

Forum:[Hack Forums](#)

Thread: [Admin Page Finder \[Completely Featured\] \[Semi-Tutorial\]](#)

Malum™ - December 27, 2016, 4:35 pm

So, I saw [NinjaLeaks' Admin Page Finder](#) and it seemed a bit basic and it was one of his first projects. So I thought I'd improve upon it and re-do it. Something fun and easy to do. :)

What I Used:

[Lists](#)

[For Each](#)

[Environment Exit Codes](#) (Forgive me for any mistakes, one of my first times using it).

[Stream Readers](#)

[Stream Writers](#)

[How To Determine If Valid URL](#)

[SaveFileDialogs](#)

[DialogResults](#)

[Command Line Arguments](#)

[WebRequest](#)

[\\$ Operator instead of String.Format](#)

Any other feature should be the absolute basics for beginners to understand (If statements, System.IO.Path/File, etc).

Screenshot:

Spoiler [\(Click to View\)](#)

Code: Spoiler [\(Click to View\)](#)

Code Imports System

Imports System.IO

Imports System.Net

Imports System.Threading

Imports System.Windows.Forms

Module Program

Private Const ERROR_BAD_ARGUMENTS As Int16 = &H1

Private Const ERROR_INVALID_FILE As Int16 = &H2

Private Const ERROR_CANT_READ_FILE As Int16 = &H3

Private Const ERROR_INVALID_URL As Int16 = &H4

Private Const ERROR_FAILED_SEARCH As Int16 = &H5

Private Pages As New List(Of String)()

Private Duplicates As Integer = 0

Private Timeout As Integer = 5000

Sub Main(ByVal args As String())

Console.Title = "Admin Page Finder"

```

Console.WriteLine("Author: Malum")
Console.WriteLine("Date: 12/26/2016" + Environment.NewLine)

If (args.Length < 1) Then
    FatalError("Must input a file to read admin pages from!", ERROR_BAD_ARGUMENTS)
End If

For Each f As String In args
    If (Not File.Exists(f) And args.Length < 1) Then
        FatalError($"{Path.GetFileNameWithoutExtension(f)} does not exist in the current
context!", ERROR_INVALID_FILE)
    ElseIf (Not File.Exists(f)) Then
        Log($"{Path.GetFileNameWithoutExtension(f)} does not exist in the current context!",
ConsoleColor.Red)
    End If

    Log($"Reading admin pages from {Path.GetFileName(f)}..")

    Try
        Using sReader As New StreamReader(f)
            Dim Line As String = sReader.ReadLine()

            Do While (Not Line Is Nothing)
                If (Pages.Contains(Line)) Then
                    Duplicates += 1
                Else
                    Pages.Add(Line)
                End If

                Line = sReader.ReadLine()
            Loop

            sReader.Close() : sReader.Dispose()
        End Using

    Catch ex As Exception
        FatalError($"Error! {ex.Message}", ERROR_CANT_READ_FILE)
    End Try

    Log($"Successfully read all {Pages.Count} admin pages!")
Next

Log($"Removed {Duplicates} duplicates from all lists!")

Console.Write($"[{DateTime.Now.ToShortTimeString}] Enter Timeout (In Milliseconds)> ")

Dim sTimeout As String = Console.ReadLine()

If (Not ValidTimeout(sTimeout)) Then
    Log($"{sTimeout}' is not a valid timeout. Default remains at 5000ms", ConsoleColor.Red)
Else
    Timeout = Convert.ToInt32(sTimeout)
End If

