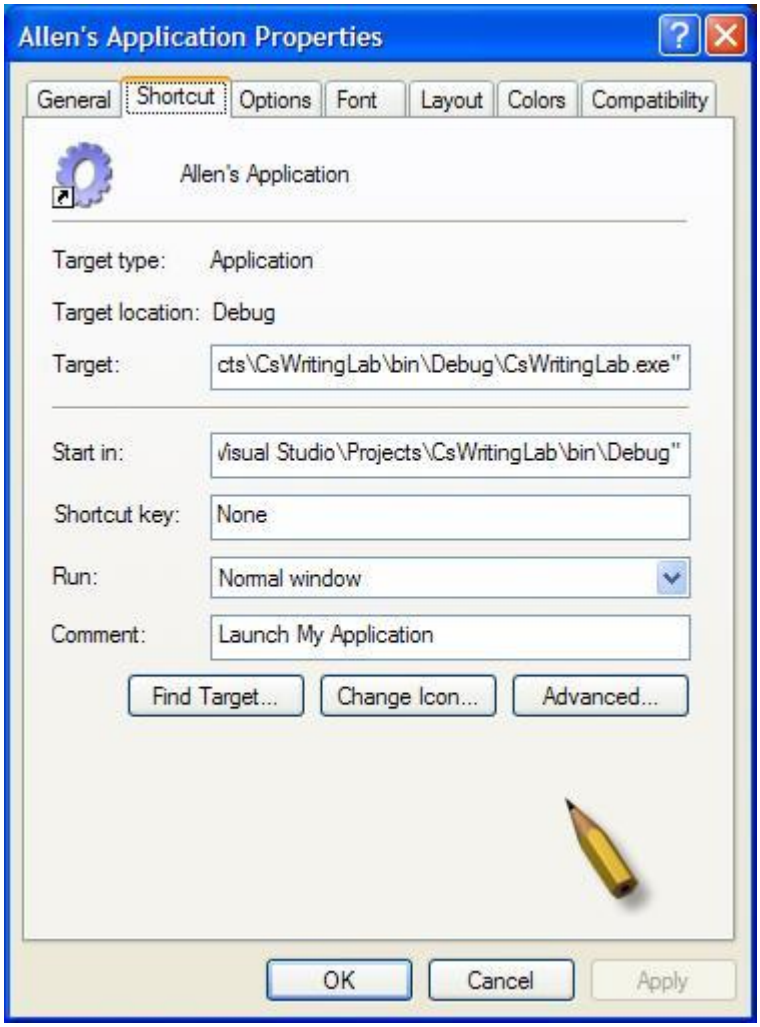
创建快捷方式对于绝大多数 Windows 用户来说都是小菜一碟了，然而，这项工作却为程序员带来不少麻烦。.NET 没有提供简便直接的创建快捷方式的方法，那么在 .NET 中我们如何为应用程序创建快捷方式呢？

1. 快捷方式文件

快捷方式实质上是一个扩展名为 .LNK 的文件。右击一个快捷方式文件并选择属性，跳到快捷方式选项卡，如图 2 所示：



你可以看到一个快捷方式包含如下数据：

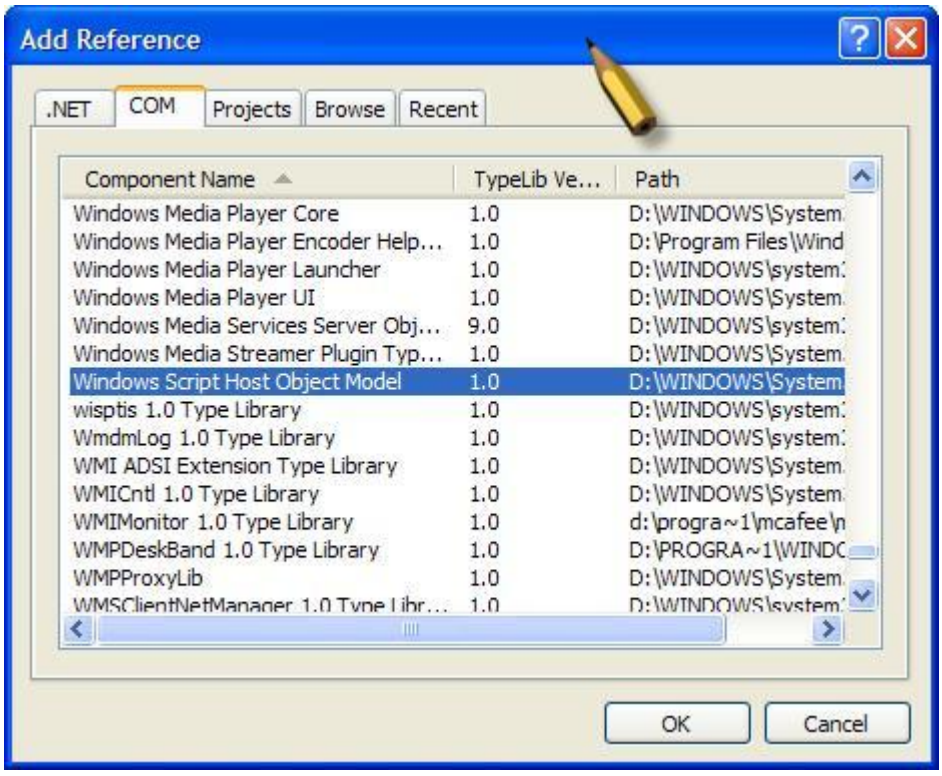
- 快捷方式的名字
- 快捷方式所指向的目标所在的位置
- 快捷方式所指向的目标的工作目录
- 激活该快捷方式的热键

