

FORM TEMPLATES IN MICROSOFT ACCESS

Presented by
Mike Wolfe of
nolongerset.com

THURSDAY
JUNE 15



Denver Area Access Users Group

Who am I?



- MS Access / SQL Server developer since 2007 (16 years)
- Owner of Grandjean & Braverman, Inc. since 2016 (7 years)
- International speaker (Vienna, Madrid, UK, USA)
- Write **daily** articles on Access and VBA at **nolongerset.com**
- **Microsoft Access MVP** since November 2021



What to Expect Today

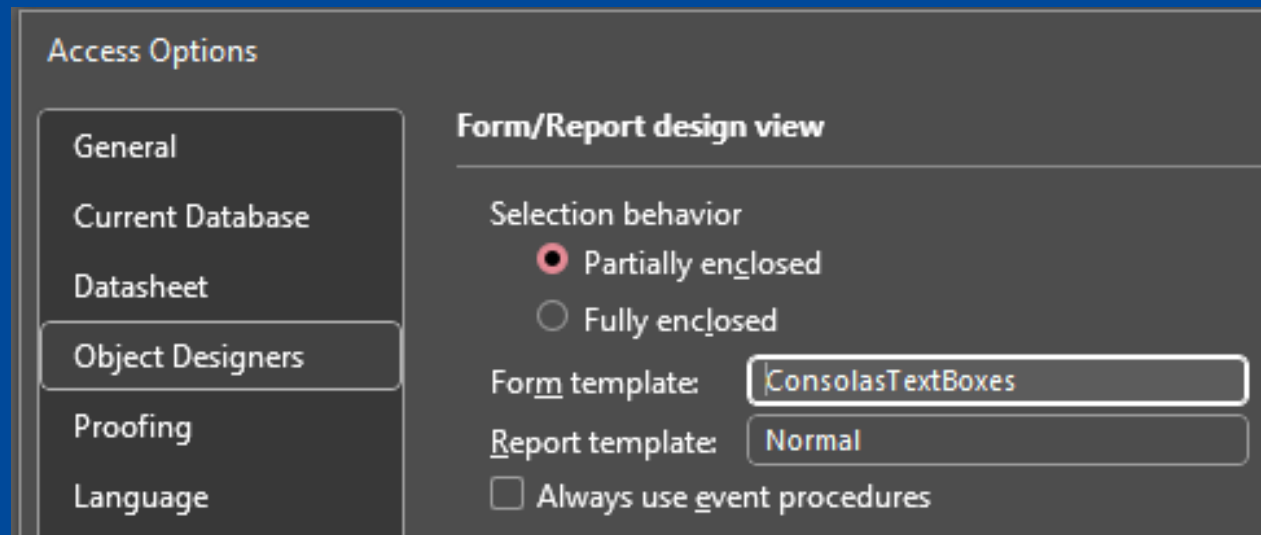
- Introduction: Importance of a consistent UI design
- Approach 1: Built-in "Form template" setting
- Approach 2: Making copies of existing forms
- Approach 3: Creating form "skeletons" via VBA
- Approach 4: Generating entire forms via VBA
- Pros and cons of each approach
- BONUS: Access Themes
- Screenshots and Code Samples

Consistent UI Design

- The Mark of a Professional Application
- A Good UI is Like a Good Naming Convention
- Consistency is More Important than Individual Choices
 - Record Navigation: Built-in or Custom?
 - Application Navigation: Switchboard or Ribbon?
 - Editable Control Style: Sunken or Flat?
 - Font Choices: Sans Serif or Monospace?
- Consistency is Hard to Maintain without a Plan

#1: Built-in Form Template Setting

- By default, Access will look for a form named “Normal”.
- However, you can customize the form template to something else:
 - File > Options > Object Designers > Form template
- This is a **Machine-Wide** option, not just the current database



Built-in Form Template Setting

PRO

- Simple and easy to implement.
- Allows customization of the default form template.
- Consistent font choice can enhance user input readability.

