HYPERION

AN ANTF PLATFORM

Back to basics

Simplify VB/C# OE code

Be More Creative!

MYPERION

AN ANTF PLATFORM

Understanding

MyPerion Framework

The MyPerion Framework is a powerful, portable DLL driven framework designed to deliver fast and reliable solutions aimed at helping VB & C# developers simplify and automate a variety of operations in their projects and to create a standard for virtual OE's and other projects.

Legal Information

This software is subject to international copyright laws and is protected under both international and domestic applicable laws.

This software is intened to help aspiring VB/C# programmers to simplify often tedious coding operations.
This software package is inteneded for individuals, hobbyists, academic uses (Not Corporate Use)

Violations of this agreement include doing the following with ANTF MyPerion. While there are many terms, here is a few of the big ones:

- Reverse Engineering the DLL(s) or source code
- Mass distribution on non-ANTF sites or spaces
- Selling MyPerion on non-sactioned platforms without written consent from the developer.
- Changing the product and trying to pass it off as your own
- Not crediting MyPerion's use in your project comments
- Using the software package in a corporate application without written consent from the developer.

A More detailed EULA can be found on our website (https://antfcorp.com)

For Support or Contact

- https://antfcorp.com/support
- support@antfcorp.com

For more information on US copyright code. Click Here

©opyright

Table of Contents

Integrating Into VB.Net/C#

Add MyPerion to your current project and get working

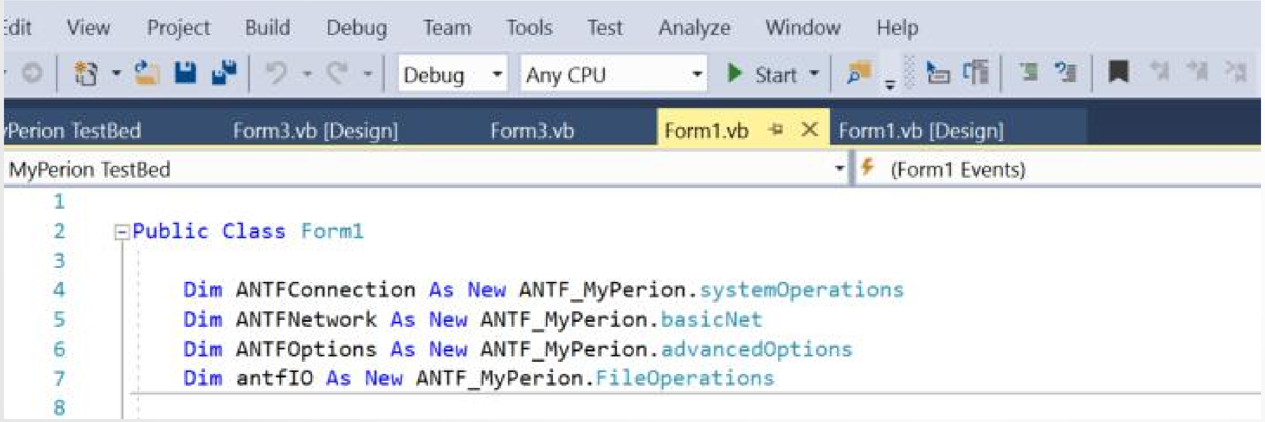
MYPERION
AN ANTF PLATFORM

Adding MyPerion to a VB Project

Adding MyPerion to your VB project is Easy! Simply do the following

1. Click on My Project, then References
2. Click the “Add” Button within the References Tab,a dialog will Appear
3. Navigate to C:\ANTF Group\ANTF MyPerion Framework or wherever you chose to install MyPerion and add the ANTF MyPerion.dll file to your project

Now that that part is done, simply go into your Windows form or class, or module, a nd create an instance of the DLL See Below. That's all there is to it



```

Public Class Form1

    Dim ANTFConnection As New ANTF_MyPerion.systemOperations
    Dim ANTFNetwork As New ANTF_MyPerion.basicNet
    Dim ANTFOptions As New ANTF_MyPerion.advancedOptions
    Dim antfIO As New ANTF_MyPerion.FileOperations

