

# Unity资源管理系统

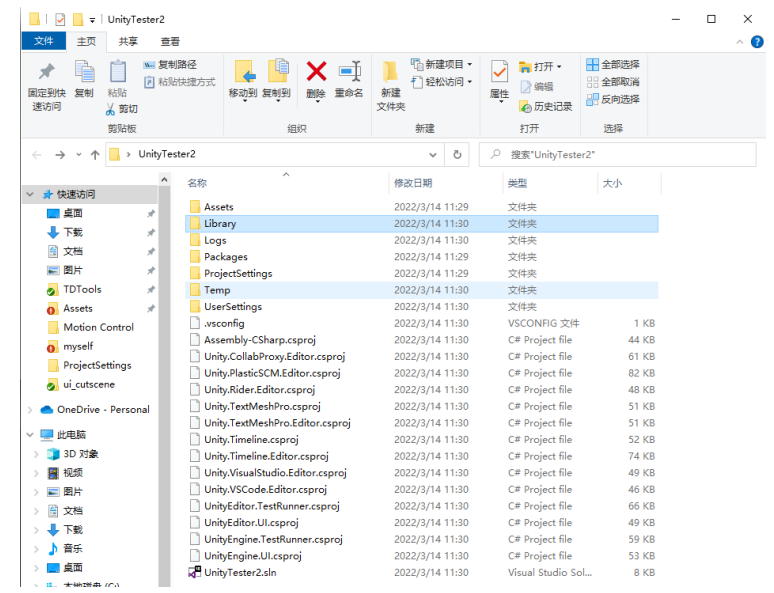
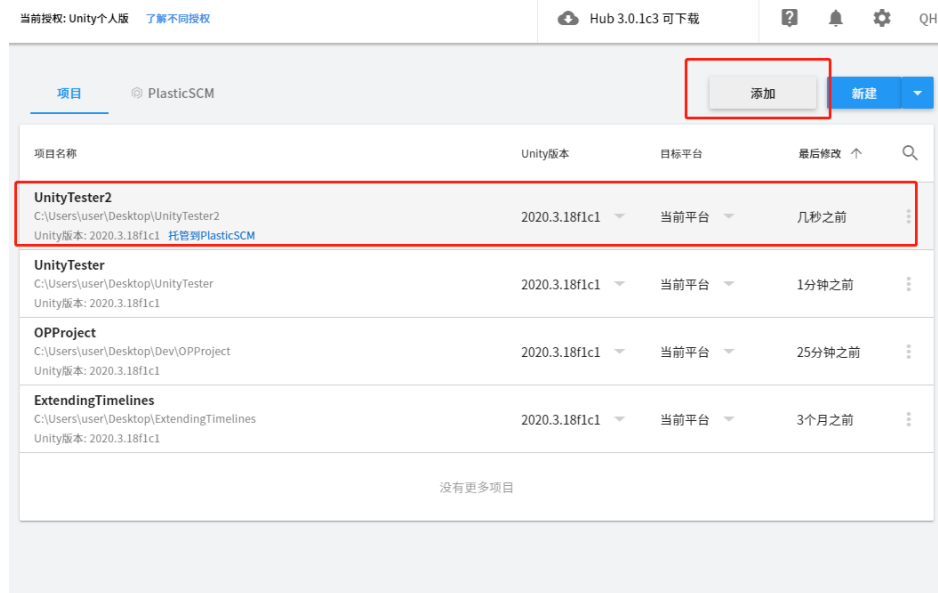
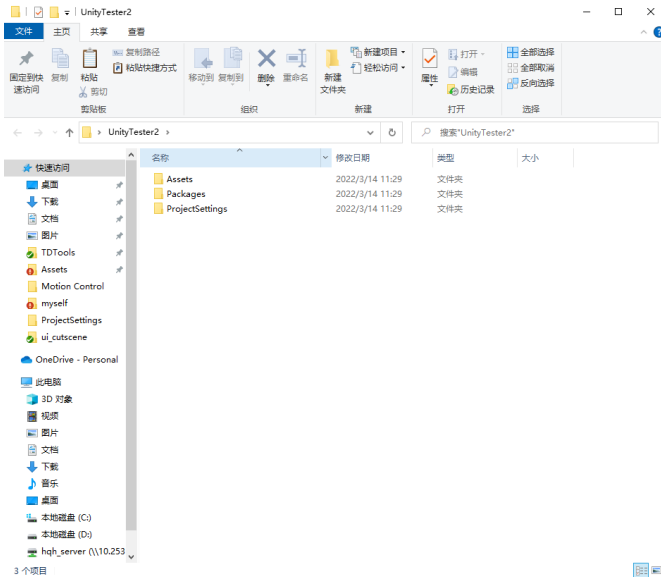
# Unity中的文件夹

- 当我们创建一个空项目时，Unity会自动为我们创建几个文件夹
  - Assets: 资源文件夹，所有原始资源都必须放在此文件夹中才能参与编译，包括场景、代码、配置、库、动画、脚本等等
  - Library: Unity编译后产生的文件，用于cache（增量编译）
  - Logs: 日志
  - Packages: 记录已安装的package信息
  - ProjectSettings: Unity中的各项设定（可修改）
  - Temp: 临时文件
  - UserSettings: 用户设定，如页面布局

