

**Microsoft VS .NET 2005**

***C# .NET NOTEBOOK*** <sup>®</sup>

**Author: SREEKANTH.P**

Visit Us@:[www.dotnetsparkles.wordpress.com](http://www.dotnetsparkles.wordpress.com)

E-mail: [sreekanth@ dotnetsparkles.wordpress.com](mailto:sreekanth@dotnetsparkles.wordpress.com)

## Profile

### Professor's Profile:

**Proff.B.NAGARAJU** is a Sr DotNet Consultant for MNC's and Trainer in Naresh Information Technologies, Hyderabad, A.P. India. Started with Dbase III+, FoxPro, COBOL, C, ORACLE and today well-versed with top Microsoft Technologies ASP.NET, C#.NET also VB.NET, SQL SERVER. Latest Developments AJAX, Silver Light, LIVE etc. Interested to implement always new and wanted technologies. Professor has own Development Team for some projects. Love to teach more technical skills to the professionals.

Views Of Proff.B.Nagaraju

## DOT NET HISTORY

<u>Version</u>	<u>Year of Release</u>	<u>Named As</u>
1...Net (beta)	mid of 2000	Visual Studio .Net 2003
2...Net 1.0	2002	Visual Studio .Net 2003
3...Net 1.1	2003	Visual Studio .Net 2003
4...Net 2.0	Nov2005	Visual Studio .Net 2005
	In India Dec2005/Jan 2006	(supports only 2.0, 3.0+extensions)
5...Net 3.0	2007	Visual Studio .Net 2005 +
6...Net 3.5 (beta)	2007	Visual Studio .Net Orcus (Supports 2.0, 3.0, 3.5)

### Note:

-“.Net 3.0=2.0+WPF+WF+WCF+InfoCard”

-WPF: Windows Presentation Foundation (For Next generation User Experiences).

-WCF: Windows Communication Foundation (For service oriented development).

-WF: Windows Workflow Foundation (For Business process modeling).

-INFOCARD (WCS): Windows Card Space (For Digital Identity Management).

### INSTALLATION GUIDE:

1. Install IIS (Internet Information Service).
2. Keep the VS.NET 2005 CD1 press setup then continue
3. Then keep the VS.NET2005 CD2 when the installation displays a dialogue“pls insert the CD2”.

Note: The installation will install an integrated “SQL Server 2005” along the VS 2005 Installation.

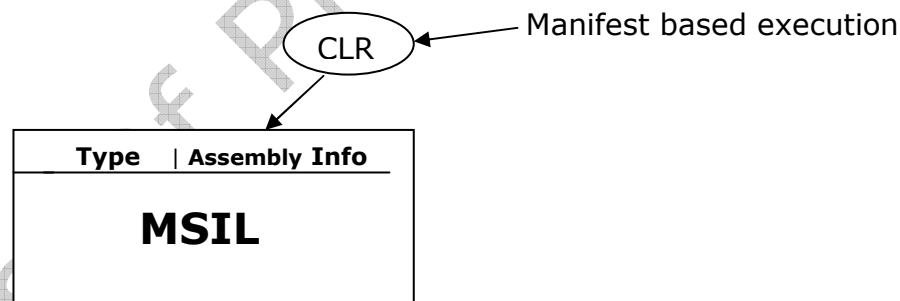
**Meta Data in Assemblies:**Type Description:

- (i) Classes
- (ii) Base Classes
- (iii) Implement Interfaces
- (vi) Data Members
- (v) Methods

Assembly Description

Name	}	Manifest
Version		
Culture		
Other Assemblies		
Security Permissions		
Exported Types		

Manifest: Assembly Manifest is a small portion that containing the above.

