* UnityContainer

[CLSCompliant(true)]

[DebuggerDisplay("{DebugName()}")]

public class UnityContainer : IUnityContainer, IDisposable

{

public UnityContainer();

public IEnumerable<IContainerRegistration> Registrations { get; }

public IUnityContainer Parent { get; }

public IUnityContainer AddExtension(UnityContainerExtension extension);

public object BuildUp(Type typeToBuild, object existing, string nameToBuild, params ResolverOverride[] resolverOverrides);

public object Configure(Type configurationInterface);

public IUnityContainer CreateChildContainer();

public void Dispose();

public bool IsRegistered(Type type, string name);

public IUnityContainer RegisterInstance(Type registeredType, string name, object instance, LifetimeManager lifetimeManager);

public IUnityContainer RegisterType(Type typeFrom, Type typeTo, string name, LifetimeManager, InjectionMember[] injectionMembers);

public object Resolve(Type typeToBuild, string nameToBuild, params ResolverOverride[] resolverOverrides);

protected virtual void Dispose(bool disposing);

}