名称	修改日期	类型	大小
Assets	2022/3/14 11:04	文件夹	
Library	2022/3/14 11:04	文件夹	
Logs	2022/3/14 11:04	文件夹	
Packages	2022/3/14 11:04	文件夹	
ProjectSettings	2022/3/14 11:04	文件夹	
Temp	2022/3/14 11:04	文件夹	
UserSettings	2022/3/14 11:04	文件夹	
.vsconfig	2022/3/14 11:04	VSCONFIG 文件	1 KB
Unity.CollabProxy.Editor.csproj	2022/3/14 11:04	C# Project file	61 KB
Unity.PlasticSCM.Editor.csproj	2022/3/14 11:04	C# Project file	82 KB
Unity.Rider.Editor.csproj	2022/3/14 11:04	C# Project file	48 KB
Unity.TextMeshPro.csproj	2022/3/14 11:04	C# Project file	51 KB
Unity.TextMeshPro.Editor.csproj	2022/3/14 11:04	C# Project file	51 KB
Unity.Timeline.csproj	2022/3/14 11:04	C# Project file	52 KB
Unity.Timeline.Editor.csproj	2022/3/14 11:04	C# Project file	74 KB
Unity.VisualStudio.Editor.csproj	2022/3/14 11:04	C# Project file	49 KB
Unity.VSCode.Editor.csproj	2022/3/14 11:04	C# Project file	46 KB
UnityEditor.TestRunner.csproj	2022/3/14 11:04	C# Project file	66 KB
UnityEditor.UI.csproj	2022/3/14 11:04	C# Project file	49 KB
UnityEngine.TestRunner.csproj	2022/3/14 11:04	C# Project file	59 KB
UnityEngine.UI.csproj	2022/3/14 11:04	C# Project file	53 KB
UnityTester.sln	2022/3/14 11:04	Visual Studio Sol...	7 KB

# 可删除的文件夹

- 如果需要项目进行迁移，可以将除了Assets、Packages和ProjectSettings之外的文件夹删除，下次Unity就会根据这三个文件夹的信息自动编译生成其他文件夹



# AssetBundle

- AssetBundle是一类资源的集合，对Assets进行有序地归档
  - 它是针对平台的
  - 它不包含代码（但是可以包含序列化文件）
  - 它支持在运行时加载
  - 它表达了Assets之间的依赖
- 将AssetBundle与代码分隔是为了更加高效地索引、使用Assets，提高运行时加载资源的效率  
(你是否发现了BundleRes这个文件夹下不包含代码文件？)
- 然而，使用AssetBundle需要对资源打包进行合理规划
  - 为此，Unity推出了Addressable Assets System，但似乎没有维护了……

# Asset/Object

- 简单来说，Asset就是存储在Assets文件夹中的文件，它仅仅指资源本身，包含Texture、Model、Audio Clip、Materials、Fbx等等
  - Native: 像Material这种Asset直接存储了Unity调用所需的数据，使用时无需处理
  - Non-native: 像Fbx这种Asset在使用时首先需要被Unity处理成可以运行的格式
- Object则是针对Unity运行时而言，在运行时，Unity会把Asset**实例化**为Object，从而使用，比如mesh、sprite、AudioClip等等
- ScriptableObject和MonoBehaviour是最重要的两类Object
- Asset与Object是一对多的关系，一般来说，一个Asset文件（比如prefab）包含多个Object

