

# EXTENDING A SCRIPTING LANGUAGE FOR VISUAL BASIC FORMS

*Robert M. Siegfried, Denis Diakoniarakis, Kenneth G. Franqueiro and Amol Jain*  
*Department of Mathematics and Computer Science*  
*Adelphi University*  
*South Avenue*  
*Garden City, NY 11530*  
[siegfrir@panther.adelphi.edu](mailto:siegfrir@panther.adelphi.edu)

**Keywords** - Visual Basic, Visual Programming, Graphical User Interfaces, Blind Programmers

## **Abstract**

A prototype scripting language was introduced in 2002 to assist the blind in creating forms for Visual Basic programs in a much simpler fashion; currently, creating these forms are particularly difficult for the blind and visually impaired. The language is expanded to include the Visual Basic objects that were omitted from the original grammar for the scripting language. A second prototype of the compiler for the scripting language is now available online. An example and a formal grammar are included.

## **Introduction**

Computer programming has been a relatively accessible career for the blind, particularly before the proliferation of graphical user interfaces (GUIs). Despite the widespread use of Windows and other GUIs, most blind programmers still work on text-based systems. More than 90% of the blind programmers in the database of the American Federation for the Blind (AFB) work in mainframe environments.<sup>1</sup>

The original goal of the project was to create a platform-independent, programming language that was suitable for use by blind programmers for rapid application development (RAD); no such language currently exists<sup>2</sup>. However, the feedback from members of the Blind Programming list indicated that they did not need their own programming language; what they needed were tools that would help them create forms in Visual Basic. The original grammar for the language was proposed in 2002<sup>3</sup> and subsequently revised as the project continued. The original prototype compiler was completed in August 2003<sup>4</sup>. At that time, a request for comments was posted on the **blindprogramming** mailserv list and made available online, together with a brief manual; generally, comments have been favorable.

It was evident from the beginning of the project that the language was only prototypical; none of the original developers are blind and none could know definitively what works best for blind programmers. Also, only five of the various object classes available in Visual Basic version 6.0 were implemented. The language was then expanded to include all the classes of objects that one can include in Visual Basic forms. The current prototypical compiler, referred to as "version 0.3" was completed in August 2004.

## **Using the Compiler**

The compiler is a console application, run separately from Visual Basic. The form file that it produces is then included in the Visual Basic project. The executable file for the compiler is **molly.exe**. The files containing the form scripts use the extension **.fms** (for **form script**). After creating the form script using any text editor (such as Notepad), the form script can be compiled a command prompt window using the command

```
molly FileName.fms
```

where *FileName.fms* is the name of the form script file. If the file's name is **test.fms** the command would be:

```
molly test.fms
```

This will produce a standard form file **test.frm**, which can then be included in a Visual Basic project.

## **Syntax for the Scripting Language**

The grammar for the scripting language appears in Appendix A. Form scripts can be organized as a set of either rows or columns containing any of the objects that can appear in a Visual Basic form. Most

of the properties are set by default; many of the others are limited to a small set of values from the programmers can choose.

