

Visual Web Development

Terry Marris January 2009

18 Dates

We look at some Date methods and the Web Server Calendar.

18.1 The Birthday Book

Forgetting your wife, girlfriend's or mother's birthday is a disaster waiting to happen. We need a birthday book. Some entries in the birthday book might be:

Tom	16 June
Dick	12 May
Harry	5 September
Ann	9 February
Sue	23 July

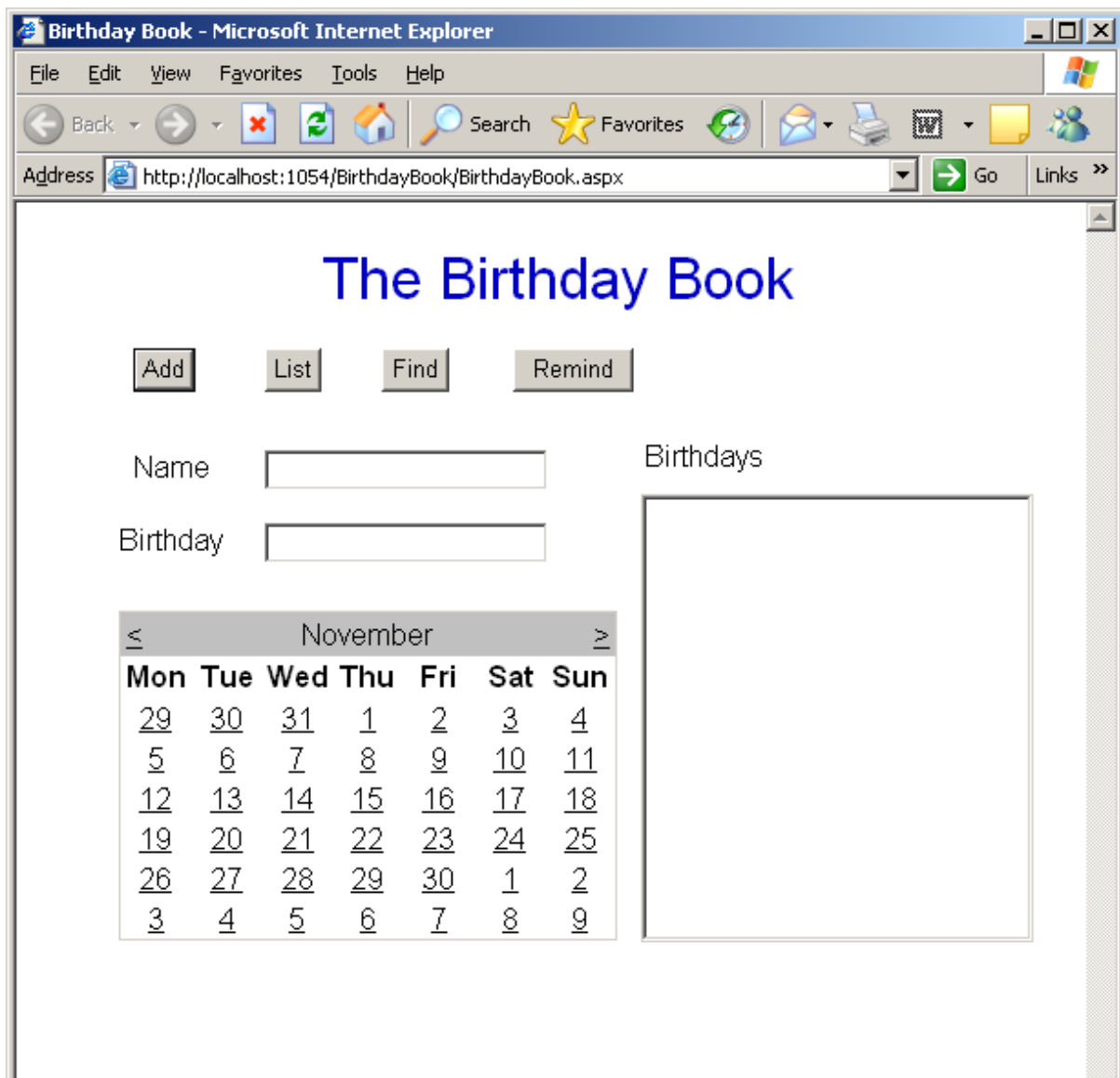
We want to add a new birthday.

