

PDMS call Operating System Command

eryar@163.com

1.Introduction

AVEVA 提供了三种二次开发的方式：DARs, PML 和 AVEVA .Net (C#)。DARs 是 Fortran 的函数库，也可以用 C/C++来开发，依赖的库少，但是只能用于控制台的程序，不能使用在 MFC 或 Qt 程序中，即不能有友好的用户界面 (GUI)，所以对于提取 PDMS 数据，然后进行一些处理的小程序，此方法还比较适用。

AVEVA 的 PML 和常见的 Python, Tcl/Tk, Lua 等一样是一种解释型的脚本语言，其只能在 PDMS 软件中被解释执行。也相当于批处理文件对于 Windows 操作系统。PDMS 自带的很多程序就是用 PML 开发的。脚本语言的显著优势就是将常用的命令封装出来，这样在测试程序新功能时，不需要编译源码，只需要编写脚本，提高软件开发效率；使用脚本时需要将功能暴露出来，这样会对接口的合理性做更多考虑，使软件架构更清晰等。缺点就是对于复杂算法，执行效率较低。还有就是 PML 完全依赖 AVEVA 软件，脱离 AVEVA 就像离开空气的飞机一样飞不起来。

AVEVA .Net 是 AVEVA 推出的一种更方便的开发方式。结合 C#丰富的库及相对于 PML 而言更加高效，使用 C#在 AVEVA 上可以快速地开发出应用程序。且 PML.Net 使得在 PML 中调用 C#的库成为可能。比如用 C#可以直接读定 Word, Excel, 甚至 AutoCAD 的 DWG 文件，所以使用 C#对 PDMS 进行开发优势明显，但也有不足之处，那就是程序调试不方便，当 C#的库 DLL 被 PDMS 加载后，不能卸载，不能对源码实时修改，开发效率有所影响。

PDMS 提供了在 PML 中调用系统命令的命令：syscom。借助这个 PML 命令，为 PDMS 的开发又提供了一种新架构 (New Architecture)。本文详细介绍 syscom 命令的用法，及如何使用这条命令来为 PDMS 的二次开发指出一条新路。

2.How to...

SYSCOM 也是一条 PML 命令，其用法和其他 PML 命令一样，只是其参数是 Windows 中的 DOS 命令。如在命令窗口中输入命令 `syscom 'dir'`，则可以在控制台窗口中显示出当前文件夹中的内容：

