

# VB.NET: Send A Letter With SendLetter()

## Instructions

1. Create a new Console Application project and add a web reference
2. Copy the entire code below, delete anything written in your application and paste the code there
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyDescription* with your own text to help you identify this letter in your activity log (optional).
  - o Replace *MyFileExtension* with the extension of the document you are using for this test.
  - o Replace *MyPathToFile* with a path to your test PDF file. Remember, for method *SendLetter* you must use a document where the recipient's address already appears so it will show through the envelope window. You can use one of our Sample Letter Documents.
  - o *MyWorkMode* determines if your letter will be sent as Production, Development or will use your user's default Work Mode setting. Read more about Production and Development Work Modes.
4. Start your application. It will start, encode your document as Base64 and send a letter to the PostalMethods *SendLetter* Web Service method. If the response is a positive number, you will be able to see your letter in the PostalMethods Control Panel.

If the response is negative number, check the Web Service Status Codes section.

That's it - you have successfully sent a letter to PostalMethods. Congratulations.

```
Imports System.IO

Module Module1

    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const MyDescription As String = "Sending a letter using VB.NET"
        Const MyFileExtension As String = ".pdf"
        Const MyPathToFile As String = "C:\MyFile.pdf"
        Const MyWorkMode As com.postalmethods.api.WorkMode =
com.postalmethods.api.WorkMode.Default

        Dim objPM As New com.postalmethods.api.PostalWS()
        Dim lngResult As Long
        '
        ' Read your binary document. In this example: C:\MyFile.pdf
        '
        Dim objFR As FileStream
        objFR = New FileStream(MyPathToFile, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite)
        Dim B(objFR.Length - 1) As Byte
        objFR.Read(B, 0, B.Length)
        objFR.Close()
        objFR = Nothing
        '
        ' Invoke the SendLetter method
        '
        lngResult = objPM.SendLetter( _
                                MyUsername, _
                                MyPassword, _
                                MyDescription, _
```

```

        MyFileExtension, _
        B, _
        MyWorkMode)

    ,
    ' Check response status
    ,
    If lngResult > 0 Then
        ,
        ' A positive value means the message was successfully queued for
processing.
        ' The PostalMethods Letter ID is returned.
        ,
        Console.WriteLine("LetterID is: " & lngResult)
    Else
        ,
        ' A negative value means an error occurred.
        ' See the PostalMethods Status Codes:
http://www.postalmethods.com/statuscodes#webservice.
        ,
        Console.WriteLine("Error is:" & lngResult)
    End If

    ' In debug mode, the following lines prevent your console application from
closing automatically upon termination
    Console.WriteLine("Hit ENTER to terminate application")
    Console.ReadLine()

End Sub

End Module

```

## VB.NET: Send A Letter with SendLetterAndAddress()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyDescription* with your own text to help you identify this letter in your activity log (optional).
  - o Replace *MyFileExtension* with the extension of the document you are using for this test.
  - o Replace *MyPathToFile* with a path to your test PDF file. Remember, method *SendLetterAndAdress* will add the recipient's address so it would show through the envelope window. You can use one of our Sample Letter Documents.
  - o *MyWorkMode* determines if your letter will be sent as Production, Development or will use your user's default Work Mode setting. Read more about Production and Development Work Modes.
  - o Use the existing postal recipient or replace with another valid postal address. Leave an empty string ("" ) where no value is needed.
4. Start your application. It will start, encode your document as Base64 and send a letter to the PostalMethods *SendLetterAndAddress* Web Service method. If the response is a positive number, you will be able to see your letter in the PostalMethods Control Panel. If the response is negative number, check the Web Service Status Codes section.