**\*\*Manifest Based Execution\*\***

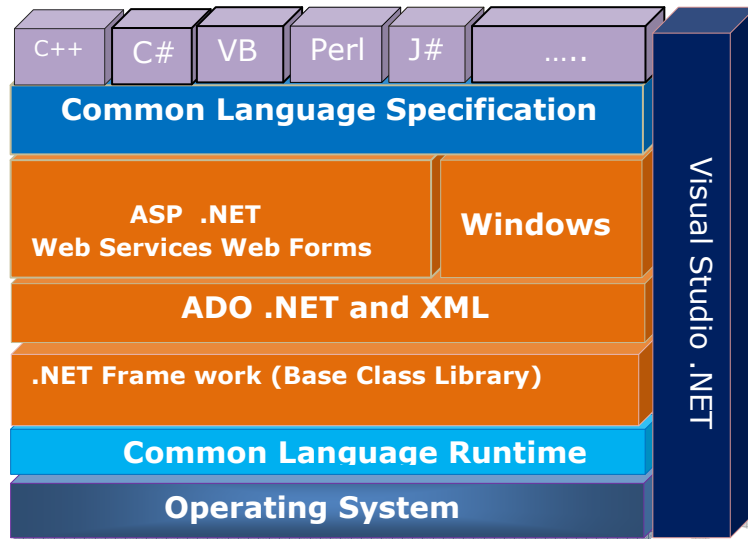
When we compile the program the CLR is going to check at first in assembly info (like index in a book) then it will go to MSIL and executes, this process is called as manifest based execution. Whereas in .Net 1.0, 1.1 at first the compiler checks for the type of assembly and then goes to assembly info and then to MSIL to executes.

**MSIL:** Microsoft Intermediate Language.

**CAS:** Code Access Security.

Caspol.exe: It's a .Net utility allows us to Define/Remove Permissions

### **.NET Framework standard Compliance**



#### Note:

- CLR manages code execution at runtime
- Memory management, Thread management etc..

### **.NET Framework (Base class libraries)**

- Object-oriented collections of re-useable types
- Collections, I/O, Strings ...
- Standard and Common classes
- Simple to use.
- Remoting and many more.

### **ADO .NET and XML**

- Access relational database
- Disconnected data model
- Work with XML Types of Assemblies

### **ASSEMBLIES**

There are two types of Assemblies

**(i) Private Assemblies**

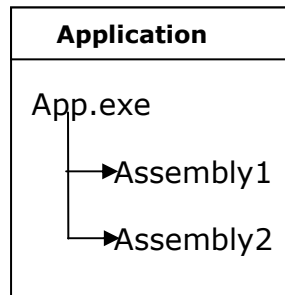
**(ii) Public Assemblies**

(i) Private Assemblies (or) Satellite Assemblies:

-Only visible for the applications (i.e. for .exe files only)

- Stored in application folder or in sub-directories
- Isolated from changes made to the system (changes made to OS or system will not affect the process)
- No naming required (except the applications names are unique)

Ex: if a file named as convert.exe is built by using the math.dll library then if we want to run the convert.exe file we should load the math.dll file in the same folder or directory in this private assembly.



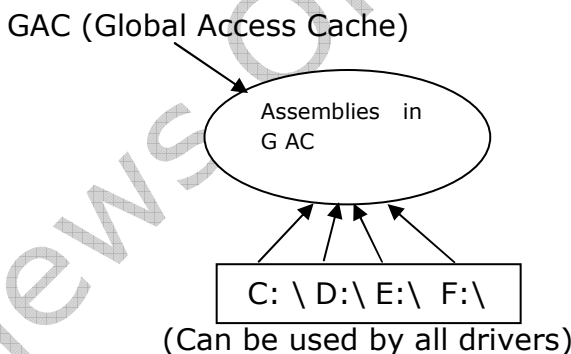
#### Properties:

- (1) Private Assemblies will run the files currently in that folder only.
- (2) If we copied only the .exe file from one system to another or to another drive (Ex: C: to D) the file will not be compiled until we copy the .dll file (its Library file) to that location.

#### (ii)Public\Shared\Global Assemblies:

- .Net doesn't have any third type assembly.
- When ever we install the .Net frame work it occupies a memory called as GAC (Global Assembly Cache).
- Cache memory is a programmed memory (i.e.it is allocated separately to the each Program likes to OS -200 MB, for programs -100MB,for wm player 20 MB etc.)

