

# Visual Programming and the Blind: The Challenge and the Opportunity

Robert M. Siegfried  
Department of Mathematics and Computer Science  
Adelphi University  
Garden City, NY 11530 USA  
(516)877-4482  
siegfrir@panther.adelphi.edu

## ABSTRACT

The proliferation of graphical user interfaces has had a dramatic impact on the ability to work as a programmer. It is particularly difficult for the blind to create forms for visual programming applications, such as Visual Basic. A scripting language is introduced that enables the blind to create Visual Basic forms without needing to specify a great deal of detail and without needing the “point and click” approach that they cannot use. Related issues of accommodating the blind in a computer science course are also discussed.

## Categories and Subject Descriptors

K.3.2 [Computers and Education]: Computer and Information Science Education - *Computer science education*.

K.4.2 [Computers and Society]: Social Issues - *Assistive technologies for persons with disabilities*.

## General Terms

Human Factors, Languages

## Keywords

Visual Basic, Visual Programming, Graphical User Interfaces, Blind Programmers.

## 1. INTRODUCTION

With the development of text-based, interactive computing in the 1960s, the command-line interface became the standard development environment that most computer professionals used. This began to change in the 1980s with the development of the graphical user interface at Xerox PARC [2], and led to X-Windows (and other GUIs for UNIX systems), the Macintosh operating system and Microsoft Windows [5]. These innovations allowed Bill Gates to meet the career goal of making the computer user-friendly enough for his mother to use [9]. Paradoxically, while the GUI made it easy for the typical person to use and to program computers, it made it much more

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGCSE'06, March 3-5, 2006, Houston, Texas, USA.  
Copyright 2006 ACM 1-59593-259-3/06/0003...\$5.00.

difficult for the blind to work as computer professionals.

Computer programming has long been considered a relatively accessible career path for the blind. Starting in the mid-1960s, the University of Manitoba ran a program for many years in which they trained blind students as programmers. The course had a near 100% retention rate with most of the students successfully finding employment [3]. The American Federation for the Blind currently has 130 blind programmers in their database [1].

### 1.1 The Blind and Computer Programming

Blind programmers were able to compete with their sighted counterparts when mainframes were the most commonly used computers. Visually impaired programmers had been trained and conditioned to use specialized tools to enable them to write programs. There were special screen readers that converted text into speech or into Braille. Thus, the blind could read directions and instructions that were necessary for software development. With these aids, blind programmers were able to work as effectively as their sighted counterparts. However, the computer industry's transition to graphical interfaces has had a deleterious effect on the blind programmers. Many have been falling behind. Screen readers have not always been upgraded as quickly as GUIs have. This has made it more difficult for the blind to work on newer platforms [1].

Many people have preconceived ideas about what blind programmers can and cannot do. The author had a conversation about this work with a personal friend, a programmer who could not understand how a blind person could work as a programmer because of the need to write down so much information used in developing software. It was only after further thought that this man was able to accept the idea that the blind can work successfully in the field. These attitudes are quite common and frequently make it difficult for the blind that get hired as programmers.

The blind are not without resources, however. The National Federation for the Blind has been an advocate for the blind, playing a major role in the passing of the 1997 amendments to the Individuals with Disabilities Education Act [7]. This law ensures that the blind, like others with disabilities, receive an appropriate education and have access to appropriate adaptive technology. The NFB's lobbying ensured that the blind have access to Braille instruction except where it is not appropriate.

The Internet has made the world a smaller place for the blind as it has for most of us. The American Federation for the Blind (<http://www.afb.org>), the National Federation for the Blind

(<http://www.nfb.org>) and Lighthouse International (<http://www.lighthouse.org>) have web sites with information available to the blind and sighted public. There is also the Blind Programming web site (<http://blindprogramming.com>) that provides the blind with access to various software tools and documentation and other information that make it easier for the blind to program and use computers productively. The mailing list associated with the site now has discussion threads archived on their site.

Adaptive technologies have also helped the blind work with computers. These include Braille displays and screen readers such as JAWS. Also, the legally blind, whose corrected vision measures no better than 20/200, can use screen enlargers that show a portion of the normal display at many times the normal size. For many people, this can be enough to allow them to make productive use of a computer.