That's it - you have successfully sent a letter to PostalMethods. Congratulations.

```

Imports System.IO

Module module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const MyDescription As String = "Sending a letter using VB.NET"
        Const MyFileExtension As String = "pdf"
        Const MyPathToFile As String = "C:\MyFile.pdf"
        Const MyWorkMode As com.postalmethods.api.WorkMode =
com.postalmethods.api.WorkMode.Default
        Const MyAttention1 As String = "George Washington"
        Const MyAttention2 As String = ""
        Const MyCompanyName As String = "The White House"
        Const MyAddress1 As String = "1600 Pennsylvania Ave NW"
        Const MyAddress2 As String = ""
        Const MyCity As String = "Washington"
        Const MyState As String = "DC"
        Const MyPostalCode As String = "20500"
        Const MyCountry As String = "United States of America"

        Dim objPM As New com.postalmethods.api.PostalWS()
        Dim lngResult As Long
        '
        ' Read your binary document. In this example: C:\MyFile.pdf
        '
        Dim objFR As FileStream
        objFR = New FileStream(MyPathToFile, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite)
        Dim B(objFR.Length - 1) As Byte
        objFR.Read(B, 0, B.Length)
        objFR.Close()
        objFR = Nothing
        '
        ' Invoke the SendLetterAndAddress method
        '
        lngResult = objPM.SendLetterAndAddress( _
            MyUsername, _
            MyPassword, _
            MyDescription, _
            MyFileExtension, _
            B, _
            MyWorkMode, _
            MyAttention1, _
            MyAttention2, _
            MyCompanyName, _
            MyAddress1, _
            MyAddress2, _
            MyCity, _
            MyState, _
            MyPostalCode, _
            MyCountry)

        '
        ' Check response status
        '
        If lngResult > 0 Then
            ' A positive value means the message was successfully queued for
processing.
            ' The PostalMethods Letter ID is returned.
            Console.WriteLine("LetterID is: " & lngResult)
        End If
    End Sub
End Module

```

```

Else
    '
    ' A negative value means an error occurred.
    ' See the PostalMethods Status Codes:
    http://www.postalmethods.com/statuscodes#webservice.
    '
    Console.WriteLine("Error is:" & lngResult)
End If

' In debug mode, the following lines prevent your console application from
closing automatically upon termination
Console.WriteLine("Hit ENTER to terminate application")
Console.ReadLine()

End Sub

End Module

```

## VB.NET: Send A Letter Using A Template

### Instructions

1. First, you need to upload a letter template to be used with merge data provided in your Web Service request. Download the invoice template sample (.MHT) and upload it to your account using the [UploadFile](#) method (see [code sample](#)).
2. Create a new Console Application project and add a web reference.
3. Copy the entire code below, delete anything written in your application and paste the code there.
4. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyDescription* with your own text to help you identify this letter in your activity log (optional).
  - o Replace *MyFileExtension* with a reference to the template you uploaded in step #1.
  - o Do not change *MyXMLData*.
  - o *MyWorkMode* determines if your letter will be sent as Production, Development or will use your user's default Work Mode setting. Read more about Production and Development Work Modes.
  - o Use the existing postal recipient or replace with another valid postal address. Leave an empty string ("") where no value is needed.
5. Start your application. It will start, encode your document as Base64 and send a letter to the PostalMethods `SendLetterAndAddress` Web Service method. If the response is a positive number, you will be able to see your letter in the PostalMethods Control Panel. If the response is negative number, check the Web Service Status Codes section.

That's it - you have successfully sent a letter to PostalMethods using a pre-uploaded template and by providing your merge data. Congratulations.

```

Imports System.IO

Module module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const MyDescription As String = "Sending an invoice template using VB.NET"
        Const MyFileExtension As String = "MyTemplate:PurchaseInvoice.mht"
    End Sub
End Module

