

# AVEVA PMLNet Guide

[eryar@163.com](mailto:eryar@163.com)

**Abstract.** AVEVA PMLNet allows you to instantiate and invoke methods on .NET objects from PML proxy objects. The PML proxy objects behave just like any other PML object. The paper gives a hello world example to demonstrate the PMLNet usage.

**Key Words.** AVEVA, PMLNet, PML, AVEVA .Net, Managed C++

## 1. Introduction

AVEVA PMLNet 允许在 PML 代理对象中实例化或调用 .Net 对象的方法。代理 PML 对象的定义通过 .NET 类运行时创建出来，其与 .Net 类有类似的方法；代理 PML 类方法的参数只能是指定的几种，这几种类型的参数与 .Net 类型对应。代理 PML 对象的用法及功能表现与其他 PML 对象相同。

可被 PML 调用的组件（即动态库 dll）通过命令 IMPORT 加载到 AVEVA PDMS/Marine 中来。组件可由任意 .Net 语言来定义，例如：Managed C++（托管 C++）、C# 或 VB .NET 等。PMLNet 通过反射（reflection）来加载指定的组件。PMLNet 引擎（Engine）只会加载包含有自定义属性标记 PMLNetCallable 的组件中的类和方法。为了在 .Net 类中定义代理 PML 类必须满足一定的条件（rules）。

综上所述，在满足 PMLNet 一定条件下创建的 .Net 组件中的类是可以被 PML 使用的，即可以在 PML 代码中调用 .Net 组件中的资源。通过这种方式，可以获得以下几点优势：

- ❖ 在简单易学的 PML 中使用 .Net 库中的海量资源，如 Excel 的读写库等；
- ❖ 程序关键部分用 .Net 实现，代码的保密性相对 PML 而言要好很多；
- ❖ 由于 PMLNet 引擎支持 .Net 的组件，即 C++、VB.Net 写的库都可以被 PML 调用；

本文主要通过一个简单例子来说明 PML 代理类的定义方法，掌握后可以扩展到在 PML 中使用托管 C++，代码保密性更好且速度更快。这样就可以在 PML 中应用更广泛的资源，来提高程序的开发效率。

由于本人水平所限，文中的错误不妥之处在所难免，敬请不吝指教，将不胜感激。欢迎讨论交流，共同进步。

## 2. Design Details

下图所示为如何在 PDMS/Marine 中使用 PMLNet 实现自定义。有一些 .NET API 可以用来访问当前数据库任务，显示列表 drawlist，几何 geometry 和其他功能。用户可以通过 C# 的 API 来访问 PDMS/Marine，**但是直接在 C# 中调用 PML 是不可能的**（It is not possible to directly to call PML from C#）。然而 AVEVA 提供了一个事件机制来允许 PML 去订阅（subscribe）C# 发出的事件（events），如下图虚线所示。