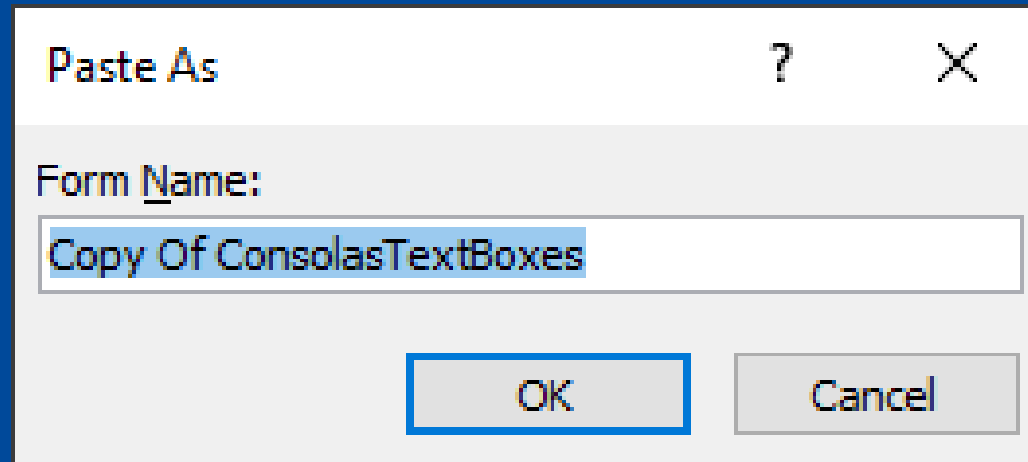
CON

- Limited to a one form template.
- Machine-wide setting, lacking application-specific customization.
- Potential bugs with forms created from custom templates.

Rating: ★★★★★

#2: Copying Existing Forms

- Create one or more forms to use as starting points for other forms.
- To create a new form, select an existing form, [Ctrl] + [C] to copy, [Ctrl] + [V] to paste.
- Give the new form its own name.



Copying Existing Forms

PRO

- Ability to have multiple form templates.
- Customization of templates on an application-specific basis.
- No known bugs or unexpected behavior.

CON

- Difficult to isolate customized settings from defaults.
- Harder to maintain UI consistency within an app.
- Harder to maintain UI consistency across apps.

Rating: ★★★★★

#3: Form Skeletons with VBA

- Create a function to generate a form from scratch.
- Set common settings and customize others based on the form type.
- Includes support for many types of forms:
 - Continuous vs. Single
 - Bound vs. Unbound
 - Report Preview Forms
 - Generic “Get” Forms (think: InputBox replacements)

Form Skeletons with VBA



PRO

- Multiple form templates.
- Easily maintained consistency across templates and applications.
- Clear visibility of explicitly set properties.
- Version control friendly.

CON

- Requires a bigger upfront investment of time to set up the function.
- Not always intuitive to determine the appropriate VBA properties and methods.

Rating: ★★★★★

#4: Generating Forms with VBA

- Create class modules that can build entire forms from scratch.
- Use method chaining to boost signal-to-noise ratio (Fluent API).
- Works well if you need to generate *a lot* of simple forms.
- Controls are placed using relative positioning to simplify layout.

Generating Forms with VBA

PRO

- Ultimate consistency.
- Efficient form creation.
- Easily insert controls into existing form layouts.
- Encapsulated complexity.
- Enhanced functionality.
- Reusability.

CON

- *Significant* upfront investment.
- Risk of front-end corruption.
- Complex form designs are more difficult to create in code than doing it by hand.

Rating: ★★★★★

#5: BONUS: Access Themes

- Themes help maintain consistency in the following areas:
 - Colors
 - Fonts
 - Icons
 - Images
 - Button styles
- Not my specialty, Peter Cole is the expert
- Website: <https://thememydatabase.co.uk>
- Video: [Themes in Access - Access Lunchtime](#)

DesignNewForm() Function Parameters

- **{NONE}**: The default is to create a bound form in Single Form view.
- **IsContinuous**: Determines if the form should be displayed in continuous view.
- **IsUnbound**: Specifies whether the form is unbound (not bound to a data source).
- **IsPreview**: Indicates if the form is a preview form.
- **IsGetForm**: Determines if the form is a "get" form for retrieving information.
- **IsLookup**: Specifies if the form is a lookup form.
- **IsUnboundAddForm**: Indicates if the form is an unbound add form.