```

```

Const MyXMLData As String = "<MergeData><ReturnAddress>ABC Hardware Inc.<br
/>93S. Jackson Street<br />Seattle WA 98104-2818<br />United
States</ReturnAddress><Date>July 09, 2009</Date><InvoiceID>8830023-RRX-
3200985</InvoiceID><Payment>Net 30</Payment><MakeFor>ABC
Hardware</MakeFor><ProductID1>ED-2343</ProductID1><Description1>Paslode Cordless
Framing Nailer, 30 Degree Paper
Collated</Description1><Quantity1>2</Quantity1><Price1>379.21</Price1><Amount1>758.42<
/Amount1><ProductID2>FR-9485</ProductID2><Description2>Porta-Nails 2 In. Flooring
Nails Master Carton (1000
items)</Description2><Quantity2>10</Quantity2><Price2>11.20</Price2><Amount2>112.00</A
mount2></MergeData>"

Const MyWorkMode As com.postalmethods.api.WorkMode =
com.postalmethods.api.WorkMode.Default
Const MyAttention1 As String = "George Washington"
Const MyAttention2 As String = ""
Const MyCompanyName As String = "The White House"
Const MyAddress1 As String = "1600 Pennsylvania Ave NW"
Const MyAddress2 As String = ""
Const MyCity As String = "Washington"
Const MyState As String = "DC"
Const MyPostalCode As String = "20500"
Const MyCountry As String = "United States of America"

Dim objPM As New com.postalmethods.api.PostalWS()
Dim lngResult As Long
'
' Encode your XML data as binary
'
Dim B As Byte() = System.Text.Encoding.UTF8.GetBytes(MyXMLData)
'
' Invoke the SendLetterAndAddress method
'
lngResult = objPM.SendLetterAndAddress( _
    MyUsername, _
    MyPassword, _
    MyDescription, _
    MyFileExtension, _
    B, _
    MyWorkMode, _
    MyAttention1, _
    MyAttention2, _
    MyCompanyName, _
    MyAddress1, _
    MyAddress2, _
    MyCity, _
    MyState, _
    MyPostalCode, _
    MyCountry)

'
' Check response status
'
If lngResult > 0 Then
' A positive value means the message was successfully queued for
processing.
' The PostalMethods Letter ID is returned.
Console.WriteLine("LetterID is: " & lngResult)
Else
' A negative value means an error occurred.

```

```

        ' See the PostalMethods Status Codes:
        http://www.postalmethods.com/statuscodes#webservice.
    '
    Console.WriteLine("Error is:" & lngResult)
End If

    ' In debug mode, the following lines prevent your console application from
    closing automatically upon termination
    Console.WriteLine("Hit ENTER to terminate application")
    Console.ReadLine()

End Sub

End Module

```

## VB.NET: Get The Generated Letter with GetPDF()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace "ID" with the LetterID for which you would like to get the PDF.
4. Start your application. If the response is -3000, you will see the status report. If the response is any other number, check the Web Service Status Codes section.

```

Imports System.IO

Module Module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const ID As Integer = 0

        Dim objPM As New com.postalmethods.api.PostalWS()
        Dim objResult As com.postalmethods.api.GetPDFResult

        '
        ' Invoke the GetPDF method
        '
        objResult = objPM.GetPDF(MyUsername, MyPassword, ID)

        '
        ' Check response status
        '
        If objResult.ResultCode = -3000 Then
            ' -3000 means that the PDF was successfully retrieved. Enter the full path
            to save it in e.g. "c:\mypdf.pdf".
            Console.WriteLine("Successfully retrieved PDF. Please enter a path to save
it in.")
            Dim strPath As String = Console.ReadLine()

```

```

    '
    'Save the file
    '
    Dim objFS As New FileStream(strPath, FileMode.Create, FileAccess.Write)
    objFS.Write(objResult.FileData, 0, objResult.FileData.Length)
    objFS.Close()
    objFS = Nothing
Else
    '
    ' A value less than -3000 means an error occurred.
    ' See the PostalMethods Status Codes:
    http://www.postalmethods.com/statuscodes.
    '
    Console.WriteLine("Error is:" & objResult.ResultCode)
End If

' In debug mode, the following lines prevent your console application from
closing automatically upon termination
Console.WriteLine("Hit ENTER to terminate application")
Console.ReadLine()

End Sub
End Module