Tom	16 June
Dick	12 May
Harry	5 September
Ann	9 February
Sue	23 July
Sam	28 May

We want to find the birthday of a named person: Ann's birthday is 9 February; Maya's birthday is unknown. Maya is not in the birthday book.

We want a reminder when a birthday falls within one week of today. If today is 2 February 2008 we want to be reminded that Ann's birthday is on the 9 February. It would help us if we also knew what day (Monday, Tuesday, Wednesday, ..) Ann's birthday falls on this year so that we can buy her a birthday card.

The interface looks something like this.



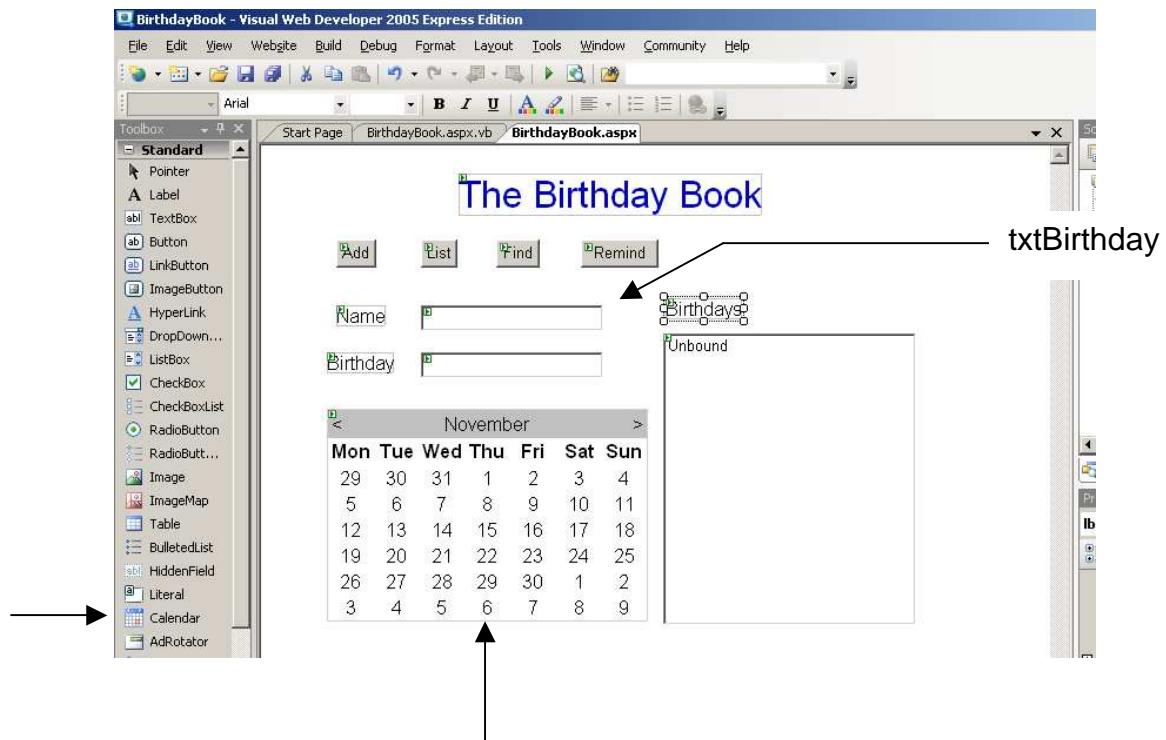
To add a new entry to the birthday book, the user types in a name, selects a date from the calendar and clicks Add.

