System.Reflection.Assembly Class

```
[ILAsm]
.class public serializable Assembly extends System.Object

[C#]
public class Assembly
```

Assembly Info:

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 2.0.x.x
- *Attributes:*
    - CLSCompliantAttribute(true)

Summary

Defines a `System.Reflection.Assembly`, which is a reusable, versionable, and self-describing building block of an application.

Inherits From: System.Object

Library: RuntimeInfrastructure

Thread Safety: This type is safe for multithreaded operations.

Description

An assembly is a reusable, versionable, self-describing deployment unit for types and resources. Assemblies are the fundamental units of deployment, and consist of collections of types and resources that are built to work together and form logical units of functionality.

An assembly consists of the following two logical elements:

- The sets of types and resources that form some logical unit of functionality.

- A manifest, which is the metadata that describes how the types and resources of an assembly relate and what they depend on to work properly.

The following information is captured in an assembly manifest:

- `Identity`. An assembly's identity includes its simple name (also called its weak name), a version number, an optional culture if the assembly contains localized resources, and an optional public key used to guarantee name uniqueness and to "protect" the name from unwanted reuse.

- `Contents`. Assemblies contain types and resources. The manifest lists the names of all the types and resources that are visible outside the assembly, along with information about where they can be found within the assembly.

- `Dependencies`. Each assembly explicitly describes other assemblies that it is dependent upon. Included in this dependency information is the version of each dependency that was present when the manifest was built and tested. In this way the "known good" configuration is recorded and can be reverted to in case of failures due to version mismatches.

- `Requested Permissions`. As an assembly is being built, the assembly records the set of permissions that the assembly requires to run.

[*Note:* For additional information about assemblies, see Partition II of the CLI Specification.]

Assembly.CreateInstance(System.String) Method

```
[ILAsm]
.method public hidebysig instance object CreateInstance(string typeName)

[C#]
public object CreateInstance(string typeName)
```

Summary

Locates the specified type from this assembly and creates an instance of it using case-sensitive search.

Parameters

Parameter	Description
typeName	The name of the type to locate.

Return Value

An instance of `Object` representing the type, or `null` if *typeName* is not found.

Exceptions

Exception	Condition
System.ArgumentException	*typeName* is the empty string ("") or "\0anything".
System.ArgumentNullException	*typeName* is `null`.

Assembly.GetType(System.String) Method

```
[ILAsm]
.method public hidebysig virtual class System.Type GetType(string name)

[C#]
public virtual Type GetType(string name)
```

Summary

Returns the `System.Type` object with the specified name defined in the current assembly.

Parameters

Parameter	Description
name	A `System.String` containing the name of the type defined in the current assembly.

Return Value

A `System.Type` object that represents the specified type, or `null` if the specified `System.Type` was not found.

Behaviors

As described above.

Exceptions

Exception	Condition
System.ArgumentException	*name* is equal to `System.String.Empty` or starts with the null character ('\0').
System.ArgumentNullException	*name* is `null`.

Assembly.GetTypes() Method

[ILAsm]
```
.method public hidebysig virtual class System.Type[] GetTypes()
```

[C#]
```
public virtual Type[] GetTypes()
```

Summary

Returns the types defined in the current assembly.

Return Value

An array of type `System.Type` containing all of the types defined in the current assembly.

Assembly.Load(System.String) Method

[ILAsm]
```
.method public hidebysig static class System.Reflection.Assembly
Load(string assemblyString)
```

[C#]
```
public static Assembly Load(string assemblyString)
```

Summary

Loads the specified assembly.

Parameters

Parameter	Description
assemblyString	A `System.String` containing the name of the assembly.

Return Value

The loaded `System.Reflection.Assembly`.

Exceptions

Exception	Condition
System.ArgumentNullException	*assemblyString* is `null`.
System.ArgumentException	*assemblyString* is equal to `System.String.Empty` or starts with the null character ('\0').
System.IO.FileNotFoundException	The `System.Reflection.Assembly` identified by *assemblyString* was not found.
System.BadImageFormatException	The `System.Reflection.Assembly` identified by *assemblyString* is not a valid assembly.

Assembly.ToString() Method

```
[ILAsm]
.method public hidebysig virtual string ToString()

[C#]
public override string ToString()
```

Summary

Returns a `System.String` representation of the value of the current instance.

Return Value

A `System.String` representation of the current instance. The string takes into account the current system culture.

Description

This method returns the `System.Reflection.Assembly.FullName` of the current assembly.

[*Note:* This method overrides `System.Object.ToString`.]

Example

The following example demonstrates the use of the `System.Reflection.Assembly.ToString` method in an assembly compiled into a file named "HelloWorld".

[C#]

```
using System;
using System.Reflection;

public class AssemblyExample {
 public static void Main() {

 Assembly a = Assembly.Load("helloworld");
 Console.WriteLine(a.ToString());
 }
}
```

The output is

```
HelloWorld, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
```

Assembly.FullName Property

```
[ILAsm]
.property string FullName { public hidebysig virtual specialname string
get_FullName() }

[C#]
public virtual string FullName { get; }
```

Summary

Gets the full name of the assembly.

Property Value

A `System.String` containing the full name of the assembly.

Description

This property is read-only.

Behaviors

As described above.

Default

The full name is returned in the following format:

<assemblyTextualName>, Version=<major.minor.build.revision>, Culture=neutral, PublicKeyToken=<publicKeyToken>

[*Note:* The *<assemblyTextualName>* section of the string contains the textual name of the assembly, and is equivalent to the name of the file into which the assembly manifest is compiled. This name does not change even if the file with the assembly manifest is later renamed. For additional information about assembly manifests, see Partition II of the CLI Specification.

For information on the *Version* information in the full name of a `System.Reflection.Assembly`, see `System.Version`.

The *<publicKeyToken>* is a `System.String` containing the value of the public key token in hexadecimal format. A `null` *publicKeyToken* indicates that the current assembly is private. For additional information about public keys and public key tokens, see Partition II of the CLI Specification.

]

Usage

This property is used by the `System.Reflection.Assembly.ToString` method.

Example

The following example demonstrates using the `System.Reflection.Assembly.FullName` property to get the full name of an assembly compiled into a file named "HelloWorld".

[C#]

```
using System;
using System.Reflection;

public class AssemblyExample {
 public static void Main() {

 Assembly a = Assembly.Load("helloworld");
 Console.WriteLine(a.FullName);
 }
}
```

The output is

```
HelloWorld, Version=0.0.0.0, Culture=neutral, PublicKeyToken=null
```