

# 命令类型插件示例：PhoneCommand

- 1. 描述
- 2. 创建命令类型插件



## 1. 描述

以创建PhoneCommand插件为例，介绍如何创建命令类型插件。  
查看完整代码请参见：<https://gitee.com/huozige-china/phone-command>。

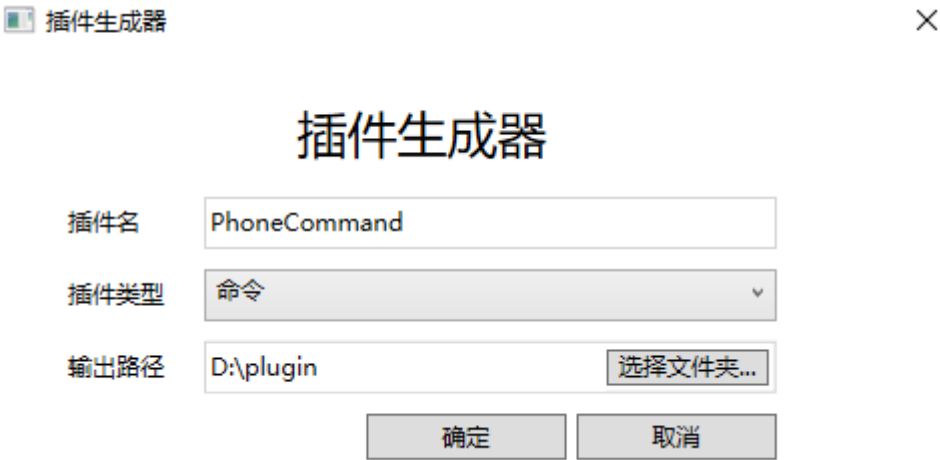


## 2. 创建命令类型插件

### 操作步骤

1

下载PluginsProjectCreator.zip，解压后运行ForguncyPluginCreator.exe，在弹出的对话框中，输入您的插件名称，选择插件类型为命令，并设置插件输出的路径。  
设置完成后，单击“OK”。



2

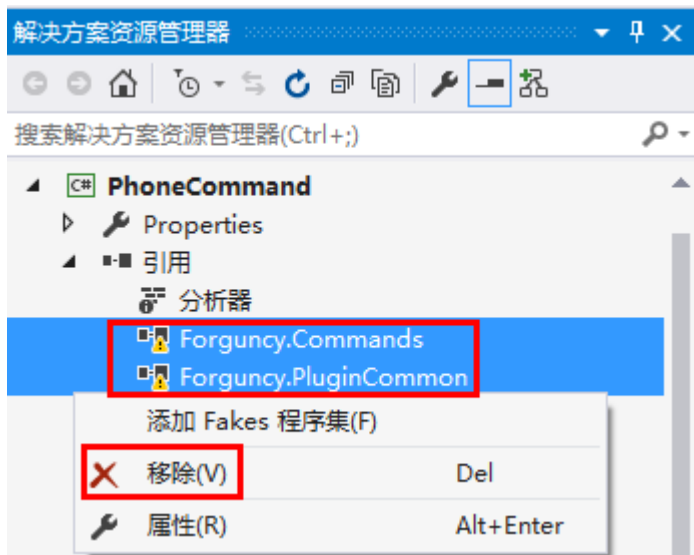
打开您创建的插件所在的文件夹，使用Visual Studio打开.csproj文件。

(D:) > plugin > PhoneCommand

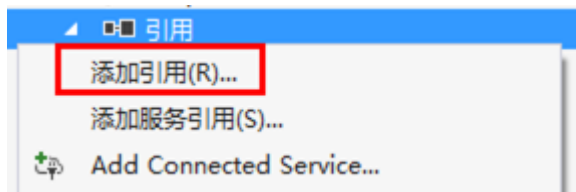
| 名称                  | 修改日期             | 类型                  | 大小   |
|---------------------|------------------|---------------------|------|
| Properties          | 2020/10/21 17:29 | 文件夹                 |      |
| Resources           | 2020/10/21 17:29 | 文件夹                 |      |
| Tools               | 2020/10/21 17:29 | 文件夹                 |      |
| PhoneCommand.cs     | 2020/10/21 17:29 | Visual C# Sourc...  | 1 KB |
| PhoneCommand.csproj | 2020/10/21 17:29 | Visual C# Projec... | 4 KB |
| PluginConfig.json   | 2020/10/21 17:29 | JSON File           | 1 KB |

