

Asset Manager

如果我们想通过一个资产文件管理一组资源的加载和卸载，这个资产文件首选的类型就是UPrimaryAssetData。因为UPrimaryAssetData继承自UDataAsset，通过创建DataAsset资产，可以将一个UObject对象实例化为资产文件。通过这个资产文件引用一组资源，然后通过另外一个类AssetManager对PrimaryAssetData进行管理，即可实现一整套的资源加载和卸载的管理系统。

例如：一个PrimaryAssetData资产包含一件武器的相关资源的硬引用关系，其中包含武器的蓝图（模型，材质），缩略图等。然后通过AssetManager加载PrimaryAssetData引用的这组资产。

PrimaryAssetData

通常，在AssetManager中我们不会直接通过使用PrimaryDataAsset来加载它引用的资源，而是为PrimaryAssetData指定一个PrimaryAssetId，然后通过这个Id来获取和加载对应的引用资源。可以通过重写UObject的方法GetPrimaryAssetId来为PrimaryAssetData指定一个PrimaryAssetId。

PrimaryAssetId

PrimaryAssetId由PrimaryType和PrimaryName组成。通常，根据需求自定义PrimaryAsset的常用类型，而PrimaryName可以使用资产名。

```
virtual FPrimaryAssetId GetPrimaryAssetId() const override
{
    return FPrimaryAssetId(PrimaryType, GetFName());
}
```

AssetManager

PrimaryAssetLabel