A Single View Bound Form

- Header and footer have our standard blue background
- Labels align to the right
- Includes:
 - Record selector
 - Navigation buttons
 - Min/max/close buttons

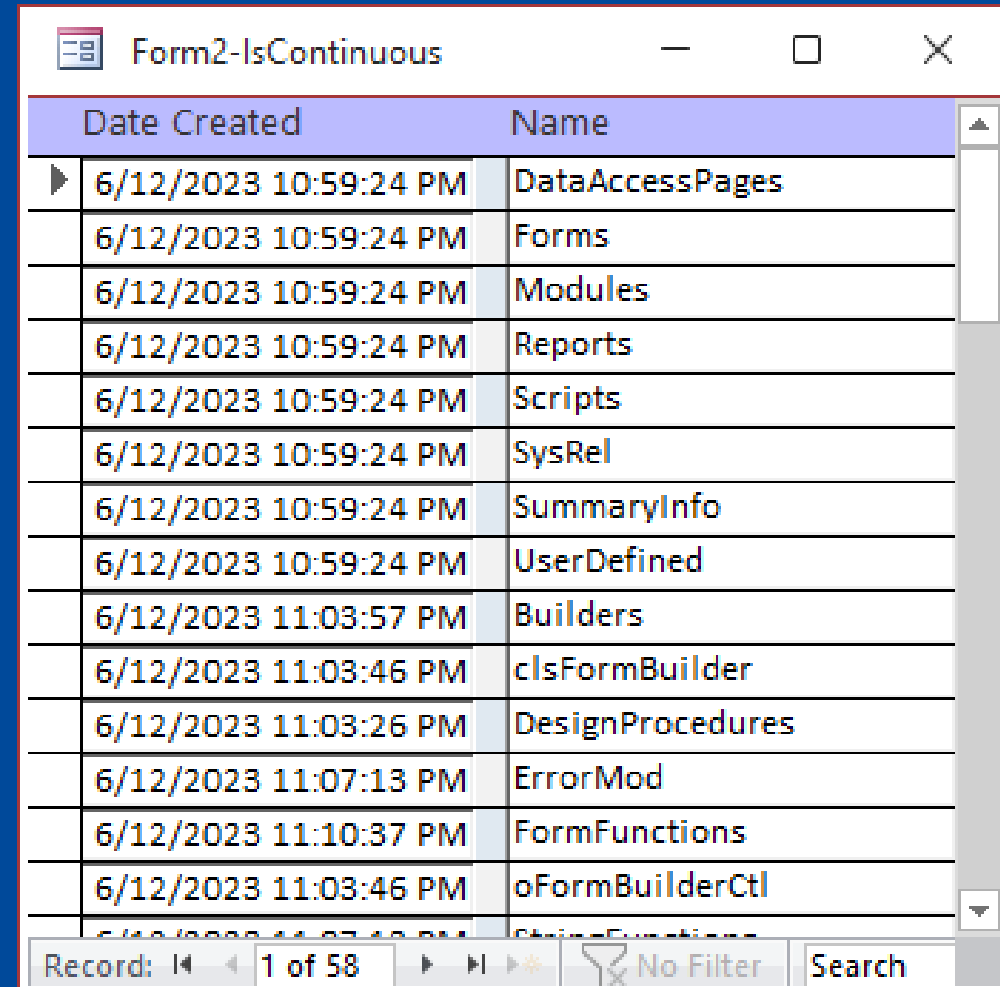
The screenshot shows a Windows application window titled "Form1-Defaults" with a standard Windows title bar (minimize, maximize, close buttons). The form has a purple header bar with the text "Default: A Single View Bound Form". The main area contains a list of fields with labels on the right and values in text boxes on the left. The fields are: DateCreate (6/12/2023 10:59:24 PM), DateUpdate (6/12/2023 10:59:24 PM), Flags (0), ForeignName (empty), Id (-2147483648), Name (DataAccessPages), Owner (empty), ParentId (251658240), RmtInfoShort (empty), and Type (3). At the bottom, there is a footer bar with a record selector showing "Record: 1 of 58", navigation buttons, a filter icon with "No Filter", and a search box.