For the blind, the basic techniques of writing and testing a program have not changed significantly. Most use standard text-based editors and then the command-line prompt to compile their programs. Many programmers will even work in Visual Basic as long as there is a text file with which to work. Tools that require pointing and clicking are unsuitable for use by the blind [1].

## 1.2 Working in Visual Basic

Visual Basic was introduced by Microsoft in 1991 [6] and has become a popular application because it allows users to create forms by “pointing and clicking.” This popularity has been enhanced by a large number of third party software components available for it. The current version is part of Microsoft’s Visual Studio .NET and it shares a common runtime library with the other components, including Visual C++. Earlier versions of Visual Basic used a text file to store information about the forms that an application uses, allowing programmers to make changes in the form without the point and click approach. Forms are stored in text format with the form’s properties and its member objects’ properties listed together with their values. While this allowed one to change these values fairly easily, it is difficult to design a form by creating such a text file. For example, the size of a form and its position on the screen were set by specifying the form’s height and width in twips (twentieths of a point). In the current version, position and size of the form and its constituent objects is specified in pixels, which standardizes the size of a form on a screen, regardless how the monitor’s resolution is set.

While it is possible for the blind to create forms by specifying the necessary details about the form, in practice, it is difficult to do. There is a great deal of detail that must be considered to avoid having one object overlap another. This is a difficult task for a sighted person, even with the use of trial and error to see how the resulting form is displayed. It was originally proposed that a platform-independent, RAD programming language be developed that was suitable for use by blind programmers; the members of the Blind Programming mailing list insisted that a scripting language for Visual Basic forms was more important.

## 2. THE SCRIPTING LANGUAGE AND COMPILER

The grammar for such a scripting language was originally proposed in 2001 [8]. A compiler for the original prototype of

the language was developed in 2003, with a compiler for the full language in 2004. These compilers produced form files in the earlier format that was used by Visual Basic version 6. A compiler that produced form files in the .NET format was developed during the summer of 2005.

### 2.1 The Compiler

The compiler is a console application, run separately from Visual Basic and then included in the Visual Basic project. The executable file for the compiler is `molly.exe` (the Molly compiler was named for Molly Siegfried Lind, the author’s aunt who lost her sight to glaucoma). 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 in 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.

### 2.2 The Scripting Language

The basic layout of a form script is given in figure 1. The screen is divided into three rows and three columns: the three rows are top, middle and bottom and the three columns are left, center and right. This allows the user to place the form in different areas of the screen without having to measure or to use trial and error.

```
Form FormName ζ
Location = VerticalAttrib HorizontalAttrib ζ
Caption = "Caption on Title Bar of Form" ζ
Organization = { Rows or Columns } ζ
SECTION
  SectionAttributes
END ' Comments appear after an
... ' apostrophe until the end of the line
END
```

Where ζ indicates a carriage return.

**Figure 1. The basic layout of a form script.**

Most forms are organized in rows or columns, but usually not both within the same form. For this reason, the user specifies whether the form’s organization is in rows or columns. After the organization is specified, each section (either a row or column, depending on which one the programmer has chosen), is declared with one or more object in the section, which are automatically laid out sequentially within the section. Their exact placement depends on the particular object and its own space requirements. The programmer can specify as few or as many objects as desired in any given section, as long as they all fit within a window. Similarly, the only limit on the number of sections is their ability to fit within the window.

Although the initial prototype allowed the programmer to use any combination of only five object types, the language now includes virtually all standard object types that appear in Visual Basic.

### 2.3 Specifying Objects

Each of the eleven objects that can be specified has its own syntax because the key properties differ from one object to another. The general syntax for an object is:

```

Object Type ObjectName
Properties
End
    
```

where *ObjectType* is CommandButton, TextBox, ComboBox, Frame, CheckBox, ListBox, Timer, DriveListBox, FileListBox, DirListBox, ScrollBar. *ObjectName* is the name that the object will have within the Visual Basic form. It must be unique within the form and should follow the standard name conventions for Visual Basic objects.

