

Essential Links

Unity NuGet Package	https://www.nuget.org/packages/Unity/
Unity Registration by Convention NuGet Package	https://www.nuget.org/packages/Unity.RegistrationByConvention/
Overview	https://www.tutorialsteacher.com/ioc/unity-container
GitHub Repo	https://github.com/unitycontainer/unity

Example Code

```
public interface ICar
{
    int Run();
}

public class Car : ICar
{
    private int _miles;
    public int Run()
    {
        return ++_miles;
    }
}

public class Driver
{
    private ICar _car;
    public Driver(ICar car)
    {
        _car = car;
    }
    public void RunCar()
    {
        Console.WriteLine("Running {0} - {1} mile ",
            _car.GetType().Name, _car.Run());
    }
}
```

Register and Resolve

```
IUnityContainer container = new UnityContainer();
container.RegisterType<ICar, Car>();
//Resolves dependencies and returns the Driver
Driver driver= container.Resolve<Driver>();
driver.RunCar();
```

Constructor Injection

```
[InjectionConstructor]
public Driver(ICar car)
{
    _car = car;
}
```

Property Injection

```
[Dependency]
public ICar Car { get; set; }
```

Register Types

```
//NuGet Package Unity.RegistrationByConvention
//The interface and class names must match
//ICar and Car
public static void RegisterTypes(IUnityContainer
    container)
{
    container.RegisterType(
        AllClasses.FromLoadedAssemblies(),
        WithMappings.FromMatchingInterface,
        WithName.Default);
}
```

Register Types In Assembly

```
RegisterTypesInAssembly(container,
    typeof(Car).Assembly);
public static void RegisterTypesInAssembly
    (IUnityContainer container, Assembly assembly)
{
    IEnumerable<Type> types = assembly.GetTypes()
        .Where(o => o.IsClass
            && !o.IsGenericType
            && !o.IsAbstract
            && o.GetInterfaces().Any());
    foreach (Type type in types)
    {
        container.RegisterType(type.GetInterfaces()
            .First(), type);
    }
}
```

