



Visual Basic.NET and EPANET Toolkit Usage

[EPANET
Toolkit-Users](#)

[atifselcuk](#) 2015-07-18 07:24:03 UTC #1

This EPANET Forum Category is intended to find Q&A and share the experiences for "Visual Basic.NET" and "EPANET Toolkit" Users.

- There are less number of software developers using "Visual Basic.NET". Most of the developers uses "C". So it becomes difficult to find someone to ask and probably to find an exact answer. Lots of developers gives answers in "C". But you can't find its equalance in "Visual Basic.NET". Then it becomes difficult to understand each other.
- I believe it will be fast and easy way to ask and to find a solution for "Visual Basic.NET" users. And also to share experiences directly.
- Depending on my experiences, it is not necessary to merge This Catagory with another category, or split it into more categories? But if someone creates more catagories like "Visual C and EPANET Toolkit Usage", "Delphi and EPANET Toolkit Usage" or something like this. It will be hepful to software developers.

In my professional works; I am developing softwares for water supply network design. I use "Microsoft Visual Studio 2010" and "Visual Basic .NET Framework". I am using "epanet2.dll" and "epanet2.bas" Visual Basic files to use EPANET. During this works I recognized some declaration problems. I will share them with you next time.

Best Regards

[atifselcuk](#) 2015-07-18 07:45:28 UTC #2

Hi,

I am developing softwares for water supply network design. I use "Microsoft Visual Studio 2010" and "Visual Basic .NET Framework". I am using two Visual Basic files ("epanet2.dll" and "epanet2.bas") to use EPANET. During this works I recognized some declaration problems.

AS-issue-01>EPA

The declaration in "epanet2.bas", it didn't work as below;

Declare Function ENgetcount Lib "epanet2.dll" (ByVal Code As Long, ByVal Value As Long) As Long

I changed it as following and it works;

Declare Function ENgetcount Lib "epanet2.dll" (ByVal Code As Int32, ByVal Value As Int32) As Int32

AS-issue-02>EPA
In "epanet2.bas";
Declare Function ENgetnodeid Lib "epanet2.dll" (ByVal Index As Long, ByVal ID As String) As Long
My usage;
Declare Function ENgetnodeid Lib "epanet2.dll" (ByVal Index As Int32, ByVal ID As Int32) As Int32

AS-issue-03>EPA
In "epanet2.bas";
Declare Function ENgetnodevalue Lib "epanet2.dll" (ByVal Index As Long, ByVal Code As Long, ByVal Value As Single) As Long
My usage;
Declare Function ENgetnodevalue Lib "epanet2.dll" (ByVal Index As Int32, ByVal Code As Int32, ByVal Value As Single) As Int32

AS-issue-04>EPA
In "epanet2.bas";
Declare Function ENgetlinkid Lib "epanet2.dll" (ByVal Index As Long, ByVal ID As String) As Long
My usage;
Declare Function ENgetlinkid Lib "epanet2.dll" (ByVal Index As Int32, ByVal ID As Int32) As Int32

AS-issue-05>EPA
In "epanet2.bas";
Declare Function ENgetlinkvalue Lib "epanet2.dll" (ByVal Index As Long, ByVal Code As Long, ByVal Value As Single) As Long
My usage;
Declare Function ENgetlinkvalue Lib "epanet2.dll" (ByVal Index As Int32, ByVal Code As Int32, ByVal Value As Single) As Int32

AS-issue-06>EPA
In "epanet2.bas";
Declare Function ENrunH Lib "epanet2.dll" (ByVal T As Long) As Long
My usage;
Declare Function ENrunH Lib "epanet2.dll" (ByRef T As Int32) As Int32

AS-issue-07>EPA
In "epanet2.bas";
Declare Function ENnextH Lib "epanet2.dll" (ByVal Tstep As Long) As Long
My usage;
Declare Function ENnextH Lib "epanet2.dll" (ByRef Tstep As Int32) As Int32

Whenever I find anything else, I will share with you.

