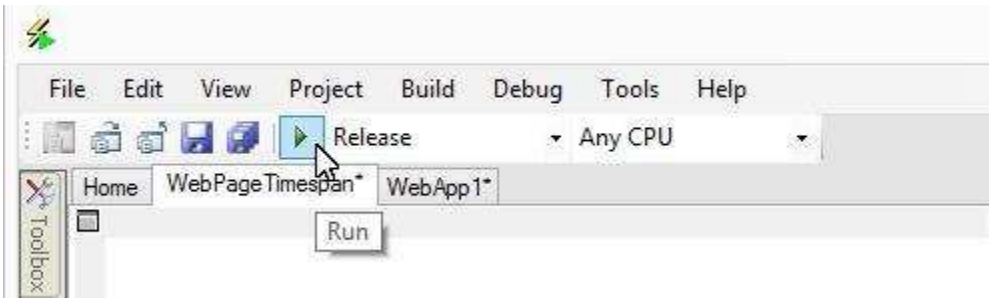
Click OK to create the action:



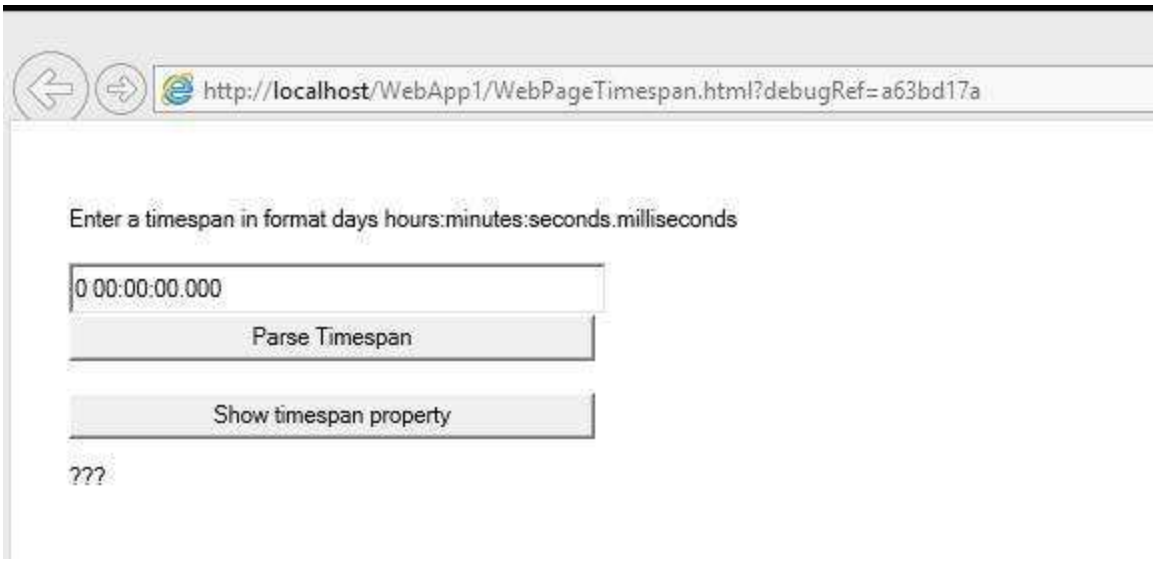
The action is created and assigned to the button:



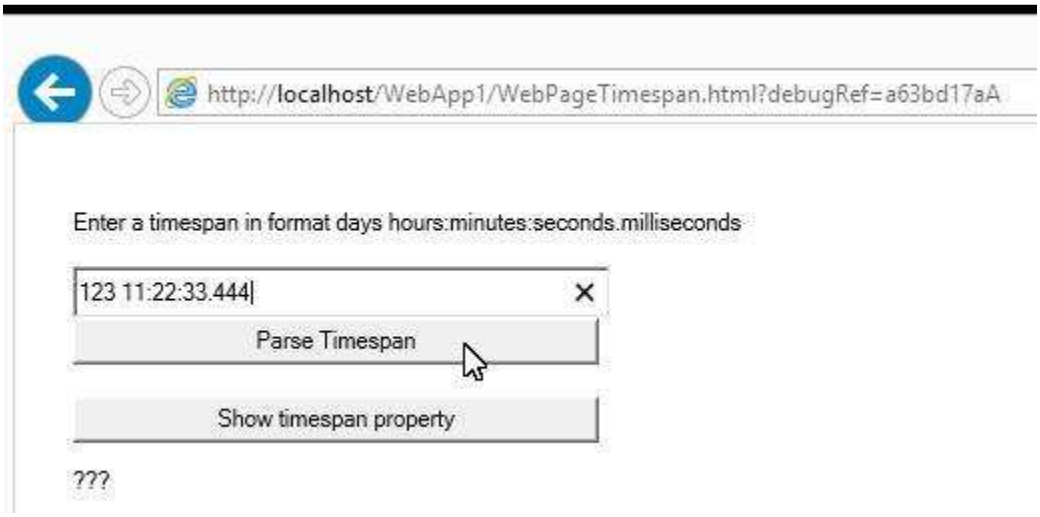
Now launch the web page to try it:



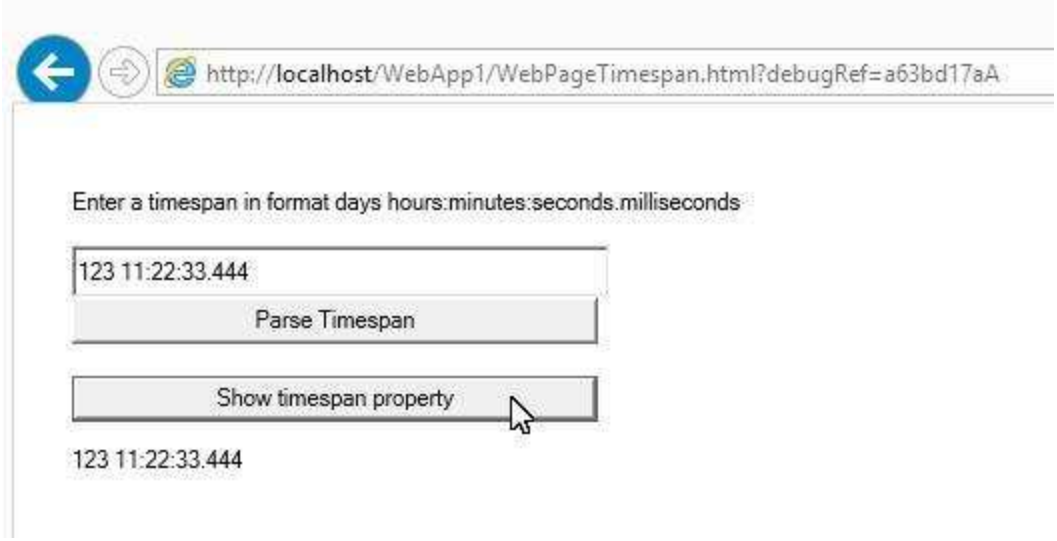
The web page appears:



Enter a timespan value and click “Parse Timespan”:

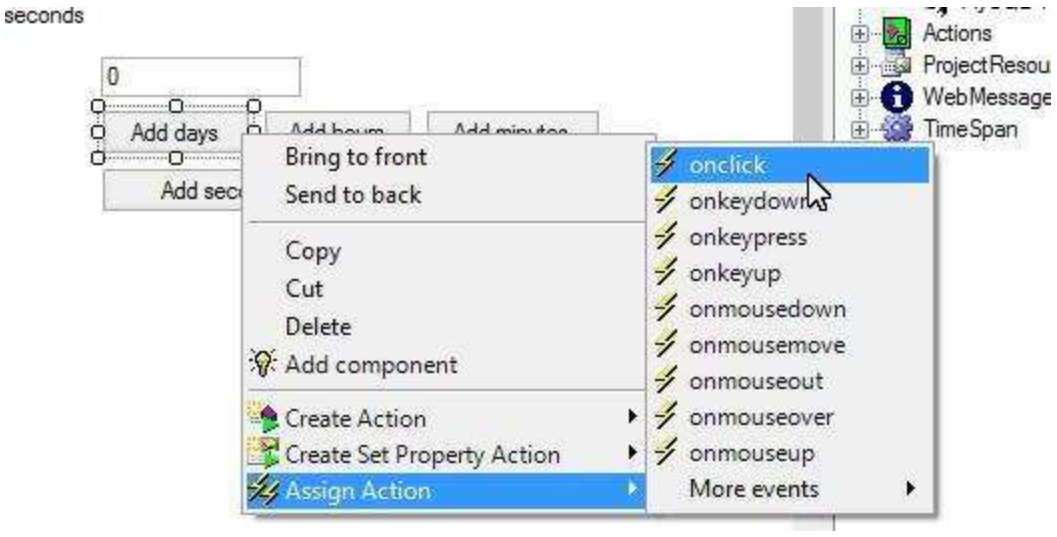


Now click “Show timespan property”, we can see that it shows what we entered:

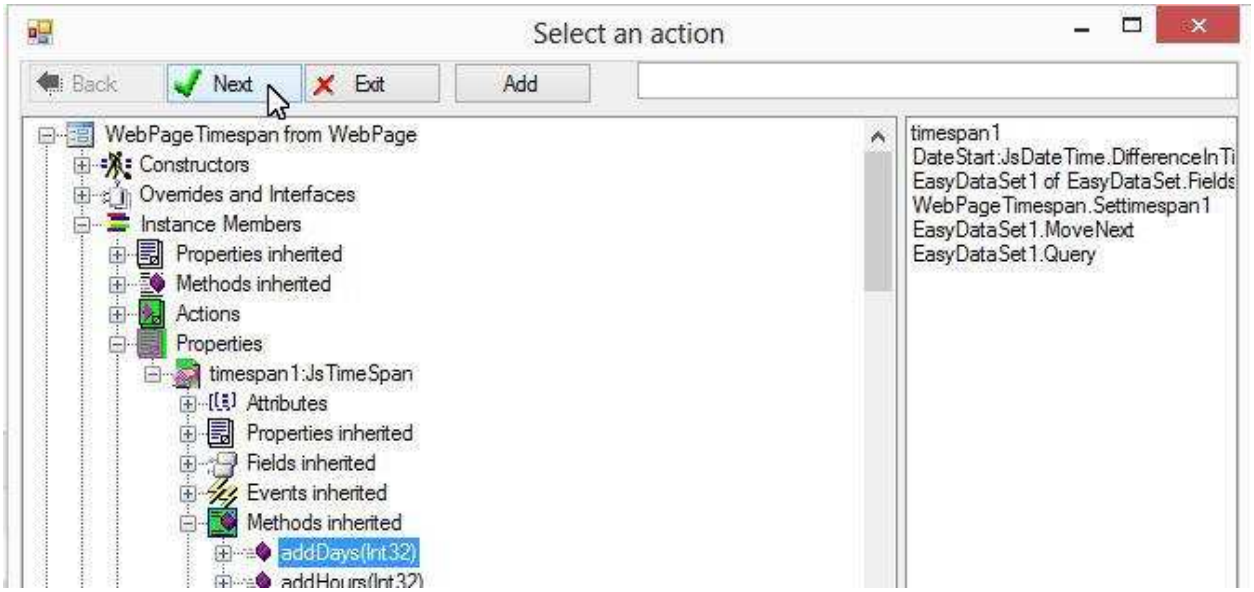


Add value to timespan

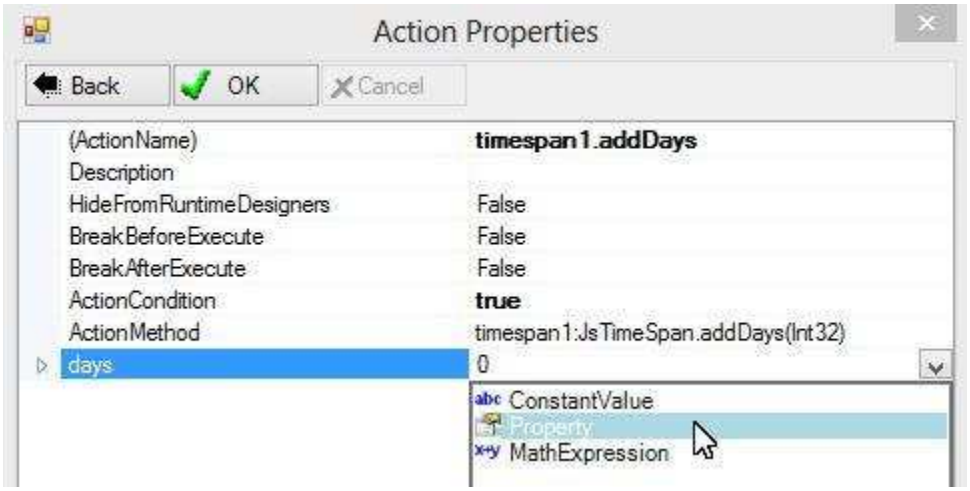
We use a text box for entering a number. We create a set of buttons in the web page. Each button executes an action to add the number to the timespan. To program button “Add days”, right-click the button; choose “Assign Action”; choose “Click” event:



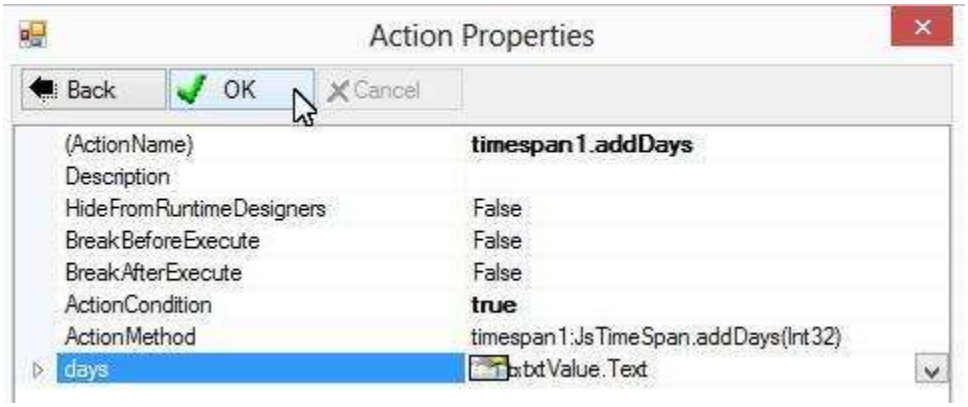
Choose method “addDays”:



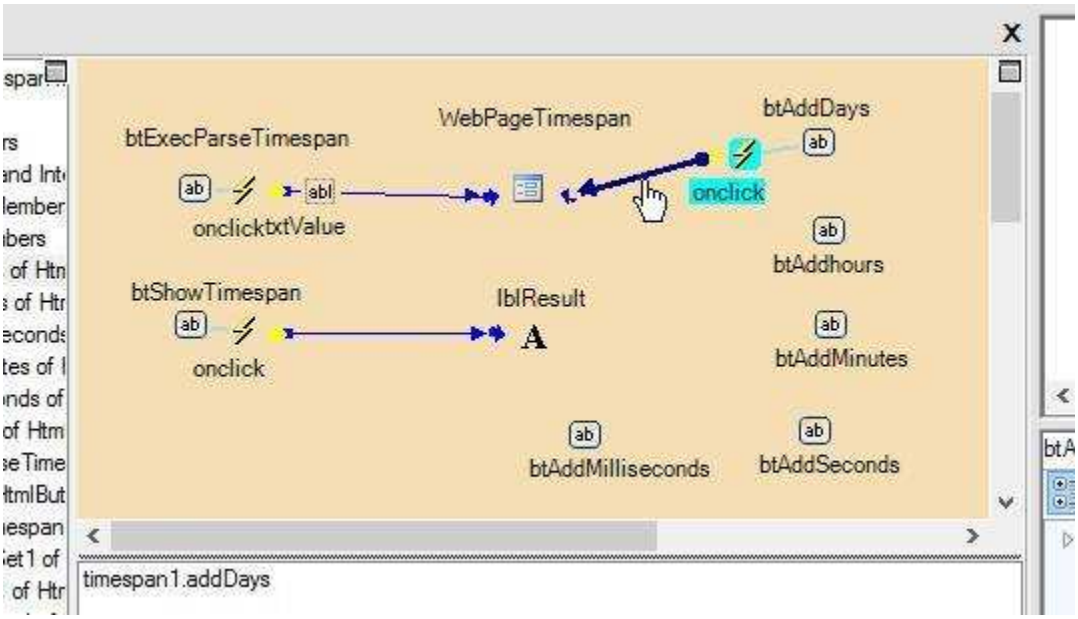
Use the Text property of the Textbox as “days”:



