Purpose

You can use Amos 4.0 as a component in the computer programs you write. It is a relatively easy process because Amos lets you work in a general-purpose programming language instead of having to learn some special language. To use Amos 4.0 with your program, you need to program in a language or environment that can control automation servers, e.g., Microsoft Visual Basic, Microsoft C++, the SPSS scripting facility, SAS, DEC Visual FORTRAN or Borland Delphi.

The program that you write can carry out a single Amos analysis, or multiple Amos analyses. It can also include statements that:

- specify a model
- give the location of data files
- select options, such as: which discrepancy function to use, whether to obtain bootstrap confidence intervals, and so on.

This section demonstrates in detail how you can call Amos 4.0 from the Microsoft Visual Basic Professional Edition, Version 5.0. It assumes that Microsoft Visual

---

1 At the time of this writing, there have not been any attempts to use Amos 4.0 with SAS, Visual Fortran, or Borland Delphi.
Basic Professional Edition is installed on your system. The Visual Basic files for this tutorial are StartVB.vbp (VB project) and StartVB.bas (VB code). You will find both located in the Tutorial subdirectory underneath the Amos 4.0 program directory.

1. Create a new Visual Basic project

First, launch Visual Basic. In the New Project window, click on the New tab and double-click on Standard EXE.

2 Microsoft offers several other Visual Basic packages with varying capabilities. Most of these can be used with Amos 4.0, but different steps may be required to reference the Amos Engine.
2. Remove Form1

If this is the first time you have used Visual Basic since its installation, the Visual Basic window will probably look something like this:

Many of the elements in the Visual Basic window are useful only when you are writing a program that has a graphical interface. These elements are not needed for this tutorial, so we will get rid of them.
In the **Project1** window, use the *right* mouse button to click on **Form1** (Form1) as indicated by the mouse pointer in the preceding figure. When the following pop-up menu appears, select **Remove Form1**:
3. Hide unnecessary windows

Hide the VB toolbar by clicking the X button indicated by the mouse pointer in the following figure:

Also, hide the three sub-windows titled Project - Project1, Properties - Project1, and Form Layout - Project1. (Do not hide the sub-window titled Immediate.)
The resulting **Visual Basic** window should look like this:
4. Create a code window for your program

On the menu, select **Project** and then **Add Module**:

![Diagram showing the menu selection process for adding a module in Microsoft Visual Basic](image)
In the **Add Module** window, select the **New** tab and double-click on **Module**:
The **Visual Basic** window will then look something like this:
5. Tell Visual Basic how to find the Amos Engine

In order to let Visual Basic know that you want to use the Amos Engine, select **Project** from the main menu, and then **References**:
When the References - Project1 dialog appears, make sure that the Amos Engine box is has a check mark in it. If necessary, click on the Amos Engine box to add a check mark. Then press the OK button to close the dialog:

![References - Project1 dialog]

6. Enter your program

Type the following program in the large panel in the Visual Basic window.

```vbnet
Option Explicit

Sub main()
    Dim Sem As New AmosEngine
    Sem.TextOutput
    Sem.BeginGroup Sem.Dir & "Tutorial\Hamilton.dbf"
    Sem.Structure "SAT = (w1) Education + (w2) Income + (1) Other"
    Sem.FitModel
    Debug.Print "Regression weight 1 is "; Sem.ParameterValue("w1")
    Debug.Print "Regression weight 2 is "; Sem.ParameterValue("w2")
    Debug.Print "Chi square = "; Sem.Cmin
    Debug.Print "Degrees of freedom = "; Sem.df
End Sub
```
Instead of typing these commands, you can also insert the `StartVB.bas` file from Amos 4.0’s `Tutorial` subdirectory. With the code window active, run **Edit → Insert File...** and select `StartVB.bas` located by default in the `C:\Program Files\Amos 4\Tutorial` directory. The `Sem.Dir` is an Amos Engine property that returns the path to where the Amos program is found. The VB string expression:

```
Sem.Dir & "Tutorial\Hamilton.dbf"
```

pinpoints exactly where the data file resides.

The **Visual Basic** window should look like this:
7. Run the program

To run the program, select **Run** from the main menu, and then **Start with Full Compile**:

![Image of Visual Basic interface showing Start With Full Compile option]

When the program has finished running, the output from the `Debug.Print` statements appears in the **Immediate** panel:

![Image of Immediate window showing output]

8. Try out the help system

Before closing the **Visual Basic** window, try the help system. Click somewhere within one of the words that describes an Amos method in your program:

- TextOutput
- BeginGroup
- Structure
- FitModel
- ParameterValue
- Cmin
- Df

Then press the **F1** key. A dialog box will appear giving documentation for that method. To get a list of all available methods, press the **F2** key to display **Visual Basic's Object Browser**. In the **Classes** listbox, click on **AmosEngine**.
All the available methods will appear in the listbox labeled **Members of ‘AmosEngine’**:

Click once on a method to see a short summary of its function in the bottom panel. Press **F1** to see the full documentation of the method.

---

**Beyond the basics**

Still seeking more information about how to use **Amos 4.0** as a component in the computer programs you write? Extensive, advanced reference material for the Amos API is also included in the **Amos 4.0 Programming Reference Guide**. This guide is available as a downloadable Portable Document Format (PDF) file from the Amos 4.0 CD-ROM or as a Help file from either the **Amos Graphics** or **Amos Basic** Help menu.