Best Regards

[atifselcuk](#) 2015-07-18 08:03:04 UTC #3

Hi,

I am sending 1 more issue for "Visual Basic .NET" users. I am using "epanet2.dll" and "epanet2.bas" files to use EPANET.

AS-issue-08>EPA"
The function "ENgetnodeid(i, id)" works for the cases "i<=256", stops when "i>256". I think "i" is declared as "byte" in "epanet2.dll". I am sending a subroutine that I used.

Are there anyone having the same problem?

Best Regards,

Public Sub sub_ENgetnodeid_TEST()

```
Try
'Open the toolkit & obtain a hydraulic solution
Call ENOpen("C:\test\TO-EPANET.INP", _
            "C:\test\FROM-EPANET.RPT", "")

'---4.6---/* Solve hydraulics */
Call ENSolveH()

Dim i As Int32, NumNodes As Int32
Dim t As Int32, tstep As Int32
Dim p As Single
Dim id As String
Dim errkodu As Int32
Call ENGetcount(EN_NODECOUNT, NumNodes)
MsgBox("NumNodes...<" & NumNodes & ">")

Call ENOpenH()

'---4.5---/* Initialize hydraulics */
Call ENinitH(0)

Do
'---4.4---/* Make a single period run */
Call ENrunH(t)
For i = 1 To NumNodes
    errkodu = ENgetnodeid(i, id)
    MsgBox("i.....=" & i & "/" & NumNodes & vbCrLf & vbCrLf _
           & "id.....=" & id & vbCrLf _
           & "errkodu...=" & errkodu)

    Call ENgetnodevalue(i, EN_PRESSURE, p)
    MsgBox("i....=" & i & "/" & NumNodes & vbCrLf & vbCrLf _
           & "t....=" & t & vbCrLf _
           & "id...=" & id & vbCrLf _
           & "p....=" & p)
Next i
    Call ENnextH(tstep)
Loop While (tstep > 0)

Call ENcloseH()
Call ENclose()
End Try
```

End Sub

[LRossman](#) 2015-07-20 15:05:31 UTC #4

The “epanet2.bas” file was meant to be used with VB6 or VBA (Visual Basic for Applications), not with VB.Net. To make it compatible with VB.Net you can make the following changes and save the resulting file as “epanet2.vb”:

1) Add the following lines to the top of the file:

```
Imports System.Runtime.InteropServices
Imports System.Text
Module Epanet2
```

- 2) Replace all occurrences of "GLOBAL" with "Public".
- 3) Replace all occurrences of "LONG" with "INT32".
- 4) Specify all function arguments as "ByRef" unless they are already specified as "ByVal".
- 5) Replace "ByVal ID As String" in ENgetpatternid, ENgetnodeid, and ENgetlinkid with "ByVal ID As StringBuilder".
- 6) Replace "ByVal ErrMsg As String" in ENgeterror with "ByVal ErrMsg As StringBuilder".
- 7) In ENsetpattern, replace "ByRef F As Any" with "ByRef F As Single".
- 8) Add "End Module" to the end of the file.

Use epanet2.vb as a module added to your VB.Net projects. Whenever you need to get the text of a pattern, node or link ID using the getXXXid functions, declare the variable to be passed into the function as follows:

```
Dim Id As New StringBuilder(32)
```

and use Id.ToString to get the String representation of the ID. Do something similar for the ErrMsg argument in ENgeterror, but size it to at least 80 characters.

[samhatchett](#) 2015-07-21 13:14:59 UTC #5

Thanks all for the discussion. [@atifselcuk](#) it would be very helpful if you could file these issues on the issues tracker: <https://github.com/OpenWaterAnalytics/EPANET/issues> so that they are not lost in a discussion forum. I've created the .NET compatibility milestone: <https://github.com/OpenWaterAnalytics/EPANET/milestones/VB.NET%20compatibility> so that the issues can be linked to it. This will just help everyone benefit from these discussions.