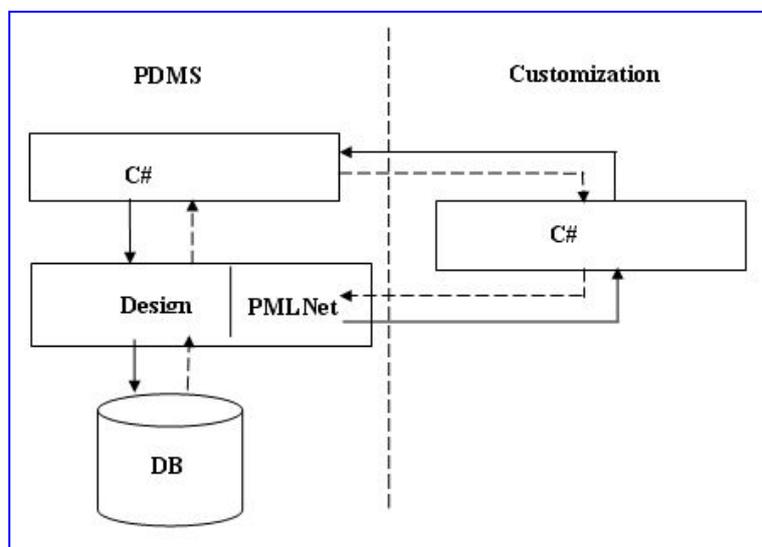


Figure 2.1 Using PMLNet

使用 PMLNet 有如下限制：

- ❖ 只有标记了 PMLNetCallable 且满足一定条件的 .Net 类才能被 PML 调用；
- ❖ 模块切换并不保留 .NET 对象，核心的 C++ 或 FORTRAN 对象在模块切换时也不会被保留；
- ❖ PML 调用 .NET 方法时只能传入指定类型的变量，其他类型不支持，如 DIRECTION, ORIENTATION 等；
- ❖ 在 .NET 中调用 PML 是不允许的，唯一的办法就是通过 .NET 的事件来调用 PML；
- ❖ It is not possible to enter 'partial' namespaces as you might in C# and expect them to be concatenated；

并不是所有的 PML 对象都可以传递到所调用的 .NET 方法中去，只有下表的 PML 类型的变量可以传递到调用 .NET 对象的方法中去：

PML	.NET
REAL	double
STRING	string
BOOLEAN	bool
ARRAY (these can be sparse and multi-dimensional)	Hashtable
OBJECT Any existing PML Net instance	PMLNetCallable class

Figure 2.2 Only PML variables types maybe passed to methods of .NET class

### 3. Using PMLNet

AVEVA 提供了一个简单的 PMLNet 例子，程序名为 PMLNetExample，将这个例子的代码例出如下所示：

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;
using System.Collections;

using Aveva.PDMS.PMLNet;

namespace Aveva.Pdms.Examples
{
    [PMLNetCallable()]
    public class PMLNetExample
    {
        [PMLNetCallable()]
        public event PMLNetDelegate.PMLNetEventHandler PMLNetExampleEvent;

        [PMLNetCallable()]
        public PMLNetExample()
        {
        }

        [PMLNetCallable()]
        public void Assign(PMLNetExample that)
        {
            //No state
        }

        [PMLNetCallable()]
        public void RaiseExampleEvent()
        {
            ArrayList args = new ArrayList();
            args.Add("ExampleEvent");
            if (PMLNetExampleEvent != null)
                PMLNetExampleEvent(args);
        }

        [PMLNetCallable()]
        public void Method()
        {
            MessageBox.Show("Called Method");
        }
    }
}
```

如上述代码所示，需要暴露给 PML 的方法或属性都要标记上 PMLNetCallable。将上述代码编译后生成一个组件放到 PDMS/Marine 的安装目录下，就可以写一个 PML 小程序来测试效果了。

在 CommandWindow 中输入如下图所示的命令：

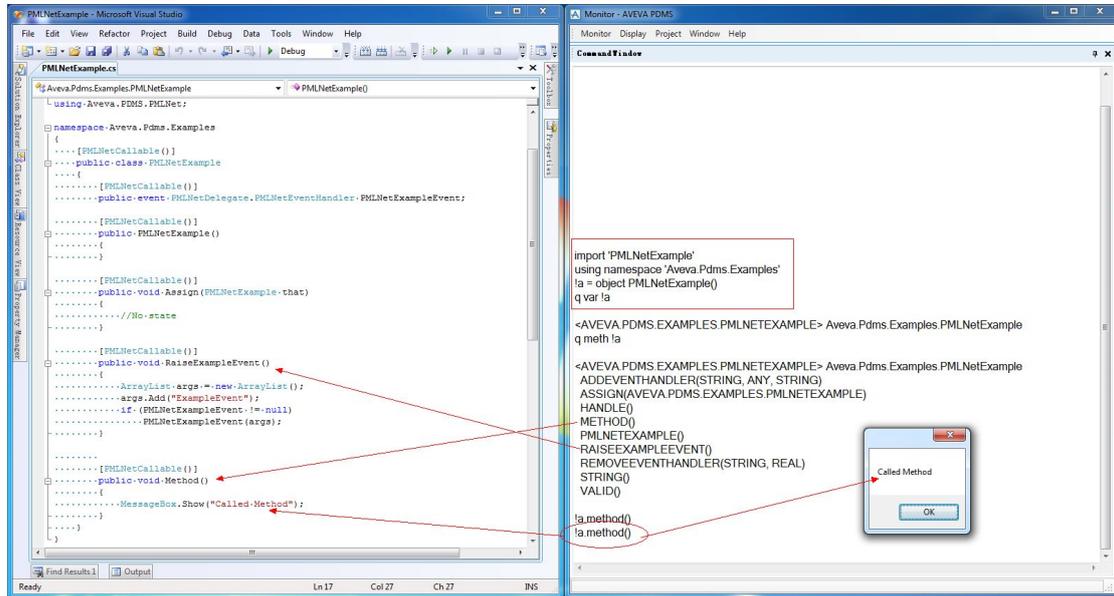


Figure 3.1 Using PML proxy object in PML

在 PML 中使用代理 PML 对象主要分为以下步骤：

- ❖ 使用 IMPORT 命令导入组件；