- 快捷方式所指向的目标运行时的窗口风格（普通、最大化和最小化）
- 该快捷方式的描述性文字
- 快捷方式的图标所在的位置

2. 使用 WSH 创建快捷方式

2.1 添加 WSH 的引用

添加引用的方法非常简单，右击你的项目并选择添加引用，选择 COM 选项卡并选择 Windows Script Host Object Model，如图 3 所示：



2.2 创建你的快捷方式

创建一个快捷方式的完整代码如下：

```
// Code #01

using System;
using IWshRuntimeLibrary;

class Program
{
    static void Main(string[] args)
    {
        WshShell shell = new WshShell();
```

```
        IWshShortcut shortcut = (IWshShortcut)shell.CreateShortcut(
            Environment.GetFolderPath(Environment.SpecialFolder.DesktopDirectory) +
            "\\" + "Allen's Application.lnk"
            );
        shortcut.TargetPath = System.Reflection.Assembly.GetExecutingAssembly().Location;
        shortcut.WorkingDirectory = System.Environment.CurrentDirectory;
        shortcut.WindowStyle = 1;
        shortcut.Description = "Launch Allen's Application";
        shortcut.IconLocation = System.Environment.SystemDirectory + "\\" + "shell32.dll, 165";
        shortcut.Save();
    }
}
```

首先，我们创建一个 WshShell 的实例对象，接着通过该对象的 CreateShortcut 方法来创建 IWshShortcut 接口的实例对象，传递给 CreateShortcut 方法的参数是将要创建的快捷方式的完整路径（包括该快捷方式的名字）。接下来，我们就要设置 IWshShortcut 实例对象的相关属性值了。

2.3 设置快捷方式的属性

2.3.1 TargetPath

该属性仅用于设置或者读取快捷方式的目标所在的位置。Code #01 中，将要创建的快捷方式指向本应用程序。

2.3.2 WorkingDirectory

该属性指定应用程序的工作目录，当用户没有指定一个具体的目录时，快捷方式的目标应用程序将使用该属性所指定的目录来装载或保存文件。

2.3.3 WindowStyle

该属性指定快捷方式的目标应用程序的窗口为普通（原始）状态、最小化或者最大化。对比图 1 中 Run 下拉菜单的项，该属性的值及其意义如下：

Value	Meaning
1	Normal window
3	Maximized
7	Minimized

2.3.4 Description

该属性设置用于设置或者读取快捷方式的附加说明。

2.3.5 IconLocation

该属性用于指定快捷方式的图表的所在位置，它的值包含一个完整的路径和一个索引值。Code #01 中，快捷方式的图标被设定为系统文件夹中的 Shell32.dll 文件所包含的第 165 个图标。

2.4 生成快捷方式

CreateShortcut 仅仅创建一个 IWshShortcut 的实例对象，它不会为你生成任何快捷方式，当一切就绪后，你必须调用 IWshShortcut.Save 方法才能生成快捷方式文件。

3. 活用快捷方式简化操作

想象一下你的应用程序支持命令行参数组合运行，例如：

```
App  /out:output.txt  /sortby:date  /desc
```

而你又希望只需要轻轻按下 CTRL+ALT+F11 就可以直接启动这项功能，这时你需要用到 IWshShortcut 的 Arguments 和 Hotkey 两个属性了：

```
// Code #02

shortcut.Arguments = "/out:output.txt /sortby:date /desc";
shortcut.Hotkey = "CTRL+ALT+F11";
```

注意：Hotkey 属性的值里面不能包含空格。

你可以在应用程序里面像平常那样处理命令行参数：

```
// Code #03

class Program
{
    static void Main(string[] args)
    {
        foreach (string arg in args)
        {
            Console.WriteLine(arg);
        }
    }
}

// Output:
//
// /out:output.txt
// /sortby:date
// /desc
```

现在，你把这个快捷方式放到桌面，每当你需要运行的时候，轻轻的按下 CTRL+ALT+F11 就行了.