3

在解决方案资源管理器中将Forguncy.Commands和Forguncy.PluginCommon移除掉。

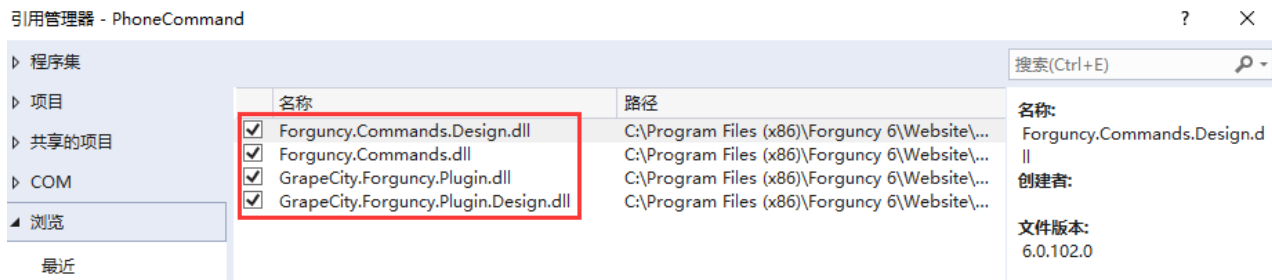


4 在解决方案资源管理器中，“引用”上右击，选择添加引用。

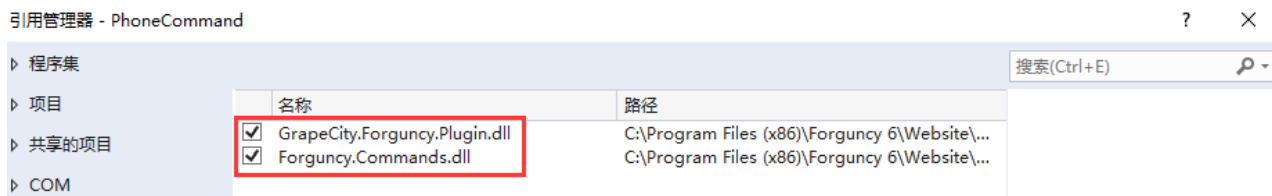


5 在右下角单击“浏览”，选择活字格安装文件所在的bin文件夹，如果是默认安装，则路径为“C:\Program Files (x86)\Forguncy 6\Website\designerBin”，找到以下文件，将其添加到解决方案资源管理器中，

- 如果需要自定义命令编辑器或实现ICommandChecker接口，请添加Forguncy.Commands.dll，Forguncy.Commands.Design.dll，GrapeCity.Forguncy.Plugin.dll，GrapeCity.Forguncy.Plugin.Design.dll 这四个文件；



- 在其他情况下，请添加Forguncy.Commands.dll，GrapeCity.Forguncy.Plugin.dll 这两个文件；



并且将这些文件的属性中的“复制本地”设置为“False”。



6

在PhoneCommand.cs文件中添加代码。

添加“提取号码单元格”、“提取策略”和“提取号码类型”三个属性，且该命令可在普通单元格、页面加载和表格中使用。

```
[Icon("pack://application:,,,/PhoneCommand;component/Resources/Icon.png"
)]
public class PhoneCommand : Command
{
    [DisplayName("")]
    [FormulaProperty(true)]
    public object ExtractPhoneNumberCell { get; set; }
    [DisplayName("")]
    public ExtractPolicy ExtractPolicy { get; set; }
    [DisplayName("")]
    public ExtractPhoneType ExtractPhoneType { get; set; }

    public override string ToString()
    {
        return "";
    }

    public override CommandScope GetCommandScope()
    {
        return CommandScope.Cell | CommandScope.ListView |
CommandScope.PageLoad;
    }
}
public enum ExtractPolicy
{
    [Description("")]
    ExtractFirstOne,
    [Description("")]
    ExtractLastOne
}
public enum ExtractPhoneType
{
    [Description("")]
    Mobile,
    [Description("")]
    Phone
}
}
```

