

8.8 技巧 101: MATLAB 与 C# 混合编程

8.8.1 技巧用途

C# 是一种简洁、安全的面向对象的语言，开发人员可以使用它来构建在 .NET Framework 上运行的各种安全、可靠的应用程序。

MATLAB Builder NE 是 MATLAB Compiler (MATLAB 编译器) 的一个扩展产品。.NET 程序员可以使用任何支持公共语言规范 (CLS) 的语言 (如 C#、VB.NET 等) 开发的程序来访问 MATLAB Builder NE 编译的 MATLAB 函数。

MATLAB Builder NE 将 MATLAB 函数转换为封装了 MATLAB 代码的 .NET 方法。每一个 MATLAB Builder NE 组件包含一个或多个类，每一个类提供了指向 M 函数的接口。.NET 组件提供了符合 CLS 规范的、封装了 M 代码的一类方法。

本技巧介绍在 C# 程序中调用 MATLAB 函数的方法。

8.8.2 技巧实现

下面以 MATLAB 帮助文件中的示例程序为例来说明如何在 C# 应用程序中使用由 MATLAB Builder NE 所创建的组件。该示例程序所用到的文件存放在如下的路径：

matlabroot\toolbox\dotnetbuilder\Examples\VSversionnumber\PlotExample

① 在 MATLAB 中编写函数 M 文件 drawgraph.m，并加入相应的帮助信息，用于解释函数的用途和参数，其代码如下：

```
function drawgraph(coords)
% DRAWGRAPH Plot a curve from the specified x and y coordinates contained in the coords array.
% DRAWGRAPH (coords) Plots a curve from the specified coordinate values
% in a MATLAB figure window.
% This file is used as an example for the MATLAB Builder NE product.

% Copyright 2001 - 2006 The MathWorks, Inc.
% $ Revision: 1.1.6.3 $ $ Date: 2007/12/03 22:07:24 $
% 使用输入的 2xN 的数组来绘制图形
plot(coords(1,:), coords(2,:));
```

② 打开 MATLAB 的部署工具 (Deploy Tool)，以便编译 M 文件。

在命令窗口中输入 `deploytool` 命令,即可打开 MATLAB Compiler 对话框;或者单击“应用程序”导航栏右侧的下拉箭头,在下拉列表中选择“库编译器”,如图 8.8-1 所示。打开的 MATLAB Compiler 窗口如图 8.8-2 所示。



图 8.8-1 从导航栏打开 MATLAB Compiler



图 8.8-2 MATLAB Compiler 窗口

③ 选择编译类型。在“应用程序类型”选择框中选择.NET Assembly,表示使用 MATLAB Builder NE 编译器产品来编译.NET 组件。

④ 向项目中添加要编译的 M 文件。在图 8.8-2 所示的对话框中,单击“导入的函数”右侧的“+”按钮,添加要编译的 M 文件 drawgraph.m。

⑤ 输入库的名称、类的名称等信息。在“库的名称”输入框中输入 PlotComp 作为库的名称,即编译出来的库为 PlotComp.dll;在“组件的命名空间的名称”输入框中输入 PlotComp;重命名类名为 plotter,方法的名称为 drawgraph.m。

⑥ 编译选项设置。单击图 8.8-2 所示 MATLAB Compiler 界面上方的 Settings 按钮,打开 Settings 对话框,如图 8.8-3 所示,用户可以设置输出文件夹等信息。在此选择默认值。

