**跨AppDomain运行代码**

在应用程序域之间的数据是相对独立的，当需要在其他AppDomain当中执行当前AppDomain中的代码时，可以使用CrossAppDomainDelegate委托,通过AppDomain的DoCallBack方法即可

static void Main(string[] args)

{

    Console.WriteLine("CurrentAppDomain start!");

    // 建立新的应用程序域对象

    AppDomain newAppDomain = AppDomain.CreateDomain("newAppDomain");

    // 创建委托方法

    CrossAppDomainDelegate crossAppDomainDelegate=new CrossAppDomainDelegate(MyCallBack);

    // 绑定 DomainUnload 的事件处理方法

    newAppDomain.DomainUnload += (obj, e) =>

    {

        Console.WriteLine("NewAppDomain unload!");

    };

    // 调用委托

    newAppDomain.DoCallBack(crossAppDomainDelegate);

    AppDomain.Unload(newAppDomain) ;

    Console.ReadKey();

}

static public void MyCallBack()

{

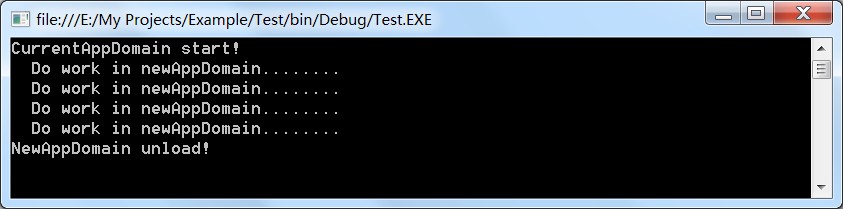
    string name = AppDomain.CurrentDomain.FriendlyName;

    for(int n=0;n<4;n++)

    Console.WriteLine(string.Format( "  Do work in {0}........" , name));

}

运行结果



**跨AppDomain的线程**

线程存在于进程当中，它在不同的时刻可以运行于多个不同的AppDomain当中。

下面的例子将介绍一下如何跨AppDomain使用线程

新建一个项目Example

static void Main(string[] args)

{

    var message = string.Format("  CurrentThreadID is:{0}\tAppDomainID is:{1}",

        Thread.CurrentThread.ManagedThreadId, AppDomain.CurrentDomain.Id);

    Console.WriteLine(message);

    Console.Read();

}

然后再新建一个ConsoleApplication项目，在此项目中新一个AppDomain对象，在新的AppDomain中通过ExecuteAssembly方法执行Example.exe程序。

static void Main (string[] args) {

    // 当前应用程序域信息

    Console.WriteLine ("CurrentAppDomain start!");

    ShowMessage ();

    // 建立新的应用程序域对象

    AppDomain newAppDomain = AppDomain.CreateDomain ("newAppDomain");

    // 在域 AppDomain 中载入并执行程序集

    // 调用该方法，当前线程就会进入到 Example.exe 中，执行 main 方法

newAppDomain.ExecuteAssembly ("Example.exe");

    AppDomain.Unload (newAppDomain);

    Console.ReadKey ();

}

public static void ShowMessage () {

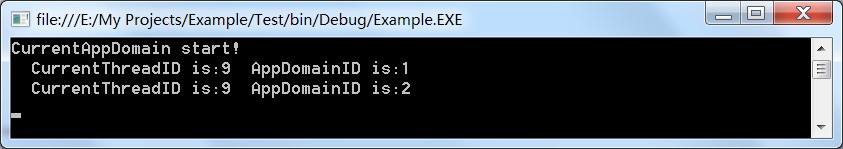
    var message = string.Format ("  CurrentThreadID is:{0}\tAppDomainID is:{1}",

        Thread.CurrentThread.ManagedThreadId, AppDomain.CurrentDomain.Id);

    Console.WriteLine (message);

}

运行结果



可见，ID等于9的线程在不同时间内分别运行于AppDomain 1与AppDomain 2当中。