To show all the entries in the birthday book the user clicks List - the entries are shown in the list box titled Birthdays.

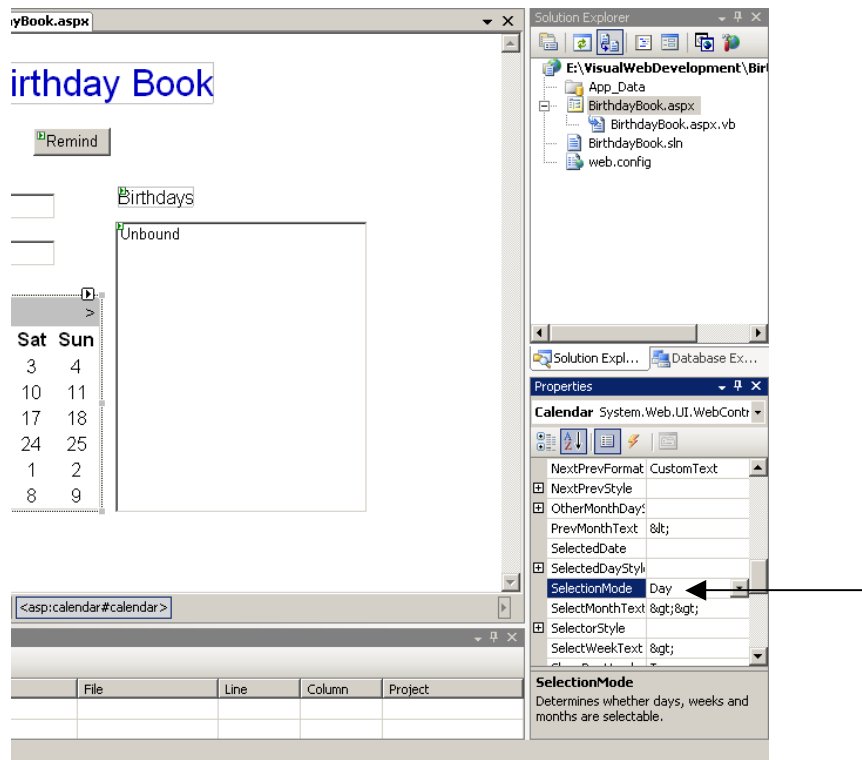
To find a birthday the user enters the person's name and clicks Find.

For a list of birthdays in the coming week the user clicks Remind.

The txtBirthday text box is set to read only.



The Web Server calendar control is dragged from the toolbox.



And its SelectionMode property is set as Day.

18.2 The Birthday Class

An entry in the birthday book is a pair of objects: a name and a date. We use a class to bind the two together.

```
Class Birthday
  Private strName As String
  Private dteBirthday As Date

  Public Sub New(ByVal strName As String, ByVal dteBirthday As Date)
    Me.strName = strName
    Me.dteBirthday = dteBirthday
  End Sub

  Public Function getName() As String
    Return strName
  End Function

  Public Function getBirthday() As Date
    Return dteBirthday
  End Function

  Public Overrides Function ToString() As String
    Return strName + ", " + dteBirthday.ToLongDateString()
  End Function
End Class
```

The class is named Birthday. It has two fields, strName and dteBirthday. Fields are also known as instance variables. The Date class is provided by VB.

The class has one constructor, identified by the New keyword. It is used to create new objects or instances of the class when given a name and a birthday. Me. is used to identify field names. For example, Me.strName = strName copies the parameter strName value into the instance variable Me.strName.

The class has methods to make the field values accessible. getName() returns the value stored in the name field and getBirthday() returns the value stored in the birthday field.

The ToString() function provides a specialised method for returning the combined fields as one string. ToLongDateString() generates a string something like 4 November 2007. ToShortDateString() generates a string something like 4/11/07.