The syntax for two representative object declarations appears in Table 1, along with an example. Some of the declarations are very simple with only one or two properties specified. Command buttons are examples of this; other than name, the property that is specified is the caption. Other objects have a larger number of properties specified. Scroll bars, for example, require that the programmer specify whether the scroll bar is horizontal or vertical, its length (as small, medium or large), as well as its minimum, maximum, small change and large change values. It is noteworthy that in all instances, a programmer will specify fewer properties than he or she would specify if he or she were creating a Visual Basic form file directly.

**Table 1. The objects that can be appear in a form script**

Object Type	Syntax	Example
CommandButton	<pre> CommandButton cmdButtonName ζ Caption = "CommandButtonCaption" ζ End ζ                     </pre>	<pre> CommandButton cmdConvert Caption = "Convert to Metric" End                     </pre>
Scrollbar	<pre> ScrollBar scrScrollBarName ζ Orientation = / Horizontal or Vertical ζ Length = { Small or Medium or Large ζ Min = ConstantValue ζ Max = ConstantValue ζ Smallchange = ConstantValue ζ Largechange = ConstantValue ζ End ζ                     </pre>	<pre> ScrollBar hsbMyScrollBarName Orientation = Horizontal Length = Large Min = 0 Max = 60 Smallchange = 1 Largechange = 5 End                     </pre>

Several object types, including textboxes, comboboxes, checkboxes, and listboxes require that the width be specified. In all these cases, it is sufficient to specify them as small, medium or large. The height, when required, is specified by the number of lines; it is limited to 5 lines or fewer. The style is specified by the number used by Visual Basic to indicate one of several different styles; these codes are standard and do not require a programmer to use trial and error to determine the appearance of the object.

Although frames in Visual Basic forms can contain objects other than option buttons, the scripting language limits frames to option buttons only. This is the most common usage of frames; this was a design decision made to simplify the scripting language. Similarly, the scripting language treats horizontal and vertical scroll bars as different instances of the same object class differing only by the orientation property. This differs from the approach taken in Visual Basic where they are treated as different object classes. A sample script, the form specified by

the script and the file created by the compiler are shown in Figure 2, 3, and 4 respectively.

```

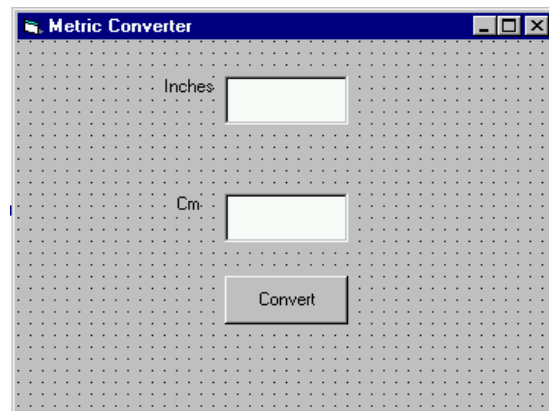
Form InToCm
Location = Top Left
Caption = "Metric Converter"
Sections = Columns
Section

TextBox txtInches
Height = 2 ' expressed in lines
Width = Medium
Label = "Inches"
END

TextBox txtCm
Height = 2
Width = Medium
Label = "Cm"
END

CommandButton cmdConvert
Caption = "Convert"
END
END ' Section
END ' Form
    
```

**Figure 2. A sample form script.**



**Figure 3. The form specified by the script**

### 3. REACTION TO THE PROJECT

It was the original goal of the project to create a platform-independent RAD programming language that was suitable for use by blind programmers since no such language existed [8]. After posting a request on the blindprogramming mailserv for feedback, the author received a dozen emails with the same essential comment, that what they really wanted was a scripting language for the creation of Visual Basic forms. Consequently, the nature of the project was changed as a direct result of the blind programming community's initial reaction to the project. The prototype compiler and its manual was posted on Adelphi University's web site (<http://www.adelphi.edu/~siegfir/molly>) and the Blind Programming mail list was notified of its availability. The comments were received indicate that the blind programmers who have read the specifications and the sample scripts believe that it has the potential to help them create Visual