```

## VB.NET: Get Letter And Postcard Metadata with GetDetails()

### Instructions

1. Create a new Console Application project and add a web reference
2. Copy the entire code below, delete anything written in your application and paste the code\_ there
3. Replace "MyUsername" and "MyPassword" with your PostalMethods user details
4. Replace "IDs" with a single ID of one of your sent mail items, multiple IDs (ID1,ID2,ID3) or a range of IDs (ID1-ID2)
5. Start your application. If the response is -3000, you will see the details report. If the response is any other number, check [the Web](#) Service Status Codes section.

```

Imports System.IO

Module Module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const IDs As String = "0" 'To query a single mail item
        'Const IDs As String = "0,0,0" 'To query multiple mail items
        'Const IDs As String = "0-0" 'To query a range of mail items

        Dim objPM As New com.postalmethods.api.PostalWS()
        Dim WSResult As com.postalmethods.api.GetDetailsResult

        '
        ' Invoke the GetLetterDetails method
        '
        WSResult = objPM.GetDetails( _
            MyUsername, _
            MyPassword, _

```

```

        IDs)
    ,
    ' Check response status
    ,
    If WSResult.ResultCode = -3000 Then
        ,
        ' -3000 means that the data was successfully retrieved.
        ,
        For i As Integer = 0 To WSResult.Details.Length - 1

            With WSResult.Details(i) Console.WriteLine("ID: {0}", .ID)
                Console.WriteLine("Price: {0}", .Price)
                Console.WriteLine("Number of sheets: {0}", .NumOfSheets)
                Console.WriteLine("Submit Time: {0}", .SubmitTime)
                Console.WriteLine("Completion Time: {0}", .CompletionTime)
                Console.WriteLine("Status: {0}", .Status)
                Console.WriteLine("Status Description: {0}", .StatusDescription)
                Console.WriteLine("Is this item a part of a batch?: {0}",
                    .IsBatch)
                Console.WriteLine("My Description: {0}", .MyDescription)
                Console.WriteLine("Letter or Postcard?: {0}",
                    .MailerType.ToString)
                Console.WriteLine("Production or Development Work Mode: {0}",
                    .WorkMode.ToString)
            End With
        Next
    Else
        ,
        ' A value less than -3000 means an error occurred.
        ' See the PostalMethods Status Codes:
        http://www.postalmethods.com/statuscodes.
        ,
        Console.WriteLine("Error is: " & WSResult.ResultCode)
    End If

    ' In debug mode, the following lines prevent your console application from
    closing automatically upon termination
    Console.WriteLine("Hit ENTER to terminate application")
    Console.ReadLine()

End Sub
End Module

```

## VB.NET: Get Letter And Postcard Delivery Details with GetDetailsExtended()

### Instructions

1. Create a new Console Application project and add a web reference
2. Copy the entire code below, delete anything written in your application and paste the code there
3. Replace "MyUsername" and "MyPassword" with your PostalMethods user details

