System.AppDomain Class

```
[ILAsm]
.class public sealed AppDomain extends System.MarshalByRefObject

[C#]
public sealed class AppDomain: MarshalByRefObject
```

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

Represents an application domain, which is an isolated environment where applications execute.

Inherits From: System.MarshalByRefObject

Library: RuntimeInfrastructure

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

Application domains, which are represented by `System.AppDomain` objects, provide isolation, unloading, and security boundaries for executing managed code.

Multiple application domains can run in a single process; however, there is not a one-to-one correlation between application domains and threads. Several threads can belong to a single application domain, and while a given thread is not confined to a single application domain, at any given time, a thread executes in a single application domain.

Application domains are created using the `CreateDomain` method. `System.AppDomain` instances are used to load and execute assemblies (`System.Reflection.Assembly`). When a `System.AppDomain` is no longer in use, it can be unloaded.

The `System.AppDomain` class implements a set of events to enable applications to respond to the following conditions:

Condition	Event
An assembly was loaded.	`System.AppDomain.AssemblyLoad`

An application domain will be unloaded.	`System.AppDomain.DomainUnload`
An unhandled exception was thrown.	`System.AppDomain.UnhandledException`

1

2

AppDomain.AssemblyLoad Event

```
[ILAsm]
.event public event AssemblyLoad

[C#]
public event AssemblyLoadEventHandler AssemblyLoad
```

Summary

Raised when an assembly is loaded.

Description

[*Note:* This event is handled by a `System.AssemblyLoadEventHandler` delegate. Information about the event is passed to the delegate in a `System.AssemblyLoadEventArgs` instance.

For additional information about events, see Partitions I and II of the CLI Specification.

]

AppDomain.DomainUnload Event

```
[ILAsm]
.event public event DomainUnload

[C#]
public event EventHandler DomainUnload
```

Summary

Raised when a `System.AppDomain` is about to be unloaded.

Description

[*Note:* This event is handled by a `System.EventHandler` delegate. Information about the event is passed to the delegate in a `System.EventArgs` instance. The delegate for this event can perform any termination activities before the application domain is unloaded.

For additional information about events, see Partitions I and II of the CLI Specification.

]

AppDomain.UnhandledException Event

```
[ILAsm]
.event public event UnhandledException

[C#]
public event UnhandledExceptionEventHandler UnhandledException
```

Summary

Raised when an exception is not caught by the default application domain.

Description

[*Note:* This event is handled by a `System.UnhandledExceptionEventHandler` delegate. Information about the event is passed to the delegate in a `System.UnhandledExceptionEventArgs` instance. The delegate provides default handling for uncaught exceptions. When this event is not handled, the system default handler reports the exception to the user and might terminate the application. For additional information, see `System.UnhandledExceptionEventArgs.IsTerminating`.]

This event is raised only for the application domain that is created by the system when an application is started. If an application creates additional application domains, specifying a delegate for this event in those applications domains has no effect.

[*Note:* For additional information about events, see Partitions I and II of the CLI Specification.]

AppDomain.CreateDomain(System.String) Method

```
[ILAsm]
.method public hidebysig static class System.AppDomain CreateDomain(string
friendlyName)

[C#]
public static AppDomain CreateDomain(string friendlyName)
```

Summary

Creates and returns a new application domain with the specified name.

Parameters

Parameter	Description
friendlyName	A `System.String` containing the friendly name of the domain.

Return Value

A `System.AppDomain` representing the newly created application domain.

Description

[*Note:* The *friendlyName* parameter is intended to identify the domain in a manner that is meaningful to humans. This string should be suitable for display in user interfaces.]

Exceptions

Exception	Condition
System.ArgumentNullException	*friendlyName* is `null`.

AppDomain.ToString() Method

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

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

Summary

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

Return Value

A `System.String` containing information about the current `System.AppDomain` instance.

Description

The string returned by this method includes the friendly name of the application domain.

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

AppDomain.Unload(System.AppDomain) Method

```
[ILAsm]
.method public hidebysig static void Unload(class System.AppDomain domain)

[C#]
public static void Unload(AppDomain domain)
```

Summary

Unloads the specified application domain.

Parameters

Parameter	Description
domain	A `System.AppDomain` representing the application domain to be unloaded.

Description

If the thread that invoked `System.AppDomain.Unload` is running in *domain*, another thread is started to perform the unload operation. If *domain* cannot be unloaded, a `System.CannotUnloadAppDomainException` is thrown in that thread, not the original thread that invoked `System.AppDomain.Unload`. However, if the thread that invoked `System.AppDomain.Unload` is running outside *domain*, that is the thread that receives the exception.

The threads in *domain* are terminated using the `System.Threading.Thread.Abort` method, which throws the thread an instance of `System.Threading.ThreadAbortException`. [*Note:* Although the thread should terminate promptly, it can continue executing for an unbounded amount of time in its `finally` clause.]

Exceptions

Exception	Condition
System.ArgumentNullException	*domain* is `null`.
System.CannotUnloadAppDomainException	*domain* could not be unloaded.

AppDomain.FriendlyName Property

```
[ILAsm]
.property string FriendlyName { public final hidebysig virtual specialname
string get_FriendlyName() }

[C#]
public string FriendlyName { get; }
```

Summary

Gets the friendly name of the current instance.

Property Value

A `System.String` containing the friendly name of the current application domain.

Description

This property is read-only.

The friendly name of a `System.AppDomain` instance created by an application is specified to the constructor. The friendly name of the default `System.AppDomain` is the name of the assembly file loaded in the application domain. The friendly name is formed by stripping the directory information from the assembly's file name. For example, if the loaded assembly has the name `"\MyAppDirectory\MyAssembly.exe"`, the friendly name of the default application domain is `"MyAssembly.exe"`.