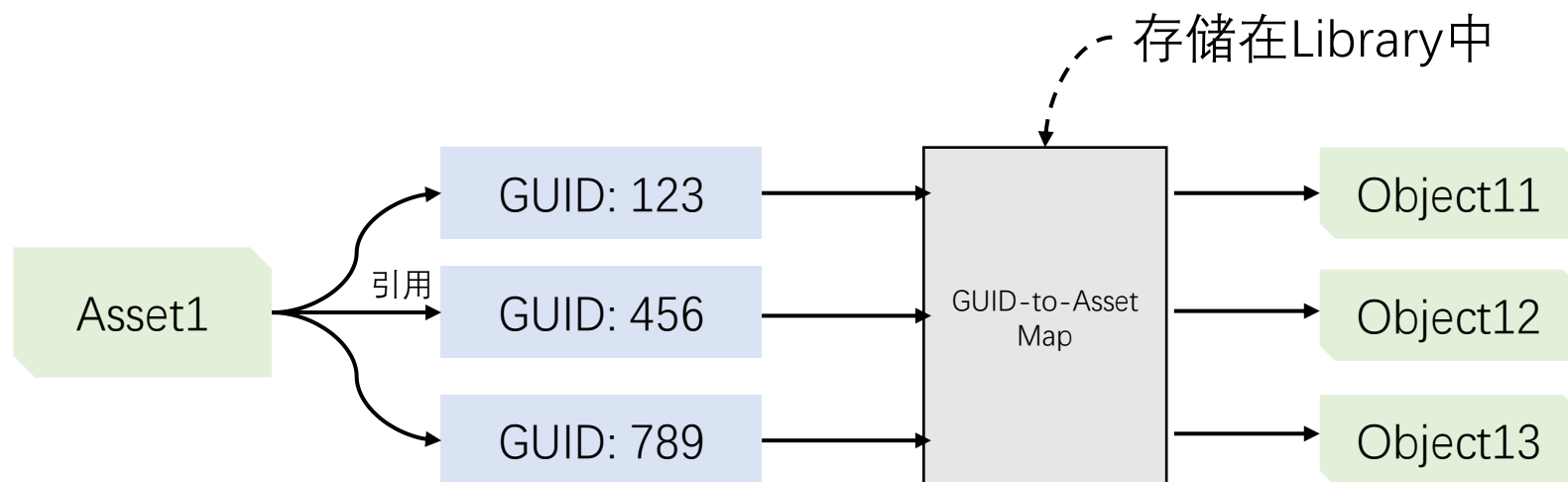
# Inter-Object references

- Object之间有相互引用的关系，可以在同一个Asset内，也可能通过另外的Asset导入
- 引用通过两个ID实现：File GUID和Local ID
  - File GUID与**Asset**绑定，保存在.meta文件中
  - Local ID标识了**Object**在Asset中的引用，一般保存在Asset文件中
- 每个.meta文件实际上保存了两个关键信息：
  - GUID，标识了每个独特的Asset
  - ImportSettings信息，即每个包含的Object导入时的设置是怎样的
- .meta文件的注意点：
  - 第一次导入自动生成
  - 在Project窗口对Asset的修改Unity会自动修改对应的.meta文件（移动、删除等）
  - 在其他文件系统（如Windows/Mac文件夹目录）中修改Asset不会自动修改.meta，需要手动调整

```
Monster_Alvida_attack01.FBX.meta X
C:\Users\user\Desktop\Dev\OPProject\Assets\Creatures\
wenghaowen, 7天前 | 3 authors (mengmingheng and others)
1  fileFormatVersion: 2
2  guid: 738e8981a4141414a9e9edc602359797
3  ModelImporter:
4    serializedVersion: 20200
5    internalIDToNameTable:
6    - first:
7      74: 7781131826771658959
8      second: Monster_Alvida_attack01
9    externalObjects: {}
10   materials:
11   materialImportMode: 2 wenghaowen, 上
12   materialName: 0
13   materialSearch: 1
14   materialLocation: 1
15   animations:
16   legacyGenerateAnimations: 4
17   bakeSimulation: 0
18   resampleCurves: 1
19   optimizeGameObjects: 0
20   motionNodeName:
21   rigImportErrors:
22   rigImportWarnings:
23   animationImportErrors:
24   animationImportWarnings:
25   animationRetargetingWarnings:
26   animationDoRetargetingWarnings: 0
27   importAnimatedCustomProperties: 0
28   importConstraints: 0
29   animationCompression: 1
30   animationRotationError: 0.5
31   animationPositionError: 0.5
32   animationScaleError: 0.5
33   animationWrapMode: 0
34   extraExposedTransformPaths: []
35   extraUserProperties: []
36   clipAnimations: []
37   isReadable: 0
38   meshes:
39   LODScreenPercentages: []
40   globalScale: 1
41   meshCompression: 0
42   addColliders: 0
43   useSRGBMaterialColor: 1
44   sortHierarchyByName: 1
```

# Why File GUIDs and Local IDs

- 为了提供灵活、且跨平台的资源管理系统
  - File GUID是具体资源的抽象位置，只要与资源绑定，那么资源在文件夹中的位置就不再重要；当GUID与对应资源的绑定丢失时，所有对该资源的引用也会丢失
  - Local ID为Asset中包含的每个Object做出了区分，对同一个Asset包含的多个Object，它们共享同一个GUID
  - Unity维护了一个GUID到资源的Map，当有引用关系时，会首先通过GUID检索到对应的Asset文件，再通过Local ID找到具体的Object