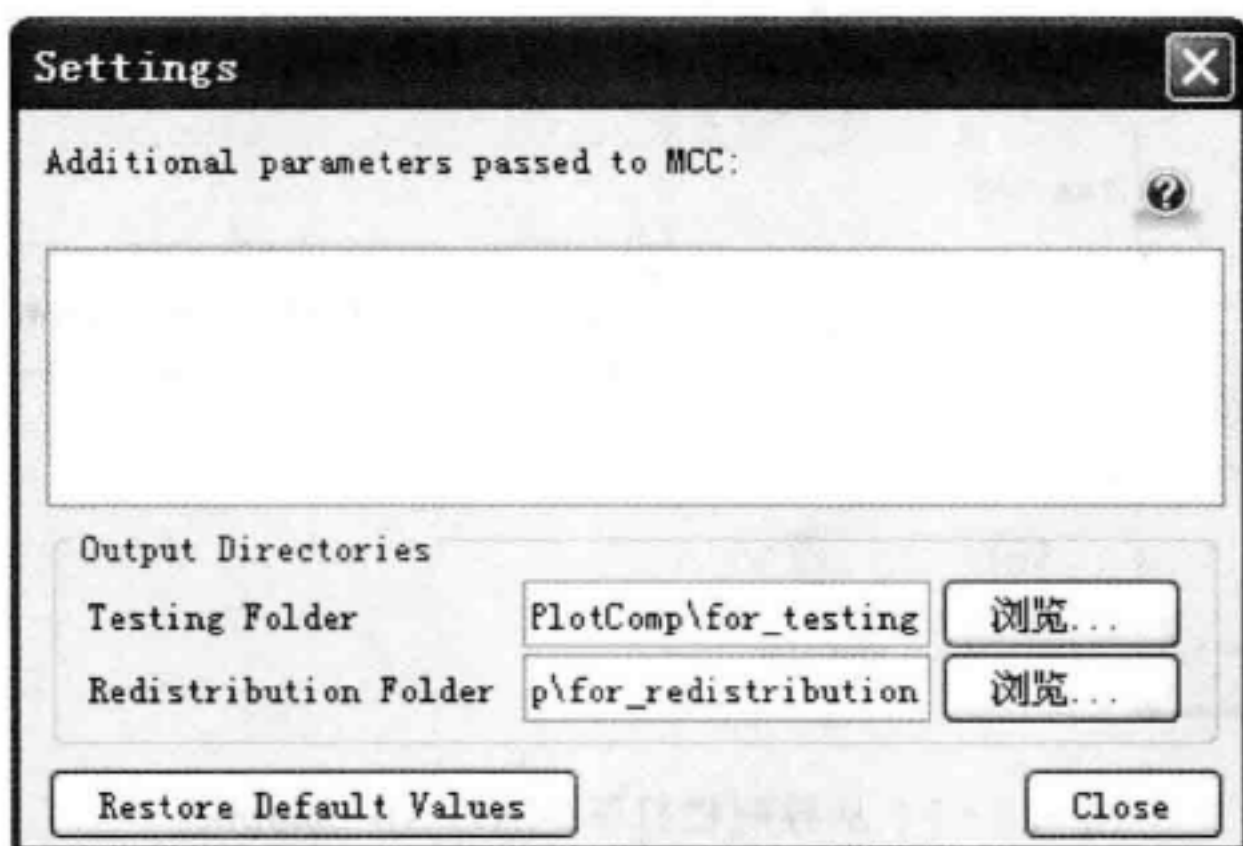


图 8.8-3 部署设置对话框

【注】 所有文件名以及路径名称都不能带有中文字符;否则,编译将出错。

⑦ 编译。单击图 8.8-2 所示窗口中的 Package 按钮,开始编译,并显示如图 8.8-4 所示的打包提示对话框。

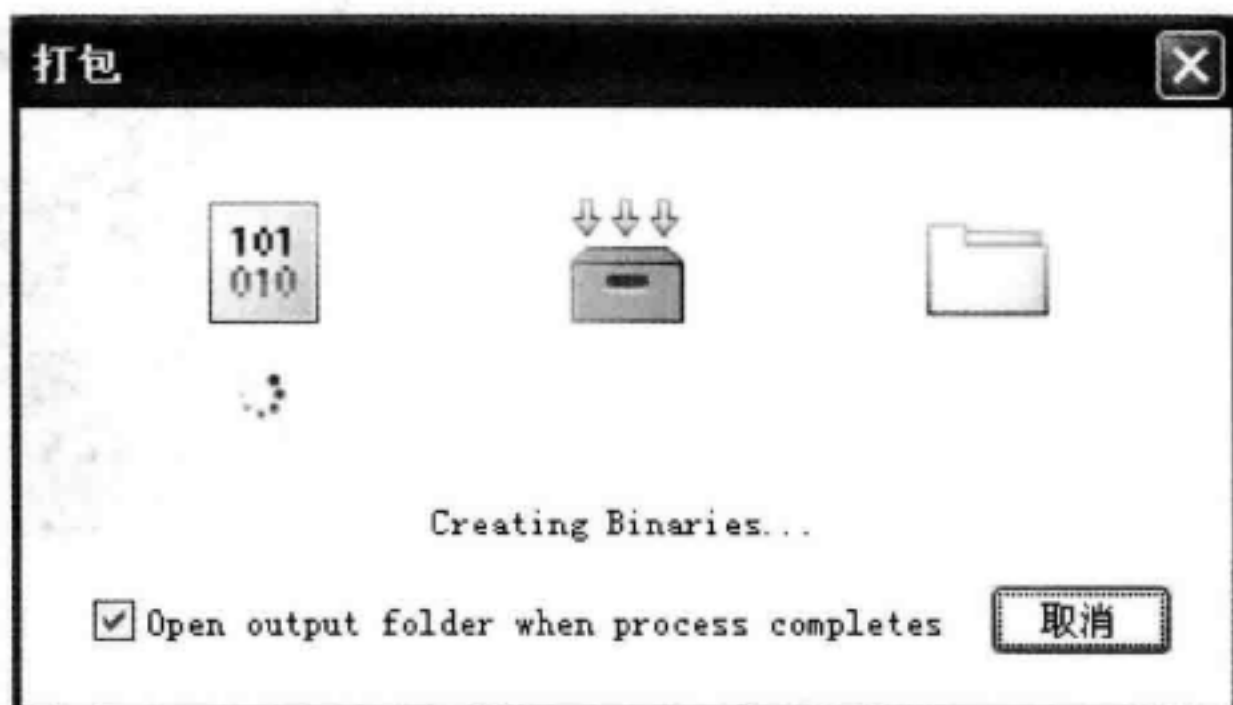


图 8.8-4 “打包”对话框

编译完成后,在输出文件夹下创建 3 个子文件夹:for_testing(用于测试)、for_redistribution_files_only(只包含发布的文件)和 for_redistributionsrc(包含打包的安装程序)。其中,for_redistribution_files_only 文件夹中的 PlotComp.dll 为生成的动态链接库文件,其封装了 Plotter 类。可以在 for_testing 文件夹中的 Plotter.cs 文件中查看 Plotter 类的代码。

⑧ 编写访问 .NET 组件的 C# 应用程序。在 Microsoft Visual Studio 2008 中创建基于 C# 的控制台应用程序。解决方案的名称为 ConsoleApplication1, 工程的名称为 matlabcsharp, Visual Studio 2008 自动生成应用程序的框架, 代码如下:

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

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

⑨ 添加对所需组件的引用。右击工程名称 matlabcsharp, 在弹出的快捷菜单中选择 Add Reference, 弹出 Add Reference 对话框, 如图 8.8-5 所示。单击 .NET 选项卡, 添加对 MathWorks .NET MWArray API 组件的引用。该组件所在的文件默认的路径为: matlabroot\toolbox\dotnetbuilder\bin\win32\v2.0\MWArray.dll。