```

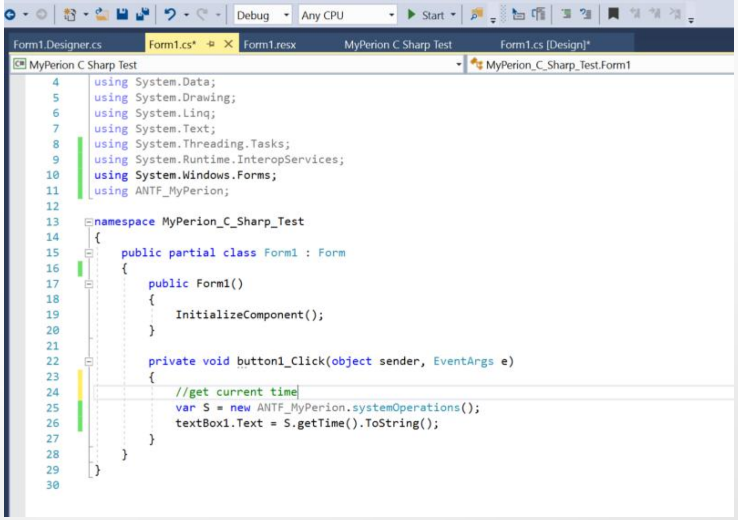
Adding MyPerion to a C# Project

Adding MyPerion to your C# project is Easy, But Tricky!

- Right click on references in the solution explorer, and choose “add reference”. a dialog appears
- Navigate to C:\ANTF Group\ANTF MyPerion Framework or wherever you chose
- Now at the top of your project type “using ANTF_MyPerion;”

Now that that part is done, simply go into your Windows form or class, or module, and create an instance of the DLL See Below as a var or object. That's all there is to it

Remember: C# is typecast heavy. Remember to convert return information as illustrated below.



```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Runtime.InteropServices;
using System.Windows.Forms;
using ANTF_MyPerion;

namespace MyPerion_C_Sharp_Test
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //get current time
            var S = new ANTF_MyPerion.systemOperations();
            textBox1.Text = S.getTime().ToString();
        }
    }
}
```

Networking

Access to addresses, domains, and more

Example connection: dim basicNet As New ANTF_MyPerion.basicNet

Call Syntax	Description	Return type
getNetworkStatus()	returns a 0 for a network connection detected, or a 1 for no connection detected.	integer
getAddress()	returns current IP address of the local machine	string
getName()	returns name of the local machine	string
getNetworkStatDump()	Returns information about network gateways, IP's, an d other prevelant info (similar to ipconfig)	string

Example connection: dim advNet As New ANTF_MyPerion.advancedOptions

Call Syntax	Description	Return type
getDomainController()	returns the name of the domain host controller of the local machine/network	string
getNetworkName()	returns name of the current internet network that the local machine is currently connected to	string
getNetSpeed()	returns average network speed result of internet connection expressed as mbps	double (rounded)

Note: *Some functions exist within this section that are currently not in production due to delays and technical issues such as downloading files. These features will be available in coming updates.*

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Get Network Status

```
Dim a As Integer = ANTFNetwork.getNetworkStatus
If a = 1 Then
    Op.Text = "No Network Detected (Offline)"
Else
    Op.Text = "Network Detected (Online)"
End If
```

Get Address

```
Op.Text = ANTFNetwork.GetAddress
```

Get Name

```
Op.Text = ANTFNetwork.GetName
```

Get Network Dump

```
Op.Text = ANTFNetwork.GetNetworkStatDump
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Get Domain Controller

```
Op.Text = ANTFOptions.GetDomainController
```

Get Network Name

```
Op.Text = ANTFOptions.GetNetworkName
```

Get network speed

```
Op.Text = ANTFOptions.getNetSpeed.ToString + " mbps"
```

File IO Operations

Read/Write/Create Files, Folders, and more

MYPERION
AN ANTF PLATFORM

Example connection: dim antfIO As New ANTF_MyPerion.fileOperations

Call Syntax	Description	Return type
readTextBasedFile(path, fileName)	returns all text content from basic text based files	string
overWriteTextBasedFile(path, fileName, content)	overwites specified basic text based files with new content. Returns a 0 for s uccess, or a 1 for failure	integer
appendTextBasedFile(path, fileName, content)	adds new content to specified basic text based files. Can return 1 for failure	integer
createBasicTextFile(path, fileName)	creates a blank new text file (.txt) in a specified directory	integer
createCustomBasicTextFile(path,fileName,extension)	creates a blank new text based file in a specified directory with a custom extension.	integer
fetchSiteData(url)	returns all the HTML source code from a specified website.	string

Example connection: dim antfIO As New ANTF_MyPerion.fileOperations

Call Syntax	Description	Return type
createDirectory(path, dirName)	allows for the creation of a new folder on the local machine or home drives such as a personal NAS. Returns a 0 for success, or a 1 for failure.	integer
createEntireDirectory(basePath, parentFolderName, nameArray())	allows for the creation a new folder and various sub folders to all be created at one utilizing an array. Returns a 0 for success, or a 1+ for failure.	integer
deleteDirectory(path, directoryName)	deletes a folder from the local machine including all its sub folders and files. Returns a 0 for success, or a 1 for failure.	integer
deleteFile(path, fileName)	deletes a file from the local machine. Returns a 0 for success, or a 1 for failure.	integer

Example connection: dim antfIO As New ANTF_MyPerion.fileOperations

Call Syntax	Description	Return type
fileFinder(file)	checks to see if a specified file exists in a specified location. Returns true or false	boolean
folderFinder(path)	checks to see if a specified directory exists in a specified location. Returns true or false	boolean
textFind(filePath, textSearch)	checks to see if specified text exists within a basic text file. Returns true or false	boolean
textFindComparetor(filePath1, filePath2, textSearch)	checks to see if specified text exists within two basic text files. (like a comparison) Returns true or false	boolean

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Read Basic Text File