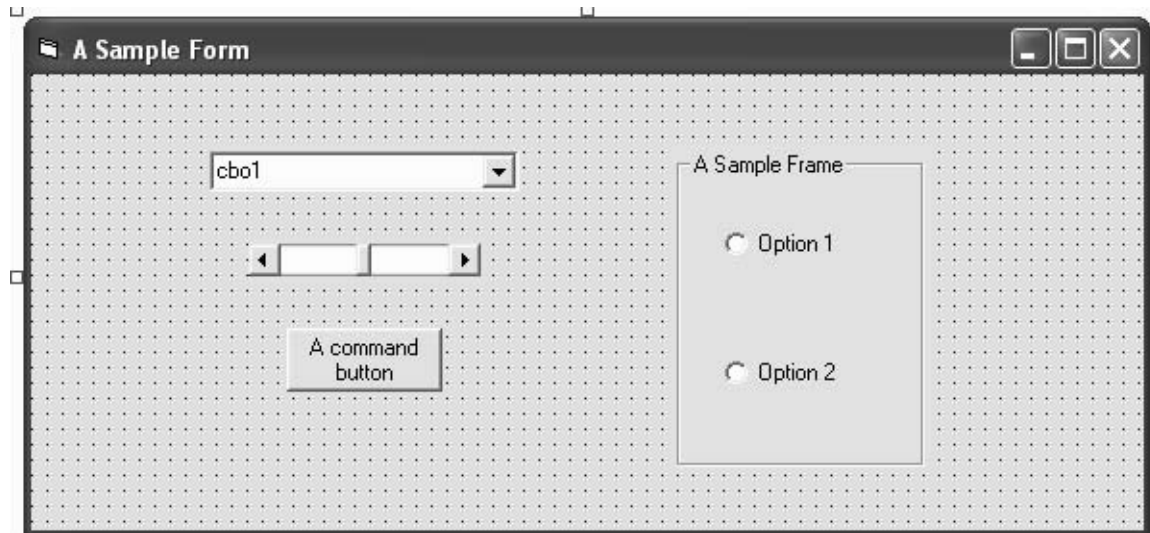


Figure 1: A sample form created by the scripting language compiler

The following script specifies the form appearing in Figure 1:

```

Form MySampleForm      ' The form's header
Location = Top Center ' It is centered in the top
                      ' third of the screen
Caption = "A Sample Form"
Organization = Columns ' The form is organized in columns

Section                ' Begins a section declaration
  ComboBox cbo1        ' A combo box declaration
    Sorted = False
    Style = 0
    Width = Large      ' Widths can be small, medium or large
  End ' combobox

  Scrollbar scrMine    ' A scroll bar declaration
    Orientation = horizontal
    Length = small
    Min = 0
    Max = 100
    Value = 50
    SmallChange = 1
    LargeChange = 5
  End ' Scrollbar

  Commandbutton cmd1   ' A command button declaration
    Caption = "A command button" ' The only option
  End ' Combobox
End ' Section

Section
  Frame fraYours      ' A Frame can only contain option buttons
    Caption = "A Sample Frame"
    Optionbutton Opt1

```

```

    Caption = "Option 1"
    Visible = true ' They both are visible
End ' Opt1

Optionbutton Opt2
    Caption = "Option 2"
    Visible = true
End ' Opt2
End ' Frame
End ' Section
End ' Form

```

### Testing and Review Process

Everyone who has worked on the project is sighted. As sensitive as the development team could be to the needs of the blind and visually impaired, it is impossible to anticipate entirely what the blind would consider more suitable for their needs. Testing has started by using the scripting language and its alternatives (i.e., “pointing and clicking” and creating .frm files for Visual Basic version 6). Testing will continue in two forms: asking the members of the **blindprogramming** mailing list for feedback on the language and compiler and this will be followed by seeking out blind high school and college students and seeking their opinions on the utility of the language and compiler.

Any comments or suggestions on this scripting language are welcome and should be sent to the author at the e-mail address appearing at the top.

### References

- <sup>1</sup>Private communication of U. Obianyo-Agu with Christa Earl, web site developer for the American Federation of the Blind in New York.
- <sup>2</sup>Private communication with Curtis Chong, Director of Technology, National Federation of the Blind.
- <sup>3</sup>Robert M. Siegfried, "A Scripting Language To Help The Blind To Program Visually", ACM SIGPLAN Notices 32(2), 2002, February, p. 53-56.
- <sup>4</sup>Robert M. Siegfried, Denis Diakoniarakis and Uchechukwu Obianyo-Agu, “Teaching the Blind to Program Visually”, Proc. ISECON 2004 v. 21 (Newport), §3265

