

# C#: Reflection

Nicolas Blanc



# What is reflection?

- Ability to discover type information at run time.
- Content of the presentation:
  - How to dynamically retrieve type information from assemblies

# Type

- Name
- Module
- IsAbstract
- IsArray
- IsClass
- IsPointer
- IsPublic
- IsValueType
- IsSerializable
- IsInterface
- IsEnum
- IsSealed
- ...

# Querying for type:

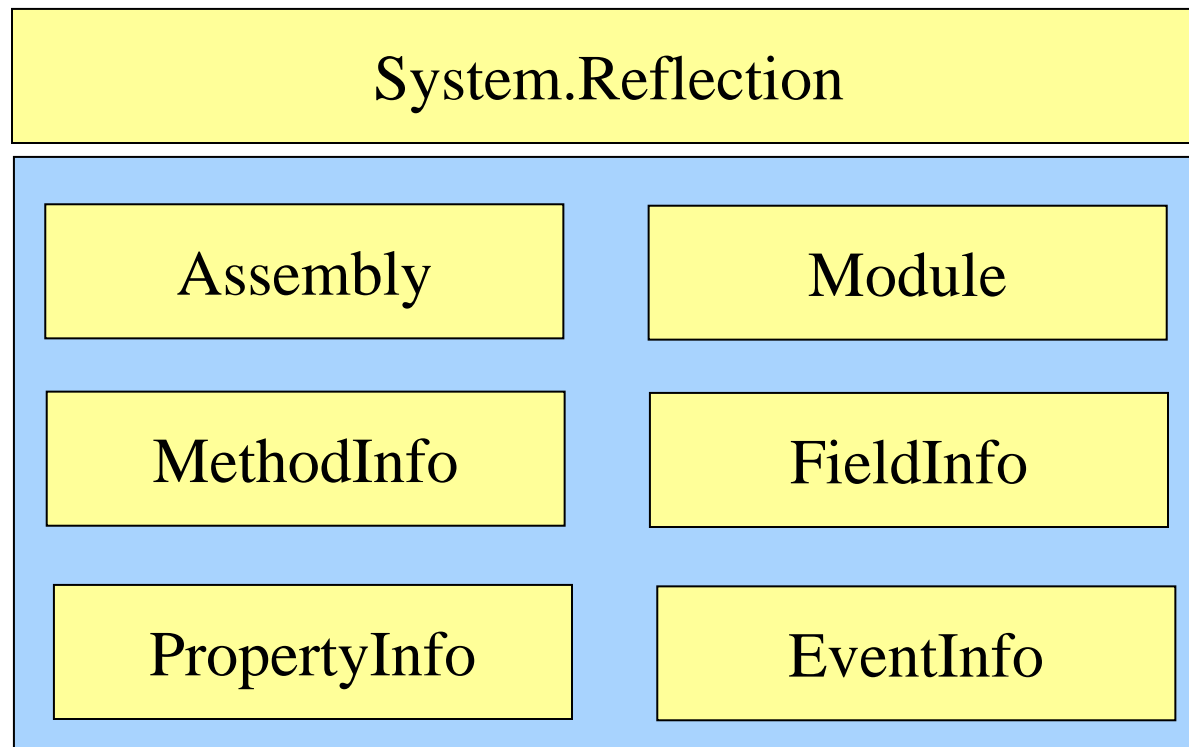
```
int i = 10;  
Type t = i.GetType();  
Console.WriteLine(t.Name);  
// Output: Int32
```

```
Type t = typeof(String);  
Console.WriteLine(t);  
// Output: System.String
```

```
class A {...}
```

```
Type t = Type.GetType("ConsoleApplication1.A", false);  
Console.WriteLine(t);  
// Output: ConsoleApplication1.A
```

# .NET Reflection API



# Assembly

- Consists of .NET *portable executable* (PE)
- Created by the C# compiler
- Two kinds of PE:
  - EXE (ConsoleApplication)
  - DLL (ClassLibrary)

# Assembly

- Use Assembly to:
  - Load assemblies
  - Explore the metadata
  - Get the currently loaded assemblies

# Querying for assemblies:

```
Assembly assem = Assembly.GetExecutingAssembly();  
Console.WriteLine("Assembly Full Name:");  
Console.WriteLine(assem.FullName);
```

# Examples

- Late Binding
- Plugin
- Dynamic Code Generation