```
Try
    Op.Text = antfIO.readTextBasedFile("C:\ANTF Group", "Doggo.txt")
Catch ex As Exception
    Op.Text = "The File Cannot Be Found"
End Try
```

Append basic Text File

```
Dim check As Integer = antfIO.appendTextBasedFile("C:\ANTF Group\", "Doggo.txt", Op.Text)
If check = 1 Or check = 2 Then
    Op.Text = ("The changes Failed! Error Code: " + check.ToString)
Else
    Op.Text = "The changes were Successful"
End If
```

Overwrite Basic Text File

```
Dim check As Integer = antfIO.overWriteTextBasedFile("C:\ANTF Group\", "Doggo.txt", Op.Text)
If check = 1 Or check = 2 Then
    Op.Text = ("The changes Failed! Error Code: " + check.ToString)
Else
    Op.Text = "The changes were Successful"
End If
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Create Blank text File

```
antfIO.createBasicTextFile("C:\ANTF Group", "Doggo")
```

Create Custom File

```
antfIO.createCustomBasicTextFile("C:\ANTF Group", "frogy", ".apv")
```

Get Site Data

```
Op.Text = antfIO.fetchSiteData("https://www.apple.com")
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Create Directory

```
Dim check As Integer = antfIO.createDirectory("C:\ANTF Group\", "MyPerion Testing")
If check = 1 Then
    Op.Text = ("Directory Creation Failed! Error Code: " + check.ToString)
Else
    Op.Text = "Directory Creation Successful"
End If
```

Create Entire Directory

```
Dim folderGrid(4) As String    'ProTip (This is a string array in VB.Net)
folderGrid(0) = "Test 1"
folderGrid(1) = "Test 2"
folderGrid(2) = "Test 3"
folderGrid(3) = "Test 4"
Dim check As Integer = antfIO.createEntireDirectory("C:\ANTF Group\MyPerion Testing\", "Directory Testing", folderGrid)
If check = 1 Then
    Op.Text = ("Directory Creation Failed! Error Code: " + check.ToString)
Else
    Op.Text = "Directory Creation Successful"
End If
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Delete Entire Directory/Folder

```
Dim check As Integer = antfIO.deleteDirectory("C:\ANTF Group\MyPerion Testing\", "Directory Testing")
If check = 1 Then
    Op.Text = ("Directory Deletion Failed! Error Code: " + check.ToString)
Else
    Op.Text = "Directory Deletion Successful"
End If
```

Delete File

```
Dim check As Integer = antfIO.deleteFile("C:\ANTF Group\", "Doggo.txt")
If check = 1 Or check = 2 Then
    Op.Text = ("File Deletion Failed! Error Code: " + check.ToString)
Else
    Op.Text = "File Deletion Successful"
End If
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Check If File Exists

```
Private Function checkFile(ByRef filepath)
        Dim check As Boolean = antfIO.fileFinder(filepath)
        If check = False Then
            Op.Text = "The File was either not created or not found"
        Else
            Op.Text = "The File was Created and Located in its Proper Directory"
        End If
End Function
```

Check If Directory/Folder Exists

```
Private Function checkDir(ByRef dirPath)
        Dim check As Boolean = antfIO.folderFinder(dirPath)
        If check = False Then
            Op.Text = "The File was either not created or not found"
        Else
            Op.Text = "The File was Created and Located in its Proper Directory"
        End If
End Function
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Find Text In File

```
Dim check As Boolean = antfIO.textFind("C:\ANTF Group\Test 1.txt", "white")
If check = False Then
    Op.Text = ("White was not Found in Test 1.txt Error Code: " + check.ToString)
Else
    Op.Text = "White was Found in Test 1.txt!"
End If
```