Field	Value
DateCreate	6/12/2023 10:59:24 PM
DateUpdate	6/12/2023 10:59:24 PM
Flags	0
ForeignName	
Id	-2147483648
Name	DataAccessPages
Owner	
ParentId	251658240
RmtInfoShort	
Type	3

Record: 1 of 58 | No Filter | Search

A Continuous View Bound Form

- Header and footer have our standard blue background
- Labels align to the left
- Includes:
 - Record selector
 - Navigation buttons
 - Min/max/close buttons

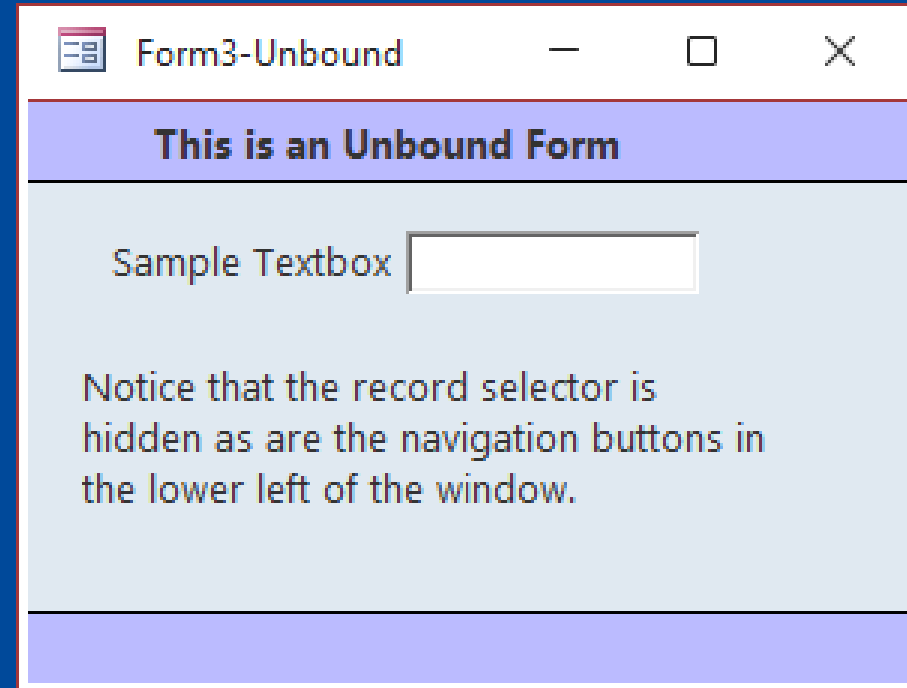


Date Created	Name
6/12/2023 10:59:24 PM	DataAccessPages
6/12/2023 10:59:24 PM	Forms
6/12/2023 10:59:24 PM	Modules
6/12/2023 10:59:24 PM	Reports
6/12/2023 10:59:24 PM	Scripts
6/12/2023 10:59:24 PM	SysRel
6/12/2023 10:59:24 PM	SummaryInfo
6/12/2023 10:59:24 PM	UserDefined
6/12/2023 11:03:57 PM	Builders
6/12/2023 11:03:46 PM	clsFormBuilder
6/12/2023 11:03:26 PM	DesignProcedures
6/12/2023 11:07:13 PM	ErrorMod
6/12/2023 11:10:37 PM	FormFunctions
6/12/2023 11:03:46 PM	oFormBuilderCtl
6/12/2023 11:03:46 PM	oFormBuilderCtl
6/12/2023 11:03:46 PM	oFormBuilderCtl

Record: 1 of 58 | No Filter | Search

An Unbound Form

- Header and footer have our standard blue background
- Includes:
 - Min/max/close buttons
- Excludes:
 - Record selector
 - Navigation buttons



Report Preview Form

- No header and footer
- Standard blue in detail section
- Includes:
 - [Preview Report] button
 - [Export to Excel] button
- Excludes:
 - Record selector
 - Navigation buttons
 - Min/Max/X buttons

Form4-ReportPreview

This is a report preview form

Start Date

End Date

Preview Report Export to Excel

Close

Generic “Get” Form

- No header and footer
- Standard blue in detail section
- Includes:
 - “Message” text box
 - [OK]/[Cancel] buttons
- Excludes:
 - Record selector
 - Navigation buttons
 - Min/Max/X buttons