4. Replace "IDs" with a single ID of one of your sent mail items, multiple IDs (ID1,ID2,ID3) or a range of IDs (ID1-ID2)
5. Start your application. If the response is -3000, you will see the details report. If the response is any other number, check the [Web Service Status Codes](#) section.

```
Imports System.IO

Module Module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const IDs As String = "0"           'To query a single mail item
        'Const IDs As String = "0,0,0"     'To query multiple mail items
        'Const IDs As String = "0-0"       'To query a range of mail items

        Dim objPM As New com.postalmethods.api.PostalWS()
        Dim WSResult As com.postalmethods.api.GetDetailsExtendedResult

        '
        ' Invoke the GetLetterDetails method
        '
        WSResult = objPM.GetDetailsExtended( _
            MyUsername, _
            MyPassword, _
            IDs)

        '
        ' Check response status
        '
        If WSResult.ResultCode = -3000 Then
            '
            ' -3000 means that the data was successfully retrieved.
            '
            For i As Integer = 0 To WSResult.Details.Length - 1
                With WSResult.Details(i)
                    Console.WriteLine("ID: {0}", .ID)
                    Console.WriteLine("Status: {0}", .Status)
                    Console.WriteLine("Status Description: {0}", .StatusDescription)
                    Console.WriteLine("Completion Time: {0}", .CompletionTime)
                    Console.WriteLine("Is Batch: {0}", .IsBatch)
                    Console.WriteLine("Mailer Type: {0}", .MailerType)
                    Console.WriteLine("MyDescription: {0}", .MyDescription)
                    Console.WriteLine("Num Of Sheets: {0}", .NumOfSheets)
                    Console.WriteLine("Price: {0}", .Price)
                    Console.WriteLine("SubmitTime: {0}", .SubmitTime)
                    Console.WriteLine("Work Mode: {0}", .WorkMode)
                    Console.WriteLine("Print Color: {0}", .PrintColor)
                    Console.WriteLine("Print Sides: {0}", .PrintSides)
                    Console.WriteLine("Postcard: {0}", .Postcard)
                    Console.WriteLine("Envelope: {0}", .Envelope)
                    Console.WriteLine("Orientation: {0}", .Orientation)
                    Console.WriteLine("Paper: {0}", .Paper)
                    Console.WriteLine("International Mailing: {0}",
                        .InternationalMailing)
                    Console.WriteLine("National Mailing: {0}", .NationalMailing)
                End With
            Next
        Else
            '
        End Sub
    End Module

```

```

        ' A value less than -3000 means an error occurred.
        ' See the PostalMethods Status Codes:
        http://www.postalmethods.com/statuscodes.
    '
    Console.WriteLine("Error is: " & WSResult.ResultCode)
End If

    ' In debug mode, the following lines prevent your console application from
    closing automatically upon termination
    Console.WriteLine("Hit ENTER to terminate application")
    Console.ReadLine()

End Sub
End Module

```

## VB.NET: Cancel Delivery Of A Letter with CancelDelivery()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace "ID" with the LetterID for which you would like to cancel.
4. Start your application. If the response is -3000, you will see the status report. If the response is any other number, check the Web Service Status Codes section.

```

Imports System.IO

Module Module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const ID As Integer = 0

        Dim objPM As New com.postalmethods.api.PostalWSSimple()
        Dim lngResult As Long

        '
        ' Invoke the CancelDelivery method
        '
        lngResult = objPM.CancelDelivery(MyUsername, MyPassword, ID)

        '
        ' Check response status
        '
        If lngResult = -3000 Then
            '
            ' -3000 means the request was successfully marked for cancellation.
            '
            Console.WriteLine("Request has been cancelled")
        Else
            '
            ' A value less than -3000 means an error occurred.

```

```

        ' See the PostalMethods Status Codes:
        http://www.postalmethods.com/statuscodes.
    '
    Console.WriteLine("Error is:" & lngResult)
End If

    ' In debug mode, the following lines prevent your console application from
    closing automatically upon termination
    Console.WriteLine("Hit ENTER to terminate application")
    Console.ReadLine()

End Sub
End Module