Find Text In Multiple Files

```
Dim check As Boolean = antfIO.textFindComparetor("C:\ANTF Group\Test 1.txt", "C:\ANTF Group\Test 2.txt", "dog")
If check = False Then
    Op.Text = ("dog was not Found in both Files Error Code: " + check.ToString)
Else
    Op.Text = "dog was Found in both files!"
End If
```

System Operations

Access System Data, Run Commands, Open Items, and Interface with compatible ANTF software, and more

Example connection: dim antfSystem As New ANTF_MyPerion.systemOperations

Call Syntax	Description	Return type
startWinExplorer()	restarts the Windows Explorer interface on Windows including the taskBar	None
killWinExplorer()	kills the Windows Explorer Interface including the taskBar and disables the Win key.	None
killProcess(processName)	kills a specified process running on the local machine.	None

Example connection: dim antfSystem As New ANTF_MyPerion.systemOperations

Call Syntax	Description	Return type
launchApp(instancePath)	launches an application, file, or other process based on its full path location or system process name.	None
launchSite(webURL)	opens the default web browser on the system and/or opens a new tab with the specified website address.	None
launchFocusedApplication(instancePath)	some applications cannot be launched by shelling an instance. Instead some rely on being opened through Windows Focusing (Normal Win. Parameters)	None

Example connection: dim antfSystem As New ANTF_MyPerion.systemOperations

Call Syntax	Description	Return type
shellControlPanel()	opens an instance of the Windows Control Panel (classic version)	None
shellPrinterSettings()	opens the control panel applet containing information about printers and applicable settings	None
shellNetworkNeighborhood()	opens the control panel applet containing information about internet connections and networks	None
shellIESettings()	opens the Windows Internet Explorer/System (Not Edge) Internet settings property menu	None
shellCMD()	opens an instance of the Windows Command Prompt	None
runCommand(command)	shells a command via Windows Command Prompt and returns a string containing the output of the command.	String
shutDownPC()	shuts down the local machine	None
rebootPC()	restarts the local machine	None

Example connection: dim antfSystem As New ANTF_MyPerion.systemOperations

Call Syntax	Description	Return type
getDate()	returns the crrent date expressed as 08/23/2018	string
getTime()	returns current time in hour, minute (AM/PM)	string
getPrecisionTime()	returns current time in hour, minute, and second (AM/PM)	string
getMonth()	returns the current month expressed as the word (January, May, August...)	string
getDay()	returns the current day of the week within the month expressed numerically (1,23,31...)	integer
getWeekDay()	returns the current day of the week expressed as the word (Monday, Friday, Snday...)	string
getYear()	returns the current year	integer
withinDST()	checks to see if today is within daylight savings time (US Only)	boolean

Example connection: dim antfSystem As New ANTF_MyPerion.systemOperations

Call Syntax	Description	Return type
getOSName()	returns the full name of the current version of Windows installed on the local machine such as (Windows 10 Home Edition)	string
getOSVersion()	returns the version instance of Windows installed on the local machine	string
getOSPlatform()	returns the Windows platfrom version	string
getCurrentWindowsUser()	returns the username of the user currently signed into Windows on the local machine	string

Example connection: dim antfSystem As New ANTF_MyPerion.systemOperations

Call Syntax	Description	Return type
getANTFMyPerionVersion()	returns the current revision numer of the current version of ANTF MyPerion that your project is currently using	string
getANTFDev()	returns copyright information for MyPerion, the developer, and contact/support information	string
checkForNebula(driveLetter)	returns a True or False answer depending on if it found an instance of ANTF Nebula (ANTF's Virtual Operating Enviroment) in the drive specified	boolean
ANTFWingITSearch(searchTerm)	opens the default browser and/or opens a new tab and uses ANTF's WingIT search API's to redirect to the website that best closely matches your search terms. If a site cannot be determined, it opens a Google search result. For m ore on ANTF WingIT, visit https://antfwingit.jimdo.com/antf-wingit/	None
ANTFPreInject(path,injectionArgs)	force injects startup arguments into comaptible ANTF applications utilizing push files/commands	None
ANTFInject(path,injectionArgs,appPath)	force injects startup arguments into comaptible ANTF applications utilizing push files/commands, and then launches said application	None

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Kill Explorer

```
ANTFConnection.killWinExplorer()
```

Start Windows Explorer

```
ANTFConnection.startWinExplorer()
```

Kill Proccess

```
ANTFConnection.killProcess("notepad")
```

Launch App

```
ANTFConnection.launchApp("C:\Windows\System32\notepad.exe")
```

Launch App Focused

```
ANTFConnection.launchFocusedApplication("C:\Windows\System32\notepad.exe")
```

