



**.DLL** (*DYNAMIC LINKED LIBRARY*)  
**IN .NET PLATFORM**  
**HOW TO CREATE & HOW TO CONSUME**  
By: Kayun Chantarasathaporn

# .DLL FILE

- DLL <= Dynamic Linked Library
- A dynamic link library (DLL) is a collection of small programs, any of which can be called when needed by a larger program that is running in the computer. (<http://www.whatis.com>)
- Characteristics
  - Do not have Main() method
  - Can not execute by itself
  - Need to be called (by other class's members)
- Usage
  - Utility tools (do not need to provide source code)
  - Device drivers
- Be careful about .DLL files since
  - They could be generated from many languages
  - They could be generated from many platforms
  - .DLL in .NET is usually not the same as .DLL from former technologies (COM, DCOM)
  - Since their originates may be from different source, they might be unable to work together (Have to check from their documents)

# .DLL IN .NET

- .DLL in .NET platform is just a class that does not have Main() method
- It can contain field (class's data member) and method (class's function member) like regular class
- If class's member is *static*, it would be called by using *class name*
- If class's member is *non-static*, it would be called by using *class instance name* (object name)

# HOW TO CREATE .NET DLL FILE

- Start Visual Studio 2005
- Create *New Windows Project* by using *Class Library* template
- From class file in project, create method
  - This method should have access modifier as *public*
  - This method can be either *static* or *non-static* member
  - This method can be either
    - non-return value method
    - return value method
    - return more than one value method (using *out* or *ref* technique)
- Compile
- There will be .dll output file in either *bin/debug* or *bin/release* folder

# HOW TO CONSUME .NET DLL FILE FROM THE SAME SOLUTION

- Add *New Project* to existing solution
- From project
  - Add reference the .dll project
- From class file in project
  - using [*namespace from .dll project*]
  - To use static method from .dll, use its class name
  - To use non-static method from .dll, create object of its class first, then use object name to call method
- Set current project to be *StartUp Project*
- Compile
- Is there any .dll file in bin/debug or bin/release of current application?

# HOW TO CONSUME .NET DLL FILE FROM DIFFERENT SOLUTION

- Start Visual Studio 2005
- Create *New Windows Project* by using *Windows Application* template
- From project
  - Add reference the .dll file
- From class file in project
  - using [*namespace from .dll file*]
  - To use static method from .dll, use its class name
  - To use non-static method from .dll, create object of its class first, then use object name to call method
- Compile
- Is there any .dll file in bin/debug or bin/release of current application?