18.3 Birthday Book Code

We declare the state variable, `alBirthdayBook` as an array list, and preserve its values from one page call to the next.

```
Partial Class _Default
    Inherits System.Web.UI.Page

    ' STATE VARIABLE
    Dim alBirthdayBook As ArrayList

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If IsPostBack = True Then
            alBirthdayBook = Session("BirthdayBook")
        Else
            alBirthdayBook = New ArrayList()
            txtName.Focus()
        End If
    End Sub

    Protected Sub Page_Unload(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Unload
        Session("BirthdayBook") = alBirthdayBook
    End Sub
```

We hold the Birthday objects, instances of the Birthday class, in the array list.

On start up we set the focus to be the name text box.

When the user clicks the Add button, we get the input name and the selected date, create a new Birthday object (or instance) and add it to the birthday book. Then we clear the input fields. A problem that needs fixing is that it is possible to add a birthday for someone with no name.

```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
    Dim strName As String = txtName.Text
    Dim dteBirthday As Date = Calendar.SelectedDate
    Dim birthday As Birthday = New Birthday(strName, dteBirthday)
    alBirthdayBook.Add(birthday)
    txtName.Text = ""
    txtBirthday.Text = ""
    txtName.Focus()
End Sub
```

When the user makes a selection from the calendar, we place the date selected by the user into the txtBirthday text box. This procedure is started when you double click on the calendar when in design mode.

```
Protected Sub Calendar_SelectionChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles Calendar.SelectionChanged
    Dim dteBirthday As Date = Calendar.SelectedDate
    txtBirthday.Text = dteBirthday.ToLongDateString()
End Sub
```

All the birthdays stored in the birthday book are displayed in the list box when the user clicks List. We go through the birthday book entry by entry. For each entry we convert it to a string and add it to the list box.

```
Protected Sub btnList_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnList.Click
    lstBirthdays.Items.Clear()
    For Each entry As Birthday In alBirthdayBook
        Dim strBirthday As String = Convert.ToString(entry)
        lstBirthdays.Items.Add(strBirthday)
    Next
End Sub
```

In the Find method we again go through the birthday book entry by entry. For each entry we extract its name and compare it with the name we are looking for. If we find the name we are looking for, we add the entry to the list box for display. If we reach the end of the birthday book and still have not found the name we are looking for, we display "birthday not found".

```
Protected Sub btnFind_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnFind.Click
    lstBirthdays.Items.Clear()
    Dim strNameIn = txtName.Text
    strNameIn = strNameIn.ToLower()
    For Each entry As Birthday In alBirthdayBook
        Dim strName As String = entry.getName()
        strName = strName.ToLower()
        If strName.Equals(strNameIn) Then
            txtBirthday.Text = entry.getBirthday().ToLongDateString()
            Dim strBirthday As String = Convert.ToString(entry)
            lstBirthdays.Items.Add(strBirthday)
            txtName.Text = ""
            txtBirthday.Text = ""
            Exit Sub
        End If
    Next
    lstBirthdays.Items.Add(txtName.Text.ToString() + ", birthday not found")
    txtName.Text = ""
    txtBirthday.Text = ""
End Sub
```

When the user clicks on the Remind button we look for all birthdays that fall within the next seven days of today, whatever that day is. First, we get today's date.

```
Dim dteToday As Date = Date.Now()
```

Then, for each entry in the birthday book we:

- extract its birthday

```
Dim dteBirthday As Date = entry.getBirthday()
```

- find the number of days between today and the birthday

```
Dim difference As Long =  
    DateDiff(DateInterval.Day, dteToday, dteBirthday)
```

DateDiff() returns a Long integer. Since DateInterval is specified to be Day it returns the number of days between the two dates. This number is negative if the first date comes after the second, and positive if the first date comes before the second.