7

在Resources/PhoneCommand.js中添加代码：

```

var PhoneCommand = (function (_super) {
    __extends(PhoneCommand, _super);
    function PhoneCommand() {
        return _super !== null && _super.apply(this, arguments) || this;
    }

    PhoneCommand.prototype.execute = function () {
        // Get settings
        var commandSettings = this.CommandParam;
        var extractPolicy = commandSettings.ExtractPolicy;
        var extractPhoneNumberCell =
commandSettings.ExtractPhoneNumberCell;
        var extractPhoneType = commandSettings.ExtractPhoneType;
        var text = this.evaluateFormula(extractPhoneNumberCell);
        var mobileNumbersArray = [];
        var phoneNumbersArray = [];
        if (extractPhoneType === ExtractPhoneType.Mobile) {
            var mobileNumberMatches =
text.match(/\\d{3}\\s*\\d{4}\\s*\\d{4}/g);
            if (mobileNumberMatches && mobileNumberMatches.length > 0) {
                mobileNumberMatches.map(function (temp) {
                    mobileNumbersArray.push(temp);
                });
            }
        } else if (extractPhoneType === ExtractPhoneType.Phone) {
            var phoneNumberMatches =
text.match(/((\\+?\\s*\\d{2,3}\\s*\\-?\\s*)?(\\d{2,3})\\s*\\-?\\s*)(\\d{7,8})(\\s*\\-?\\s*(\\d{3,}))?/g);
            if (phoneNumberMatches && phoneNumberMatches.length > 0) {
                phoneNumberMatches.map(function (temp) {
                    temp = temp.replace(/\\-/g, "");
                    phoneNumbersArray.push(temp);
                });
            }
        }
        var array = mobileNumbersArray;
        if (extractPhoneType === ExtractPhoneType.Phone) {
            array = phoneNumbersArray;
        }
        if (array.length > 0) {
            if (extractPolicy === ExtractPolicy.ExtractFirstOne) {
                window.location.href = "TEL:" + array[0];
            } else if (extractPolicy === ExtractPolicy.ExtractLastOne) {
                window.location.href = "TEL:" + array[array.length - 1];
            }
        }
    }

    return PhoneCommand;
})(Forguncy.CommandBase));
var ExtractPolicy = {
    ExtractFirstOne: 0,
    ExtractLastOne: 1
};
var ExtractPhoneType = {
    Mobile: 0,
    Phone: 1
}

```



```
// Key format is "Namespace.ClassName, AssemblyName"
Forguncy.CommandFactory.registerCommand("PhoneCommand.PhoneCommand,
PhoneCommand", PhoneCommand);
```

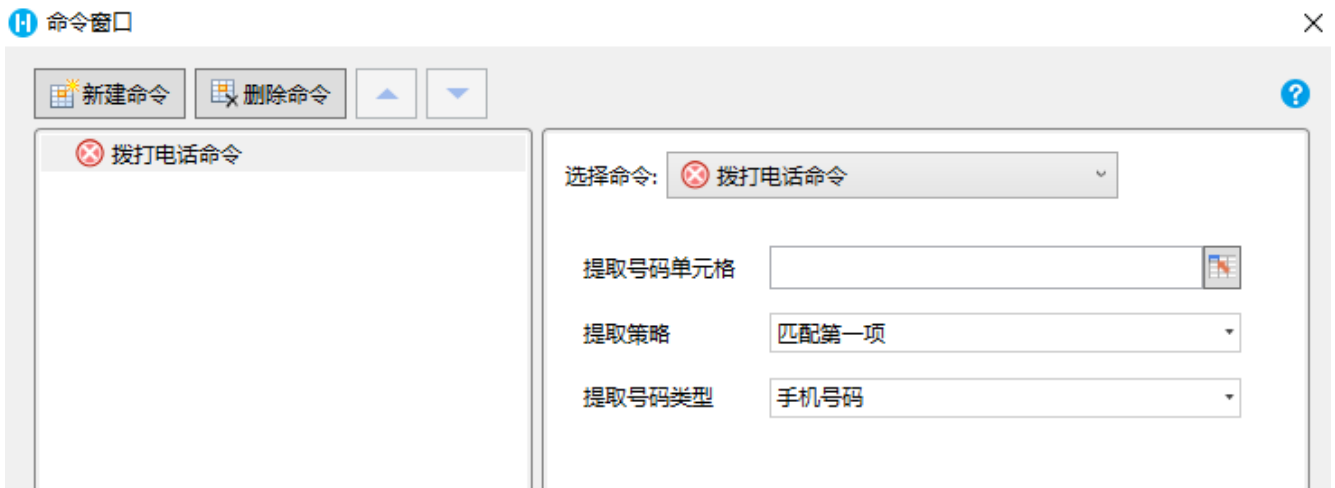
8

您可以自定义插件图标、logo、介绍等信息。编辑完成后，在插件名称上右击，选择“生成”或者“重新生成”。完成后重启活字格设计器，新建的插件就会安装到活字格设计器中。



结束

您可以在页面中设置一个按钮，编辑其命令为“拨打电话命令”，并进行其他设置。运行页面后，在手机上就可以单击此按钮来提取号码并选择呼叫、复制或添加到手机通讯录等。



[回到顶部](#)