利用AOP我们很容易的就能实现类似Log等业务无关的横切关注点，避免了这些代码在项目中不断的重复，利用.NET DLR 我们可以很容易的就实现一个简单的AOP，代码如下：

```
using System;
using System.Dynamic;
using System.Reflection;

namespace Zxf.Practices
{
    public class DynamicProxy : DynamicObject
    {
        Object m_Object;

        public DynamicProxy(Object obj)
        {
            m_Object = obj;
        }

        public override bool TryInvokeMember(InvokeMemberBinder binder, object[] args, out
        {
            try
            {
                Console.WriteLine("Begin: " + binder.Name);
                result = m_Object.GetType().InvokeMember(binder.Name, BindingFlags.InvokeM
                Console.WriteLine("end: " + binder.Name);
                return true;
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.ToString());
                throw;
            }
        }
    }
}
```

我们可以在项目中这样使用这个AOP类：

```
    using System;
    using Zxf.Practices;

    class Program
    {
        static void Main(string[] args)
        {
            dynamic proxy = new DynamicProxy(new TestClass());
            proxy.test();
        }
    }

    public class TestClass
    {
        public void test()
        {
            throw new Exception("test");
        }
    }
```