

Implicit & Explicit linking of DLL

Implicit Linking Application

Library of Implicit Linking

```
using System;

namespace Marvellous
{
    // Class declaration
    public class Hello
    {
        // Entry point function
        public void fun()
        {
            // Display string on console
            Console.WriteLine("Marvellous Infosystems : Inside DLL");
        }
    }
}
```

Client of Implicit Linking

```
using System;

using Marvellous;

class Demo
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Inside Client application");

        // Create object of class from DLL
        Hello hobj = new Hello();

        // Call the exported method from DLL
        hobj.fun();
    }
}
```

Explicit Linking Application

Library of Explicit Linking

```
using System;

public class Hello
{
    public void fun()
    {
        // Display string on console
        Console.WriteLine("Marvellous Infosystems : Inside DLL");
    }
}
```

Client of Explicit Linking

```
using System;
using System.Reflection;
using System.Collections.Generic;

class Demo
{
    public static void Main(string[] args)
    {
        Console.WriteLine("Inside Client application");

        try
        {
            Assembly DLL = Assembly.LoadFile(@"Path of DLL");

            Console.WriteLine("DLL loaded in memory...");

            Type[] types = DLL.GetTypes();

            Console.WriteLine("Data types from DLL gets fetched...");

            foreach (Type typ in types)
            {
                Console.WriteLine("Data type from DLL is"+typ);

                object obj = Activator.CreateInstance(typ);

                Console.WriteLine("Object gets created of class"+obj);

                MethodInfo mi = typ.GetMethod("fun");

                Console.WriteLine("Address of method fun gets fetched..");
            }
        }
    }
}
```

```
        mi.Invoke(obj, null);  
  
        Console.WriteLine("Method call complete");  
    }  
}  
catch (Exception e)  
{  
    Console.WriteLine(e);  
}  
}  
}
```