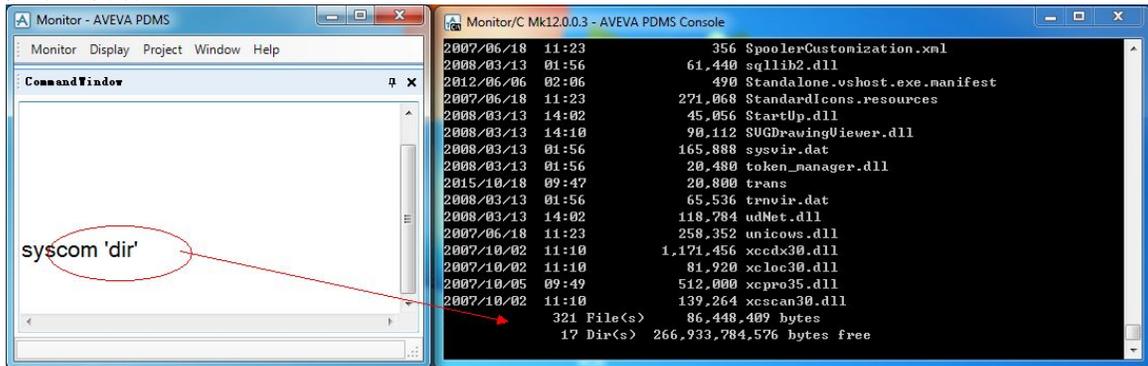


Figure 2.1 syscom usage

同理，输入 `syscom 'help'` 则可以显示出常见 DOS 命令的使用说明：

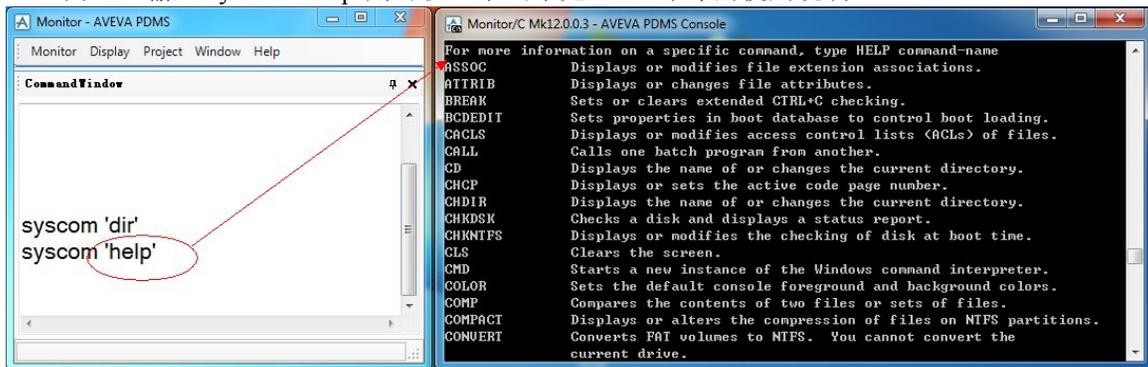


Figure 2.2 syscom call DOS command directly

由此可见，使用 `syscom` 相当于运行了一条 DOS 命令。因为 DOS 命令可以调用程序，如打开记事本，所以使用 `syscom` 也应该可以打开记事本。调用命令 `syscom 'notepad'` 即可以了，如下图所示：

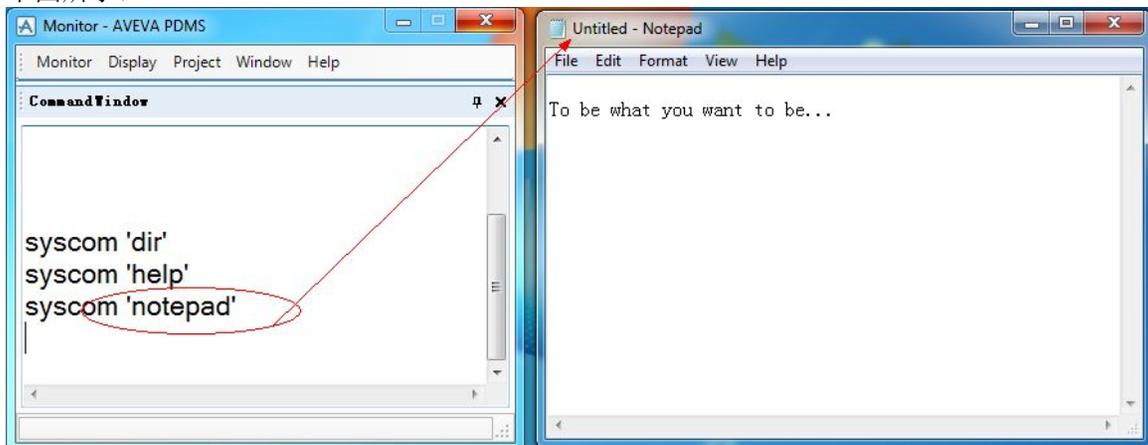


Figure 2.3 syscom to open a notepad

打开记事本后，可以看见 PDMS 程序处于等待状态。只有将记事本关闭，PDMS 程序才恢复正常。如何使 PDMS 程序不用等待调用的程序是否执行完毕呢？PDMS 中也提供了方法，那就是在程序名后加个 `&`。如 `syscom 'calc&'`，就可以打开计算器程序而不用去等待是否将其关闭。