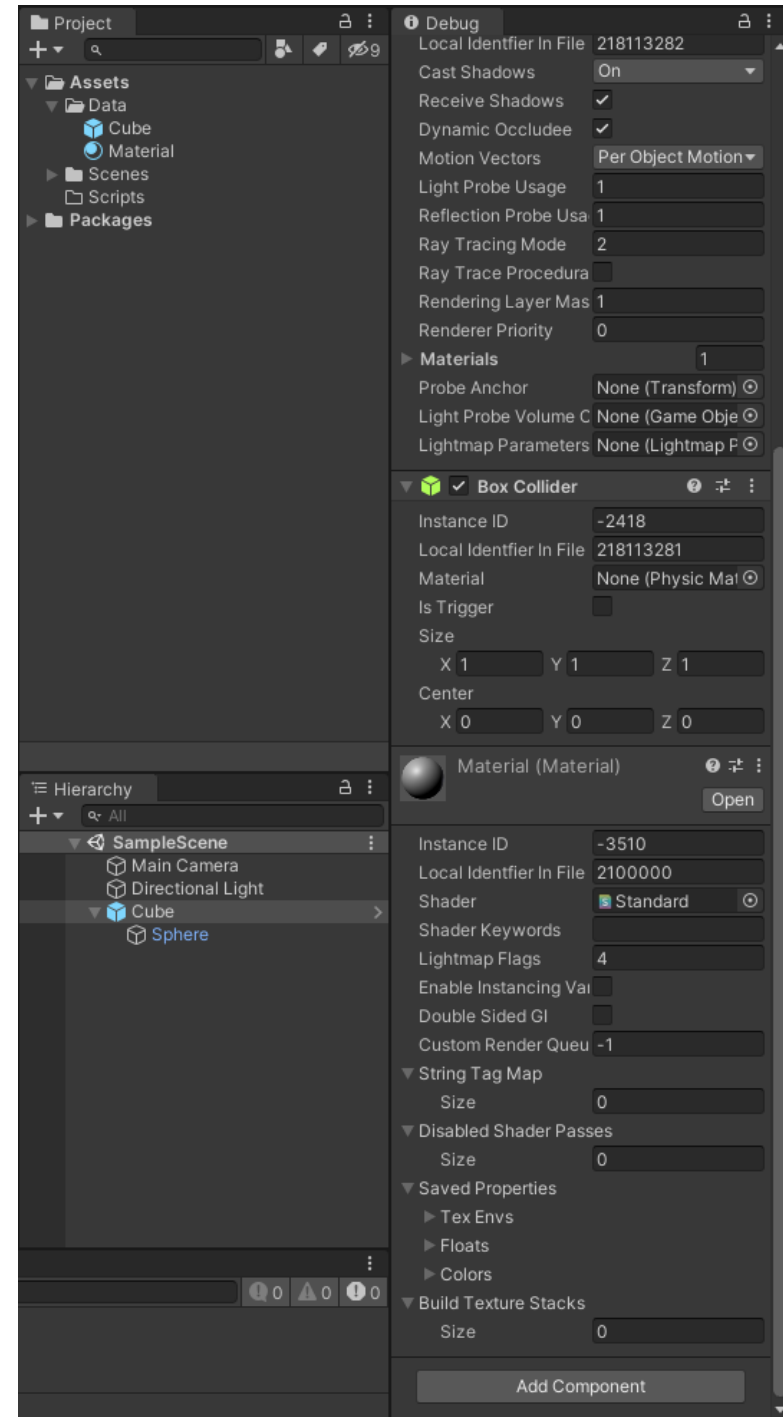
# Instance ID

- 运行时，每次都检索GUID和Local ID会产生效率问题
- 将GUID和Local ID转化为一个单独的Instance ID并把对应的Object增加到cache中
  - 是一个递增的整数
  - 维护了与GUID和Local ID的关系
  - 维护了对应的Object



# Example

- 在Scene中创建了一个Cube，并将一个Sphere作为它的child，再新建一个material挂在Cube上，最后把Cube创建为prefab



# Example

- 首先打开Material.meta文件查看
- 有GUID和ImportSettings信息
  - 可以通过AssetDatabase.GUIDToAssetPath和AssetDatabase.AssetPathToGUID对GUID和资源相互转换
  - mainObjectFileID则是它的Local ID: 2100000
- 再打开Cube.prefab.mata文件查看，同样包含了自身的GUID

Cube.prefab	2022/3/14 15:48	PREFAB 文件	6 KB
Cube.prefab.meta	2022/3/14 15:48	META 文件	1 KB
Material	2022/3/14 15:47	Microsoft Acces...	3 KB
Material.mat.meta	2022/3/14 15:47	META 文件	1 KB

```
Material.mat.meta X
C: > Users > user > Desktop > UnityTester > Assets > Data > Material.mat.meta
1 fileFormatVersion: 2↓
2 guid: 4914be0dea8931f429956c7248cf4db7↓
3 NativeFormatImporter:↓
4   externalObjects: {}↓
5   mainObjectFileID: 2100000↓
6   userData: ↓
7   assetBundleName: ↓
8   assetBundleVariant: ↓
```

```
Cube.prefab.meta X
C: > Users > user > Desktop > UnityTester > Assets > Data > Cube.prefab.meta
1 fileFormatVersion: 2↓
2 guid: ff9cf0d4ba241c541add3823fe98af0c↓
3 PrefabImporter:↓
4   externalObjects: {}↓
5   userData: ↓
6   assetBundleName: ↓
7   assetBundleVariant: ↓
8
```

# Example

• 现在打开Cube.prefab, 有如下组成部分:

- GameObject
  - Transform
  - MeshFilter
  - MeshRenderer
  - SphereCollider
- 子物体Sphere相关

- GameObject
  - Transform
  - MeshFilter
  - MeshRenderer
  - BoxCollider
- 自身Cube相关
- Class ID

```
Cube.prefab X
C: > Users > user > Desktop > UnityTester > Assets > Data > Cube.prefab
1 %YAML 1.1↓
2 %TAG !u! tag:unity3d.com,2011:↓
3 --- !u!1 &4317575759400118404↓
4 > GameObject:↓...
22 --- !u!4 &4317575759400118405↓
23 > Transform:↓...
36 --- !u!33 &4317575759400118648↓
37 > MeshFilter:↓...
44 --- !u!23 &4317575759400118407↓
45 > MeshRenderer:↓...
85 --- !u!135 &4317575759400118406↓
86 > SphereCollider:↓...
98 --- !u!1 &4317575759452987768↓
99 > GameObject:↓...
117 --- !u!4 &4317575759452987772↓
118 > Transform:↓...
132 --- !u!33 &4317575759452987771↓
133 > MeshFilter:↓...
140 --- !u!23 &4317575759452987770↓
141 > MeshRenderer:↓...
181 --- !u!65 &4317575759452987769↓
182 > BoxCollider:↓...
194
```

# Example

Sphere GameObject本身的LID

```
--- !u!1 &4317575759400118404↓
GameObject:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  serializedVersion: 6↓
  m_Component:↓
    - component: {fileID: 4317575759400118405}↓
    - component: {fileID: 4317575759400118648}↓
    - component: {fileID: 4317575759400118407}↓
    - component: {fileID: 4317575759400118406}↓
  m_Layer: 0↓
  m_Name: Sphere↓
  m_TagString: Untagged↓
  m_Icon: {fileID: 0}↓
  m_NavMeshLayer: 0↓
  m_StaticEditorFlags: 0↓
  m_IsActive: 1↓
--- !u!4 &4317575759400118405↓
Transform:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  m_GameObject: {fileID: 4317575759400118404}↓
  m_LocalRotation: {x: -0, y: -0, z: -0, w: 1}↓
  m_LocalPosition: {x: 0, y: 0, z: 0}↓
  m_LocalScale: {x: 1, y: 1, z: 1}↓
  m_Children: []↓
  m_Father: {fileID: 4317575759400118404}↓
  m_RootOrder: 0↓
  m_LocalEulerAnglesHint: {x: 0, y: 0, z: 0}↓
```

MeshFilter的LID

```
--- !u!33 &4317575759400118648↓
MeshFilter:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  m_GameObject: {fileID: 4317575759400118404}↓
  m_Mesh: {fileID: 10207, guid: 0000000000000000e000000000000000, type: 0}↓
```