Basic forms without the aid of a sighted person. However, there were questions about future versions. Since this was only a prototype, there was never any doubt that additional features would be added to the language. Since that time, the remaining Visual Basic objects were added to the language and the compiler was revised twice: once to include these additional objects and once to create forms in the format used by Visual Studio .NET.

```

Begin VB.Form frmInToCm
  Caption           = "Metric Converter"
  ClientHeight     = 3750
  ClientLeft       = 60
  clients          = 345
  ClientWidth      = 5265
  LinkTopic        = "Form1"
  ScaleHeight      = 3750
  ScaleWidth       = 5265
  StartUpPosition = 3 'Windows Default
Begin VB.CommandButton cmdConvert
  Caption           = "Convert"
  Height            = 495
  Left              = 2040
  TabIndex         = 4
  Top               = 2400
  Width             = 1215
End
Begin VB.TextBox txtCm
  Height           = 495
  Left             = 2040
  TabIndex        = 2
  Top              = 1560
  Width           = 1215
End
Begin VB.TextBox txtInches
  Height           = 495
  Left             = 2040
  TabIndex        = 0
  Top              = 360
  Width           = 1215
End
Begin VB.Label lblCm
  AutoSize         = -1 'True
  Caption          = "Cm"
  Height           = 195
  Left             = 1560
  TabIndex        = 3
  Top              = 1560
  Width           = 225
End
Begin VB.Label lblInches
  AutoSize         = -1 'True
  Caption          = "Inches"
  Height           = 195
  Left             = 1440
  TabIndex        = 1
  Top              = 360
  Width           = 480
End
End

```

Figure 4. The .frm file created by the compiler

## 4. DISCUSSION

A cursory examination of the blindprogramming.com web site removes any doubt that the blind and visually impaired can program or that they are not interested in programming in modern platforms that include GUIs. The most difficult part of

GUI programming for the blind remains the actual design of the interface. A glance at the discussion threads on programming includes several on Java, different versions of Visual Basic as well as HTML. It is important to remember that the scripting language project was in direct response to what the blind themselves said that they needed.

Most people tend to think of the blind and visually impaired as synonymous; that is not necessarily the case. Legal blindness was defined above; there are several definitions of visual impairment, all of which defined specific limitations in vision [4]. While some people with visual impairment may be able to read a computer screen even with enlargement due to the nature of their visual impairment, other can do so and do not need screen readers or Braille displays. Generally, the blind will need these assistive technologies.

Blind students in college classroom are frequently scattered among the general population. There may be only a few at a college or there may be a larger number with only a few majoring in a given department. Sometimes this complicates the delivery of necessary services; sometimes it makes it difficult for faculty to be aware of where accommodating resources on a campus may be located. One student said that "they believe that if there are elevators and wheelchair ramps, they've done their job [Anonymous, personal communication]."

The Individuals with Disabilities Education Act requires that accommodations be made for people with disabilities. We are fortunate to live in an era where we can provide them with much of the technology to live independent, productive lives. This should include being able to work in computer science, despite the changes that the profession has seen in the past twenty years.

## 5. REFERENCES

- [1] Alexander, Steve, "Blind Programmers Face An Uncertain Future", *ComputerWorld* (November 6, 1998), p. 1.
- [2] Ceruzzi, Paul E., *A History of Modern Computing*, The MIT Press, Cambridge, MA, 2000.
- [3] Hodson, Bernard, "Sixties Ushers in Program To Train Blind Programmers", *Computer World Canada*, 11 June 2004.
- [4] Leonard, Robin, *Statistics on Vision Impairment: A Resource Manual*, 5<sup>th</sup> edition, May 2002, Lighthouse International. Retrieved September 8 from <http://www.lighthouse.org/downloads/researchstats.pdf>.
- [5] Palfreman, Jon and Doron Swade, *The Dream Machine: Exploring the Computer Age*, BBC Books, London, 1991.
- [6] *PC Magazine*, 2004. Retrieved June 17 from <http://www.pcmag.com/article2/0,1759,15012,00.asp>
- [7] Riccobono, Mark A., "A Brighter Future for Blind Children", *Braille Monitor*, February 2004.
- [8] Siegfried, Robert M., "A Scripting Language To Help The Blind To Program Visually", *ACM SIGPLAN Notices* 37, 2, February 2002, 53-56.
- [9] Wallace, James and Jim Erickson, *Hard Drive: Bill Gates and the Making of the Microsoft Empire*, HarperBusiness, New York, 1993.