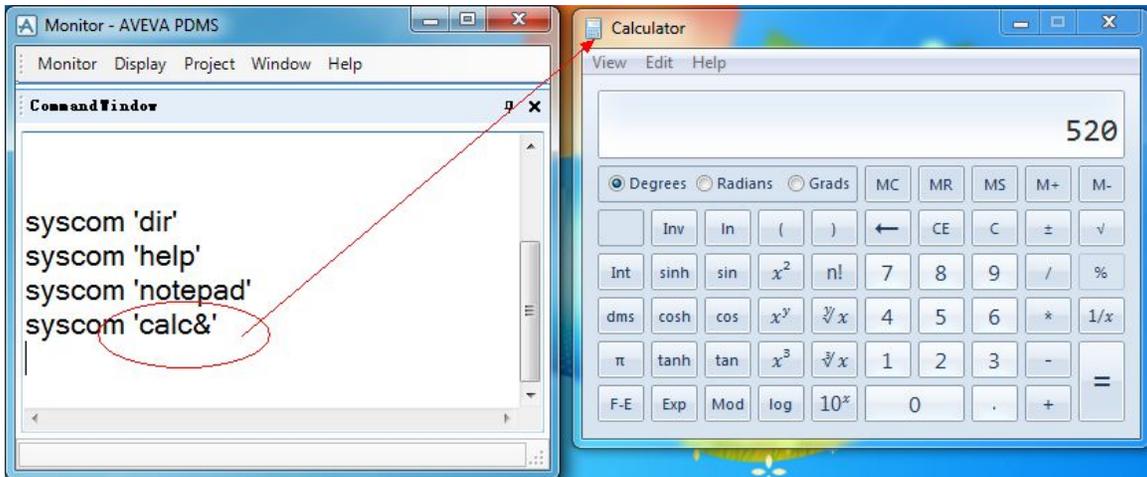


Figure 2.4 syscom to call calculator

3.New Architecture

有了这条命令，把 PDMS 的扩展性又提高了，因为可以用 syscom 来调用 Windows 中的脚本即批处理文件。这样就可以用批处理来调用其他程序，由此摆脱因为 AVEVA .NET(C#)开发时加载 DLL 后调试的不便，也可摆脱编程语言的限制，只要是一个程序都可以通过批处理来调用，即可以用 C/C++或 C#甚至 VB 来编写程序，再用通过 syscom 来调用。所以通过 syscom 调用批处理的方式也是一种全新的开发方式，下面通过一个简单的例子来进行说明如何将 PML 的变量传递给批处理。将下列代码保存为 syscom.bat 放在 PDMS 安装目录：

```
echo off

set param1=%1

echo echo from syscom.bat...
echo %param1%
```

