VB.NET
Threading

This document explains how to create a thread procedure, runn the procedure in a thread and the stop the thread using Visual Basic.

The first thing you must do is to include threading in your application. This is simply done by at the first of the projects code insert the lines

```
Imports System
Imports System.Threading
```

These lines include the threading functionality to your application.

The next step is to define a thread in the global namespace. This accomplished by expanding the code where it says

```+Windows Form Designer Generated Code```

(click on the +). In this region you will see definitions for all of the visual components you drawn on your form as well as other definitions. Anywhere in this area you would insert the code

```
Dim testthread as Thread
```

This defines a variable testthread as type Thread which will be the name of the thread that you will create when you wish to run the thread.

Now, you will need to write a thread procedure. This is the code you wish to run within the thread. I will show you my little thread procedure

```
Public Sub threadproc()
    Dim i as Integer
    threadbox.Text = "Running"
    i = 1
    Do
        Loop While (i = 1)
End Sub
```
This is defined as threadproc. It has an integer variable i defined. The first statement places the text Running in a textbox that was drawn in the main form and renamed threadbox. It then goes into a continuous loop to keep the thread busy to demonstrate that even though the thread is busy you can still interact with the main form without encountering the redraw problems that would normally occur if this code was not run in a thread.

Next, we need to create the thread and run the thread in the background. This is accomplished by running the following:

```csharp
    testthread = New Thread(New ThreadStart(AddressOf Me.threadproc))
    testthread.IsBackground = True
    testthread.Start()
```

This code creates a thread called testthread and assigns the starting address of threadproc as the entry point of the thread, it says to run it in the background and the start the execution of the thread.

The last step is to write a procedure to stop the thread. In my sample, I run the code to start the thread within an if statement which doesn't allow the rest of the program know that the thread will be created and started. If this is your case you will have to write the stop procedure.

```csharp
    Private Sub StopThread()
        testthread.Abort()
        threadbox.Text = "Stopped"
    End Sub
```

The abort thread kills the thread process, so it can be started again at a later time.