- if the difference is between zero and seven we display the birthday in the list box.

```
If difference >= 0 And difference < 7 Then  
    lstBirthdays.Items.Add(entry.ToString())  
End If
```

- if we reach the end of the birthday book and have not found any birthdays within the next seven days, we display "None found".

```
If lstBirthdays.Items.Count() = 0 Then  
    lstBirthdays.Items.Add("None found")  
End If
```

The entire procedure is shown below.

```

Protected Sub btnRemind_Click(ByVal sender As Object, ByVal e As
    System.EventArgs) Handles btnRemind.Click
    ' Displays all birthdays 7 days from today
    lstBirthdays.Items.Clear()
    Dim dteToday As Date = Date.Now()
    For Each entry As Birthday In alBirthdayBook
        Dim dteBirthday As Date = entry.getBirthday()
        Dim difference As Long =
            DateDiff(DateInterval.Day, dteToday, dteBirthday)
        If difference >= 0 And difference < 7 Then
            lstBirthdays.Items.Add(entry.ToString())
        End If
    Next
    If lstBirthdays.Items.Count() = 0 Then
        lstBirthdays.Items.Add("None found")
    End If
End Sub

```

The code in its entirety is shown below.

```

Partial Class _Default
    Inherits System.Web.UI.Page

    ' STATE VARIABLE
    Dim alBirthdayBook As ArrayList

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Load
        If IsPostBack = True Then
            alBirthdayBook = Session("BirthdayBook")
        Else
            alBirthdayBook = New ArrayList()
            txtName.Focus()
        End If
    End Sub

    Protected Sub Page_Unload(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Me.Unload
        Session("BirthdayBook") = alBirthdayBook
    End Sub

    Class Birthday
        Private strName As String
        Private dteBirthday As Date

        Public Sub New(ByVal strName As String, ByVal dteBirthday As Date)
            Me.strName = strName
            Me.dteBirthday = dteBirthday
        End Sub

        Public Function getName() As String
            Return strName
        End Function

        Public Function getBirthday() As Date
            Return dteBirthday
        End Function
    End Class