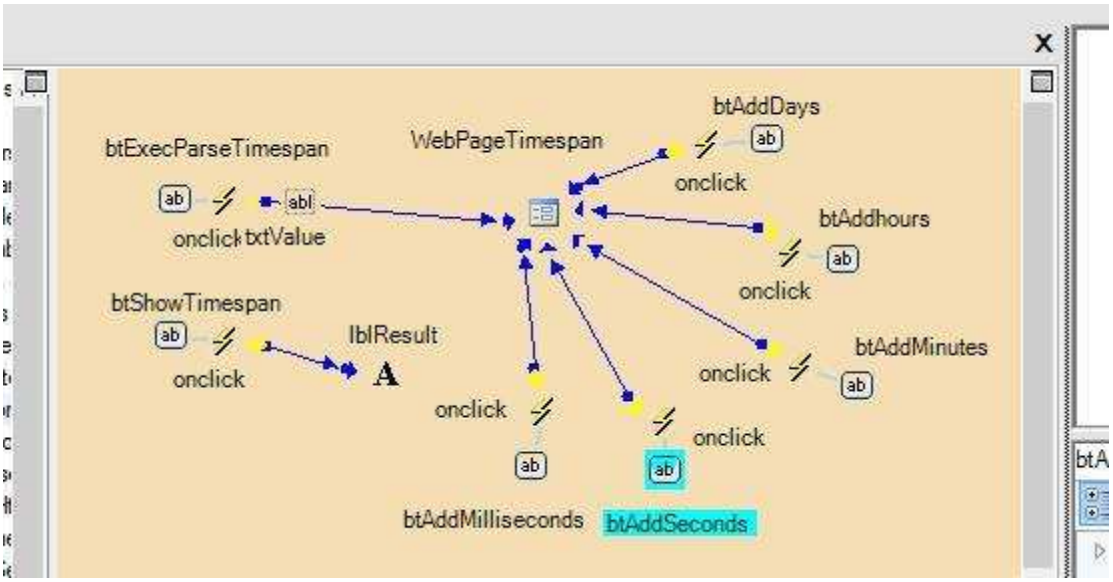
Click OK to create the action:



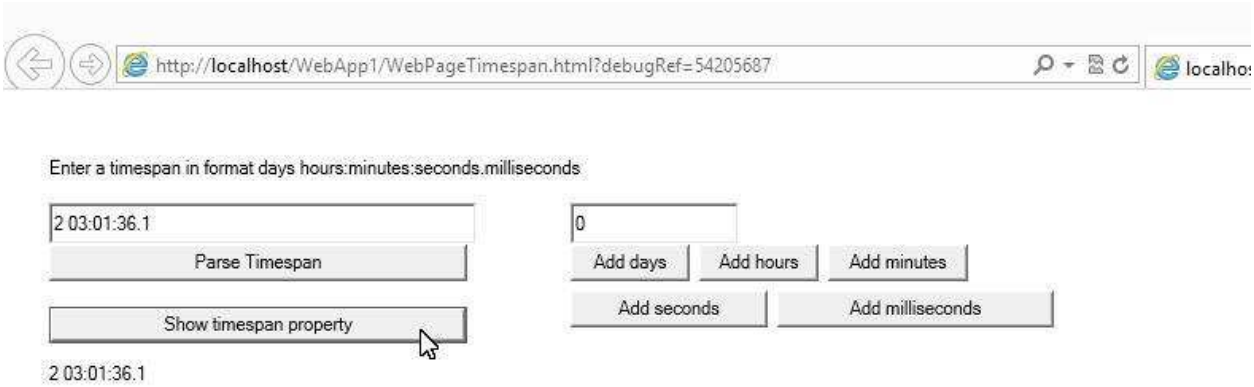
The action is created and assigned to the button:



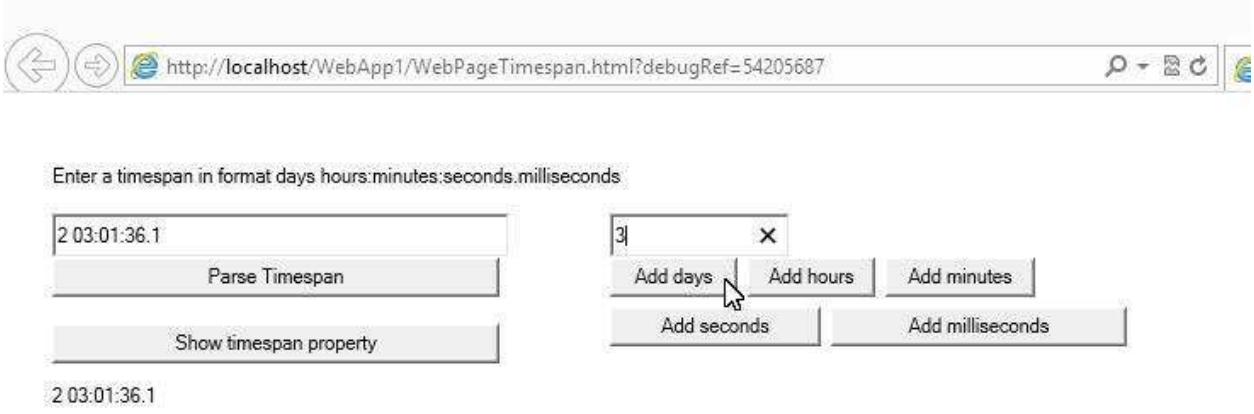
We may do the programming for other buttons, using methods addHours, addMinutes, addSeconds, and addMilliseconds respectively:



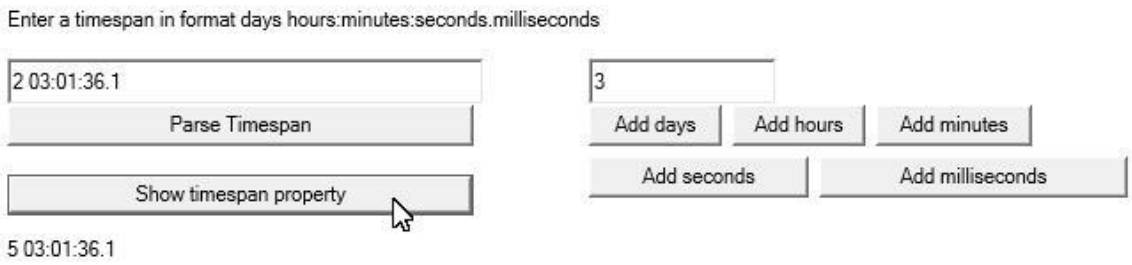
Run the web project, the web page appears. Enter a timespan value:



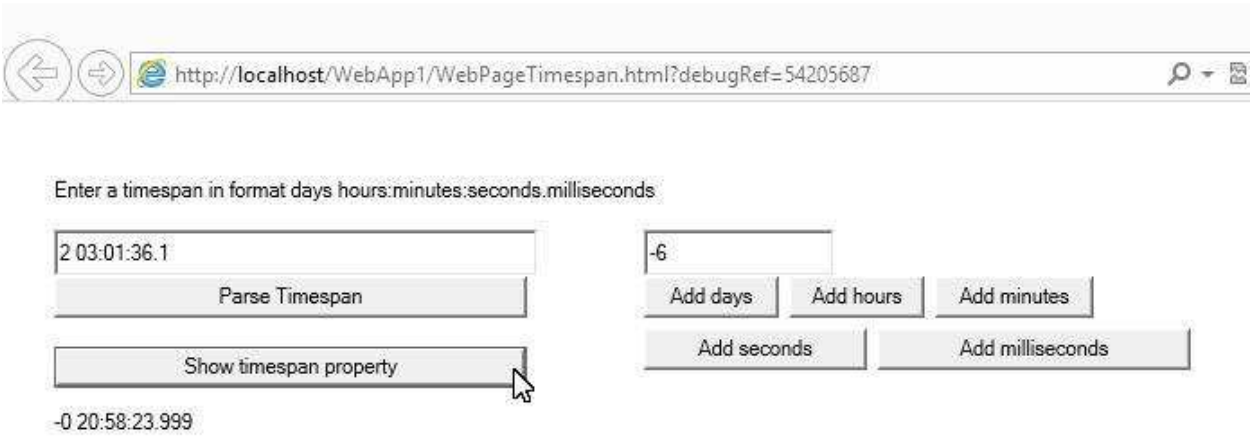
Use “Add days” button:



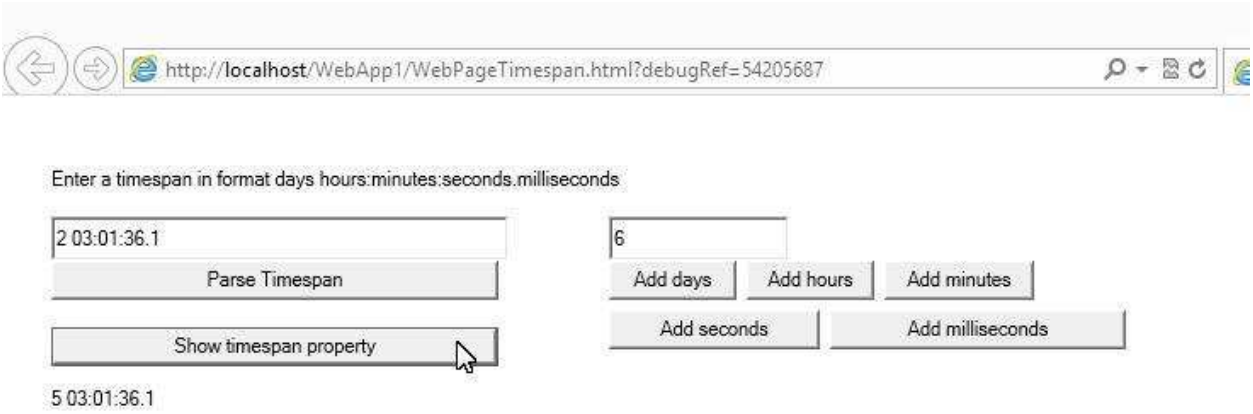
The days part changed:



We may also add a negative value:

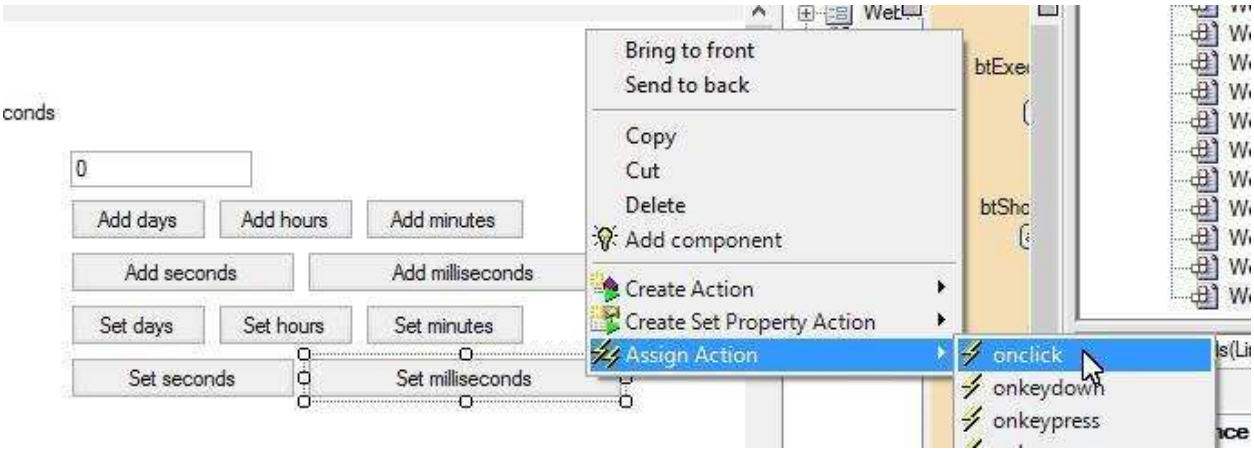


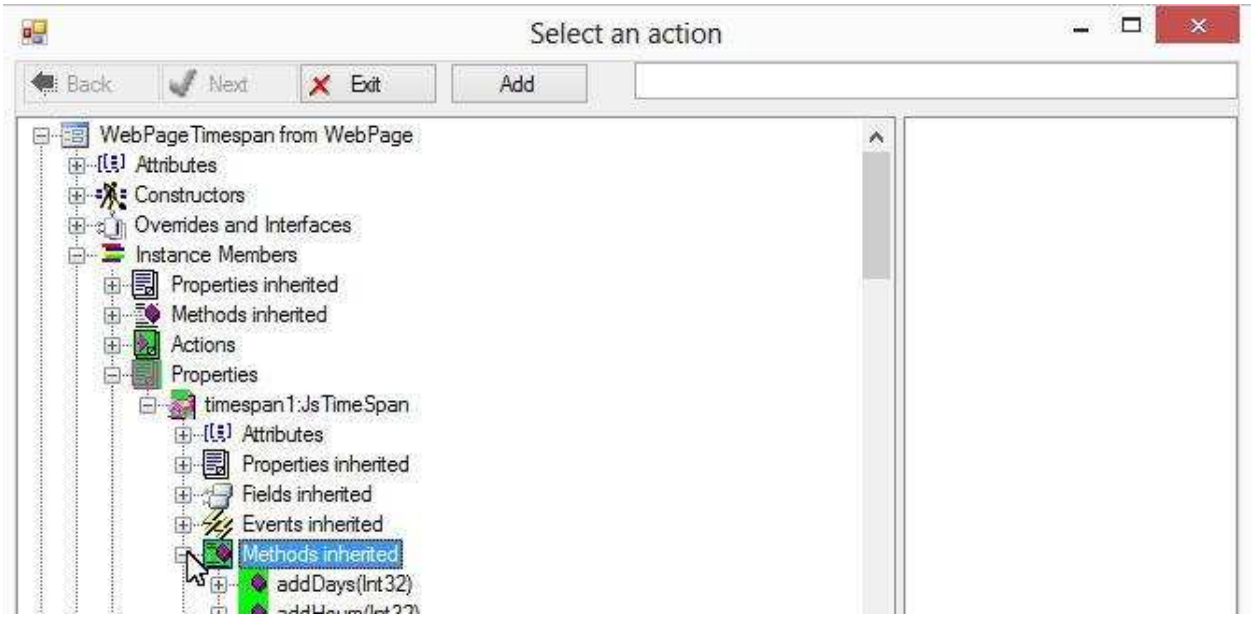
Add a positive value of the same magnitude, we can see that it restores to the original timespan:

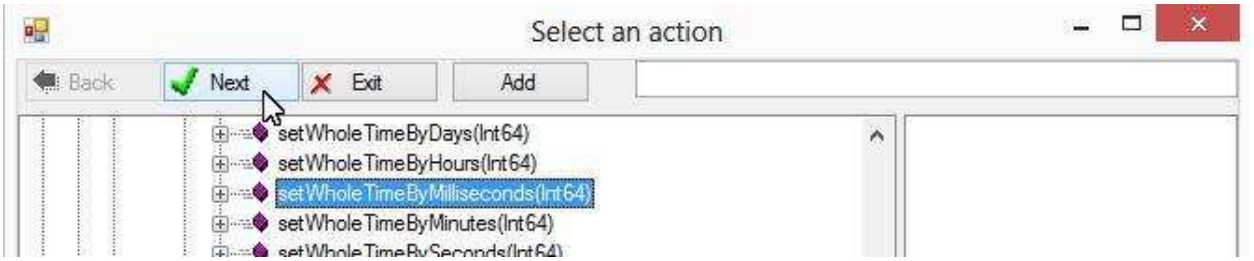


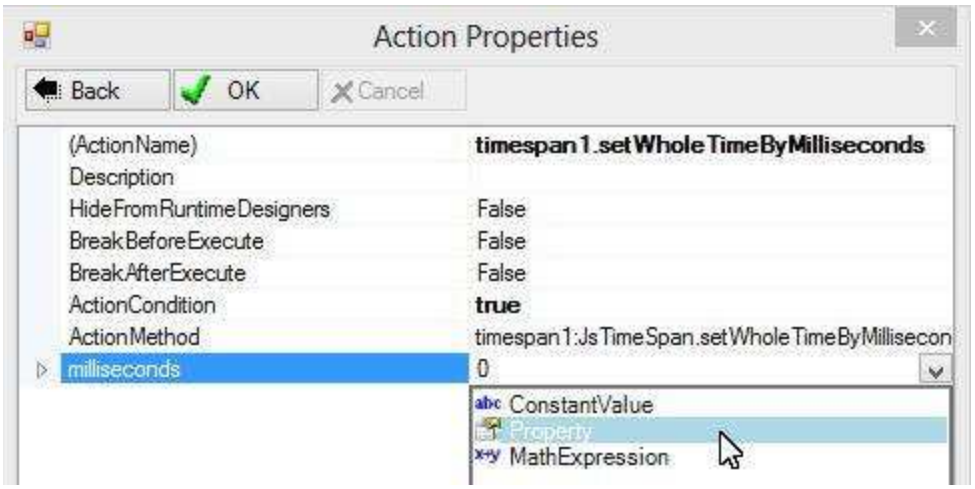
Set whole timespan value

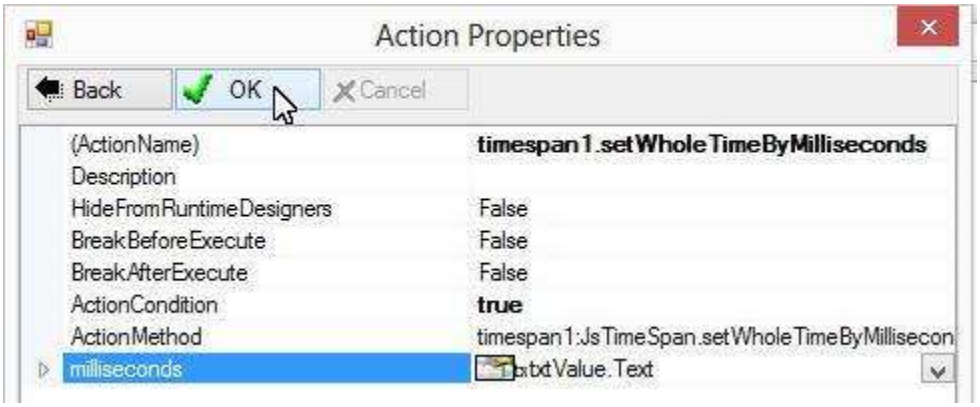
In the previous samples, a timespan is modified by adding values to its components (days, hours, minutes, seconds, or milliseconds). We may also set the whole value of the timespan, by using setWholeTimeByDays, setWholeTimeByHours, setWholeTimeByMinutes, setWholeTimeBySeconds, or setWholeTimeByMilliseconds. We can do the same programming as we did previously, using these methods.


conds
0
Add days
Add hours
Add minutes
Add seconds
Add milliseconds
Set days
Set hours
Set minutes
Set seconds
Set milliseconds
Bring to front
Send to back
Copy
Cut
Delete
Add component
Create Action
Create Set Property Action
Assign Action
onclick
onkeydown
onkeypress

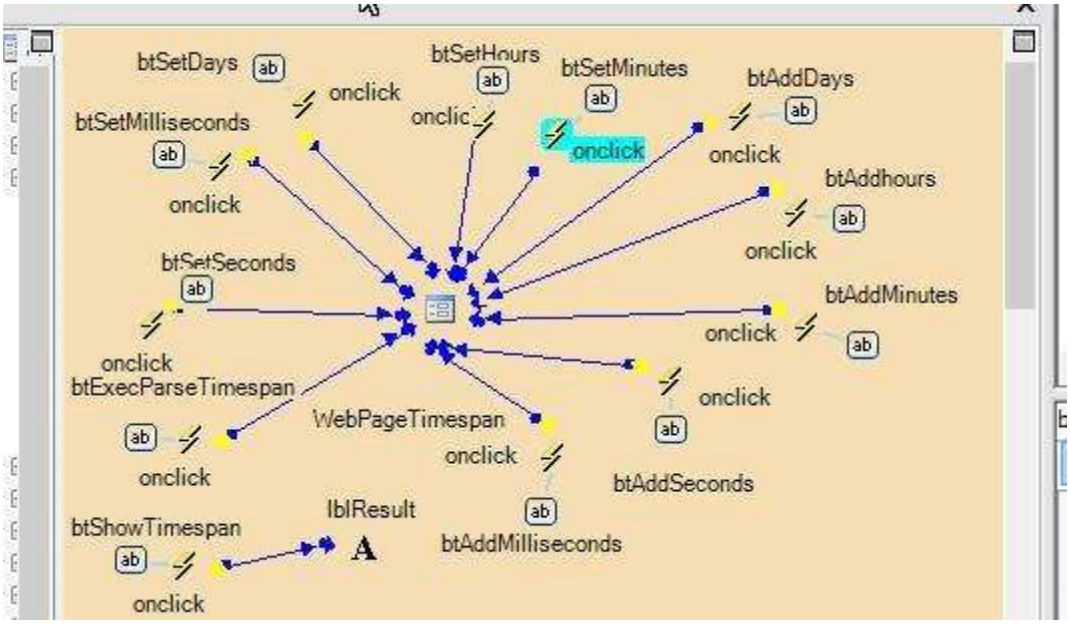

Select an action
Back
Next
Exit
Add
WebPageTimespan from WebPage
Attributes
Constructors
Overrides and Interfaces
Instance Members
Properties inherited
Methods inherited
Actions
Properties
timespan1:JsTimeSpan
Attributes
Properties inherited
Fields inherited
Events inherited
Methods inherited
addDays(Int32)


Select an action
Back
Next
Exit
Add
setWholeTimeByDays(Int64)
setWholeTimeByHours(Int64)
setWholeTimeByMilliseconds(Int64)
setWholeTimeByMinutes(Int64)

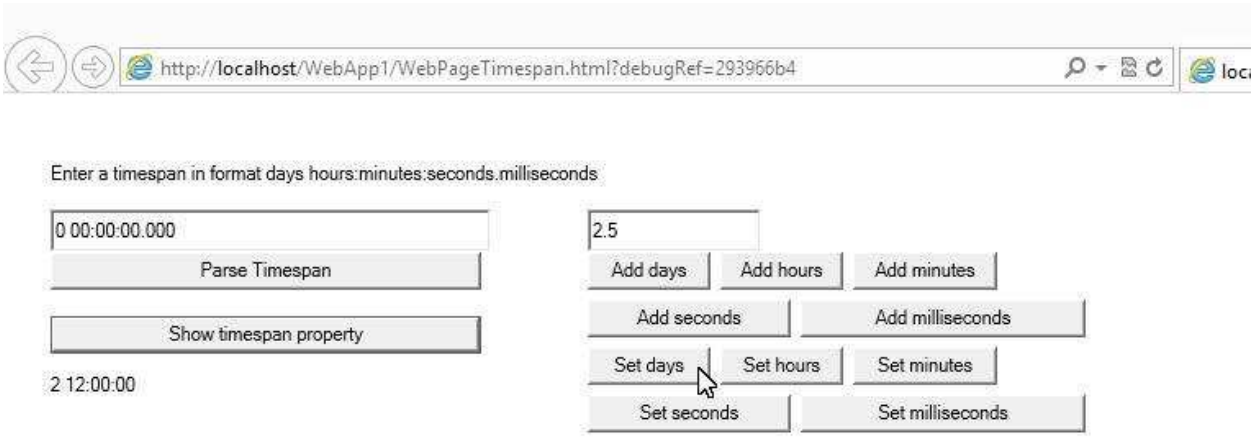




The action is created and assigned to the button. Do it for other buttons:

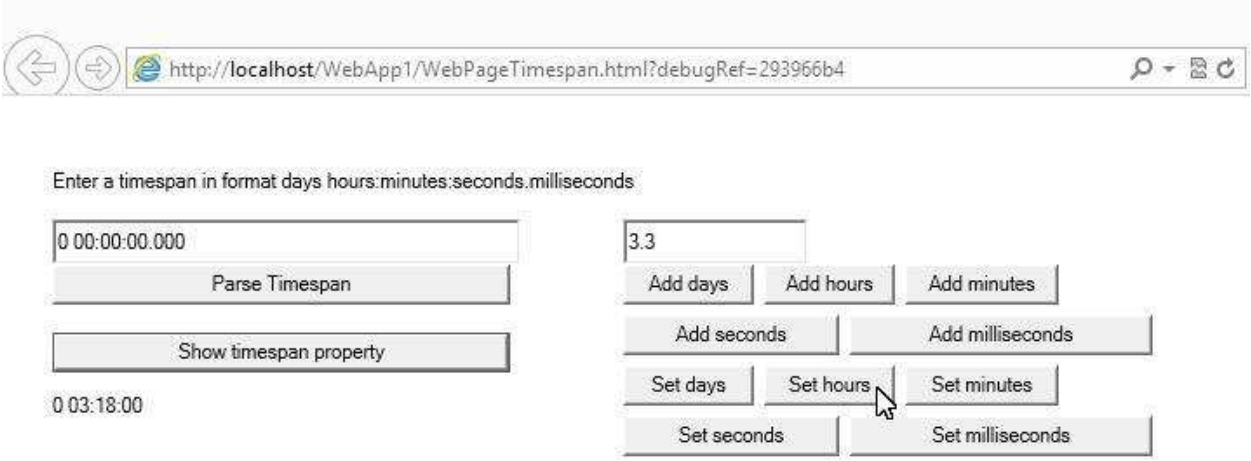


Launch the project and show the web page. Enter a value and click “Set days”. Then click “Show timespan property”:



We can see that 2.5 days is shown as 2 days and 12 hours.

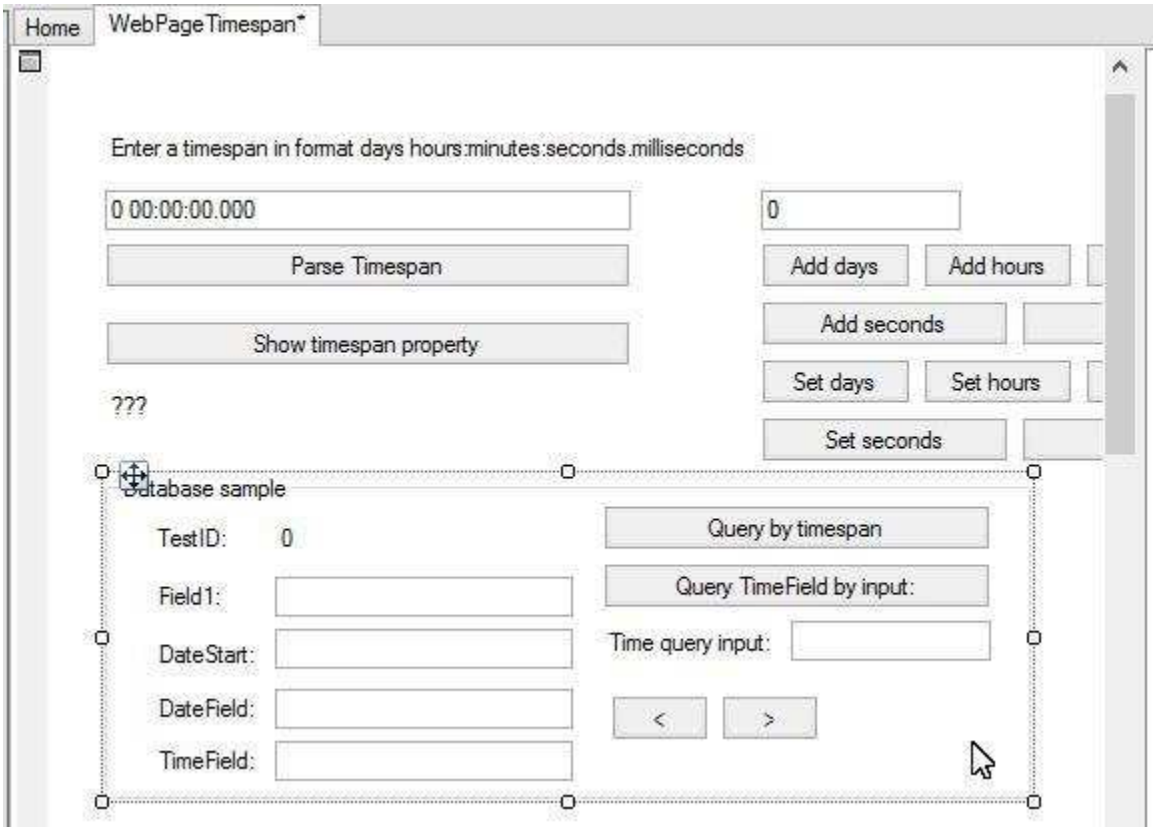
Let’s try another value:



We see that 3.3 hours is shown as 3 hours and 18 minutes.

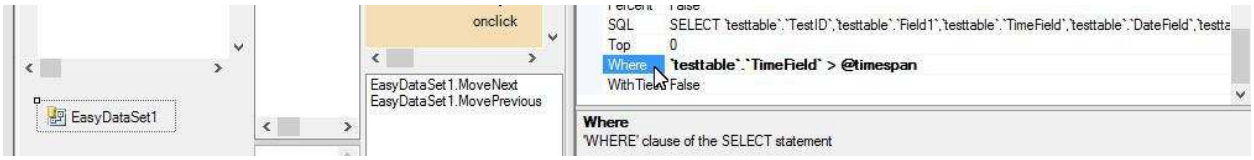
Use timespan in databases

A TIME field of a MySQL database will be represented by a timespan object in your programs. We may query and calculate with timespan fields. Let's use an EasyDataSet to query a database and use data-binding to show data on the page.



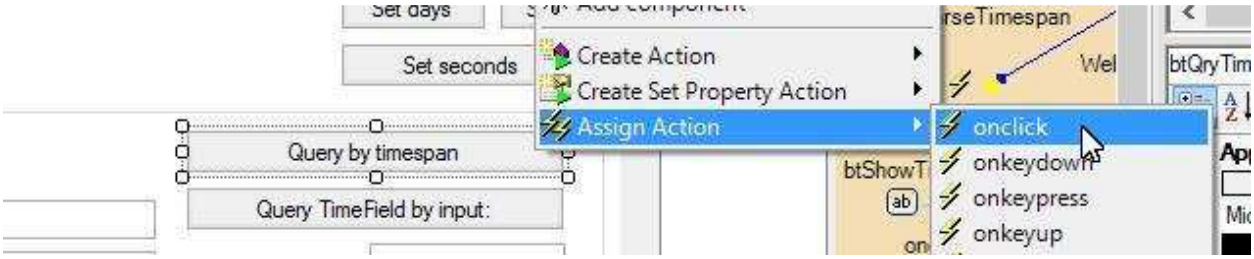
Query with timespans

We use two query samples. The EasyDataSet we use on the web page includes a query parameter @timespan:

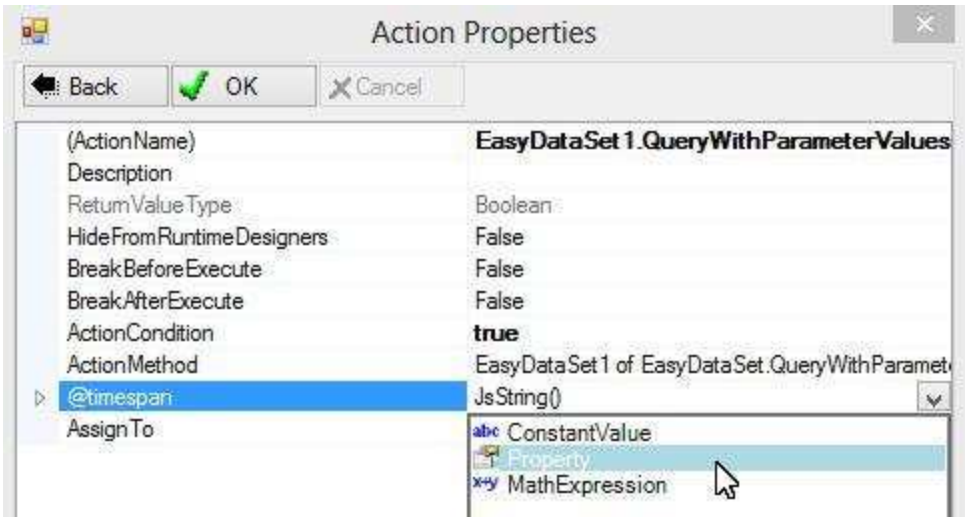


For web database programming, see http://www.limnor.com/support/webDatabaseProgramming.pdf.