Sphere有这些Component

Sphere Transform的LID

Transform所属的  
GameObject的LID, 也就是  
Sphere的LID

父级Transform的LID

MeshRenderer的LID

```
--- !u!2 &4317575759400118407↓
MeshRenderer:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  m_GameObject: {fileID: 4317575759400118404}↓
  m_Enabled: 1↓
  m_CastShadows: 1↓
  m_ReceiveShadows: 1↓
  m_DynamicOccludee: 1↓
  m_MotionVectors: 1↓
  m_LightProbeUsage: 1↓
  m_ReflectionProbeUsage: 1↓
  m_RayTracingMode: 2↓
  m_RayTraceProcedural: 0↓
  m_RenderingLayerMask: 1↓
  m_RendererPriority: 0↓
  m_Materials:↓
    - {fileID: 10303, guid: 0000000000000000f000000000000000, type: 0}↓
  m_StaticBatchInfo:↓
    firstSubMesh: 0↓
    subMeshCount: 0↓
  m_StaticBatchRoot: {fileID: 0}↓
  m_ProbeAnchor: {fileID: 0}↓
  m_LightProbeVolumeOverride: {fileID: 0}↓
  m_ScaleInLightmap: 1↓
  m_ReceiveGI: 1↓
  m_PreserveUVs: 0↓
  m_IgnoreNormalsForChartDetection: 0↓
  m_ImportantGI: 0↓
  m_StitchLightmapSeams: 1↓
  m_SelectedEditorRenderState: 3↓
  m_MinimumChartSize: 4↓
  m_AutoUVMaxDistance: 0.5↓
  m_AutoUVMaxAngle: 89↓
  m_LightmapParameters: {fileID: 0}↓
  m_SortingLayerID: 0↓
  m_SortingLayer: 0↓
  m_SortingOrder: 0↓
  m_AdditionalVertexStreams: {fileID: 0}↓
```

SphereCollider的LID

```
--- !u!135 &4317575759400118406↓
SphereCollider:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  m_GameObject: {fileID: 4317575759400118404}↓
  m_Material: {fileID: 0}↓
  m_IsTrigger: 0↓
  m_Enabled: 1↓
  serializedVersion: 2↓
  m_Radius: 0.5↓
  m_Center: {x: 0, y: 0, z: 0}↓
```

# Example

Cube GameObject本身的LID

```
--- !u!1 &4317575759452987768↓
  GameObject:↓
    m_ObjectHideFlags: 0↓
    m_CorrespondingSourceObject: {fileID: 0}↓
    m_PrefabInstance: {fileID: 0}↓
    m_PrefabAsset: {fileID: 0}↓
    serializedVersion: 6↓
    m_Component:↓
      - component: {fileID: 4317575759452987772}↓
      - component: {fileID: 4317575759452987771}↓
      - component: {fileID: 4317575759452987770}↓
      - component: {fileID: 4317575759452987769}↓
    m_Layer: 0↓
    m_Name: Cube↓
    m_TagString: Untagged↓
    m_Icon: {fileID: 0}↓
    m_NavMeshLayer: 0↓
    m_StaticEditorFlags: 0↓
    m_IsActive: 1↓
--- !u!4 &4317575759452987772↓
  Transform:↓
    m_ObjectHideFlags: 0↓
    m_CorrespondingSourceObject: {fileID: 0}↓
    m_PrefabInstance: {fileID: 0}↓
    m_PrefabAsset: {fileID: 0}↓
    m_GameObject: {fileID: 4317575759452987768}↓
    m_LocalRotation: {x: 0, y: 0, z: 0, w: 1}↓
    m_LocalPosition: {x: 0, y: 0, z: 0}↓
    m_LocalScale: {x: 1, y: 1, z: 1}↓
    m_Children:↓
      - {fileID: 4317575759400118405}↓
    m_Father: {fileID: 0}↓
    m_RootOrder: 0↓
    m_LocalEulerAnglesHint: {x: 0, y: 0, z: 0}↓
```