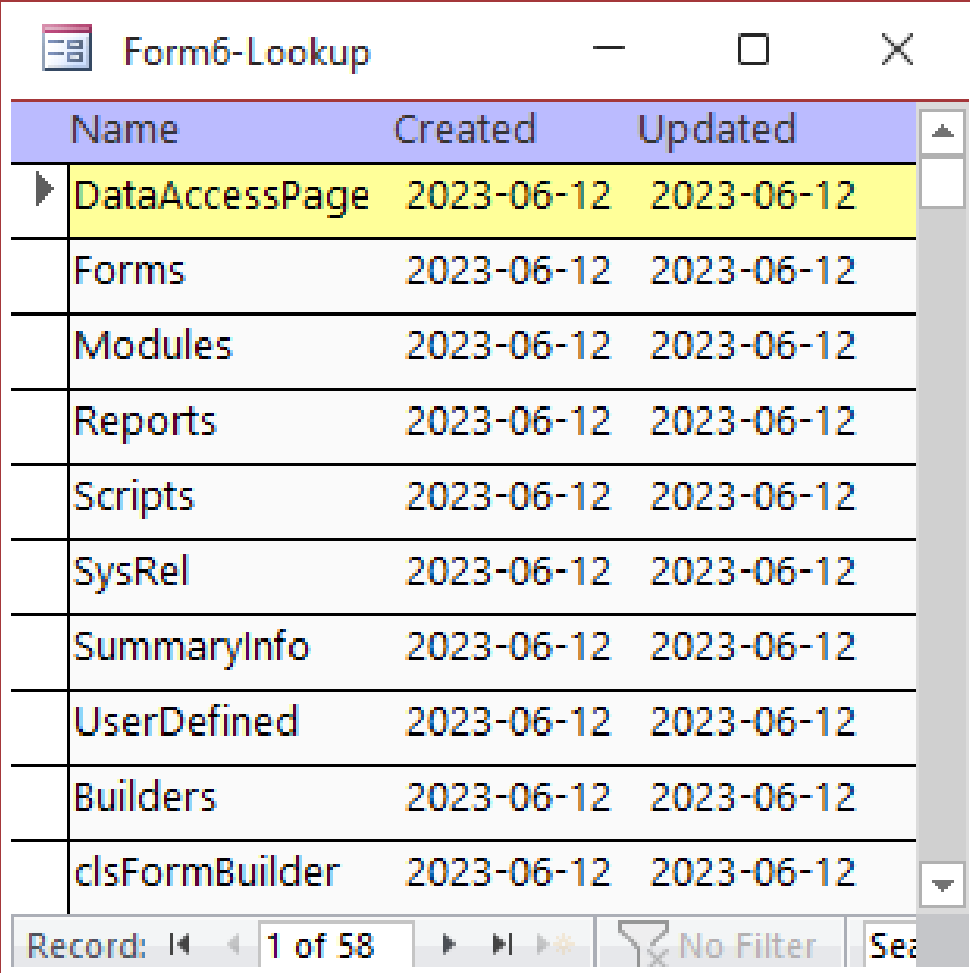
Form5-GetForm

This is a 'Get' form that is used as a generic form to get certain data from a user. Think of it as a step up from an InputBox.

Choose Unit:

Lookup Form

- Standard blue in Header section
- Includes:
 - HighlightRow() text box
 - Transparent command button
 - Record selector
 - Navigation buttons
 - Min/Max/X buttons
- Default text boxes are flat with a transparent background