Ex:



- If we want to run a program copied from another location then we should compulsorily copy the .dll file (The library file) to GAC then only we can run the program.
- Then we have to give strong name to the assembly to overcome the problem of similarity using a .NET tool "SN.exe".
- Then we have to load assembly into GAC using a tool "GACUTIL.EXE".

#### Steps for loading Assembly in GAC

- (i) Create a strong name using a .NET tool "SN.exe".

- (ii) Compile the assembly by providing this key.
- (iii) Place the compiled assembly in .NET folder called as assembly, .NET folder here means the location where we installed .NET frame work.
- (iv) Use "GACUTIL.EXE "to load the assembly into GAC.

Note: Private Assemblies are faster than the Public Assemblies.

## **Interoperability**

Date: 16\7\07

- All .net applications runtime is CLR.
- CLR code is divided
  - (i) Managed
  - (ii) Unmanaged

### **Managed Code:**

- All code i.e. executed by CLR is managed code.
  - .NET provides multiple language support that is possible with "CLS".
  - A language that is following MSIL, CLS, Net BCL, and LTS is called managed code.
  - All Code: That is compatible to CLR is called managed code.
  - CLR has a support for Un-Managed Code it should be executed by separate executing system.
- Any Code: a code is not MSIL, CLS, Net BCL, CTS is not compatible.

### UN-MANAGED CODE (VB6.0)

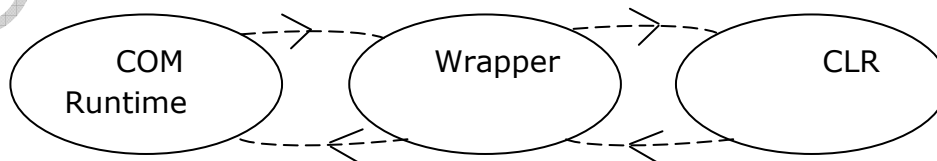
- (i) Excel
- (ii) COM Sample.Dll
- (iii) VB6.0 is not compatible to CLR
- (iv) COM run time

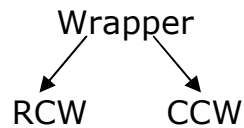
### MANAGED CODE(C#.NET)

- (i) .NET App
- (ii) dot net sample.Exe
- (iii) We want to use this COM sample.Dll in our managed code.
- (iv) CLR

### **WRAPPER:**

- wrapper is managed and wrapper is a program and it acts as a broker between COM Runtime and CLR.
- CLR talks to wrapper are responsible to take the request from CLR and send it to COM Runtime and COM Runtime sends the response to wrapper.
- it receives three files to runtime.





RCW: Runtime Callable Wrapper

CCW: COM Callable Wrapper

RCW: If we want to use COM component in .NET we go for RCW, it is mostly used.

CCW: If we want to use .NET component in some level we use CCW.

Note-We cannot run excel in Linux.

INTEROPERABILITY: - In .NET refers to consuming a COM component in .NET and .NET component in COM, .NET Provides this concept using wrappers.

- Wrappers are a piece of code that takes request and converts to any other runtime request.
- TLBIMP.Exe: -It is a .NET utility creates a wrapper for COM Libraries.

Why we use C# as .NET Compatible Language?

Date: 17/7/07

- ❖ Modern and elegant Language.
- ❖ Lot less typing than VB.Net
- ❖ Overload operators.
- ❖ Direct access to Memory.
- ❖ In line XML documentation.
- ❖ C# provides an easy way to unmanaged code (or) unsafe code (Ex: Pointers).

Why not we use C# as .NET Compatible Language?

- ❖ Slower than VB (when managed).
- ❖ Some what more complex than VB(Ex: Case Sensitivity)
- ❖ Not as familiar on the other hand VB.NET is like a new language tool.

WHY FORMS (or) WHY WE NEED FORMS?

