

自定义插件中的显示文字

- 1. 描述
- 2. 更改插件名称
- 3. 更改插件属性名
- 4. 添加属性描述
- 5. 更改枚举值的显示文字
- 6. 为一些关联属性增加分类



1. 描述

您可以自定义插件中的显示文字。



2. 更改插件名称

默认插件名与.cs文件中的类名一致。

您可重写ToString来更改插件名称，在return后输入您想更改的插件名称即可。

```
namespace MyDateCellType
{
    [Designer("MyDateCellType.MyDateCellTypeDesigner, MyDateCellType")]
    [Icon("pack://application:,,,/MyDateCellType;component/Resources/Icon.png")]
    [SupportUsingScope(PageScope.AllPage, ListViewScope.None)]
    0 个引用
    public class MyDateCellType : CellType, ICommandCellType
    {
        0 个引用
        public override string ToString()
        {
            return "LayDate";
        }
        [OrderWeight(1)]

        0 个引用
        public DateMode DateMode
    }
}
```



3. 更改插件属性名

属性编辑器的名称默认为您设置的属性名称。

您可以增加系统的DisplayNameAttribute来自定义属性编辑器名称。

```

namespace MyDateCellType
{
    [Designer(typeof(MyDateCellTypeDesigner))]
    [Icon("pack://application:,,,/MyDateCellType;component/Resources/Icon.png")]
    0 个引用
    public class MyDateCellType : CellType
    {
        0 个引用
        public override string ToString()
        {
            return "LayDate";
        }
        [DisplayName ("日期模式")]
        0 个引用
        public DateMode DateMode
        {

```



4. 添加属性描述

如果您想要添加属性的描述信息，您可以为该属性添加Description特性。

```

namespace MyDateCellType
{
    [Designer("MyDateCellType.MyDateCellTypeDesigner, MyDateCellType")]
    [Icon("pack://application:,,,/MyDateCellType;component/Resources/Icon.png")]
    0 个引用
    public class MyDateCellType : CellType
    {
        0 个引用
        public override string ToString()
        {
            return "LayDate";
        }
        [DisplayavName ("日期模式")]
        [Description("属性的描述信息")]
        0 个引用
        public DateMode DateMode
        {
            get; set;

```

运行并重启设计器后，会在属性的旁边生成一个问号图标，鼠标移动问号上会显示设置的属性描述信息。



5. 更改枚举值的显示文字

如果属性值为枚举类型，则属性编辑器默认为下拉选择框。如果您想更改枚举值的显示文字，则需要为每个枚举值添加Description特性。

```

namespace MyDateCellType
{
    [Designer("MyDateCellType.MyDateCellTypeDesigner, MyDateCellType")]
    [Icon("pack://application:,,,/MyDateCellType:component/Resources/Icon.png")]
    0 个引用
    public class MyDateCellType : CellType
    {
        0 个引用
        public override string ToString()
        {
            return "LayDate";
        }
        [DisplayName("日期模式")]
        [Description("属性的描述信息")]
        0 个引用
        public DateMode DateMode
        {
            get; set;
        }
    }
    1 个引用
    public enum DateMode
    {
        [Description("只显示日期")]
        Date,
        [Description("只显示时间")]
        Time,
        [Description("显示日期和时间")]
        DateTime
    }
}

```

重新生成插件后，在设计器将显示设置的枚举值。



6. 为一些关联属性增加分类

如果一些属性有关联，可添加活字格自定义的CategoryHeader特性为相关联的属性设置标题。

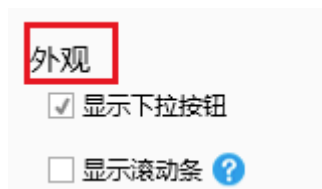
```

[CategoryHeader("外观")]
[DisplayName("显示下拉按钮")]
public Buttonvisibility DropDownButtonvisibility
{
    get;
    set;
}

[DisplayName("显示滚动条")]
[Description("拖动滚动条快速查找日期")]
public bool ShowCalendarScrollBar
{
    get;
    set;
}

```

重新启动工程，在设计器中显示如下：



[回到顶部](#)