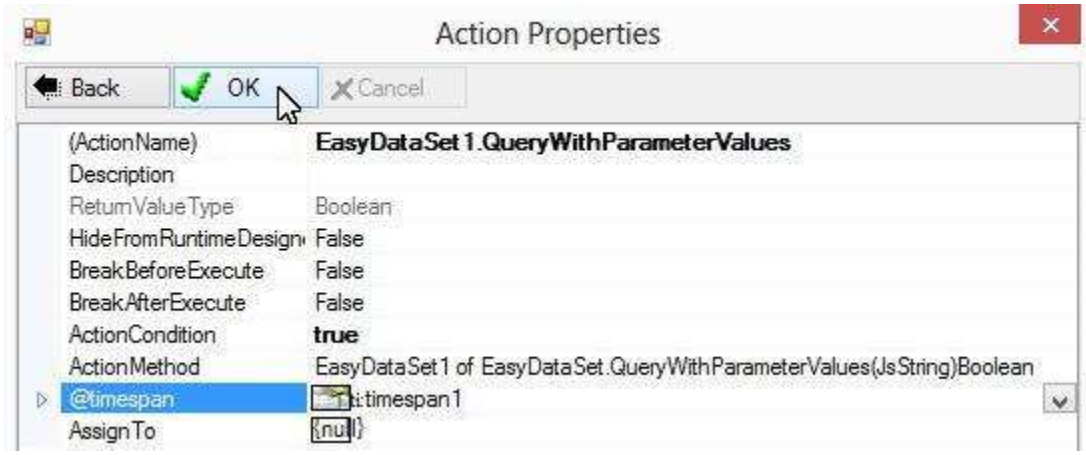
We create one action using property timespan1 for query parameter @timespan; create another action using user input from a text box for @timespan.



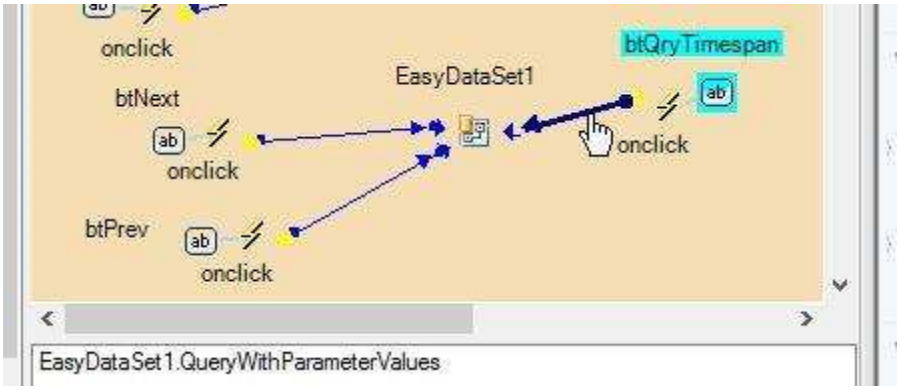
Use property timespan1 for query parameter @timespan:



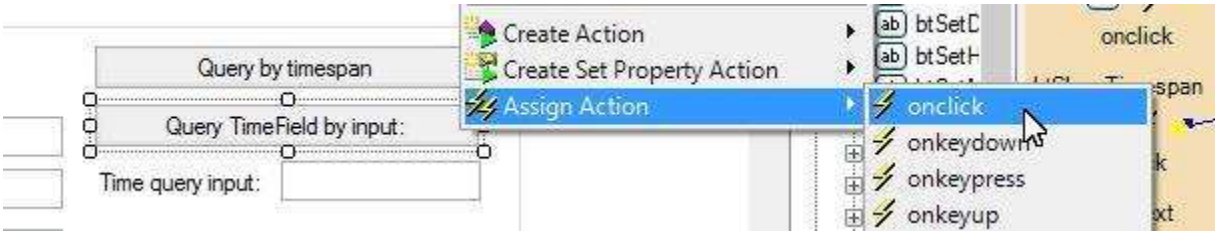
Click OK to create the action:

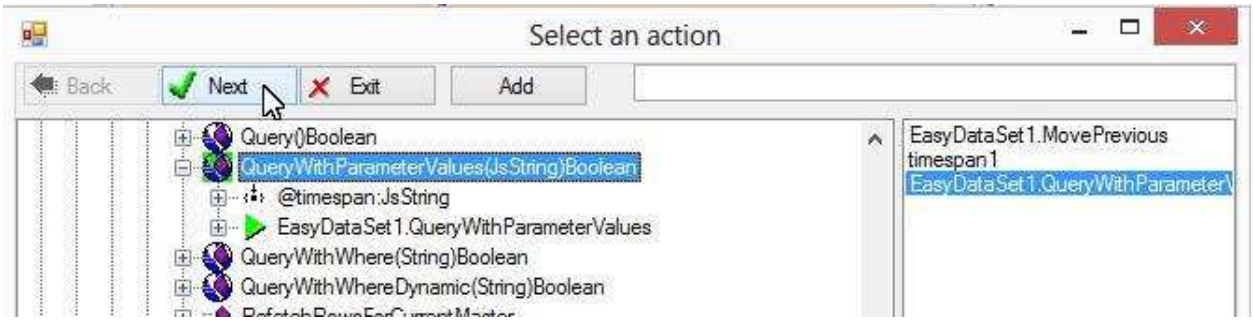


The action is created and assigned to the button:

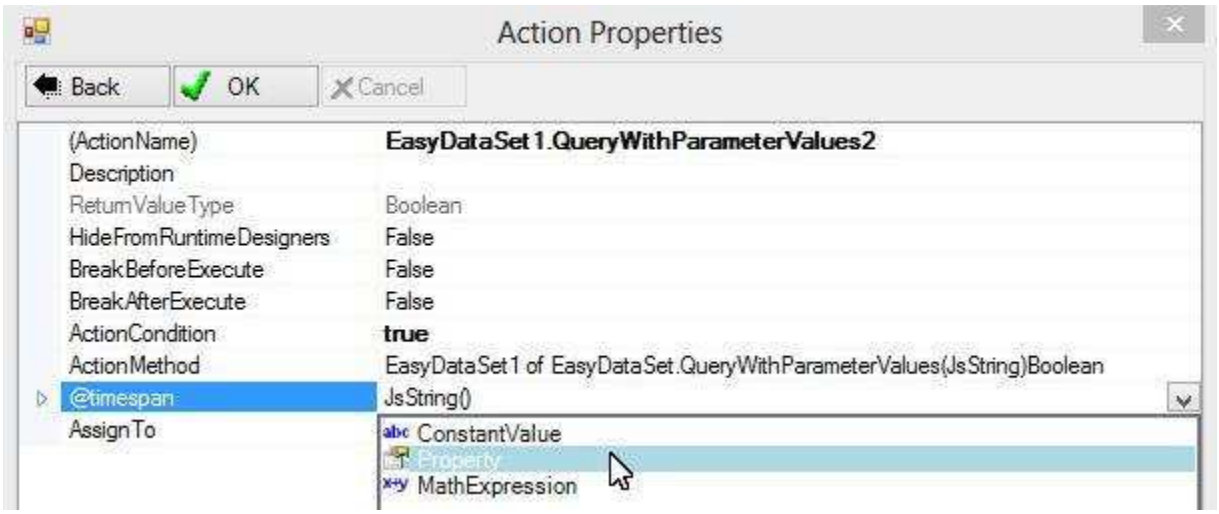


Program the other button:

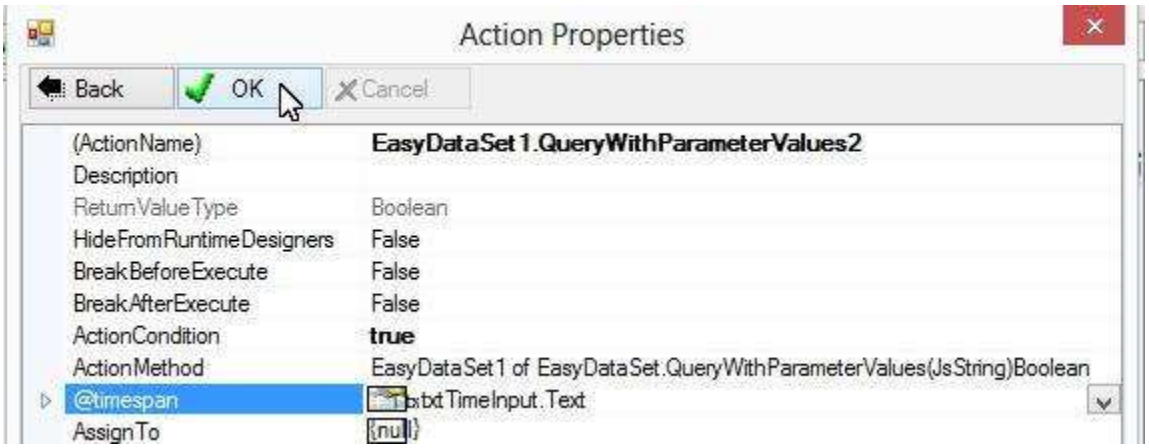




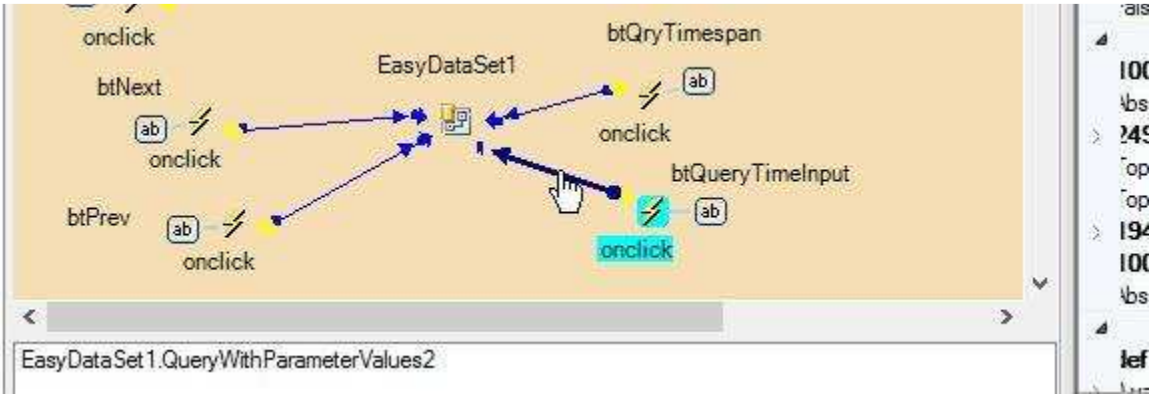
Choose Text property of the text box for query parameter @timespan:



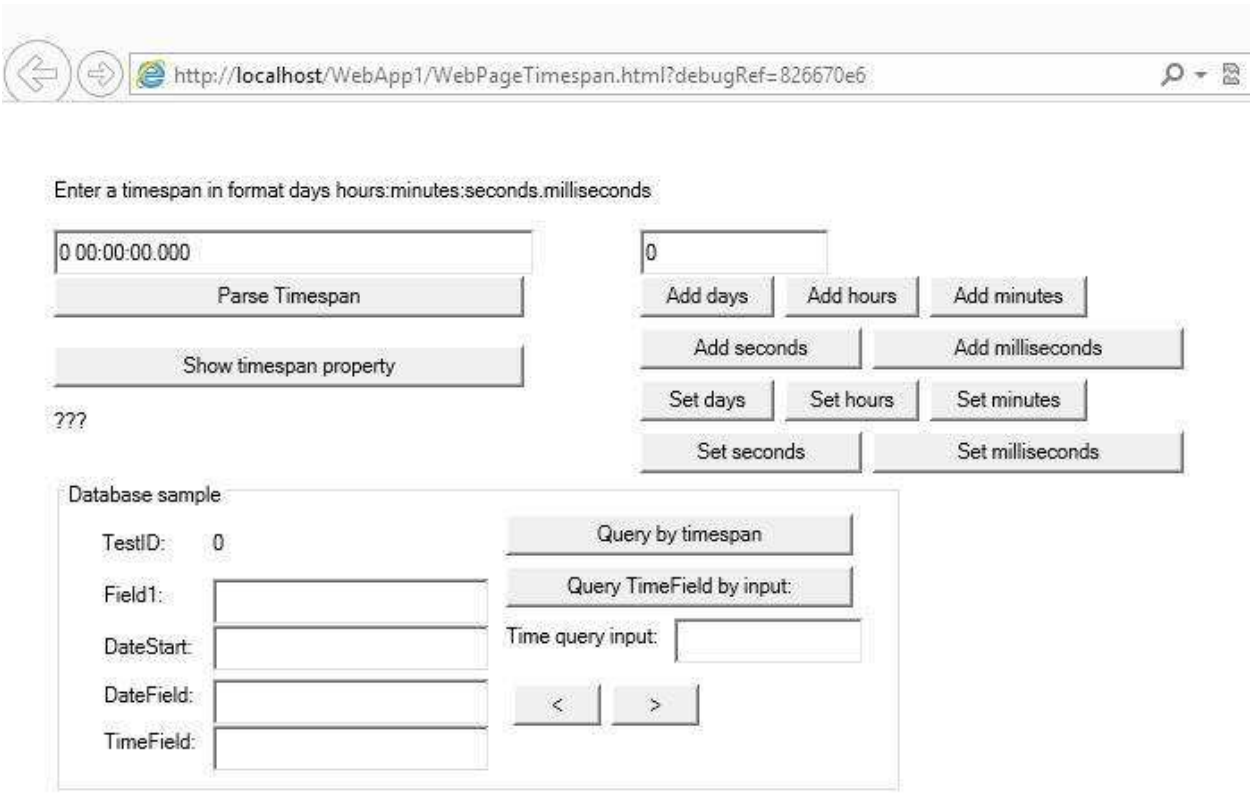
Click OK to create the action:



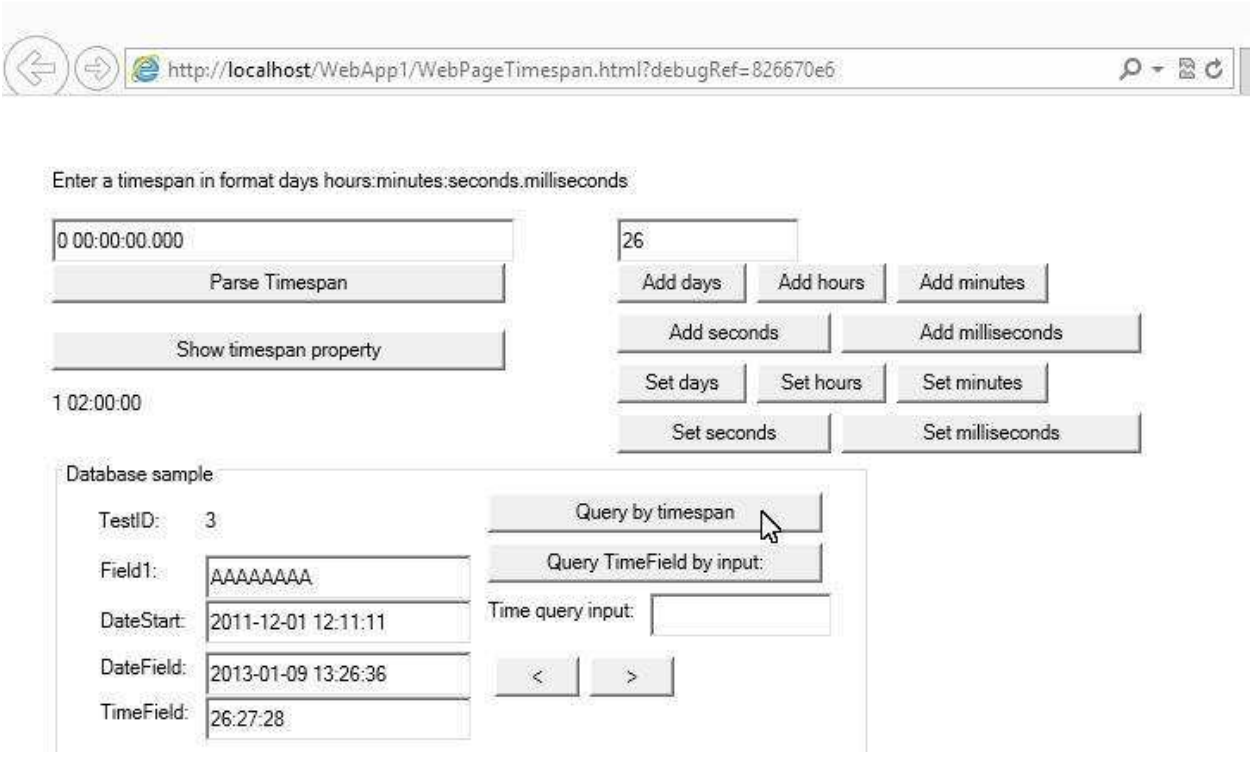
The action is created and assigned to the button:



Launch the web page.