Launch Site

```
ANTFConnection.launchSite("https://www.antfcorp.com")
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Launch Control Panel

```
ANTFConnection.shellControlPanel()
```

Launch Control Panel > Printer Settings

```
ANTFConnection.shellPrinterSettings()
```

Launch Control Panel > Network Neighborhood

```
ANTFConnection.shellNetworkNeighborhood()
```

Launch Control Panel > IE/Win Internet Settings

```
ANTFConnection.shellIESettings()
```

Launch CMD

```
ANTFConnection.shellCMD()
```

Run CMD Command

```
Op.Text = ANTFConnection.runCommand("ipconfig")
```

Reboot PC

```
ANTFConnection.rebootPC()
```

Shutdown PC

```
ANTFConnection.shutDownPC()
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Get Time

```
Op.Text = ANTFConnection.getTime
```

Get Precision Time

```
Op.Text = ANTFConnection.getPrecisionTime
```

Get Date

```
Op.Text = ANTFConnection.getDate
```

Get Year

```
Op.Text = ANTFConnection.getYear
```

Get Month

```
Op.Text = ANTFConnection.getMonth
```

Get Day (numeric)

```
Op.Text = ANTFConnection.getDay.ToString
```

Get Day (Written)

```
Op.Text = ANTFConnection.getWeekDay
```

Check Within DST

```
Dim check As Boolean = ANTFConnection.withinDST
        If check = False Then
            Op.Text = "Today's Date is Not within Daylight Savings Time (USA)"
        Else
            Op.Text = "Today's Date is within Daylight Savings Time (USA)"
        End If
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Get OS Name

```
Op.Text = ANTFConnection.getOSName
```

Get OS Version

```
Op.Text = ANTFConnection.getOSVersion
```

Get OS Platform

```
Op.Text = ANTFConnection.getOSPlatform
```

Get Current Username

```
Op.Text = ANTFConnection.getCurrentWindowsUser
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

Get MyPerion Version

```
Op.Text = ANTFConnection.getANTFDev
```

Get Developer Information

```
Op.Text = ANTFConnection.getANTFMyPerionVersion
```

Check For Nebula

```
ODim check As Boolean = ANTFConnection.checkForNebula("C")
If check = False Then
    Op.Text = "ANTF Nebula is not installed on this system"
Else
    Op.Text = "ANTF Nebula is installed on this system"
End If
```

ANTF WingIT Search Querry

```
ANTFConnection.ANTFWingItSearch("Chrysler")
```

Note: *Assume Op.Text is a TextBox object in Windows Forms but a string may be used instead*

ANTF PreInject

```
 Dim music As String = ("C:\Users\" + ANTFConnection.getCurrentWindowsUser + "\Music\MUTEMATH - Changes.mp3")
        Dim check As Integer = antfIO.overWriteTextBasedFile("C:\ANTF Group\ANTF FlaVor\", "ANTF FlavorPush.aff", music)
        If check = 1 Or check = 2 Then
            Op.Text = ("The Injection Failed! Error Code: " + check.ToString)
        Else
            Op.Text = "The Injection was Successful. Please open ANTF Flavor and make sure your push settings are on."
        End If
```

ANTF Inject

```
Dim music As String = ("C:\Users\" + ANTFConnection.getCurrentWindowsUser + "\Music\Hit My Heart.mp3")
Dim check As Integer = antfIO.overWriteTextBasedFile("C:\ANTF Group\ANTF FlaVor\", "ANTF FlavorPush.aff", music)
If check = 1 Or check = 2 Then
   Op.Text = ("The Injection Failed! Error Code: " + check.ToString)
Else
  Op.Text = "The Injection was Successful. Please open ANTF Flavor and make sure your push settings are on. Shelling App”
  ANTFConnection.launchApp("C:\ANTF Group\ANTF FlaVor\Bin\Go\ANTF Flavor Music Player.exe")
End If
```

Additional Documentation

Quick links to additional website references, links, and support tutorials for better understanding Dynamic Link Libraries (DLL), and how to integrate them into other languages

Useful Links

Use .Net DLL's in Java (Persoanl Use)

Use .Net DLL's in C++

Use .Net DLL's in Python

What are Dynamic Link Libraries (DLLS)

Note: *While these links may not fully provide the information to fully implement MyPerion in other non .Net languages, these are good starting places.*

ANTF

TOMORROW'S SOLUTIONS TODAY