```
import 'PMLNetExample'
```

其中 PMLNetExample 是生成的 dll 组件名；

- ❖ 引入命名空间；

```
using namespace 'Aveva.Pdms.Examples'
```

命名空间的名称与 C# 中命名空间对应；

- ❖ 实例化类对象；

```
!a = object PMLNetExample()
```

实例化类对象的方式与其他 PML 对象一样。

- ❖ 调用对象方法；

```
!a.method()
```

调用对象的方法也和 PML 对象一样，得到结果如下图所示：

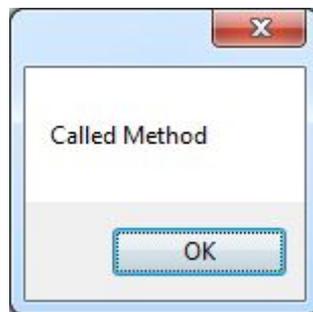


Figure 3.2 Test PML proxy object method

## 4. Rules for Calling .NET

想要在 PML 中调用 .NET 类对象，就必须遵守一定的规则。这些规则如下：

- ❖ 可被 PML 调用的 .NET 组件必须由 PMLNetCallable 标记且位于 %PDMSEXEXE% 目录中；
- ❖ .NET 组件中只有类可以被 PML 使用，结构体、接口及枚举除外；
- ❖ .NET 组件中需要被 PML 调用的类必须由 PMLNetCallable 标记；
- ❖ .NET 组件中需要被 PML 调用的方法必须由 PMLNetCallable 标记；
- ❖ .NET 组件中需要被 PML 调用的方法的参数类型必须为指定的几种类型；

PML	.NET
REAL	double
STRING	string
BOOLEAN	bool
ARRAY (these can be sparse and multi-dimensional)	Hashtable
OBJECT Any existing PML Net instance	PMLNetCallable class

Figure 4.1 Only PML variables types maybe passed to methods of .NET class

- ❖ .NET 组件中需要被 PML 调用的类和方法必须是公有的；
- ❖ .NET 组件中需要被 PML 调用的方法不支持默认参数的定义；
- ❖ .NET 组件中需要被 PML 调用的类和方法名称是区分大小写的；
- ❖ .NET 组件中需要被 PML 调用的类中必须要有 Assign() 方法；
- ❖ .NET 组件中需要被 PML 调用的类必须有一个由 PMLNetCallable 标记的公有的默认构造函数；

如果没有遵守上述条件之一，当加载相应的组件时就会报出错误，错误信息如下所示：

(46,87) PML: Object definition for XXX could not be found.

## 5. Conclusion

综上所述，在 PML 中调用 .NET 组件中的类还是很方便的，只要满足 PML 代理类定义的一些规则就可以了。

由于 .NET 组件中的类是由 .NET 语言实现的，所以托管 C++、C# 及 VB.NET 编写的组件都可以包装成代理 PML 类，进而被 PML 调用。这种方式就更加扩大了 PML 可使用的资源，如 C++ 的库都可以在 PML 中使用，且代码更保密，不易查看源码。

在 PML 中调用 .NET 组件中类是可行的，但直接在 C# 中调用 PML 却是不可行的。AVEVA 也提供了在 C# 中调用 PML 的方式：即通过事件订阅，详细请参考文档。

## 6. References

1. AVEVA .NET Customisation User Guide
2. Example of PML Callable: PMLNetExample
3. AVEVA Software Customisation Guide
4. AVEVA Software Customisation Reference Manual
5. AVEVA Data Access Routines User Guide