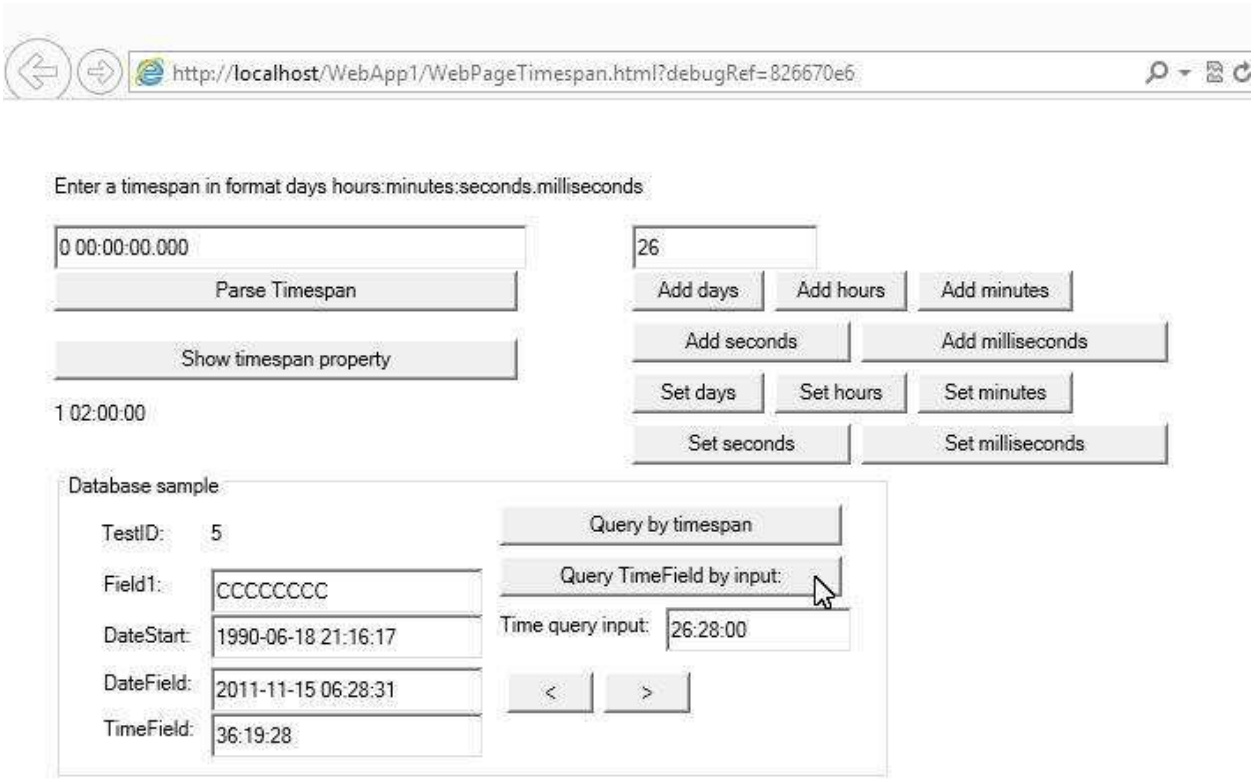


Set timespan1 to 26 hours; click “Query by timespan”:



We can see that records with TimeField greater than 26 hours are returned.

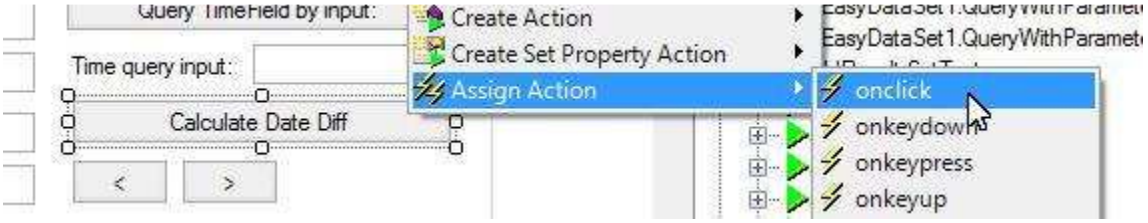
Enter a time very into the text box and click “Query TimeField by input”:

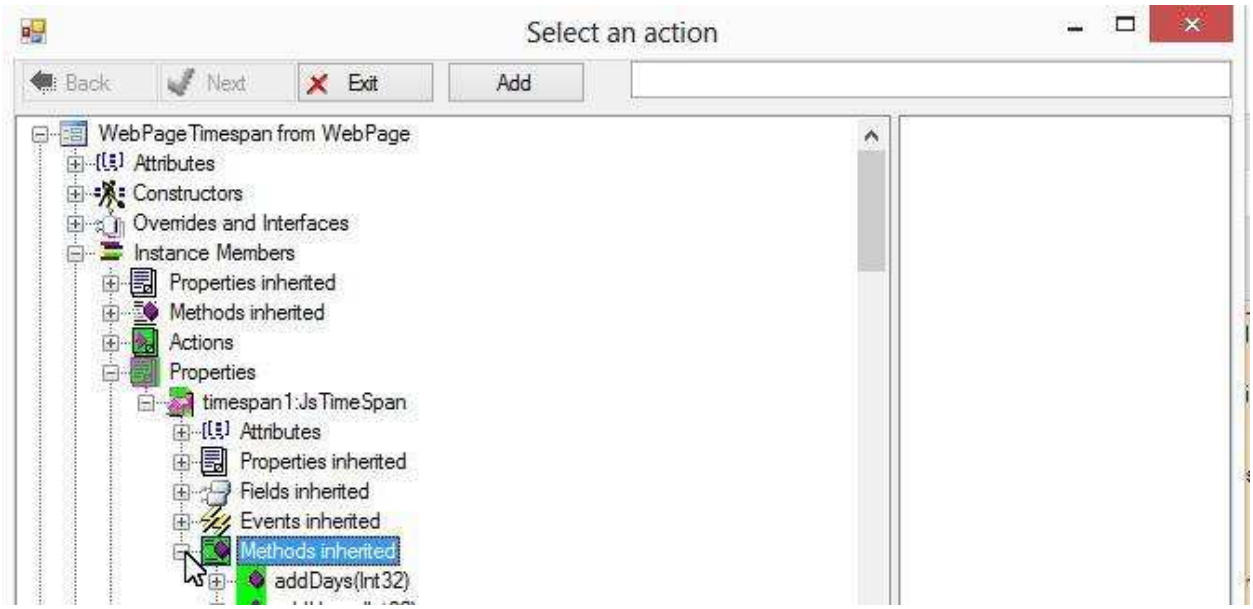


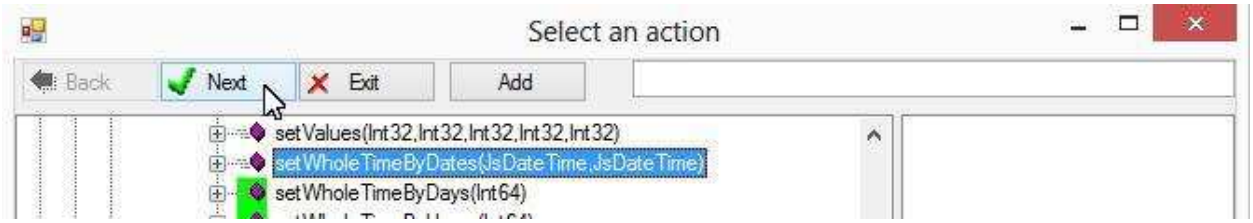
We can see that records with TimeField greater than input are returned.

Calculate date difference

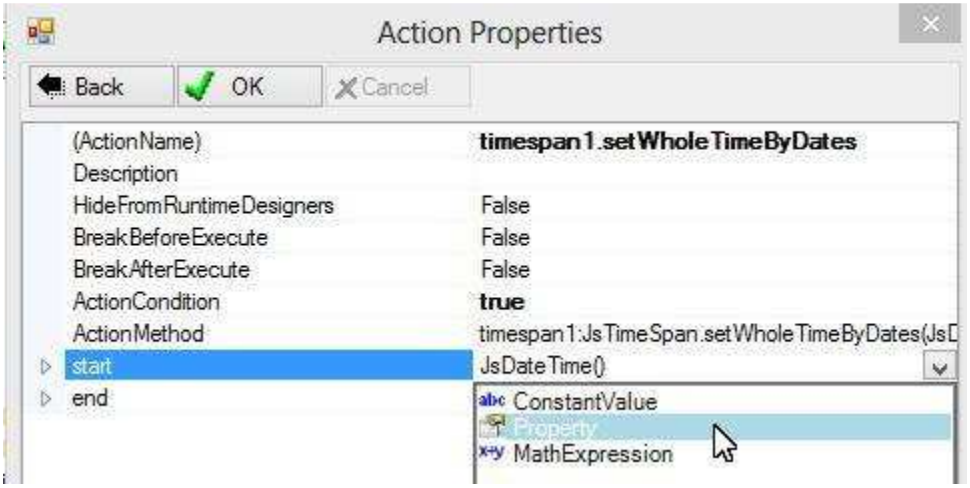
In this sample, fields DateStart and DateField are two date time fields. We may calculate the time difference between these two dates. We use a button to trigger the calculation.

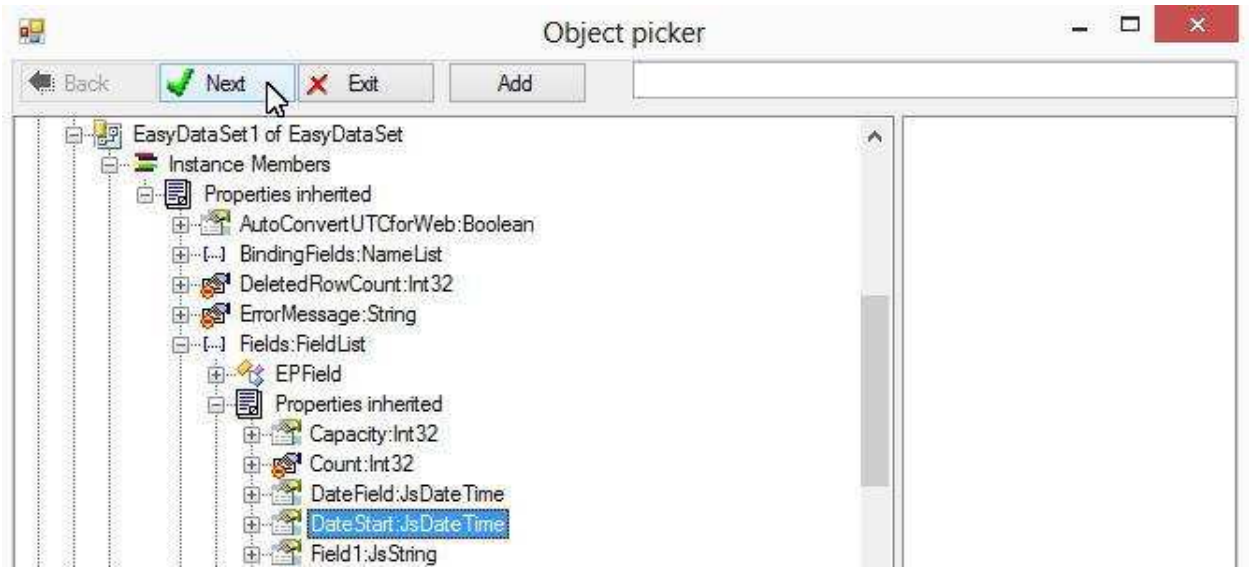


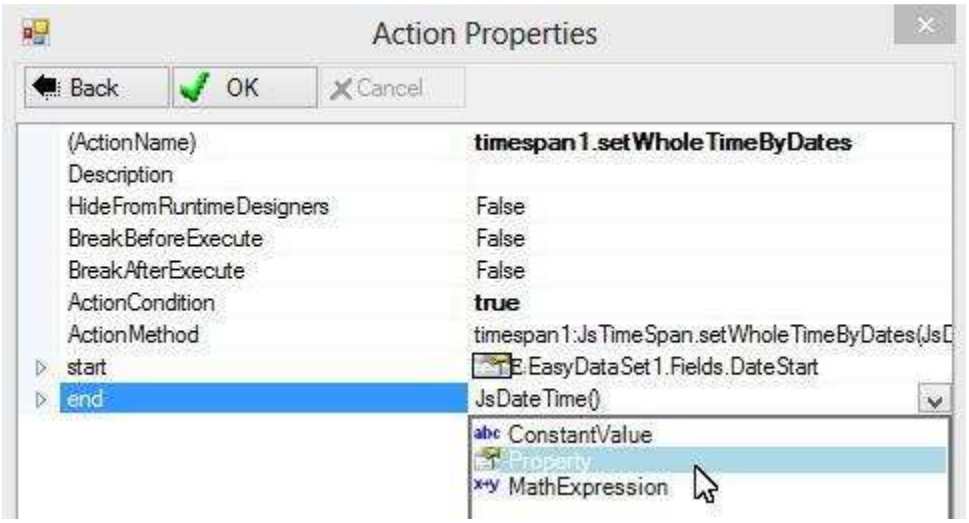


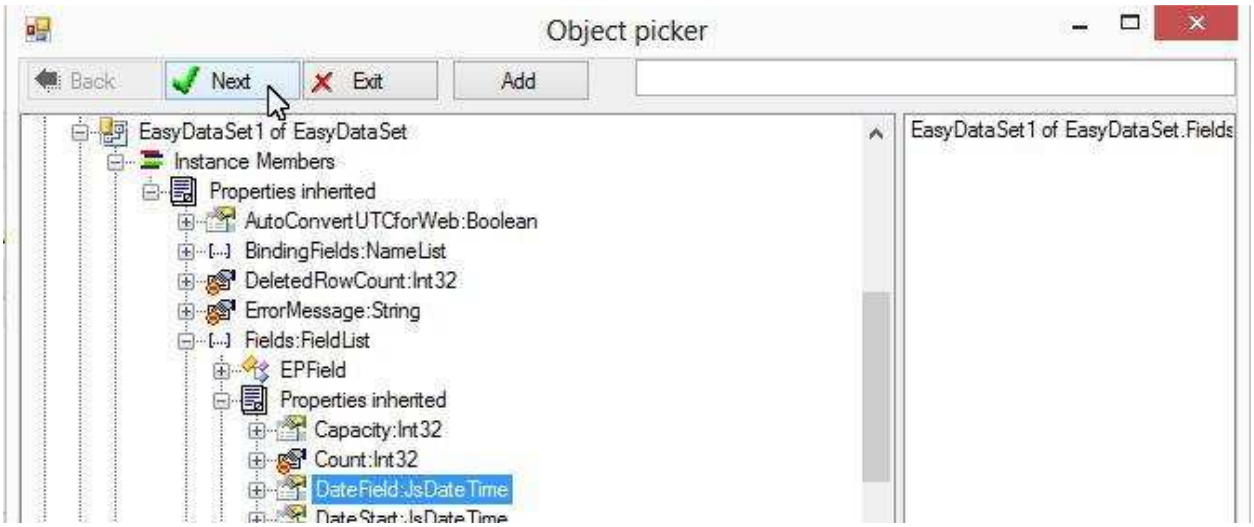


Pass the fields into the action:

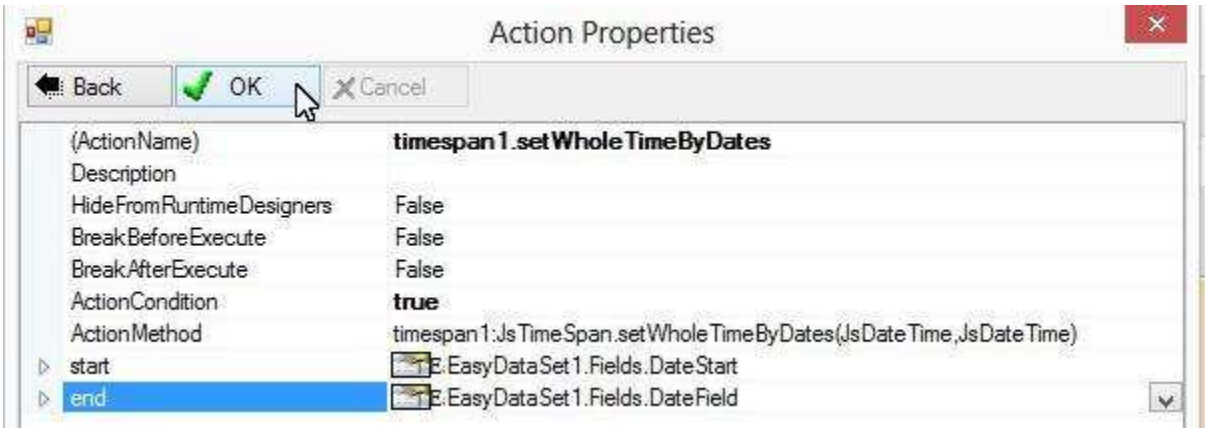




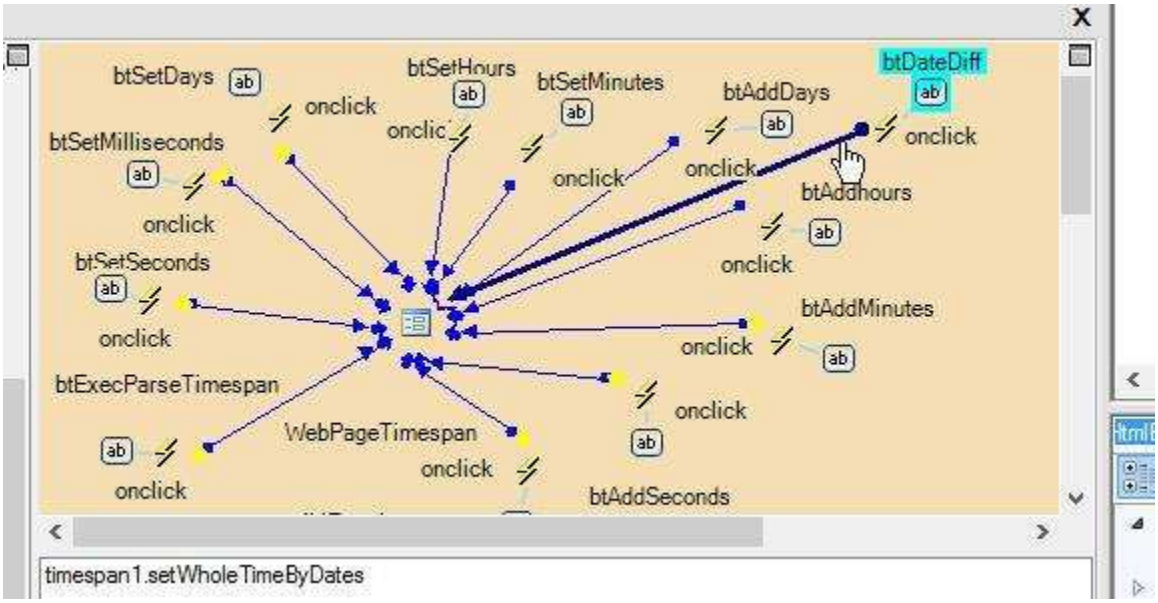




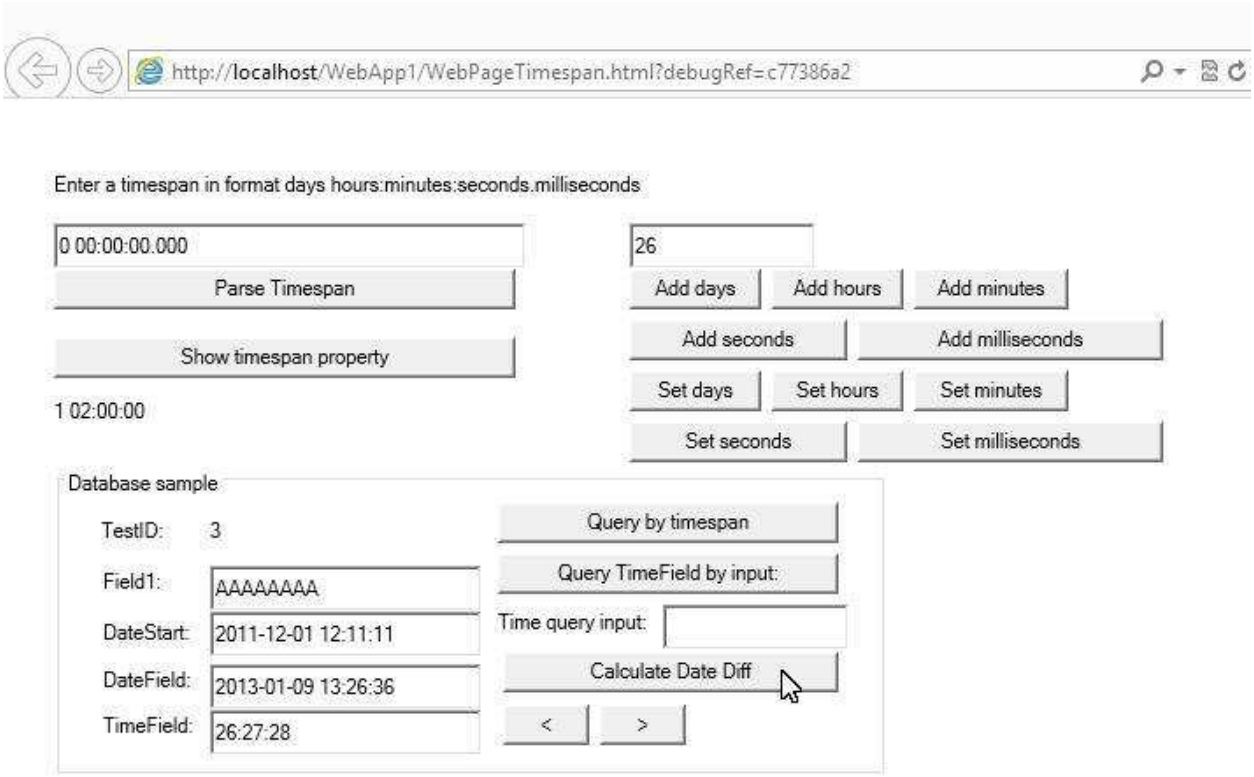
Click OK to create the action:



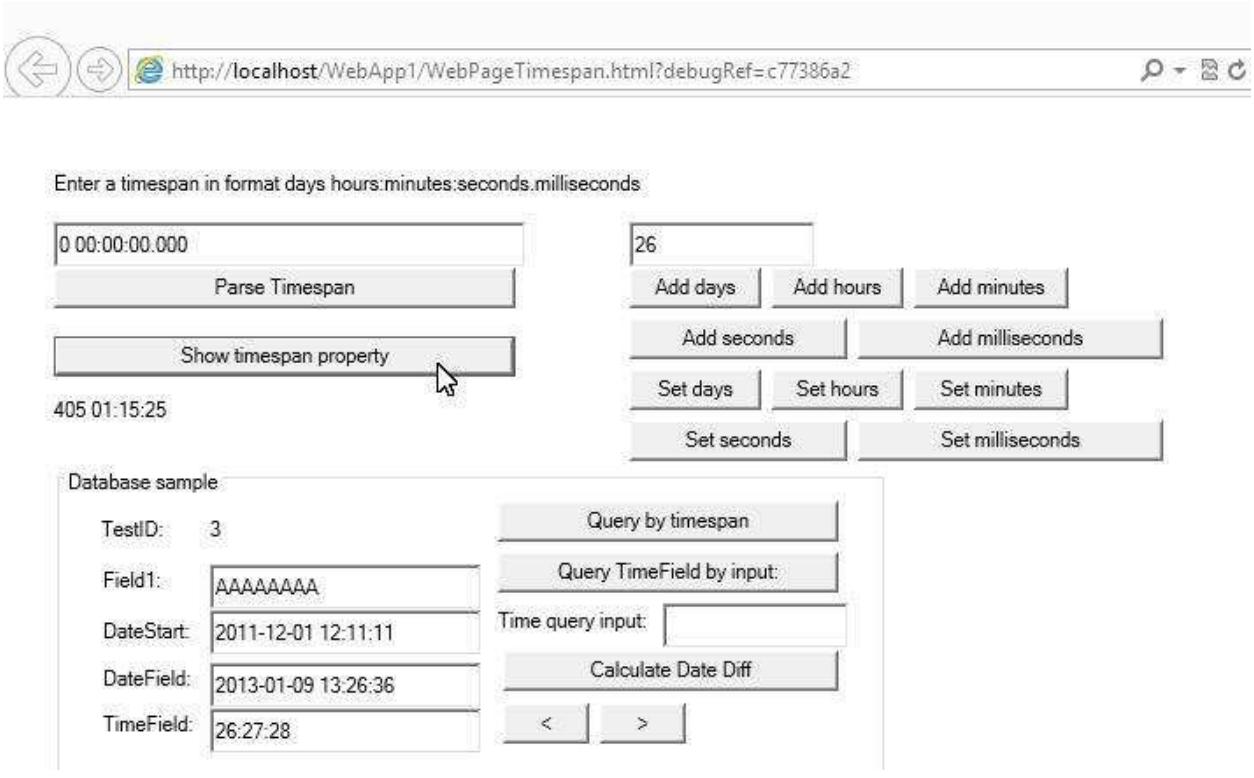
The action is created and assigned to the button:



Launch the web page. Query the database. Click “Calculate Date Diff”



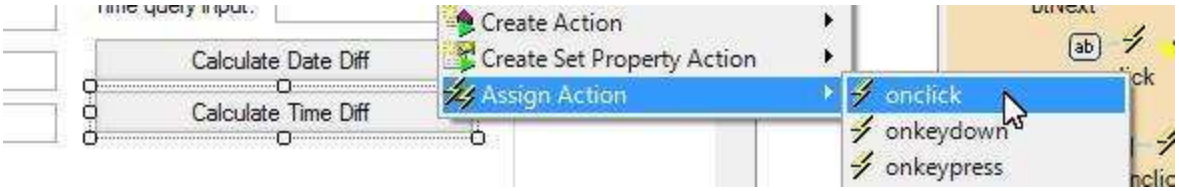
Click “Show timespan property”:



We can see that the timespan between the two dates is 405 days one hour 15 minutes and 25 seconds.

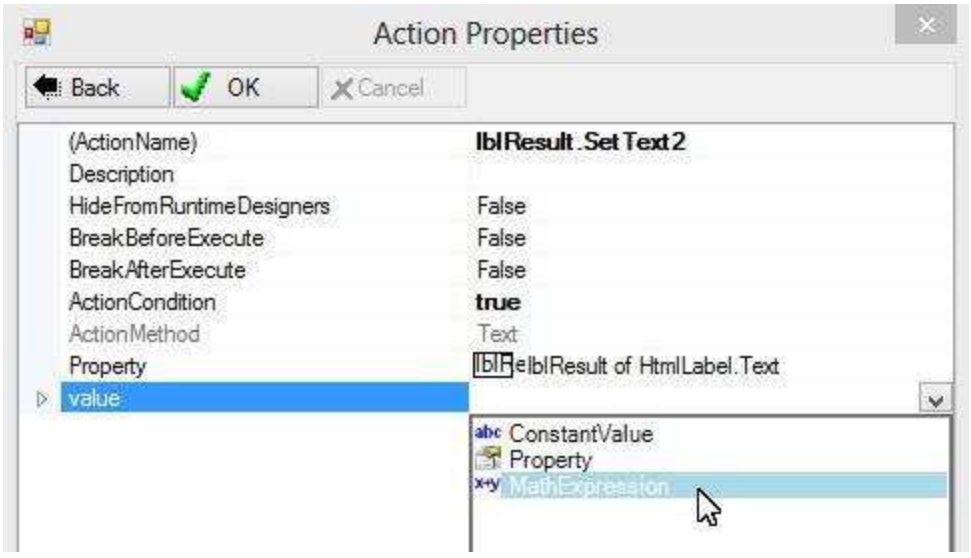
Calculate timespan differences

A timespan object may calculate its difference with another timespan object and express the difference in days, hours, minutes, seconds, milliseconds, or in another timespan object. Suppose we show on a label, in minutes, the time difference between property timespan1 and the database field TimeField; and we use a button to trigger the calculation:

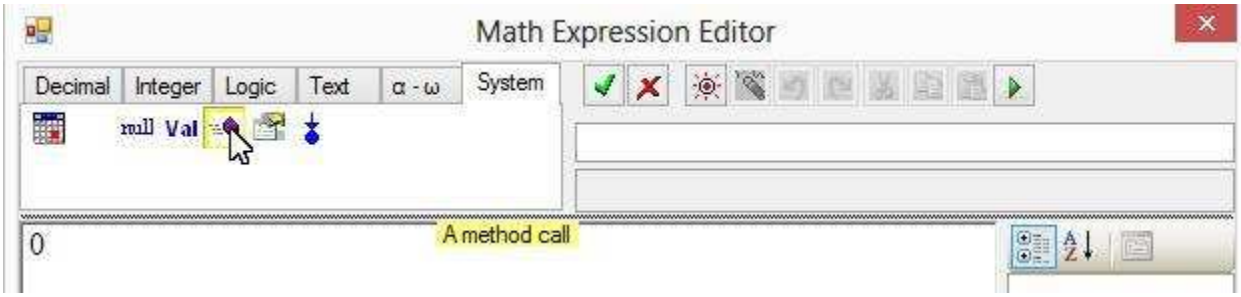


Select Text property of the label for setting the time difference on it:

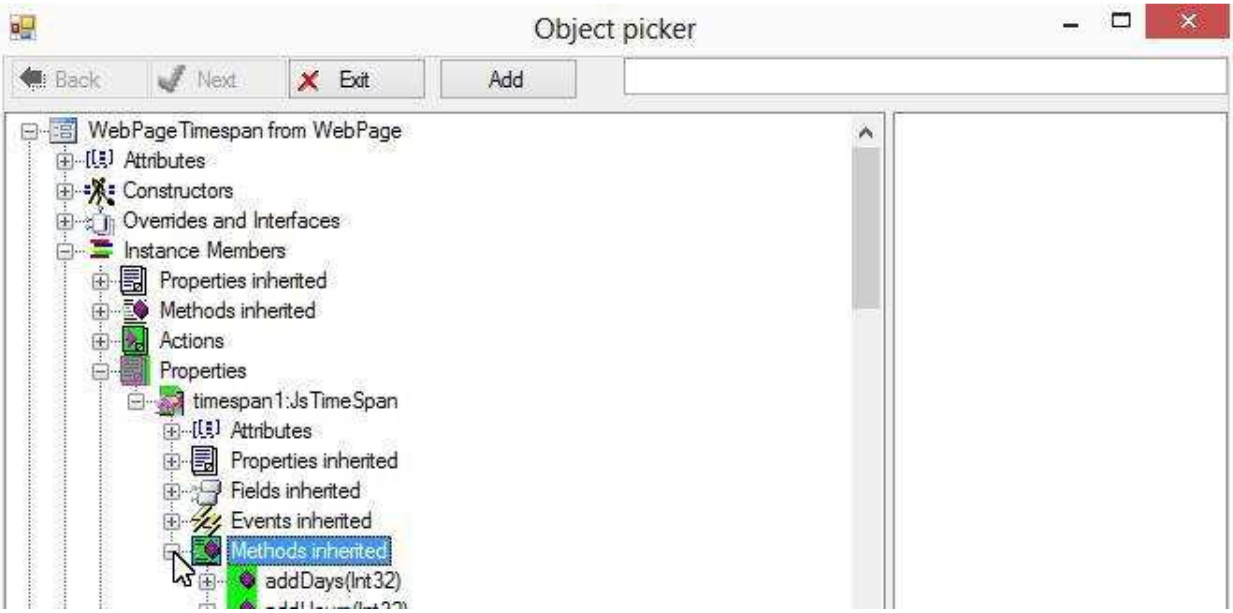
For “value” of the action, select “Math Expression” so that we may execute methods:

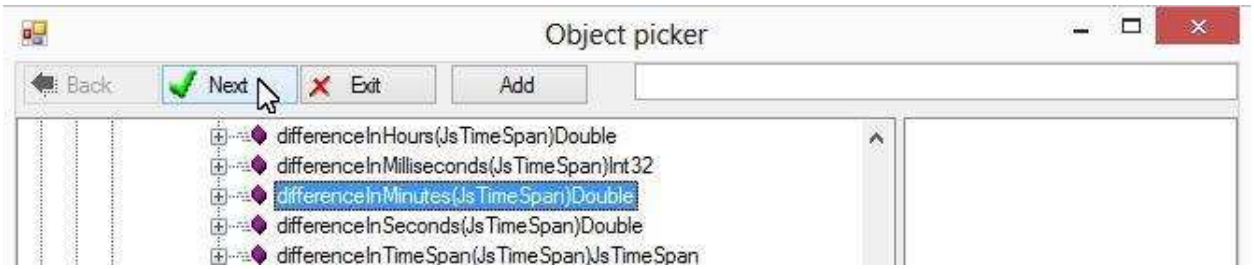


Click the method icon to execute a method:

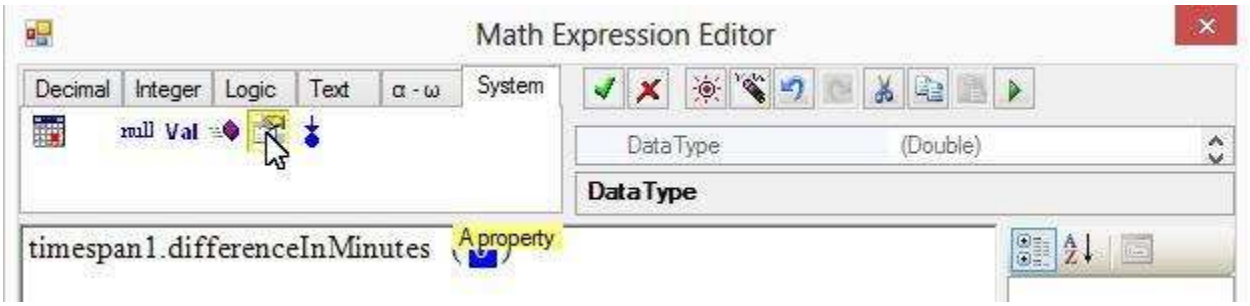


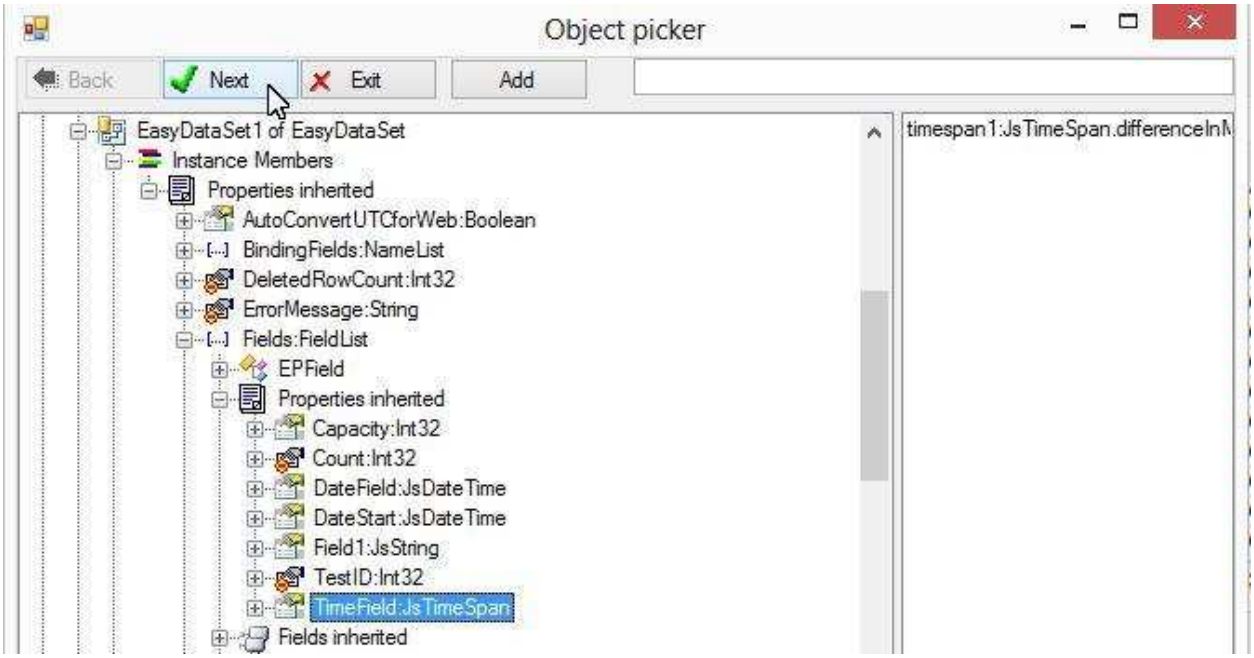
Select "`differenceInMinutes`" of timespan1:

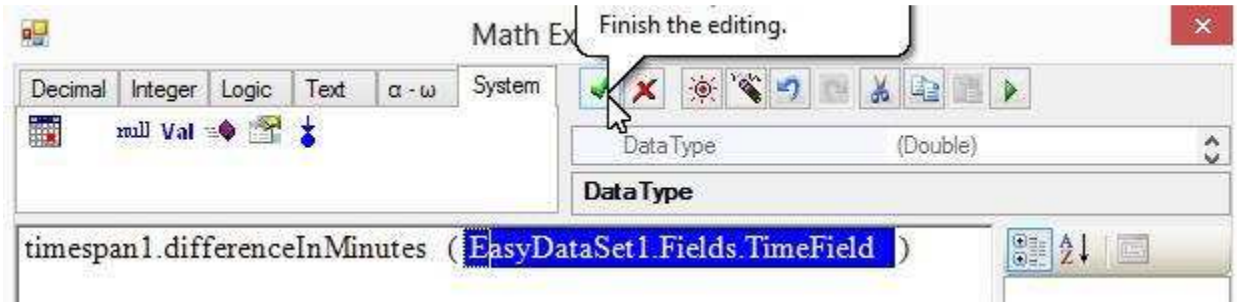




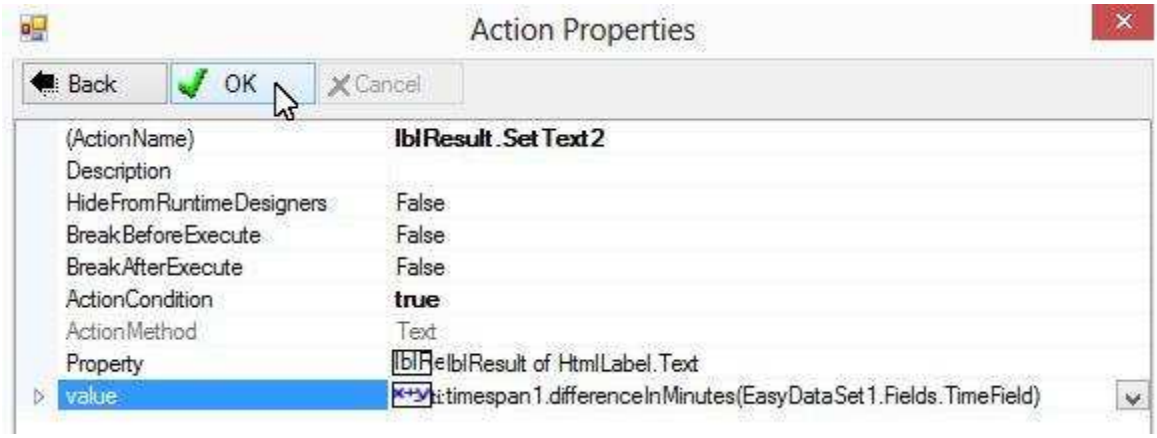
Select the method parameter and click the property icon to select the TimeField of the EasyDataSet:



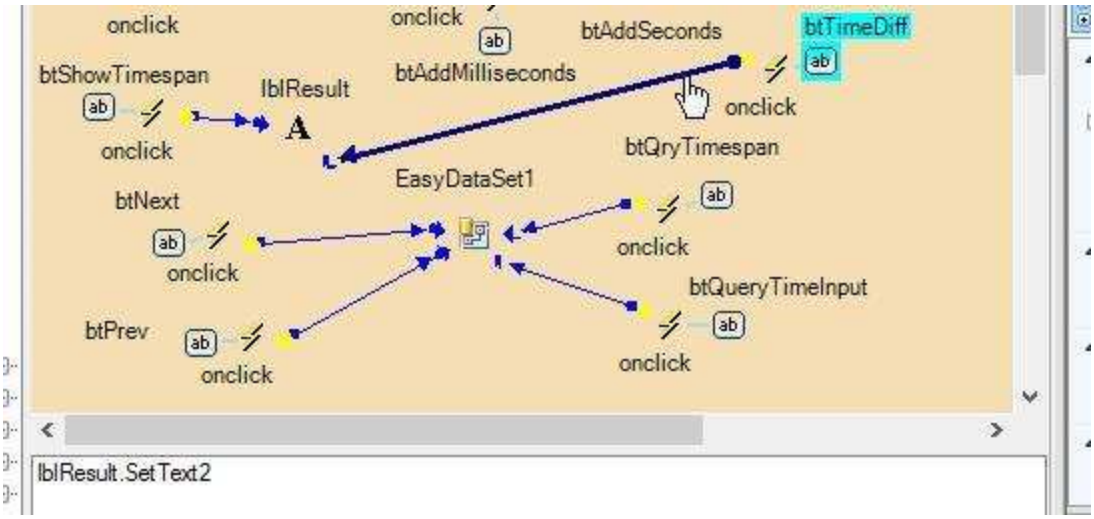




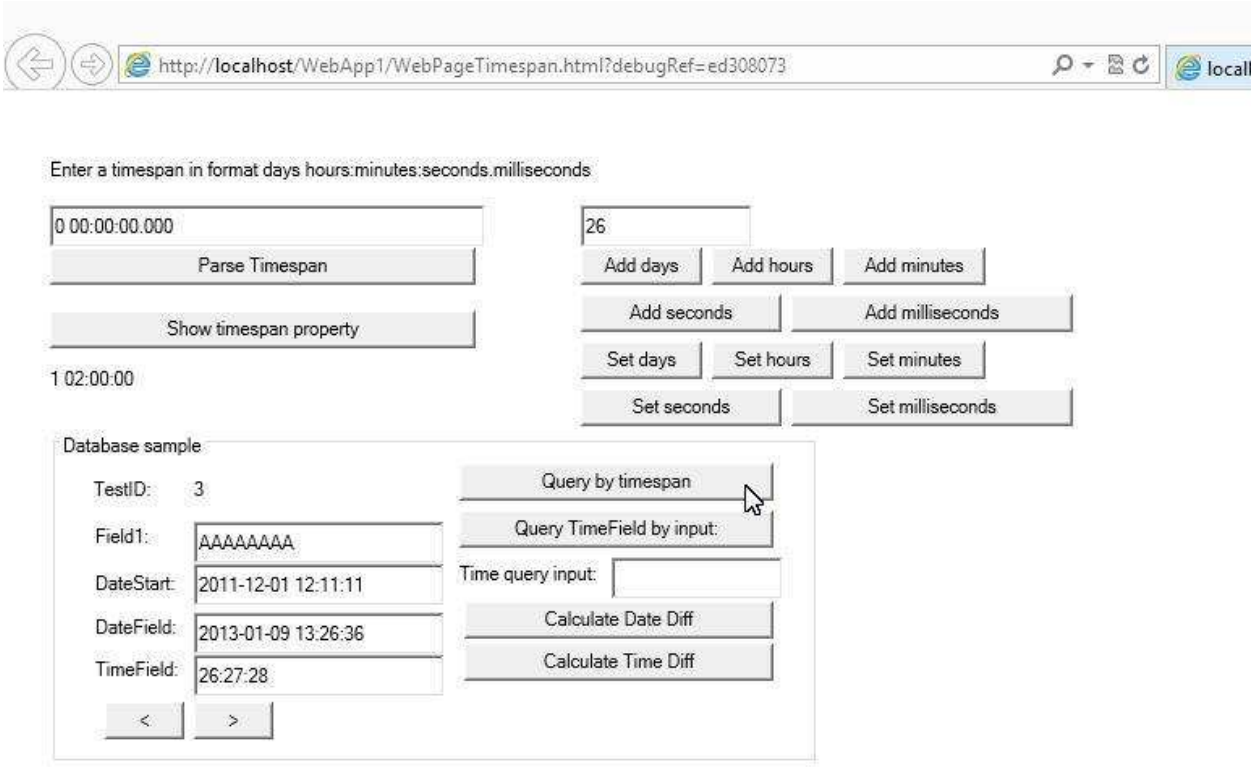
Click OK to create the action:



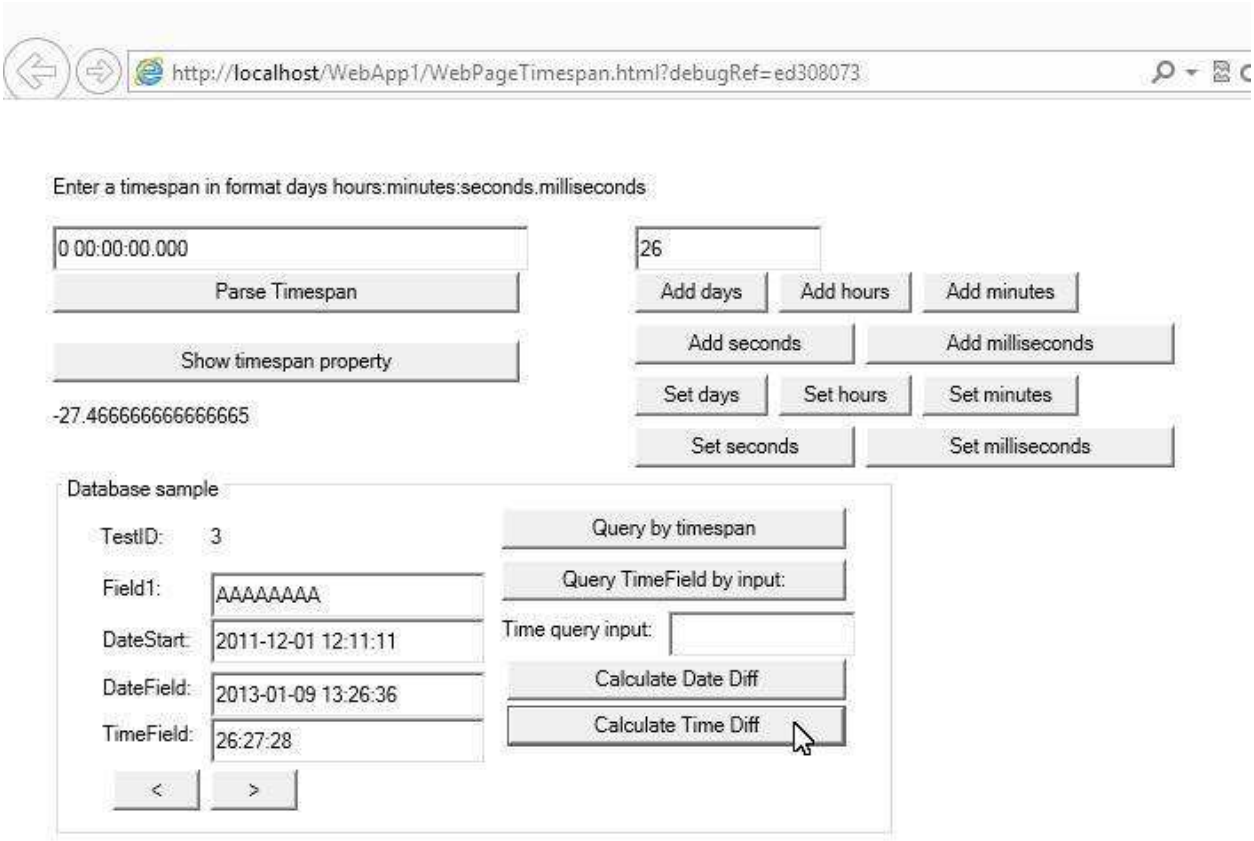
The action is created and assigned to the button:



Launch the web page and query the database:



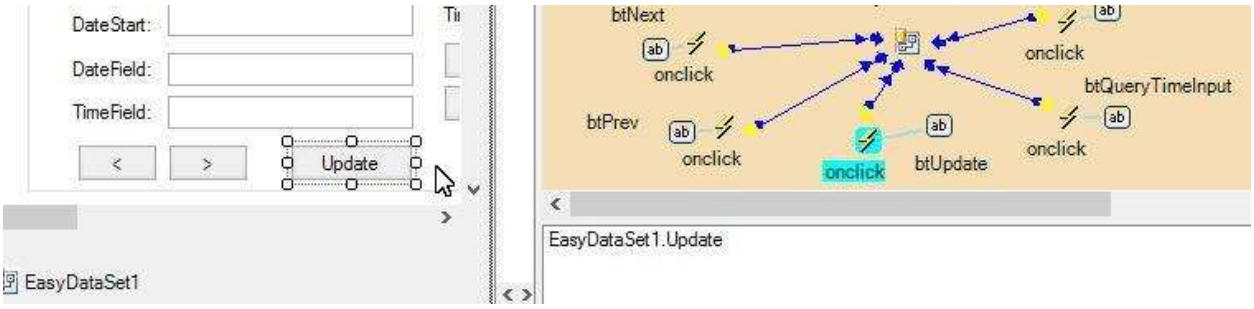
Click “Calculate Time Diff” to show timespan difference in minutes:



We see that the result is -27.46666666 minutes between “26:27:28” and “26:00:00”, because 28 seconds is 0.46666666 minutes.