在 PDMS 中输入命令：

```
syscom 'syscom.bat Hi,SYSCOM'
```

则会在 PDMS 的控制台上输出如下图所示：

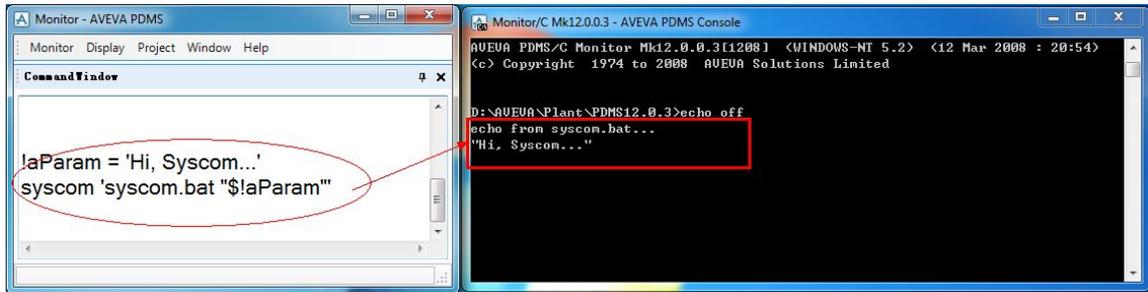


Figure 3.1 pass parameter to batch file

通过这种方式就可以将 PDMS 中的数据传递到自己的程序中了。下面以一个最简单的 C++ 程序为例来说明。

```
/*
 *   Copyright (c) 2015 Shing Liu All Rights Reserved.
 *
 *       File : main.cpp
 *       Author : eryar@163.com
 *       Date : 2015-10-27 21:00
 *
 *   Description : a simple C++ program to test PDMS syscom.
 *
 */

#include <iostream>

int main(int argc, char* argv[])
{
    for (int i = 0; i < argc; ++i)
    {
        std::cout << "parameter: " << argv[i] << std::endl;
    }
}
```

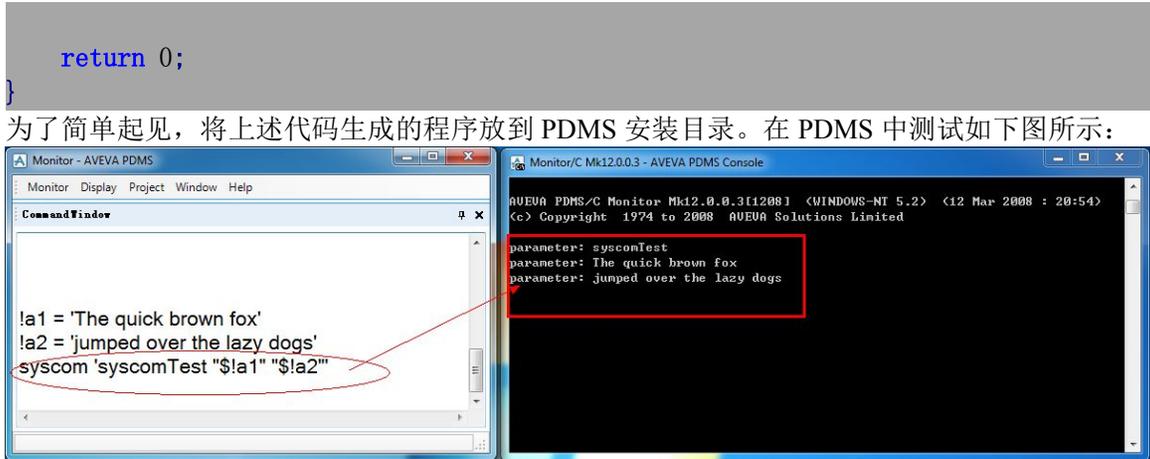


Figure 3.2 pass parameters to C++ program

从上面的例子可以看出，通过 `syscom` 可以调用任何应用程序，不管其用什么语言编写的。甚至可以直接将 PML 变量传递到程序中。这种方式来对 PDMS 进行开发比用 AVEVA .Net (C#) 来开发更方便，因为其调试很方便。如果处理的数据量比较大，可以将这些数据导出成文件，通过文件的方式来传递也是很方便的。

所以通过 `syscom` 调用批处理或应用程序或由批处理来调用程序的开发方式效率更高，且代码保密性好，把程序的关键部分用 C 或 C++ 或其他编程语言来实现，比 PML 的保密性好。因为程序和 PDMS 没有什么关联，所以即使不运行 PDMS，也是可以使用程序的。更重要的是程序调试方便，开发效率高。因此，这种开发方式可以说是一种全新地、方便地开发方式。

4. Conclusion

综上所述，通过详细介绍 PML 调用系统命令的命令 `syscom` 来对 PDMS 开发的几种方式进行对比。通过对比，说明使用 `syscom` 来调用程序的方式更为便利，且程序便于调试，保密性好。程序的规模也不受限制，可以是简单的小程序，也可以是复杂一些的应用程序。

5. References

1. AVEVA. DESIGN Reference Manual General Commands
2. AVEVA .NET Customisation User Guide
3. AVEVA Data Access Routines User Guide
4. AVEVA. Software Customisation Guide
5. AVEVA. Software Customisation Reference Manual