```

## VB.NET: Upload A File For Future Use Using UploadFile()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyFileName* with the name you would like to give the file. This name is used for future references to this file. Format: <file name>.<file extension>.
  - o Replace *MyPathToFile* with a full path to your file.
4. Start your application. If the response is -3000, you will see the status report. If the response is any other number, check the Web Service Status Codes section.

```

Imports System.IO

Module module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"
        Const MyFileName As String = "PurchaseInvoice.mht"
        Const MyPathToFile As String = "C:\PurchaseInvoice.mht"
        ' You may download this file from
        http://www.postalmethods.com/files/templates/PurchaseInvoice.mht

        ' Set account permissions for this file:
        Const MyPermissions As com.postalmethods.api.Permissions =
com.postalmethods.api.Permissions.Account
        ' Set Overwrite mode:
        Const FileUploadOverwrite As Boolean = False

        Dim objPM As New com.postalmethods.api.PostalWS()
        Dim lngResult As Long

        ' Read file to upload and convert to binary. In this case, the file is
        C:\PurchaseInvoice.mht
        ' You may download this file from
        http://www.postalmethods.com/files/templates/PurchaseInvoice.mht
        Dim MyFile As Byte() = IO.File.ReadAllBytes(MyPathToFile)
    End Sub
End Module

```

```

' Invoke the UploadFile method
lngResult = objPM.UploadFile( _
    MyUsername, _
    MyPassword, _
    MyFileName, _
    MyFile, _
    MyPermissions, _
    FileUploadOverwrite)

'
' Check response status
'
If lngResult = -3000 Then
    ' If the result code = -3000, the file was uploaded successfully and it is
    ready for use.
    Console.WriteLine("File uploaded successfully!")
Else
    ' A negative value means an error occurred.
    ' See the PostalMethods Status Codes:
    http://www.postalmethods.com/statuscodes#webservice.
    Console.WriteLine("An error occurred:" & lngResult)
End If

' In debug mode, the following lines prevent your console application from
closing automatically upon termination
Console.WriteLine("Hit ENTER to terminate application")
Console.ReadLine()

End Sub

End Module

```

## VB.NET: Delete an uploaded file using DeleteUploadedFile()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyFileName* with the name of the file, including its extension, as it was given when uploaded.
4. Start your application. If the response is -3000, the file was deleted. If the response is any other number, check the Web Service Status Codes section.

```

Imports System.IO

Module module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"

```

```

Const MyFileName As String = "PurchaseInvoice.mht"

Dim objPM As New com.postalmethods.api.PostalWS()
Dim lngResult As Long

' Invoke the DeleteUploadedFile method
lngResult = objPM.DeleteUploadedFile( _
    MyUsername, _
    MyPassword, _
    MyFileName)

'
' Check response status
'
If lngResult = -3000 Then
    ' If the result code = -3000, the file was deleted.
    Console.WriteLine("File deleted!")
Else
    ' A negative value means an error occurred.
    ' See the PostalMethods Status Codes:
    http://www.postalmethods.com/statuscodes#webservice.
    Console.WriteLine("An error occurred:" & lngResult)
End If

' In debug mode, the following lines prevent your console application from
closing automatically upon termination
Console.WriteLine("Hit ENTER to terminate application")
Console.ReadLine()

End Sub

End Module

```

## VB.NET: Get Details Of Uploaded Files Using GetUploadedFileDetails()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
4. Start your application. If the response is -3000, you will see the status report. If the response is any other number, check the Web Service Status Codes section.

```

Imports System.IO

Module module1
    Sub Main()
        Const MyUsername As String = "MyUsername"
        Const MyPassword As String = "MyPassword"

```

```

Dim objPM As New com.postalmethods.api.PostalWS()
Dim WSResult As New com.postalmethods.api.GetUploadedFileDetailsResult

' Invoke the DeleteUploadedFile method
WSResult = objPM.GetUploadedFileDetails( _
    MyUsername, _
    MyPassword)

,

' If the result code = -3000, the request was successful.
,
If WSResult.ResultCode = -3000 Then

    For i As Integer = 0 To WSResult.UploadedFiles.Length - 1
        With WSResult.UploadedFiles(i) Console.WriteLine("File Name: {0}",
            .FileName) Console.WriteLine("Description: {0}", .Description)
            Console.WriteLine("Submit Time: {0}", .SubmitTime)
            Console.WriteLine("Permissions: {0}", .Permissions)
            Console.WriteLine("Last Usage: {0}", .LastUsage)
            Console.WriteLine("=====")
        End With
    Next
    Console.WriteLine("Total number of files: " &
WSResult.UploadedFiles.Length)
    Console.WriteLine("=====")

    Else
        ,
        ' A negative value means an error occurred.
        ' See the PostalMethods Status Codes:
http://www.postalmethods.com/statuscodes#webservice.
        ,
        Console.WriteLine("An error occurred: " & WSResult.ResultCode.ToString)
    End If

    ' In debug mode, the following lines prevent your console application from
closing automatically upon termination
    Console.WriteLine("Hit ENTER to terminate application")
    Console.ReadLine()

End Sub

End Module