Modify Time field

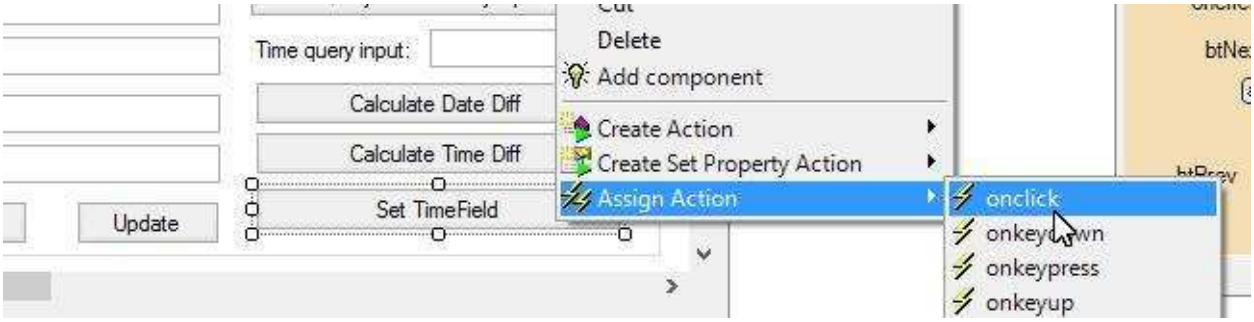
To save data modifications back to the database, we use a button to execute an Update action.



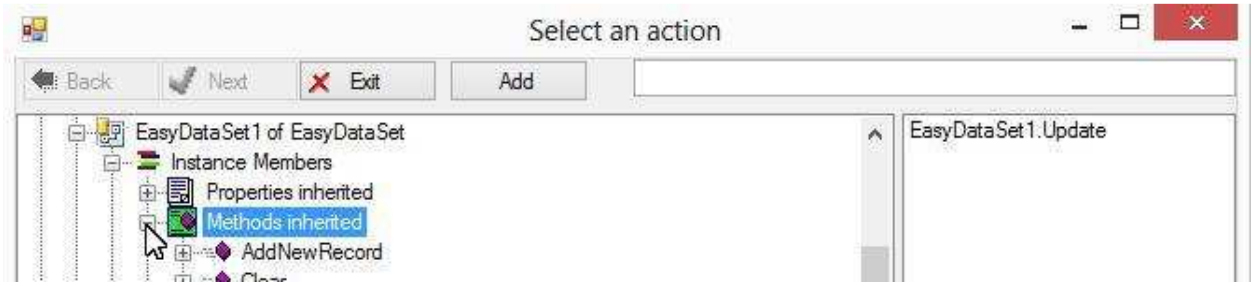
A web visitor may simply modify the value on the text box bound to a TIME field.

Your web program may also set the TIME field of the EasyDataSet programmatically, with a timespan object.

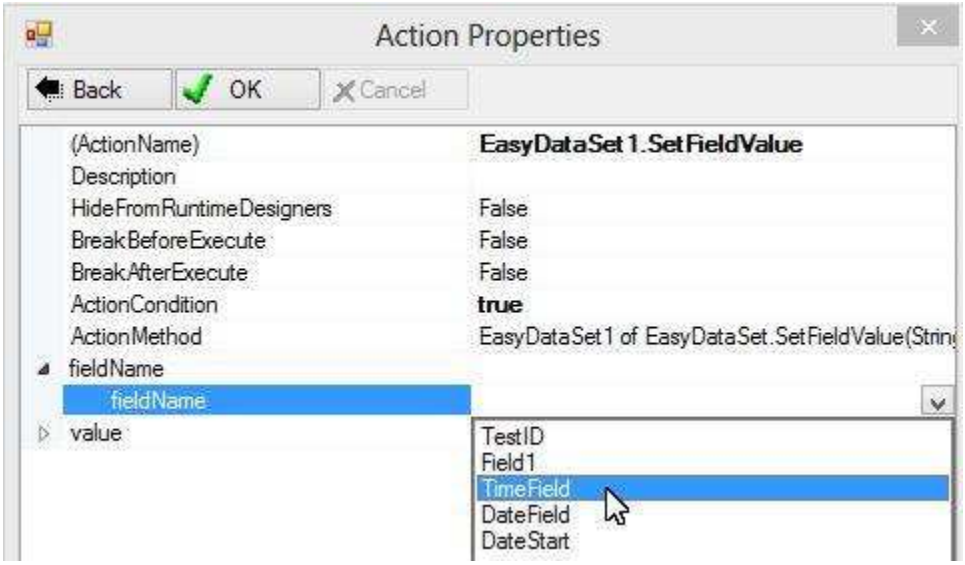
Here we use a button to pass timespan1 to field TimeField.



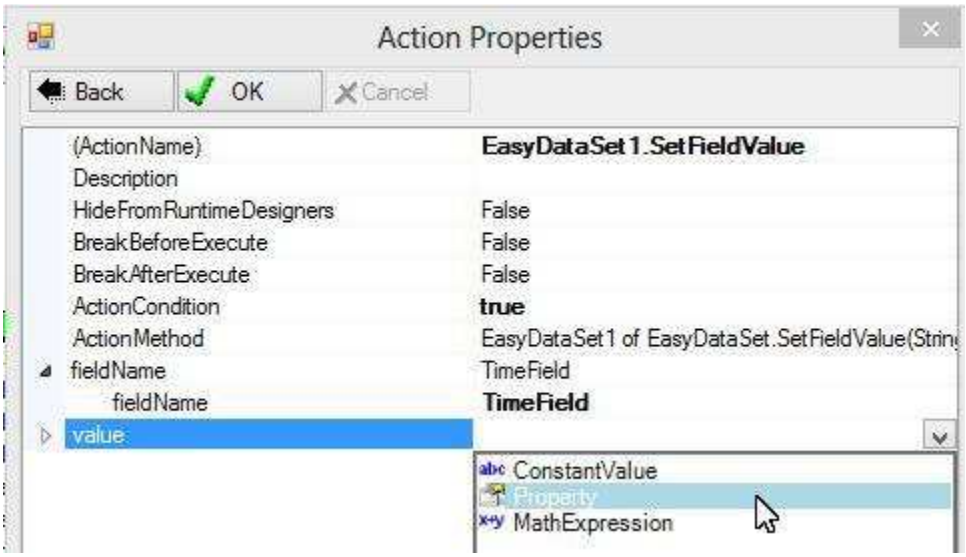
Select SetFieldValue of the EasyDataSet

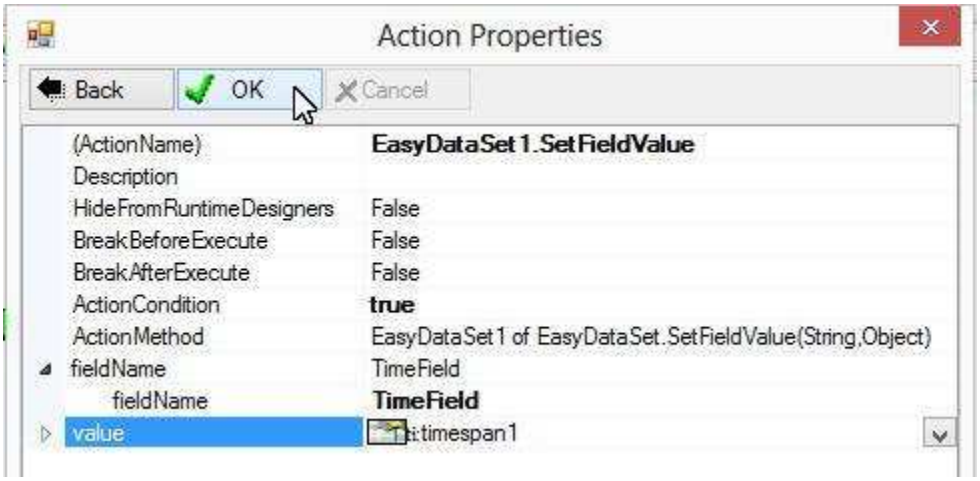


Select TimeField:



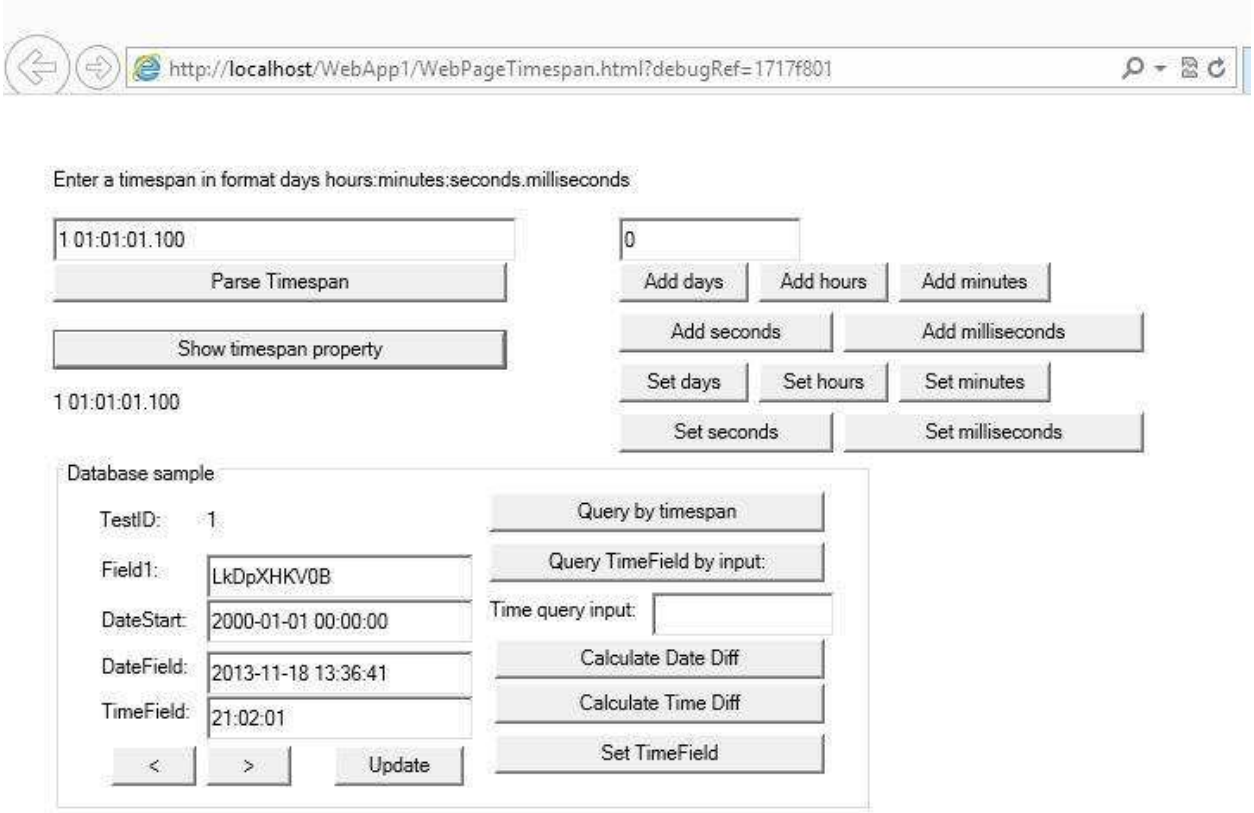
Select timespan1 for the value:



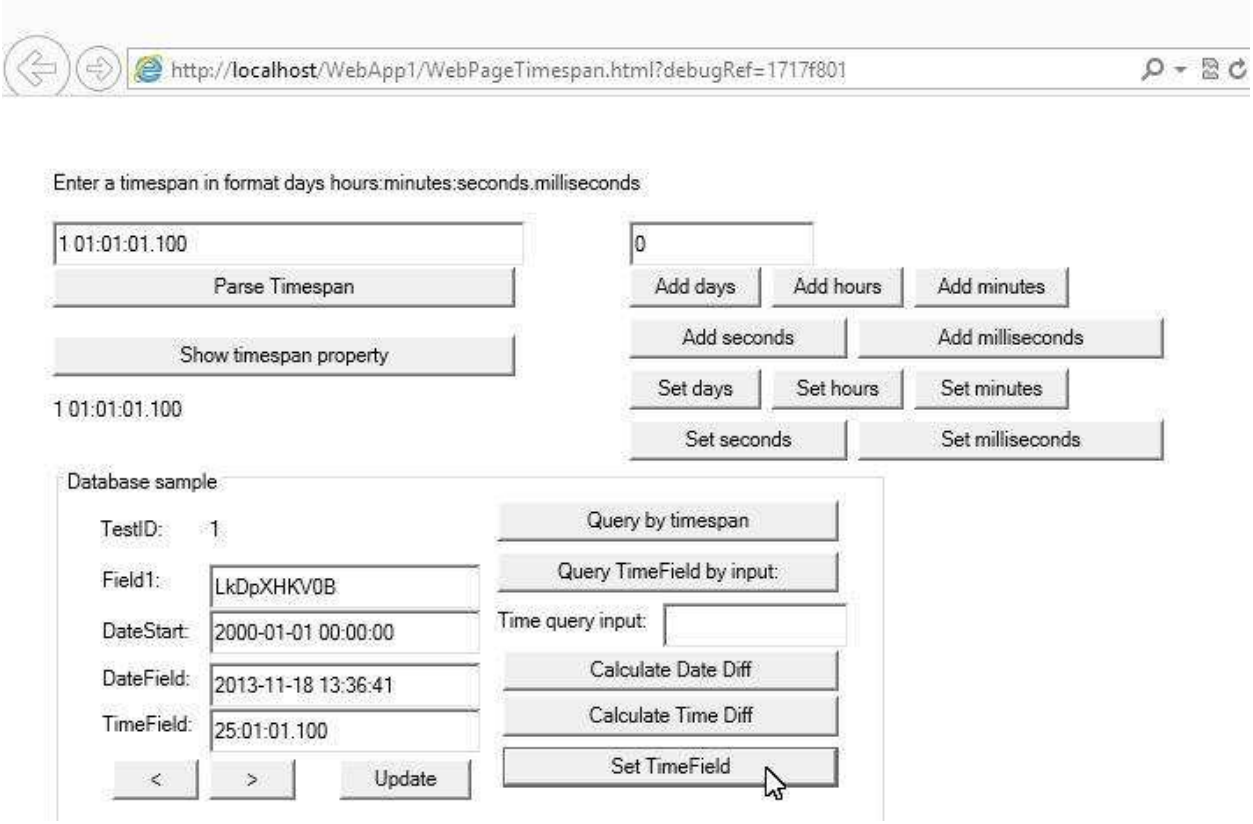


The action is assigned to the button.

Launch the web page, query the database, set value of timespan1:



Click “Set TimeField” to pass timespan1 to TimeField:



We see that the text box bound to TimeField also shows the modification.

Click “Update” button, the modifications will be saved to the database.

Feedbacks

Please send your feedbacks to support@limnor.com. Thanks!