Date: 23/7/07

-C#.NET is one of the frontend available today any front end objectives are to provide

1. To provide good designing approach.
2. To provide interaction with database for the designs.
3. Distributed environment.
  - (i) Local distribution.
  - (ii) Support for internet.

**Note:** 1. dot Net 3.0=.Net 2.0+WPF.  
2. UI+OOP (component) +DB+Ntire remoting=Frame Work.

-To achieve the first objective C#.NET has System.Windows.Forms Namespace where we have plenty of GUI Classes.

-Form is a class that allows users to design front end screen.

-Form is also like a control that acts as a container.

-A Container has a special meaning for runtime, which enables user to put other controls in it and treat them as a group.

-Only Form is not a container but we also have other controls that act as a container.

**\*COLLECTIONS:**-In .NET we have another Concept called "Collections". A collection is a group of elements (Like an array).Every collection in .Net is derived from classes and these classes when referred like an array then we call it as collection

- C# .NET even has arrays. Because every collection is derived from a class It again has members those are really very useful to make programming simple.

- Container -Collection (Container-GUI that appears in front of us).

- Container-Collection: Form containing many collections.

- Controls are a collection of Form Container.

- Syntax: Form 2 objfrm=new Form2 ( );

ObjFrm.Controls [0] =means first control in form2.

-"Controls "is a predefined collection for all controls that are called containers.

Syntax: ObjFrm.Controls [0].Controls [0]

**\*General procedure in form creation:** we select a class called "TextBox" and we put it in the form to create an object

**Toolbox:** it is a graphical representation of classes and allows user to just drag and drop to create objects.VS.NET will write code for all these actions on its own.

-Every form that we create will have two files

(i). Code file written by user.

(ii). Design file where code is auto generated by VS.NET.

-Most intrusting part is in both these files we are going to work with "single class". This is possible because C# 2.0 has introduced a concept called "PARTIAL CLASSES"

**PARTIAL CLASSES:** we can "write" or "split classes".

-C# 2.0 supports partial classes.

**\*CONTROLS**

Date: 24/7/07

-Every control is a class and class will have members.

-there are four types of members.

(i) Variables/Fields.

(ii) Methods-(Functions in C#.NET and in VB .NET procedures and Functions)

(iii) Properties.

(iv) Events.

-Important modd (Object oriented model with add on) we have

(i) Properties.

(ii) Events.

-a Property is members of classes that has read and write capabilities.

-Like function it has set of statements that executes when we set a value.

**Ex:** Button2.BackColor = Button1.BackColor;

-Properties can be set at runtime (code) or design time (properties window).there are few properties that can be set only at runtime and some times at design time

-Runtime→<Object>.<Property>=<Value>  
C=<object>.<property>

This: -is a relative reference.

This.BackColor=change color=red;

-“Mdi Parent” is a runtime property.

EVENTS: Mouse events-then press events. Then the codes will com

```
{
    Message Box. Show (“Touch me not”);
}
```

How to write Events?

-We can write events using VS.NET properties window. Just select the object (or) control and select events from properties window, provide name or simply click on the event name this results in a function (code window) where we can write the code.

-we can also write events in code where we specify the function name the event etc...

To create a control at run time:

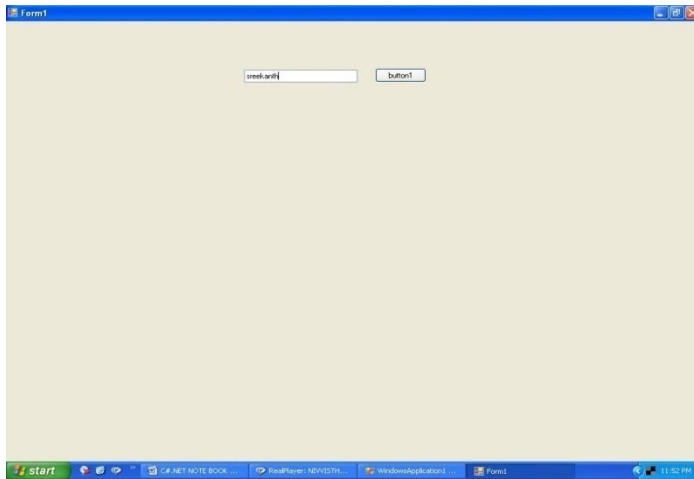
- (i) Declare an object and instantiate it.
- (ii) Set the required properties like, height, width, Text, Position etc...
- (iii) Add it to controls collection of form.