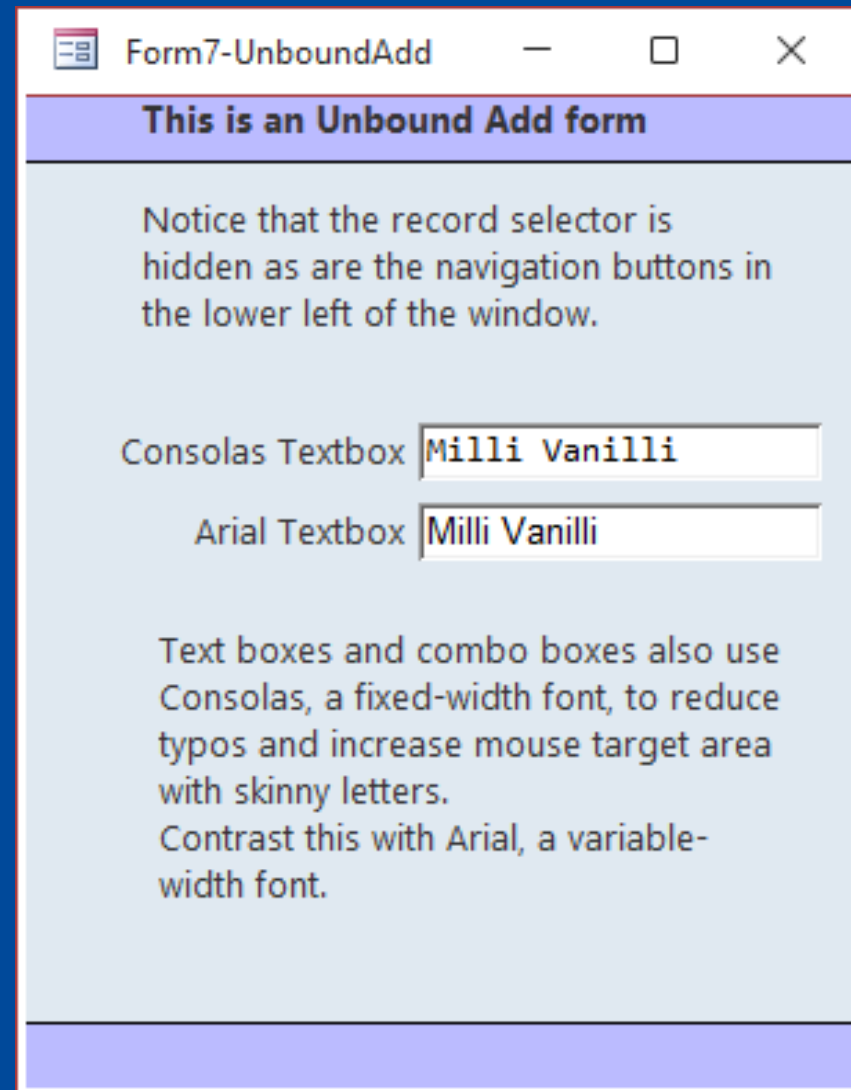


Name	Created	Updated
DataAccessPage	2023-06-12	2023-06-12
Forms	2023-06-12	2023-06-12
Modules	2023-06-12	2023-06-12
Reports	2023-06-12	2023-06-12
Scripts	2023-06-12	2023-06-12
SysRel	2023-06-12	2023-06-12
SummaryInfo	2023-06-12	2023-06-12
UserDefined	2023-06-12	2023-06-12
Builders	2023-06-12	2023-06-12
clsFormBuilder	2023-06-12	2023-06-12

Record: 1 of 58 | No Filter | Search

Unbound Add Form

- Standard blue in Header/Footer
- Excludes:
 - Record selector
 - Navigation buttons
- Text and combo boxes use a fixed width font



What is a Fluent API?

- API: Application Programming Interface
- Fluent API refers to how calling code interacts with our class modules
 - **Method chaining** combines multiple function calls on one line of code
 - IntelliSense increases efficiency when **writing code**
 - Natural language increases efficiency when **reading code**

```
Ctl.Size(14).Font("Arial").Bold.Italic.Disabled
```

How do you build a Fluent API?

1. Requires a class module
2. Use Functions instead of Property Let
3. Return type is the current class
4. Return the current class instance:

```
Set {FunctionName} = Me
```

```
Public Function Bold() As oFormBuilderCtl  
    this.Control.FontBold = True  
    this.Control.SizeToFit  
    Set Bold = Me  
End Function
```

```
Public Function Italic() As oFormBuilderCtl  
    this.Control.FontItalic = True  
    this.Control.SizeToFit  
    Set Italic = Me  
End Function
```

```
Public Function Invisible() As oFormBuilderCtl  
    this.Control.Visible = False  
    Set Invisible = Me  
End Function
```

```
Public Function Disabled() As oFormBuilderCtl  
    this.Control.Enabled = False  
    Set Disabled = Me  
End Function
```

```
Public Function Enabled() As oFormBuilderCtl  
    this.Control.Enabled = True  
    Set Enabled = Me  
End Function
```



Driving Record

Commonwealth vs. John Thomas Doe
328-2017-Criminal, et al.