有这些Component

Cube Transform的LID

子级Transform的LID

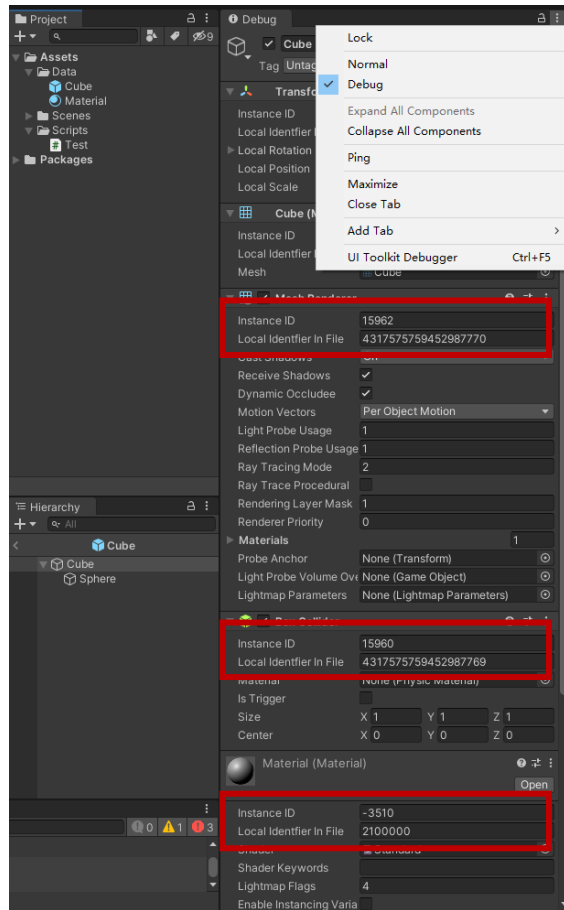
Mesh Renderer的LID

```
--- !u!23 &4317575759452987770↓
  MeshRenderer:↓
    m_ObjectHideFlags: 0↓
    m_CorrespondingSourceObject: {fileID: 0}↓
    m_PrefabInstance: {fileID: 0}↓
    m_PrefabAsset: {fileID: 0}↓
    m_GameObject: {fileID: 4317575759452987768}↓
    m_Enabled: 1↓
    m_CastShadows: 1↓
    m_ReceiveShadows: 1↓
    m_DynamicOccludee: 1↓
    m_MotionVectors: 1↓
    m_LightProbeUsage: 1↓
    m_ReflectionProbeUsage: 1↓
    m_RayTracingMode: 2↓
    m_RayTraceProcedural: 0↓
    m_RenderingLayerMask: 1↓
    m_RendererPriority: 0↓
    m_Materials:↓
      - {fileID: 2100000, guid: 4914be0dea8931f429956c7248cf4db7, type: 2}↓
    m_StaticBatchInfo:↓
      firstSubMesh: 0↓
      subMeshCount: 0↓
    m_StaticBatchRoot: {fileID: 0}↓
    m_ProbeAnchor: {fileID: 0}↓
    m_LightProbeVolumeOverride: {fileID: 0}↓
    m_ScaleInLightmap: 1↓
    m_ReceiveGI: 1↓
    m_PreserveUVs: 0↓
    m_IgnoreNormalsForChartDetection: 0↓
    m_ImportantGI: 0↓
    m_StitchLightmapSeams: 1↓
    m_SelectedEditorRenderState: 3↓
    m_MinimumChartSize: 4↓
    m_AutoUVMaxDistance: 0.5↓
    m_AutoUVMaxAngle: 89↓
    m_LightmapParameters: {fileID: 0}↓
    m_SortingLayerID: 0↓
    m_SortingLayer: 0↓
    m_SortingOrder: 0↓
    m_AdditionalVertexStreams: {fileID: 0}↓
```

引用Material的LID和GUID


# Example

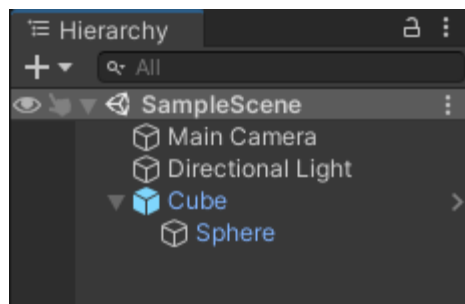
- 每个Component的Local ID都能通过Inspector面板的Debug模式看到



# Example

- 下面再打开SampleScene.unity.meta文件查看
- 还是包含了该场景的GUID
- 当打开SampleScene.unity时，会发现有很多部分
  - Local ID为1、2、3、4的部分是场景的内置Setting

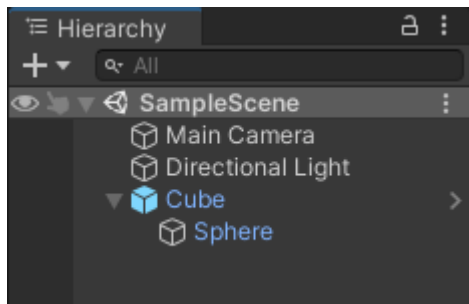
```
C: > Users > user > Desktop > UnityTester > Assets > Scenes >   
1 fileFormatVersion: 2↓  
2 guid: 9fc0d4010bbf28b4594072e72b8655ab↓  
3 DefaultImporter:↓  
4 externalObjects: {}↓  
5 userData: ↓  
6 assetBundleName: ↓  
7 assetBundleVariant: ↓  
8
```



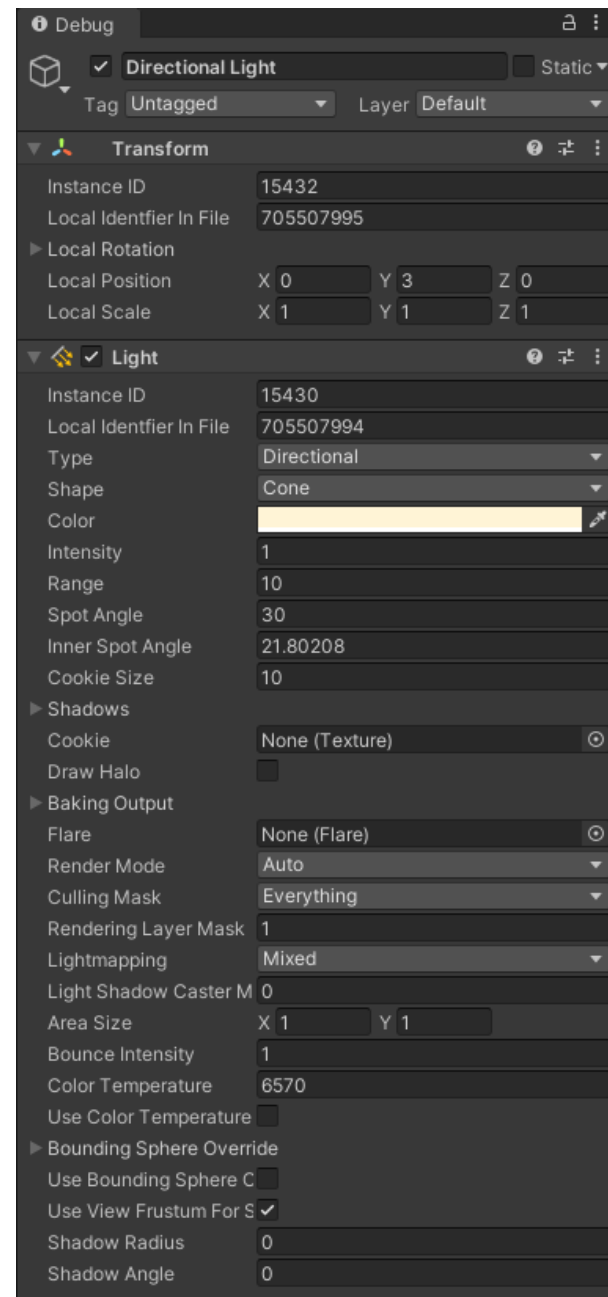
```
%YAML 1.1↓  
%TAG !u! tag:unity3d.com,2011:↓  
--- !u!29 &1↓  
> OcclusionCullingSettings:↓...  
--- !u!104 &2↓  
> RenderSettings:↓...  
--- !u!157 &3↓  
> LightmapSettings:↓...  
--- !u!196 &4↓  
> NavMeshSettings:↓...  
--- !u!1 &705507993↓  
> GameObject:↓...  
--- !u!108 &705507994↓  
> Light:↓...  
--- !u!4 &705507995↓  
> Transform:↓...  
--- !u!1 &963194225↓  
> GameObject:↓...  
--- !u!81 &963194226↓  
> AudioListener:↓...  
--- !u!20 &963194227↓  
> Camera:↓...  
--- !u!4 &963194228↓  
> Transform:↓...  
--- !u!1001 &4317575759369094264↓  
> PrefabInstance:↓...
```