```

```
Console.Write($"{DateTime.Now.ToShortTimeString}} Enter URL (EX:  
http://www.google.com)> ")
```

```
Dim Input As String = Console.ReadLine()
```

```
If (Not ValidURL(Input)) Then
```

```
    FatalError($"{Input}'is not a valid URL..", ERROR_INVALID_URL)  
End If
```

```
If Not Input.EndsWith("/") Then
```

```
    Input += "/"  
End If
```

```
Log($"Starting search for admin pages on {Input}..")
```

```
Search(Input, Pages)
```

```
Console.ReadLine()
```

```
End Sub
```

```
Function ValidTimeout(ByVal time As String) As Boolean
```

```
    For i As Integer = 0 To time.Length - 1
```

```
        If (Not Char.IsNumber(time(i))) Then
```

```
            Return False
```

```
        End If
```

```
    Next
```

```
    Return True
```

```
End Function
```

```
Function ValidURL(ByVal URL As String) As Boolean
```

```
    Return (Uri.IsWellFormedUriString(URL, UriKind.Absolute))
```

```
End Function
```

```
#Region "Log & Error"
```

```
Sub Log(ByVal Text As String)
```

```
    Log(Text, ConsoleColor.White)
```

```
End Sub
```

```
Sub Log(ByVal Text As String, ByVal Color As ConsoleColor)
```

```
    Dim cColor As ConsoleColor = Console.ForegroundColor
```

```
    Console.Write($"{DateTime.Now.ToShortTimeString}} ")
```

```
    Console.ForegroundColor = Color
```

```
    Console.Write(Text + Environment.NewLine)
```

```
    Console.ForegroundColor = cColor
```

```
End Sub
```

```
Sub FatalError(ByVal Err As String, ByVal Code As Int16)
```

```
    Log($"Error: {Err}", ConsoleColor.Red)
```

```
    Console.ReadLine()
```

```
    Environment.Exit(Code)
```

```
End Sub
```

#End Region

#Region "Admin Page Search & Export"

Sub Search(ByVal Input As String, ByVal Pages As List(Of String))

Try

Dim Count As Integer = 0

Dim fPages As New List(Of String)()

For Each p As String In Pages

If (FindPage(Input, p)) Then

fPages.Add(Input + p)

Log(\$"Found admin page at [{Input + p}]!", ConsoleColor.Green)

Count += 1

End If

Next

If (Count < 1) Then

Log("No admin pages found.")

Else

Log(\$"Found {Count} admin pages. Exporting..")

Using sfd As New SaveFileDialog() With {.Title = "Export Found Pages", .Filter = "Text
File (*.txt)*.txt"}

If (sfd.ShowDialog() = DialogResult.OK) Then

Export(fPages, sfd.FileName)

End If

End Using

End If

Catch ex As Exception

FatalError(\$"Error! {ex.Message}", ERROR_FAILED_SEARCH)

End Try

End Sub

Sub Export(ByVal Pages As List(Of String), ByVal dest As String)

Try

Using sWriter As New StreamWriter(dest) With {.AutoFlush = True}

For Each page As String In Pages

sWriter.WriteLine(page)

Next

sWriter.Close() : sWriter.Dispose()

End Using

Catch ex As Exception

End Try

End Sub

Function FindPage(ByVal URL As String, ByVal Page As String) As Boolean

Dim req As WebRequest = WebRequest.Create(URL + Page)

req.Timeout = Timeout

Try

Return (DirectCast(req.GetResponse(), HttpWebResponse).StatusDescription = "OK")

```
Catch : Return False : End Try
Return False
End Function
#End Region
End Module
```

How To Use:

Drag n' Drop text files with admin page lists. You can use more than one file.

Enter valid URL.

Wait for it to finish.

Save the results!

Some Things To Note: You can change the Timeout for the WebRequest.

There is a check if "/" exists after the inputted URL if not than it adds it.

I did my best to error check but it gets old. So Try/Catch is used quite a bit. I'm sorry :(

Just added a check for duplicates and a prints out how many were removed.

Added ability to change Timeout for WebRequest

TODO: Add a more in-depth check if Admin Page is valid.

Limit how much web traffic at a time.

Possibly multi-thread depending on how large the list is?

Create a UI version?

Any questions, please ask! Any suggestions, go ahead and say them. Politely please. :)

debug - December 28, 2016, 6:45 pm

Looks pretty good. But just glancing at your screenshot, I think you could improve it more by weeding out false positives. For instance, you could check the elements on the page for a user and password field.

Malum™ - December 28, 2016, 7:32 pm

(Dec 28, 2016 07:45 PM)debug Wrote: [\[CITE\]](#) Looks pretty good. But just glancing at your screenshot, I think you could improve it more by weeding out false positives. For instance, you could check the elements on the page for a user and password field.

That's a great idea. I'll add it to the ToDo list but in the meantime I think checking if the page exists is a good start.