[eladsal](#) 2015-07-21 14:32:03 UTC #6

[@atifselcuk](#) if you build a the module file as [@LRossman](#) suggested we can place it in the [include](#) directory of the DLL.

[atifselcuk](#) 2015-07-22 06:05:27 UTC #7

Hi Lewis,

I followed your advices, most of them works good, a few of them doesn't. "ENgetnodeid" and "ENgetlinkid" didn't work when I use "StringBuilder" definition. Stopped during run. I tried different new combinations and I reached the result. Those 2 function works good if they are like below;

```
Declare Function ENgetnodeid Lib "epanet2.dll" (ByVal Index As Int32, ByRef ID As Int32) As Int32
Declare Function ENgetlinkid Lib "epanet2.dll" (ByVal Index As Int32, ByRef ID As Int32) As Int32
```

I want to ask another thing to you; I have a test network, it has 281 nodes (node numbers are from 1 to 281) and 416 links.

AS-issue-09>EPA

When I use "ENgetnodeid" function, I get very big "ID" numbers like 3290674. Why they are not from 1 to 281. When I use "Engetlinkid" function, I get very big "ID" numbers like 3224113. Why they are not from 1 to 416.

[samhatchett](#) 2015-07-22 13:14:40 UTC #8

I'm not familiar with .NET development, but it seems like there's a major data type mismatch here, in trying to retrieve an element ID (string) as an integer ref. Maybe [@eladsal](#) would know more, but there should be a more elegant way to do this -- a quick search turns up [something like this](#), and I've also seen good things about SWIG which automagically creates C# wrappers for C libraries.

I responded to Atif privately, but in case others are interested:

ID labels in EPANET are defined as C-type character strings that can hold up to 31 characters plus a null end-of-string character. They are not integer numbers. My guess as to why Atif's ID labels were not making sense is that the "ByRef ID As Int32" argument in his function declaration is returning just the memory address of the start of the ID string and not the entire string. You have to reserve enough memory in your calling program to accept the string returned by the ENgetXXXXid function. The VB method for doing this is to use a StringBuilder object sized to hold at least 32 characters.

I had success using the following VB.Net code to list the ID names of either all nodes or all links in an input file in a listbox:

```
Private Sub ListIDs(IdType As Int32)
    Dim R As Int32
    Dim NumItems As Int32
    Dim I As Int32
    Dim Id As New StringBuilder(32)
    Dim Fname As String

    ListBox1.Items.Clear()
    Fname = TextBox1.Text
    R = ENopen(Fname, "Epanet.rpt", "")
    If R <> 0 Then
        MsgBox("Could not open input file.")
        Return
    End If
    If IdType = EN_NODECOUNT Then
        ENgetcount(EN_NODECOUNT, NumItems)
        For I = 1 To NumItems
            ENgetnodeid(I, Id)
            ListBox1.Items.Add("Node " & I & " - " & Id.ToString)
        Next I
    End If
    If IdType = EN_LINKCOUNT Then
        ENgetcount(EN_LINKCOUNT, NumItems)
        For I = 1 To NumItems
            ENgetlinkid(I, Id)
            ListBox1.Items.Add("Link " & I & " - " & Id.ToString)
        Next I
    End If
    ENclose()
End Sub
```

The declarations module used with this code contains the following:

```
Imports System.Runtime.InteropServices
```

```
Imports System.Text
```

```
Module Epanet2
```

```
Public Const EN_NODECOUNT = 0
```

```
Public Const EN_LINKCOUNT = 2
```

```
Declare Function ENopen Lib "epanet2.dll" (ByVal F1 As String, ByVal F2 As String, ByVal F3 As String) As Int32
```

```
Declare Function ENclose Lib "epanet2.dll" () As Int32
```

```
Declare Function ENgetcount Lib "epanet2.dll" (ByVal Code As Int32, ByRef Value As Int32) As Int32
```

```
Declare Function ENgetnodeid Lib "epanet2.dll" (ByVal Index As Int32, ByVal ID As StringBuilder) As Int32
```

```
Declare Function ENgetlinkid Lib "epanet2.dll" (ByVal Index As Int32, ByVal ID As StringBuilder) As Int32
```

```
End Module
```

Lew Rossman

[atifselcuk](#) 2015-07-23 12:36:13 UTC #10

Hi Lew,

I recognised my mistake by doing some tests depending on your last mail.

