**Reflection**

Metadatların runtime olaraq əl çatanlığı C#’da Reflection olaraq bilinir. Metadata, sadə dildə desək, məlumat haqqında məlumat(data about data) və ya resurs haqqında komputerləşdirilmiş məlumatdır. Datanın formatı haqqında məlumat, təşkili və ölçüsünə aid məlumatlar bu məlumatlara misaldır.

System.Reflection namespace’də yer alan Assembly partial class vasitəsi ilə bütün bu əməliyyatlar həyata keçəcək.

**\*Assembly.GetExecutingAssembly();**

Hal-hazırda daxilində yer aldığımız project‘in Assembly‘ni GetExecutingAssembly() method‘u vasitəsi ilə əldə edə bilərik.Bu method vasitəsi ilə Reference alınan project’in Assembly’nə bir başa olaraq müraciət etməyə icazə vermir.

Assembly asmb = Assembly.GetExecutingAssembly();

**\*Assembly.LoadFile(path);**

GetExecutingAssembly metod’u ilə yanaşı LoadFile(path) method’u ilə biz istədiyimiz project’in Assembly’i əldə edə bilərik.

Assembly asmb = Assembly.LoadFile(path));

Əldə etdiyimiz Assembly sayəsində orada yer alan class‘ları, o class‘lar daxilində yer alan nested class‘ları, method‘ları, field və property‘ləri, həmçinin constructor‘ları əldə edə bilərik.

**\*GetType() & GetTypes()**

Əldə olunan Assembly daxilində yer alan hər hansısa bir type asmb.GetType(typeName); vasitəsi ilə əldə olunur və geriyə Type object qaytarır.

Əldə olunan Assembly daxilində yer alan bütün typelar asmb.GetTypes(); vasitəsi ilə əldə olunur və geriyə Type[] array qaytarır.

**\*GetConstructor & GetConstructors**

Biz Assembly object’i sayəsində Constructor(lar) haqqında məlumat ala bilərik. GetConstructor və GetConstructors() method’ları bizə yardımçı olur və geriyə ConstructorInfo və ya ConstructorInfo[] qaytarır.

Class daxilində heç bir constructor yazılmayan zaman Assembly səviyyəsində boş constructor yaranır demişdik, bu method bizə bunun sübutunu göstərir.

**\*GetField & GetFields**

Public təyin olan Field’ları əldə etmək üçün BindingFlags.Public ,

Public təyin olunmayan Field’ları əldə etmək üçün BindingFlags.NonPublic ,

Non-static təyin olan Field’ları əldə etmək üçün BindingFlags.Instance ,

Static təyin olunan Field’ları əldə etmək üçün BindingFlags.NonInstance

type.GetField(fieldName, BindingFields.Public | BindingFields.NonPublic | BindingFields.Instance);

Buradan anlaşıldığı qədəri ilə biz artıq private və ya readonly olaraq təyin etdiyimiz variable’lara da müraciət edib data set və ya get əməliyyatlarını həyata keçirə bilərik.

**\*GetProperty & GetProperties**

Public təyin olan Property’ləri əldə etmək üçün BindingFlags.Public ,

Public təyin olunmayan Property’ləri əldə etmək üçün BindingFlags.NonPublic ,

Non-static təyin olan Property’ləri əldə etmək üçün BindingFlags.Instance ,

Static təyin olunan Property’ləri əldə etmək üçün BindingFlags.NonInstance

type.GetProperty(propertyName, BindingFields.Public | BindingFields.NonPublic | BindingFields.Instance);

**\*GetValue**

Əldə etdiyimiz property və ya field’ların value’larını əldə etmək üçün PropertyInfo və ya FieldInfo obyektinə GetValue() method’u tətbiq edilməlidir. Arqument olaraq isə hansı instance’ın property’si və ya field’ın dəyərini əldə etmək istəyiriksə o obyekti göndəririk.

**\*SetValue**

Əldə etdiyimiz property və ya field’lara dəyər set etmək üçün PropertyInfo və ya FieldInfo obyektinə SetValue(object, value) method’u tətbiq edilməlidir. Hansı obyektin property və ya field’a value set edəcəyiksə o obyekti və set edəcəyimiz dəyəri göndərəcəyik. Əgər property və ya field static olarsa o zaman object əvəzinə null dəyər göndərilməlidir.

**\*Activator.CreateInstance(type)**

Type tipindən olan object instance almaq üçün Activator.CreateInstance(type) object type’a box olunmuş obyekt qaytarır. Həmçinin generic yazılışa da malik olan method bu şəkildə də yazılır. Activator.CreateInstance<Type>();

(Type) type.Name, (Type) type.FullName

Type object‘ə .Name yazaraq həmin type‘ın adını, .FullName yazaraq isə bütöv adını(namespace ilə birlikdə) əldə edirik.