### Appendix A - A BNF grammar for the scripting language

```

Form ::= Header FormAttributes SectionAttributes SectionDeclarations end
Header ::= form id Returns
FormAttributes ::= LocationAttribute CaptionAttribute
LocationAttribute ::= VerticalAttribute HorizontalAttribute Returns
VerticalAttribute ::= top | middle | bottom
HorizontalAttribute ::= left | center | right
CaptionAttribute ::= caption = String Returns
OrgAttributes ::= organization = SectionOrg
SectionOrg ::= rows | columns
SectionDeclarations ::= SectionDeclarations SectionDeclaration | SectionDeclaration
SectionDeclaration ::= section Returns ObjectDeclarations end Returns
ObjectDeclarations ::= ObjectDeclarations ObjectDeclaration | ObjectDeclaration
ObjectDeclaration ::= CommandButtonDeclaration | TextBoxDeclaration | ComboBoxDeclaration |
    FrameDeclaration | CheckBoxDeclaration | ListBoxDeclaration | TimerDeclaration |
    DriveListBoxDeclaration | FileListBoxDeclaration | DirListBoxDeclaration |
    ScrollBarDeclaration
CommandButtonDeclaration ::= commandbutton id Returns CaptionAttribute end Returns
TextBoxDeclaration ::= textbox id Returns SizeAttributes LabelAttribute end Returns
SizeAttributes ::= HeightAttribute WidthAttribute
HeightAttribute ::= height = Number Returns

```

*WidthAttribute* ::= **width** = *Size* Returns  
*Size* ::= **small** | **medium** | **large**  
*LabelAttribute* ::= **label** = *String* Returns  
*ComboBoxDeclaration* ::= **combobox id** Returns *SortedAttribute* *StyleAttribute* *WidthAttribute* **end**  
Returns  
*ListBoxDeclaration* ::= **listbox id** Returns *SortedAttribute* *StyleAttribute* *SizeAttribute* *ColumnsAttribute*  
**end** Returns  
*SortedAttribute* ::= **sorted** = *Boolean* Returns  
*StyleAttribute* ::= **style** = *Number* Returns  
*FrameDeclaration* ::= **frame id** Returns *CaptionAttributes* *OptionDeclarations* **end** Returns  
*OptionDeclarations* ::= **optionbutton id** Returns *CaptionAttribute* *VisibleAttribute* **end** Returns  
*VisibleAttribute* ::= **visible** = *Boolean* Returns  
*Boolean* ::= **true** | **false**  
*CheckBoxDeclaration* ::= **checkbox id** Returns *CaptionAttribute* *SizeAttributes* **end** Returns  
*TimerDeclaration* ::= **timer id** Returns *IntervalAttribute* **end** Returns  
*IntervalAttribute* ::= **interval** = *Number* Returns  
*FileListBoxDeclaration* ::= **filelistbox id** Returns *SizeAttributes* **end** Returns  
*DirListBoxDeclaration* ::= **dirlistbox id** Returns *SizeAttributes* **end** Returns  
*DriveListBoxDeclaration* ::= **drivelistbox id** Returns **end** Returns  
*ScrollBarDeclaration* ::= **scrollbar id** Returns *OrientationAttribute* *LengthAttribute* *ScrollBarAttributes*  
**end** Returns  
*OrientationAttribute* ::= **orientation** = *OrientType* Returns  
*OrientType* ::= **horizontal** | **vertical**  
*LengthAttribute* ::= **length** = *Size* Returns  
*ScrollBarAttributes* ::= *MinAttribute* *MaxAttribute* *ValueAttribute* *ChangeAttributes*  
*MinAttribute* ::= **min** = *Number* Returns  
*MaxAttribute* ::= **max** = *Number* Returns  
*ValueAttribute* ::= **value** = *Number* Returns  
*ChangeAttributes* ::= *SmallChangeAttribute* *LargeChangeAttribute*  
*SmallChangeAttribute* ::= **smallchange** = *Number* Returns  
*LargeChangeAttribute* ::= **largechange** = *Number* Returns  
*Returns* ::= *Returns* ↵ ↵

String ::= " *AlphaNumeric* \* "  
AlphaNumeric ::= Letter | Digit  
Number ::= Digit Digit \*  
Letter ::= A | B | ... | Y | Z | a | b | ... | y | z  
Digit ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0

↵ indicates the newline character