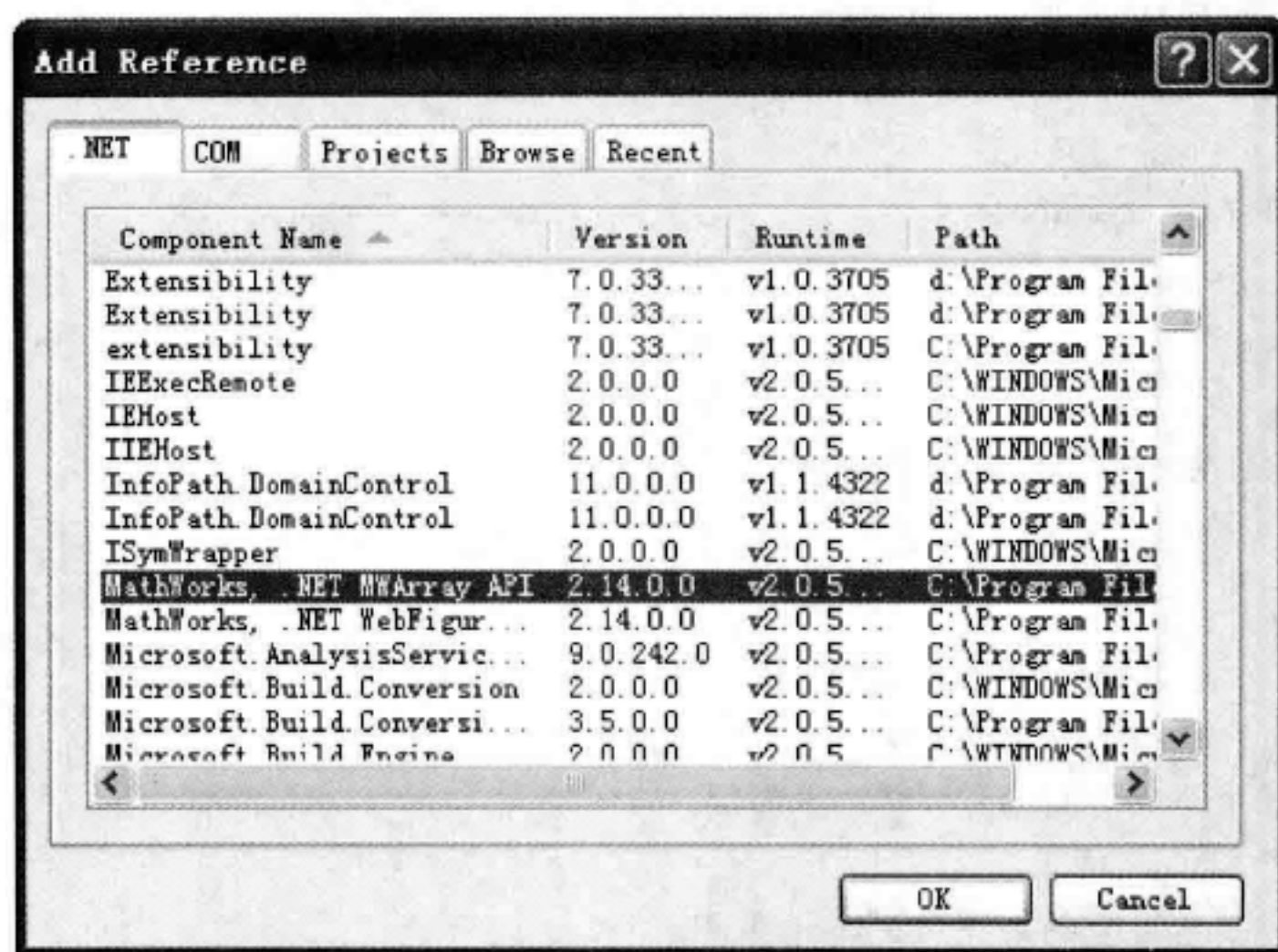


图 8.8-5 添加 MathWorks 的 .NET MWArray API 组件

同时, 单击 Browse 选项卡, 添加对由 MATLAB Builder NE 所编译创建的 PlotComp 组件的引用, 如图 8.8-6 所示。该组件所在的文件默认的路径为: MATLAB 当前目录下的 PlotComp\for_redistribution_files_only\PlotComp.dll。

⑩ 添加调用 .NET 组件的代码以及所需要的命名空间。应用程序的名称为 Program.cs, 其代码如下:



图 8.8-6 添加对由部署工具创建的组件的引用

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

% 指定组件所在的命名空间
using MathWorks.MATLAB.NET.Arrays;
using MathWorks.MATLAB.NET.Utility;
using PlotComp

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                // 定义绘图的点数
                const int numPoints = 10;

                // 为绘图数据分配本地数组
                double[,] plotValues = new double[2, numPoints];

                // 绘制 5x 对 x² 的曲线
                for (int x = 1; x <= numPoints; x++)
                {
                    plotValues[0, x - 1] = x * 5;
                    plotValues[1, x - 1] = x * x;
                }
            }
        }
    }
}
```

```
//创建一个新的 plotter 对象
plotter plotter = new Plotter();