```

# C#: Send A Letter With SendLetter()

## Instructions

1. Create a new Console Application project and add a web reference
2. Copy the entire code below, delete anything written in your application and paste the code there
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyDescription* with your own text to help you identify this letter in your activity log (optional).
  - o Replace *MyFileExtension* with the extension of the document you are using for this test.
  - o Replace *MyPathToFile* with a path to your test PDF file. Remember, for method *SendLetter* you must use a document where the recipient's address already appears so it will show through the envelope window. You can use one of our Sample Letter Documents.
  - o *MyWorkMode* determines if your letter will be sent as Production, Development or will use your user's default Work Mode setting. Read more about Production and Development Work Modes.
4. Start your application. It will start, encode your document as Base64 and send a letter to the PostalMethods *SendLetter* Web Service method. If the response is a positive number, you will be able to see your letter in the PostalMethods Control Panel.

If the response is negative number, check the Web Service Status Codes section.

That's it - you have successfully sent a letter to PostalMethods. Congratulations.

```
using System;
using System.IO;

namespace ConsoleApplication1
{
    class Module1
    {
        public static void Main()
        {
            const string MyUsername = "MyUsername";
            const string MyPassword = "MyPassword";
            const string MyDescription = "Sending a letter using C#";
            const string MyFileExtension = ".pdf";
            const string MyPathToFile = "C:\\MyFile.pdf";
            const com.postalmethods.api.WorkMode MyWorkMode =
com.postalmethods.api.WorkMode.Default;

            com.postalmethods.api.PostalWS objPM = new
com.postalmethods.api.PostalWS();
            long lngResult = 0;
            //
            // Read your binary document. In this example: C:\\MyFile.pdf
            //
            FileStream objFR = default(FileStream);
            objFR = new FileStream(MyPathToFile, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite);
            byte[] B = new byte[objFR.Length];
            objFR.Read(B, 0, B.Length);
            objFR.Close();
            objFR = null;
        }
    }
}
```

```

        //
        // Invoke the SendLetter method
        //
        lngResult = objPM.SendLetter(MyUsername, MyPassword, MyDescription,
MyFileExtension, B, MyWorkMode);
        //
        // Check response status
        //
        if (lngResult > 0)
        {
            //
            // A positive value means the message was successfully queued for
processing.
            // The PostalMethods Letter ID is returned.
            //
            Console.WriteLine("LetterID is: " + lngResult);
        }

        else
        {
            //
            // A negative value means an error occurred.
            // See the PostalMethods Status Codes:
http://www.postalmethods.com/statuscodes#webservice.
            //
            Console.WriteLine("Error is:" + lngResult);
        }

        // In debug mode, the following lines prevent your console application
from closing automatically upon termination
        Console.WriteLine("Hit ENTER to terminate application");
        Console.ReadLine();
    }
}
}