Has driver's license (even if invalid)? Yes No

Driver's License #: State: Status:

Other Photo ID:

Vehicles [Add Vehicle](#)

Year	Make/Model	Color	Plate #
v 2011	Chrysler Sebring	Silver	PA RQN 1234 Edit Delete
^ 2013	Ford Explorer	Blue	PA HBK 7002 Edit Delete

Driving Violations [Add Violation](#)

Date	Violation	State
3/21/2002	Failed to Yield	NY Edit Delete
4/3/2004	Speeding (67 in a 45)	NJ Edit Delete
5/2/2008	Failed to stop at red light	CO Edit Delete

Additional comments:

Section Status:


```

Public Sub BuildSF_Driving()
    Dim fb As clsFormBuilder
    Set fb = NewFormBuilder("Driving Record")
    fb.Form.RecordSource = "PsiDriving"

    fb.AddLabel "Has driver's license (even if invalid)?"
    fb.AddRadio("HasLicense", "Yes", -1, "No", 0).NewLine

    fb.AddTextBox(1.25, "Driver's License #:", "LicenseNum").BumpRight
    fb.AddState("State:", "LicenseState").BumpRight
    fb.AddCombo(1.25, c.RowSrc.LicenseStatus, "Status:", "LicenseStatus", "0.1").NewLine.Ctl.FillRight

    fb.AddTextBox(2.5, "Other Photo ID:", "OtherPhotoID").NewLine.Ctl.FillRight

    'Add Vehicles subform
    InsertSub fb, 3, "Vehicles", "Vehicle", "AddVehicle Me.PsiDrivingID", _
        "01_Driving_sfVehicles", "PsiDrivingID", 0.5

    'Add Driving Violations subform
    InsertSub fb, 3, "Driving Violations", "Violation", "AddDrivingViolation Me.PsiDrivingID", _
        "01_Driving_sfViolations", "PsiDrivingID", 0.5

    fb.AddTextBox(4, "Additional" & vbNewLine & "comments:", "DrivingComments", 3).NewLine.Ctl.FillRight

    AddSaveCancelButton fb
    SaveAndClose fb, "01_Driving"
End Sub

```

Vehicles

[Add Vehicle](#)

	Year	Make/Model	Color	Plate #	
▶	v 2011	Chrysler Sebring	Silver	PA RQN 1234	Edit Delete
^	2013	Ford Explorer	Blue	PA HBK 7002	Edit Delete

'Build the Vehicle subform for the Driving/Vehicle History form

```
Public Sub BuildSF_VehicleSF()
```

```
    Dim fb As clsFormBuilder
```

```
    Set fb = NewSubformBuilder(, "PsiVehicle", "PsiDrivingID")
```

```
    Dim HighlightStart As Long
```

```
    HighlightStart = fb.GetCurrentLeft
```

```
    fb.AddTextBox(0.4, "Year", "VehYear").NudgeRight.Ctl.Center
```

```
    fb.HGrow(0.6).AddTextBox 1, "Make/Model", "MakeModel"
```

```
    fb.HGrow(0.2).AddTextBox 0.75, "Color", "VehColor"
```

```
    fb.HGrow(0.2).AddTextBox 1, "Plate #", "PlateNum"
```

```
    InsertEditDeleteLinks fb, "Vehicle", HighlightStart
```

```
    SaveAndClose fb, "01_Driving_sfVehicles"
```

```
End Sub
```

The screenshot shows a Windows application window titled "Vehicle". Inside the window, there is a form with the following fields:

- Year:** A text input field.
- Make/Model:** A text input field.
- Color:** A dropdown menu with a green background and a downward arrow.
- Plate #:** A text input field.
- Primary Driver/Comments:** A larger text input field.

'Build the Vehicle edit form

```
Public Sub BuildSF_Vehicle()
    Dim fb As clsFormBuilder
    Set fb = NewEditBuilder("PsiVehicle")
    fb.Form.Caption = "Vehicle"

    fb.AddTextBox(0.5, "Year:", "VehYear").BumpRight
    fb.AddTextBox(1.5, "Make/Model:", "MakeModel").NewLine.Ctl.Require.FillRight

    fb.AddCombo(1, c.RowSrc.VehColor, "Color:", "VehColor", "1.5").BumpRight.Ctl.Require
    fb.AddTextBox(1, "Plate #:", "PlateNum").NewLine.Ctl.Require.FillRight

    fb.AddTextBox(1.75, "Primary Driver/Comments:", "DriverName").NewLine.Ctl.FillRight

    SaveAndClose fb, "01_Vehicle"
End Sub
```

Questions?

Links and Resources:

<https://nolongerset.com/daaug-form-templates/>