# Example

- 连续三个Local ID都是 Directional Light及其内部组件



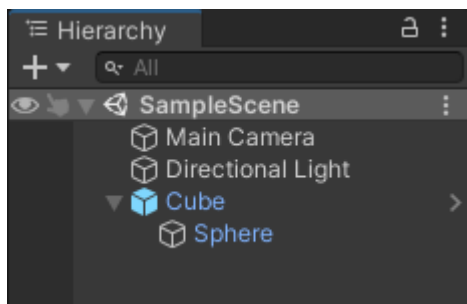
```
--- !u!1 &705507993↓
GameObject:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  serializedVersion: 6↓
  m_Component:↓
    - component: {fileID: 705507995}↓
    - component: {fileID: 705507994}↓
  m_Layer: 0↓
  m_Name: Directional Light↓
  m_TagString: Untagged↓
  m_Icon: {fileID: 0}↓
  m_NavMeshLayer: 0↓
  m_StaticEditorFlags: 0↓
  m_IsActive: 1↓
--- !u!108 &705507994↓
Light:↓...
--- !u!4 &705507995↓
Transform:↓...
```



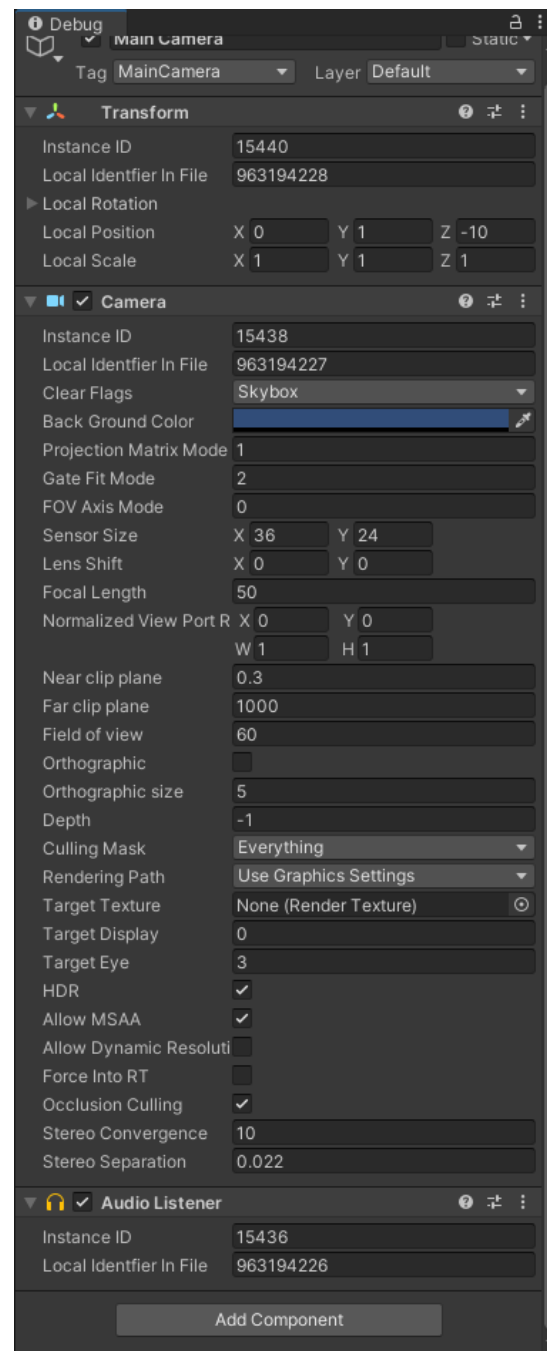


# Example

- 接下来的4个Local ID是Camera及其内部组件



```
--- !u!1 &963194225↓
▼ GameObject:↓
  m_ObjectHideFlags: 0↓
  m_CorrespondingSourceObject: {fileID: 0}↓
  m_PrefabInstance: {fileID: 0}↓
  m_PrefabAsset: {fileID: 0}↓
  serializedVersion: 6↓
  m_Component:↓
    - component: {fileID: 963194228}↓
    - component: {fileID: 963194227}↓
    - component: {fileID: 963194226}↓
  m_Layer: 0↓
  m_Name: Main Camera↓
  m_TagString: MainCamera↓
  m_Icon: {fileID: 0}↓
  m_NavMeshLayer: 0↓
  m_StaticEditorFlags: 0↓
  m_IsActive: 1↓
--- !u!81 &963194226↓
> AudioListener:↓...
--- !u!20 &963194227↓
> Camera:↓...
--- !u!4 &963194228↓
> Transform:↓...
```