I was using "ByRef ID as StringBuilder" in "ENgetnodeid" and "ENgetlinkid" functions. I assumed that those functions should return a value thru "ID" parameter. When I used "ByVal", everything started to work perfect. Thank you, this was my mistake.

I am correcting my last issue.

AS-issue-09>EPA

The correct usage of those 2 functions are below;

```
Declare Function ENgetnodeid Lib "epanet2.dll" (ByVal Index As Int32, ByVal ID As StringBuilder) As Int32
```

```
Declare Function ENgetlinkid Lib "epanet2.dll" (ByVal Index As Int32, ByVal ID As StringBuilder) As Int32
```

[atifselcuk](#) 2015-07-30 07:05:10 UTC #11

Hi,

I am sending 4 more issues for "Visual Basic .NET" users. I am using "epanet2.dll" and "epanet2.bas" files to use EPANET.

AS-issue-10>EPA"

In "epanet2.bas";

```
Declare Function ENSaveinfile Lib "epanet2.dll" (ByVal F As String) As Long
```

My usage;

```
Declare Function ENSaveinfile Lib "epanet2.dll" (ByVal F As String) As Int32
```

In "epanet2.bas";

```
Declare Function ENgetnodeindex Lib "epanet2.dll" (ByVal ID As String, ByVal Index As Long) As Long
```

My usage;

```
Declare Function ENgetnodeindex Lib "epanet2.dll" (ByVal ID As String, ByVal Index As Int32) As Int32
```

In "epanet2.bas";

```
Declare Function ENgetnodetype Lib "epanet2.dll" (ByVal Index As Long, ByVal Code As Long) As Long
```

My usage;

```
Declare Function ENgetnodetype Lib "epanet2.dll" (ByVal Index As Int32, ByVal Code As Int32) As Int32
```

In "epanet2.bas";

```
Declare Function ENgetlinknodes Lib "epanet2.dll" (ByVal Index As Long, ByVal Node1 As Long, ByVal Node2 As Long) As Long
```

My usage;

```
Declare Function ENgetlinknodes Lib "epanet2.dll" (ByVal Index As Int32, ByVal Node1 As Int32, ByVal Node2 As Int32) As Int32
```

[eladsal](#) 2015-07-30 08:15:54 UTC #12

An updated vb.net module is available on GitHub (courtesy of Lew):

github.com

[OpenWaterAnalytics/EPANET/blob/patch-2-0-13/include/epanet2.vb](https://github.com/OpenWaterAnalytics/EPANET/blob/patch-2-0-13/include/epanet2.vb)

```
'EPANET2.VB
'
'Declarations of functions in the EPANET PROGRAMMERS TOOLKIT
'(EPANET2.DLL) for use with VB.Net.

'Last updated on 7/19/15 - LR

Imports System.Runtime.InteropServices
Imports System.Text

Module Epanet2

' These are codes used by the DLL functions
Public Const EN_ELEVATION = 0      ' Node parameters
Public Const EN_BASEDEMAND = 1
Public Const EN_PATTERN = 2
Public Const EN_EMITTER = 3
Public Const EN_INITQUAL = 4
Public Const EN_SOURCEQUAL = 5
Public Const EN_SOURCEPAT = 6
```

This file has been truncated. [show original](#)

[Home](#) [Categories](#) [FAQ/Guidelines](#) [Terms of Service](#) [Privacy Policy](#)

Powered by [Discourse](#), best viewed with JavaScript enabled