```



```

        Public Overrides Function ToString() As String
            Return strName + ", " + dteBirthday.ToLongDateString()
        End Function
    End Class

    Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnAdd.Click
        Dim strName As String = txtName.Text
        Dim dteBirthday As Date = Calendar.SelectedDate
        Dim birthday As Birthday = New Birthday(strName, dteBirthday)
        alBirthdayBook.Add(birthday)
        txtName.Text = ""
        txtBirthday.Text = ""
        txtName.Focus()
    End Sub

    Protected Sub Calendar_SelectionChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Calendar.SelectionChanged
        Dim dteBirthday As Date = Calendar.SelectedDate
        txtBirthday.Text = dteBirthday.ToLongDateString()
    End Sub

    Protected Sub btnList_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnList.Click
        lstBirthdays.Items.Clear()
        For Each entry As Birthday In alBirthdayBook
            Dim strBirthday As String = Convert.ToString(entry)
            lstBirthdays.Items.Add(strBirthday)
        Next
    End Sub

    Protected Sub btnFind_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnFind.Click
        lstBirthdays.Items.Clear()
        Dim strNameIn = txtName.Text
        strNameIn = strNameIn.ToLower()
        For Each entry As Birthday In alBirthdayBook
            Dim strName As String = entry.getName()
            strName = strName.ToLower()
            If strName.Equals(strNameIn) Then
                txtBirthday.Text = entry.getBirthday().ToLongDateString()
                Dim strBirthday As String = Convert.ToString(entry)
                lstBirthdays.Items.Add(strBirthday)
                txtName.Text = ""
                txtBirthday.Text = ""
                Exit Sub
            End If
        Next
        lstBirthdays.Items.Add(txtName.Text.ToString() + ", birthday not
found")
        txtName.Text = ""
        txtBirthday.Text = ""
    End Sub

    Protected Sub btnRemind_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnRemind.Click
        ' Displays all birthdays 7 days from today
        lstBirthdays.Items.Clear()
        Dim dteToday As Date = Date.Now()
        For Each entry As Birthday In alBirthdayBook
            Dim dteBirthday As Date = entry.getBirthday()

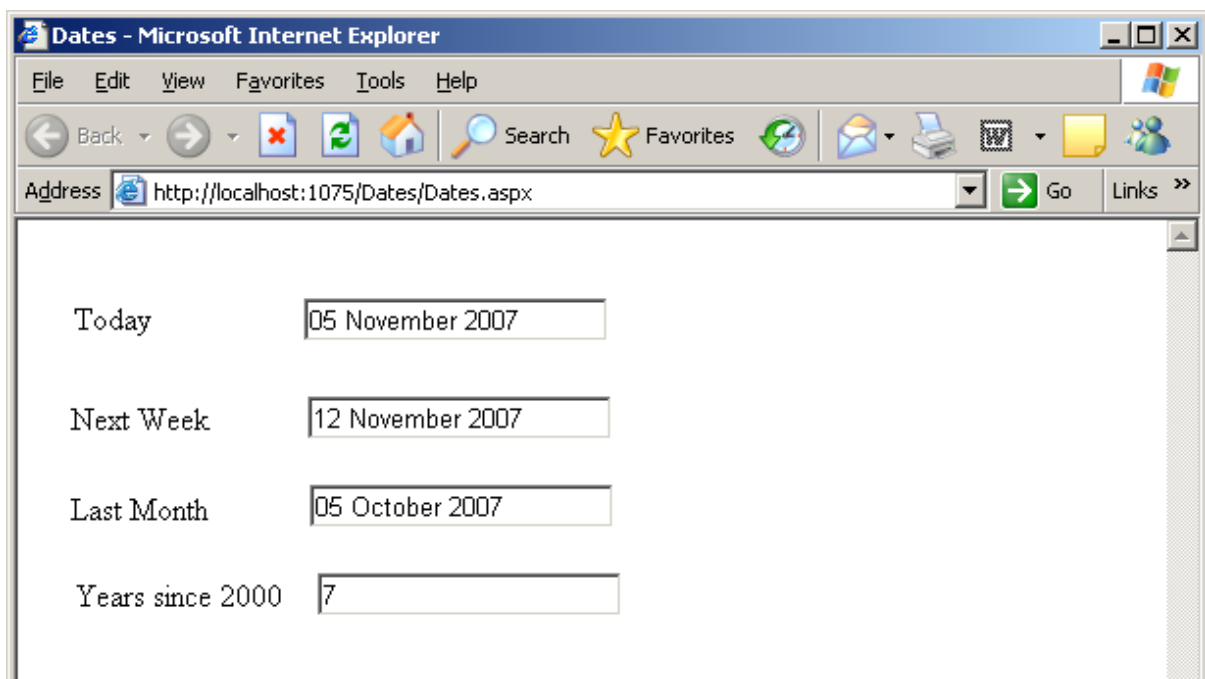
```

```

        Dim difference As Long = DateDiff(DateInterval.Day, dteToday,
dteBirthday)
        If difference >= 0 And difference < 7 Then
            lstBirthdays.Items.Add(entry.ToString())
        End If
    Next
    If lstBirthdays.Items.Count() = 0 Then
        lstBirthdays.Items.Add("None found")
    End If
End Sub
End Class

```

18.4 Some Date Methods



```

Dim today As Date = Now()
txtToday.Text = today.ToLongDateString()

Dim nextWeek As Date = DateAdd(DateInterval.Day, 7, today)
txtNextWeek.Text = nextWeek.ToLongDateString()

Dim lastMonth As Date = DateAdd(DateInterval.Month, -1, today)
txtLastMonth.Text = lastMonth.ToLongDateString()

Dim month As Integer = 1
Dim day As Integer = 1
Dim year2000 As Date = New Date(2000, month, day)
Dim yearsSince2000 As Long = DateDiff(DateInterval.Year, year2000, today)
txtYears.Text = yearsSince2000

```

18.5 Exercises

1. Try out the birthday book program shown above.
2. Try out the date methods shown above. Investigate exceptions that might occur and implement code to trap them.

18.6 Conclusion

We have looked at the Web Server Calendar and some Date methods. Next, we look at text files.