

一、添加功能入口

示例：

1. 创建Style类

2. 创建Command

UI Command包装了按钮，包括Style样式、显示文本、描述文本以及操作回调等。

创建一个继承TCommands模板的类，然后该类包含UI Command Action属性。通过实现RegisterCommands方法，在该方法中用UI_COMMAND宏定义该属性变量。然后在模块启动时，调用该方法注册该UIAction。

注册UI Action后，还需要给它映射或绑定一个回调方法。可以在RegisterCommands方法执行后，通过UICommandList对其映射绑定。

```
FCustomToolCommands::Register();
CustomToolCmdList = MakeShareable(new FUICommandList);

CustomToolCmdList->MapAction(
    FCustomToolCommands::Get().NewUIAction,
    FExecuteAction::CreateRaw(this, &EditorModule::ReimportLocTextFile),
    FCanExecuteAction()
);
```

3. 映射按钮Action UI

4. 注册按钮或菜单

//绑定按钮或菜单创建的方法到ToolMenus的RegisterStartupCallback回调方法上，可以在编辑器启动后回调到注册新菜单的方法。

```
UToolMenus::RegisterStartupCallback(FSimpleMulticastDelegate::FDelegate::CreateRaw(this, &EditorModule::RegisterMenus));
```

//注册新菜单

```
void FEditorModule::RegisterMenus()
{
    //通过ExtendMenu方法找到或注册一个ToolMenu
    UToolMenu* ToolbarMenu = UToolMenus::Get()-
>ExtendMenu("LevelEditor.LevelEditorToolBar");
    {
        //在ToolMenu中找到或添加一个指定Section
        FToolMenuSection& Section = ToolbarMenu->FindOrAddSection("Settings");
        {
            FToolMenuEntry& Entry =

            Section.AddEntry(FToolMenuEntry::InitToolBarButton(FCustomToolCommands::Get().NewUIAction));
            Entry.SetCommandList(CustomToolCmdList);
```

```
    }  
    }  
}
```

5. 注销

二、自定义资源

UAsset

- UBlueprint
- UTexture
- UMaterial
-

UFactory

创建新资产的入口类，FactoryCreateNew方法返回创建的资产

FAssetTypeActions_Base

资产描述类，OpenAssetEditor方法可以执行打开一个编辑器界面。然后把该资产对象传给编辑器。

Action类需要通过AssetTools模块的RegisterAssetTypeActions方法注册。

三、自定义编辑器

FAssetEditorToolkit

该方法定义一个编辑器界面，编辑器界面包含多个Tab，所以需要通过实现该类的RegisterTabSpawners方法将Tabs注册。

可以自定义一个Init方法，然后通过这个方法调用InitAssetEditor，通过该方法打开编辑器窗口。

图表编辑器

UEdGraph

UEdGraphSchema

UEdGraphNode->UK2Node

节点类。如果想给节点添加针脚，可以在AllocateDefaultPins中通过调用CreatePin方法生成针脚。

UEdGraphPin