Ex:

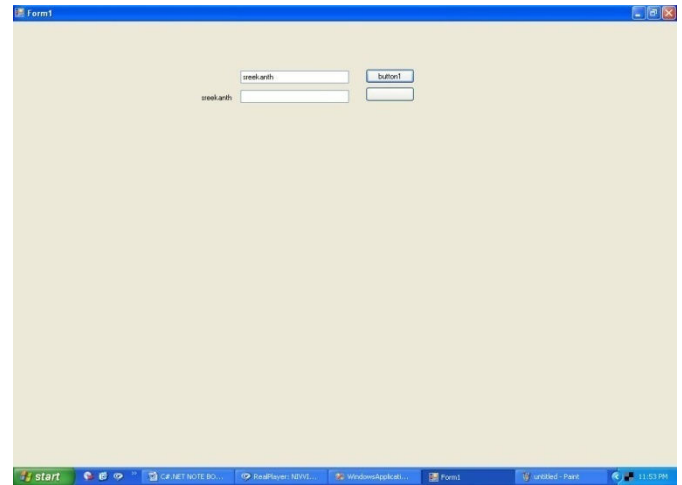
```
{
    Button dynbutton;
    dynbutton=new Button ( );
    dynbutton. Size=new size (100,100);
    dynbutto.Location=new point (50, 50);
    dynbutton. Text=“Dynamic”;
    dynbutton.BackColor=color.LightBlue;
    this.Controls.Add (dynbutton); →(Note: This control we are using here to ADD
    dynbutton. Show ( ) ;} the dynbutton to Controls)
```

Try This Ex:

-create text box and a button if we type any thing in the text box and the button is clicked the text in textbox 1 should come as label text for the dynamic textbox and a dynamic button should come. And the same should be repeated for another textbox.



STEP-1



STEP-2

### **COM INTEROPERABILITY:**

Date:3/8/07

VB 6.0, VB .NET and C3 language cannot be compatible to each other to use the libraries?

Ans: .NET supports or CLR supports COM interoperability where we can consume a COM component in .net and a .NET component is COM.

-COM interoperability is provided with the help of "WRAPPERS".

**WRAPPER:** is a code piece of code that converts a .NET request to COM and a COM request to .NET

### **Console Applications :**

13/8/07

-windows applications are Graphical User Interface based where as console applications are character based applications.

-Windows apps are event driven.

-Console Apps are flow driven.

-Windows Apps are event based executions.

-Console Apps are command line executions.

-windows Apps are more user friendly Console Apps are non user-friendly.

### **Windows Application Template:**

-In the previous examples using form class in console Apps we used System.Windows.forms Space to create a form/window.

-VS.NET provided a template called windows applications that automatically does all the work carried out by us. So that "quick start "windows App we use this template.

-When ever a new windows application is started VS.NET by default creates App template and its coding.

(i) Program.cs with static void main and also Application Object in it.

(ii) A form class called form1 that inherits Forms class of .NET BCL

VS.net makes window App development very simplified with its windows like.

(i) Solution Explorer.

(ii) Properties Window.

(i)**Solution Explorer:** hierarchal view of the project showing the current libraries referred, User

And also allowing some actions like all new item, class, form, image, Compiling Program, View  
(ii) **Properties Window:** VS .NET object oriented its also called component oriented language.

**Benefits of Component model:** Object oriented + more Customization.

- Simplified Development.
- Simplified development.
- Team Support.
- Rapid App Development.

Views Of Proff. B. Nagaraju