Using timespan to record construction labor

<http://cb-ot-devst05.ad.wsu.edu/featherman/NewFor2018/PayrollwithTimespan.aspx>

This program starts the look at timespan which is both a simple variable to use in a limited manner and a very in-depth data type that can provide a ton of functionality.

We start with the simple usage in this project and the next project shows off more of the functionality.

We use timespan in the following manner;

Timespan variable = clock out time for shift – clock in time for shift.

Timespan then can hold a number that is the result of a calculation where you subtract a date or time from one calendar control from another calendar control. For example you can subtract 3:30pm – 8:00 am to get the number of hours worked on a shift. You can subtract March 15th – March 8th to get the number of days a car was checked out. There are a ton of uses. You may even wonder how we got this far and did not use timespan. We can also see what day of the week the labor was recorded to see if overtime is justified (worked on a weekend).
Just be sure NOT to call your new project timespan, that keyword and all keywords are reserved.

Code:

'This progam hopefully gives an idea how useful it is to work with calendars and performing calculations with timespan. The start of a labor timekeeping system is kept here for a project manager.

Partial Class NewFor2018\_PayrollwithTimespan

 Inherits System.Web.UI.Page

 'we will keep track of how many labor records are recorded using this global variable (ie transaction 5)

 Public Shared gintTransactionNumber As Integer

 'The requirement was to keep track of labor by labor grade. there are 4 labor grades, framers, electricians, plumbers, and gen helpers.

 Public Shared decTotalFramers, decTotalElectricians, decTotalPlumbers, decTotalGenHelpers As Decimal

 Protected Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

 Dim decgrosspay, decPayrate As Decimal

 Dim strDate As String

 'here is another start of a timecard keeping system

 If rbllaborgrades.SelectedIndex = -1 Then

 txtOutput.Text = "Select a labor type"

 Exit Sub

 End If

 If txtStart.Text = Nothing OrElse txtEnd.Text = Nothing OrElse txtDate.Text = Nothing Then

 txtOutput.Text = "Check entries for time and date"

 Exit Sub

 End If

 'increment the labor record number

 gintTransactionNumber += 1

 'here we use the newer textboxes that are rendered as calendars and have a time input. This next line is to just format a nice date for the output

 strDate = DateTime.Parse(txtDate.Text).ToShortDateString

 'the construction firm pays overtime. Hours worked on Saturday are at time and 1/2, Sunday labor rate per hour is double the weekly rate.

 Select Case Weekday(DateTime.Parse(txtDate.Text))

 Case 1 'sunday

 decPayrate = (rblLaborgrades.SelectedValue \* 2)

 Case 2 To 6 'weekday

 decPayrate = rblLaborgrades.SelectedValue

 Case 7 'saturday

 decPayrate = (rblLaborgrades.SelectedValue \* 1.5)

 End Select

 'here is the core of the new content. Timespan is a datatype that can be instantiated like any other variable. Here we create an instance of timespan with the name ts. So ts is a variable that can hold the result of a time or date based calculation. We have to pull the date out of the textbox using datetime.parse(textbox.text)

 Dim ts As TimeSpan

 ts = DateTime.Parse(txtEnd.Text) - DateTime.Parse(txtStart.Text)

 'which makes this an eezie peezie line of code multiply the number of hours worked times the labor rate.

 decgrosspay = ts.TotalHours \* decPayrate

 txtOutput.Text &= "Record " & gintTransactionNumber & ": " & strDate & " " & rblLaborgrades.SelectedItem.Text & " worked " & ts.TotalHours & " hours on the job: " & " costing " & decgrosspay.ToString("C") & vbNewLine

 'here we increment the global variable to kee a running tally of the # hours worked by labor grade. We could have used a different term in the select case using the selecteditem.text. While the current is less typing, using selectediten.text does future proof your code. What if you add a new labor grade and the order changes? You are encouraged to change it.

 Select Case rblLaborgrades.SelectedIndex

 Case 0

 decTotalElectricians += ts.TotalHours

 Case 1

 decTotalFramers += ts.TotalHours

 Case 2

 decTotalPlumbers += ts.TotalHours

 Case 3

 decTotalGenHelpers += ts.TotalHours

 End Select

 'here is the output for the running totals.

 txtTotals.Text = "Total Hours " & vbNewLine & vbNewLine

 txtTotals.Text &= "Electricians: " & decTotalElectricians & vbNewLine

 txtTotals.Text &= "Framers: " & decTotalFramers & vbNewLine

 txtTotals.Text &= "Plumbers: " & decTotalPlumbers & vbNewLine

 txtTotals.Text &= "General Helpers: " & decTotalGenHelpers

 End Sub

 Protected Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

 'here we can clear the form for further data entry

 rblLaborgrades.SelectedIndex = -1

 txtStart.Text = Nothing

 txtEnd.Text = Nothing

 txtDate.Text = Nothing

 End Sub

End Class