

# 资产数据获取

## FAssetRegistryModule

相关模块（类）：FAssetRegistryModule

```
//通过AssetRegistryModule获取资产数据
FAssetRegistryModule& AssetRegistryModule =
FModuleManager::LoadModuleChecked<FAssetRegistryModule>("AssetRegistry");
//创建筛选条件
FARFilter AssetFilter;
AssetFilter.PackagePaths.Add("/Game");
AssetFilter.bRecursivePaths = true;
//通过筛选条件拿到数据
TArray<FAssetData> AssetData;
AssetRegistryModule.Get().GetAssets(AssetFilter, AssetData);
```

## 获取资产引用

```
TArray<FName> DependenciesArray;
AssetRegistryModule.GetDependencies(*PackageName, DependenciesArray,
EAssetRegistryDependencyType::Packages);
```

## UObjectLirbary

另一种方法：通过UObjectLirbary类

```
UObjectLirbary* ObjectLibrary;
ObjectLibrary = UObjectLibrary::CreateLibrary(UObject::StaticClass(), false,
false);
//加载指定路径下资产
ObjectLibrary->LoadAssetDataFromPath(TEXT("/Game/Resource/UI/Texture/MenuTex"));
//获取加载资产的AssetData
TArray<FAssetData> TextureData;
ObjectLibrary->GetAssetDataList(TextureData);
```

## 编辑器下资产保存

FEditorFileUtils

通过命令保存 InternalSavePackage

```
⌘ just save the package
(LogFileHelpers", ELogVerbosity::Log, FString::Printf(TEXT("Saving Package: %s"), *PackageName));
= GEngine->Exec( NULL, *FString::Printf( TEXT("OBJ SAVEPACKAGE PACKAGE=\"%s\" FILE=\"%s\" SILENT=true"), *Package

ider& SourceControlProvider = ISourceControlModule::Get().GetProvider();
```

```
FAssetData TempAssetData;  
TArray<FPackage*> PackageToSave;  
PackageToSave.Add(TempAssetData.GetPackage());  
  
UEditorLoadingAndSavingUtils::SavePackages(PackageToSave, false);
```

---

## 资产重新导入

---

相关类：FReimportManager（UnrealEd/Public/Editor.h文件下）

```
FReimportManager::Instance()->Reimport(Obj, true);
```

---

## 获取资产引用关系

---

AssetRegistryModule的GetDependencies方法

---