```

## C#: Send A Letter with SendLetterAndAddress()

### Instructions

1. Create a new Console Application project and add a web reference.
2. Copy the entire code below, delete anything written in your application and paste the code there.
3. Replace the values of the *Const* lines with your details:
  - o Replace *MyUsername* and *MyPassword* with your PostalMethods user details.
  - o Replace *MyDescription* with your own text to help you identify this letter in your activity log (optional).
  - o Replace *MyFileExtension* with the extension of the document you are using for this test.
  - o Replace *MyPathToFile* with a path to your test PDF file. Remember, method *SendLetterAndAddress* will add the recipient's address so it would show through the envelope window. You can use one of our Sample Letter Documents.
  - o *MyWorkMode* determines if your letter will be sent as Production, Development or will use your user's default Work Mode setting. Read more about [Production and Development Work Modes](#).
  - o Use the existing postal recipient or replace with another valid postal address. Leave an empty string ("") where no value is needed.
4. Start your application. It will start, encode your document as Base64 and send a letter to the PostalMethods *SendLetterAndAddress* Web Service method. If the response is a positive

number, you will be able to see your letter in the PostalMethods Control Panel.  
If the response is negative number, check the Web Service Status Codes section.

That's it - you have successfully sent a letter to PostalMethods. Congratulations.

```
using System;
using System.IO;

namespace ConsoleApplication1
{
    class Module1
    {
        public static void Main()
        {
            const string MyUsername = "MyUsername";
            const string MyPassword = "MyPassword";
            const string MyDescription = "Sending a letter using C#";
            const string MyFileExtension = "pdf";
            const string MyPathToFile = "C:\\\\MyFile.pdf";
            const com.postalmethods.api.WorkMode MyWorkMode =
com.postalmethods.api.WorkMode.Default;
            const string MyAttention1 = "George Washington";
            const string MyAttention2 = "";
            const string MyCompanyName = "The White House";
            const string MyAddress1 = "1600 Pennsylvania Ave NW";
            const string MyAddress2 = "";
            const string MyCity = "Washington";
            const string MyState = "DC";
            const string MyPostalCode = "20500";
            const string MyCountry = "United States of America";

            com.postalmethods.api.PostalWS objPM = new
com.postalmethods.api.PostalWS();
            long lngResult = 0;
            //
            // Read your binary document. In this example: C:\MyFile.pdf
            //
            FileStream objFR = default(FileStream);
            objFR = new FileStream(MyPathToFile, FileMode.Open, FileAccess.Read,
FileShare.ReadWrite);
            byte[] B = new byte[objFR.Length];
            objFR.Read(B, 0, B.Length);
            objFR.Close();
            objFR = null;
            //
            // Invoke the SendLetter method
            //
            lngResult = objPM.SendLetterAndAddress(MyUsername, MyPassword,
MyDescription, MyFileExtension, B, MyWorkMode, MyAttention1, MyAttention2,
MyCompanyName,
MyAddress1, MyAddress2, MyCity, MyState, MyPostalCode, MyCountry);

            //
            // Check response status
            //
            if (lngResult > 0)
            {
                //
                // A positive value means the message was successfully queued for
processing.
            }
        }
    }
}
```



```

// Invoke the GetPDF method
//
objResult = objPM.GetPDF(MyUsername, MyPassword, ID);

//
// Check response status
//
if (objResult.ResultCode == -3000)
{
    //
    // -3000 means that the PDF was successfully retrieved. Enter the full
    path to save it in e.g. "c:\mypdf.pdf".
    //
    Console.WriteLine("Successfully retrieved PDF. Please enter a path to
save it in.");
    string strPath = Console.ReadLine();

    //
    // Save the file
    //
    FileStream objFS = new FileStream(strPath, FileMode.Create,
FileAccess.Write);
    objFS.Write(objResult.FileData, 0, objResult.FileData.Length);
    objFS.Close();
    objFS = null;
}
else
{
    //
    // A value less than -3000 means an error occurred.
    // See the PostalMethods Status Codes:
    http://www.postalmethods.com/statuscodes.
    //
    Console.WriteLine("Error is:" + objResult.ResultCode);
}

// In debug mode, the following lines prevent your console application
from closing automatically upon termination
Console.WriteLine("Hit ENTER to terminate application");
Console.ReadLine();
}
}
}

```