//绘制图形,调用 MWNumericArray 将 C# 本地数组
//转换为 MATLAB 数值数组
plotter.drawgraph((MWNumericArray)plotValues);
//等待用户退出程序。若不加改行代码,则应用程序窗口会一闪而过
Console.ReadLine();
}

catch (Exception exception)
{
    Console.WriteLine("Error: {0}", exception);
}
}
}
```

编译并运行程序,结果如图 8.8-7 所示。

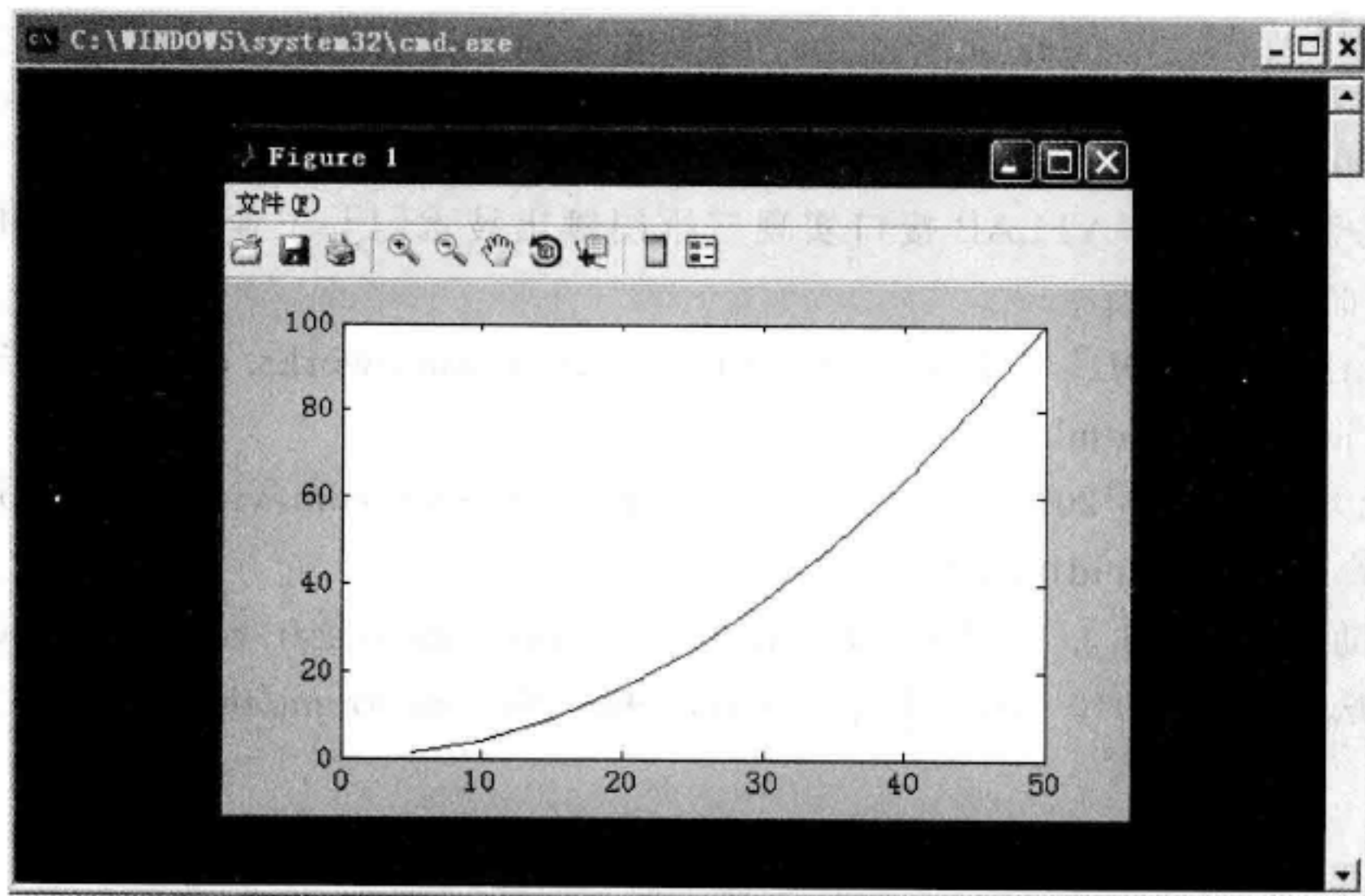


图 8.8-7 程序运行结果