

aaudio 范例: 在 aaudio 中引用 .NET 对象

```
//在 aaudio 中引用 .NET 对象
import dotNet;
var compiler = dotNet.createCompiler("C#");//创建C#语言编译器
compiler.Source = /**
namespace CSharpLibrary
{
    public class Object
    {
        public static void Test(ref double num,int [] arr){
            num = 12.3;
            arr[0] = 56;
        }
    }
}
***/
```

```
compiler.import("CSharpLibrary"); //自程序集导入名字空间
```

```
/*
以下函数都可以创建原生 .NET 对象，
这些原生对象会被封包到 DispatchableObject 对象内以避免被自动转换为 aaudio 对象。
```

DispatchableObject 对象传入 aaudio 会被自动封包为 dotNet.object 对象。
这些对象传回 .NET 会被自动解包为原生 .NET 对象。

```
dotNet.object(value,byRef)
dotNet.byte(value,byRef)
dotNet.ubyte(value,byRef)
dotNet.word(value,byRef)
dotNet.uword(value,byRef)
dotNet.int(value,byRef)
dotNet.uint(value,byRef)
dotNet.long(value,byRef)
dotNet.ulong(value,byRef)
dotNet.float(value,byRef)
dotNet.double(value,byRef)
```

这些函数创建的对象保存了对原生 .NET 对象的引用，而非直接传值。
所以简单地将上述函数的第 @2 个参数设为 true 即可支持需要引用 .NET 原生对象的输出参数（ ref,out 参数）

```
dotNet.buffer(size,value) 可直接创建封装为 dotNet.object 且支持引用传参的 buffer 对象。
*/
```

```
var num = dotNet.double(12.5,true);//创建 .NET 对象，返回 dotNet.object 对象而不是普通数值。
```

```
//也可以创建 .NET 数组
var arr = dotNet.int({1,2,3}); //创建 .NET 对象，返回 dotNet.object 对象而不是普通数组。
```

```
//调用函数，dotNet.object 对象可以直接作为 .NET 函数参数使用。
CSharpLibrary.Object.Test(num,arr);
```

```
import console;
console.log( arr[1] ) //在 aaudio 中这种数组可用下标直接读写数组成员。
```

```
//DispatchableObject 对象如果存储的是 Primitive,enum,string 类型或这些类型的普通数组则可使用 Value 属性读写原始值
console.log( num.Value );
```

```
//支持 toString(),tonumber() 转换。
console.log(tostring(num),tonumber(num));
```

```
/*
DispatchableObject 对象可调用 byRef 函数设置是否传址（ ref,out 参数），
返回值为对象自身。不指定参数则返回一个表示当前传参设置是否为传址的布尔值。
*/
num.byRef(true)
```

```
console.pause();
```