

# complete idiot guide to visual basic 6

Published: September 27, 2026 | Index ID: #-

---

## Introduction

---

When you think of modern programming, languages like Python, JavaScript, and C# often come to mind. Yet, there's a hidden gem that still powers legacy systems, automates office tasks, and serves as a stepping stone for many developers: Visual Basic 6 (VB6). Whether you're a beginner looking to dip your toes into GUI programming or a seasoned coder wanting to maintain a classic application, this **complete idiot guide to Visual Basic 6** will walk you through everything from installation to deployment.

## What is Visual Basic 6?

---

Visual Basic 6 is a 32 bit event driven programming environment released by Microsoft in 1998. It's a descendant of the original BASIC language, but it offers a powerful Integrated Development Environment (IDE) that lets developers build Windows applications quickly with drag and drop controls, pre built dialogs, and a rich set of libraries. VB6's hallmark is its simplicity: you write a few lines of code, drop a button onto a form, and the IDE takes care of the rest.

## Why Learn Visual Basic 6 Today?

---

Although newer languages have taken the spotlight, VB6 remains relevant for several reasons:

- Legacy Systems – Many enterprises still run VB6 applications that handle critical business processes.
- Rapid Prototyping – With its visual design tools, you can create functional prototypes in hours.
- Learning Curve – VB6's straightforward syntax makes it ideal for beginners transitioning to more complex languages.
- Community Support – A wealth of tutorials, forums, and code samples is available online.

## Getting Started with VB6

---

### Installing Visual Basic 6

To begin, you'll need a copy of the VB6 installer. It's no longer sold directly by Microsoft, but you can find it through trusted distributors or as part of legacy software bundles. Once you have the installer, follow these steps:

1. Run setup.exe and follow the on screen prompts.
2. Choose a destination folder (e.g., C:\VB6).
3. When prompted, install the VB6 IDE and the Microsoft Visual Basic 6.0 Runtime.
4. After installation, launch the IDE and verify that a new project can be created.

### Understanding the IDE

The VB6 IDE is divided into several panes:

- Project Explorer – Shows all components of your project.
- Properties Window – Displays properties of the selected control.
- Code Window – Where you write event handlers and procedures.
- Form Designer – Drag and drop controls onto your form.

Take a few minutes to explore these panes; familiarity with the IDE will speed up your development process.

## Basic Concepts and Terminology

---

### Projects, Forms, and Controls

A **project** is the container that holds all files and resources. Inside a project, you typically create one or more **forms**—the windows that users interact with. Forms contain **controls** such as buttons, text boxes, labels, and list boxes. Each control can trigger events (e.g., Click, Change) that you handle in the code window.

### Variables, Data Types, and Operators

VB6 uses a set of data types like Integer, Long, String, and Boolean. Variables must be declared before use:

```
Dim count As Integer
```

```
Dim name As String
```

```
Dim isActive As Boolean
```

Operators such as +, -, \*, and / perform arithmetic, while & concatenates strings.

### Procedures and Functions

Procedures are blocks of code that perform actions. A **Sub** routine does not return a value, whereas a **Function** returns a result:

```
Sub ShowMessage()  
    MsgBox "Hello, VB6!"  
End Sub
```

```
Function Add(a As Integer, b As Integer) As Integer  
    Add = a + b  
End Function
```

### Writing Your First Program

#### Creating a Simple Calculator

Let's build a quick calculator that adds two numbers. Follow these steps:

1. Create a new Standard EXE project.
2. From the Toolbox, drag two TextBox controls, one CommandButton, and one Label onto the form.
3. Set the Name property of the text boxes to txtNum1 and txtNum2, the button to btnAdd, and the label to lblResult.
4. Double click the button to open the code window and add the following:

```
Private Sub btnAdd_Click()  
    Dim num1 As Double  
    Dim num2 As Double  
    Dim sum As Double  
  
    num1 = CDbl(txtNum1.Text)  
    num2 = CDbl(txtNum2.Text)  
    sum = num1 + num2
```

```
lblResult.Caption = "Result: " & sum
End Sub
```

Run the program, enter two numbers, and click “Add.” The result appears in the label.

## Working with Forms and Controls

---

### Adding Buttons, Textboxes, and Labels

Controls are the building blocks of user interfaces. To customize them:

- Use the Properties Window to change appearance (e.g., BackColor, Font).
- Set the TabIndex to control tab order.
- Use Tag to store custom data.

### Event Handling

Every control can fire events. The most common event is Click for buttons. To handle events, double click the control or select the event from the code window's drop down menus. For example, to respond when a text box loses focus, use the Exit event:

```
Private Sub txtNum1_Exit(ByVal Cancel As MSForms.ReturnBoolean)
    If IsNumeric(txtNum1.Text) Then
        ' Valid number
    Else
        MsgBox "Please enter a numeric value."
        txtNum1.SetFocus
    End If
End Sub
```

Common Data Structures in VB6

### Arrays, Collections, and Dictionaries

VB6 supports static arrays, dynamic arrays, and the **Collection** object for storing heterogeneous data:

```
Dim names As Collection
Set names = New Collection
names.Add "Alice"
names.Add "Bob"
```

For key value pairs, use the **Scripting.Dictionary** (requires reference to *Microsoft Scripting Runtime*):

```
Dim dict As Scripting.Dictionary
Set dict = New Scripting.Dictionary
dict.Add "Key1", "Value1"
```

File Handling and Persistence

### Reading and Writing Text Files

VB6 provides simple file I/O functions. To write to a file:

```
Open "C:\temp\output.txt" For Output As #1
```

Print #1, "Hello, VB6!"

Close #1

To read:

Open "C:\temp\output.txt" For Input As #1

Dim line As String

Do While Not EOF(1)

Line Input #1, line

Debug.Print line

## Baca Juga (Artikel Terkait)

---

- [comparison yamaha fz25 vs tvs apache rtr 200 4v](http://basicgravity.com/comparison-yamaha-fz25-vs-tvs-apache-rtr-200-4v.pdf)

URL: <http://basicgravity.com/comparison-yamaha-fz25-vs-tvs-apache-rtr-200-4v.pdf>

- [comparison of static and rotary ups apc](http://basicgravity.com/comparison-of-static-and-rotary-ups-apc.pdf)

URL: <http://basicgravity.com/comparison-of-static-and-rotary-ups-apc.pdf>

- [comparison of radio direction finding technologies](http://basicgravity.com/comparison-of-radio-direction-finding-technologies.pdf)

URL: <http://basicgravity.com/comparison-of-radio-direction-finding-technologies.pdf>

- [comparison of pressure vessel codes coade](http://basicgravity.com/comparison-of-pressure-vessel-codes-coade.pdf)

URL: <http://basicgravity.com/comparison-of-pressure-vessel-codes-coade.pdf>

Tags: #complete #idiot #guide #visual #basic