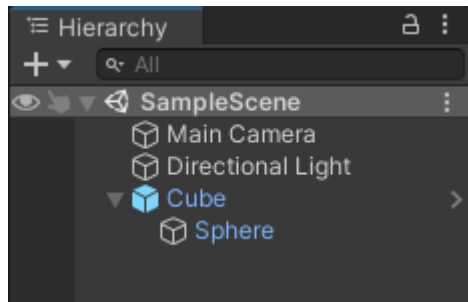


# Example

实例化时对原prefab的所有修改都会呈现在下面

- 最后的PrefabInstance与我们创建的Cube prefab有关，是这个Asset在场景中的实例化信息

- ①：在该场景中prefab的实例化LID
- ②：Cube.prefab的GUID
- ③：Cube.prefab中Cube GameObject的LID
- ④：Cube.prefab中Cube Transform的LID
- ⑤：Cube.prefab的GUID



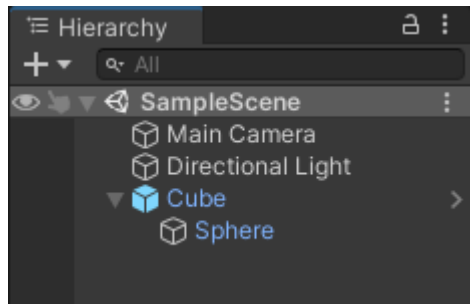
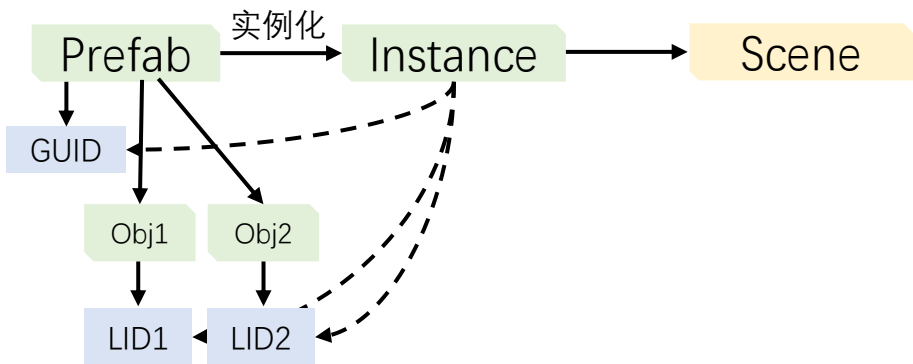
```
--- lu!1001 843175757593690942644
PrefabInstance:
  m_ObjectHideFlags: 0
  serializedVersion: 2
  m_Modification:
    m_TransformParent: {fileID: 0}
    m_Modification:
      - target: {fileID: 4317575759452987768} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_name
        value: Cube
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_RootOrder
        value: 1
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalPosition.x
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalPosition.y
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalPosition.z
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.w
        value: 1
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.x
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.y
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.z
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalEulerAnglesHint.x
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalEulerAnglesHint.y
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalEulerAnglesHint.z
        value: 0
    objectReference: {fileID: 0}
  m_RemovedComponents: []
  m_SourcePrefab: {fileID: 100100000, guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3}
```

# Example

实例化时对原prefab的所有修改都会呈现在下面

- 最后的PrefabInstance与我们创建的Cube prefab有关，是这个Asset在场景中的实例化信息

- ①：在该场景中prefab的实例化LID
- ②：Cube.prefab的GUID
- ③：Cube.prefab中Cube GameObject的LID
- ④：Cube.prefab中Cube Transform的LID
- ⑤：Cube.prefab的GUID



```
--- lu!1001 843175757593690942644
PrefabInstance:
  m_ObjectHideFlags: 0
  serializedVersion: 2
  m_Modification:
    m_TransformParent: {fileID: 0}
    m_Modification:
      - target: {fileID: 4317575759452987768} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_name
        value: Cube
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_RootOrder
        value: 1
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalPosition.x
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalPosition.y
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalPosition.z
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.w
        value: 1
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.x
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.y
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalRotation.z
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalEulerAnglesHint.y
        value: 0
      - target: {fileID: 4317575759452987772} guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3
        propertyPath: m_LocalEulerAnglesHint.z
        value: 0
    objectReference: {fileID: 0}
  m_RemovedComponents: []
  m_SourcePrefab: {fileID: 100100000, guid: ff9cf0d4ba241c541add3823fe98af0c, type: 3}
```

# 一个实际例子

- 需求：找到.anim文件对应的.fbx文件
- 已有.anim文件，通过调用API获取.asset文件，再通过.asset中存储的fbx文件对应的GUID获取fbx文件

```
Monster_Alvida_attack01.asset x Monster_Alvida_attack01.FBX.meta x
C:\Users\user\Desktop\Dev\OPProject\Assets\BundleRes\EditorAnimation\Monster_Alvida\Monster_Alvida
wenghaowen, 7天前 | 2 authors (mengmingheng and others)
1 %YAML 1.1
2 %TAG !u! tag:unity3d.com,2011:
3 --- !u!114 &11400000
4 MonoBehaviour:
5   m_ObjectHideFlags: 0
6   m_CorrespondingSourceObject: {fileID: 0}
7   m_PrefabInstance: {fileID: 0}
8   m_PrefabAsset: {fileID: 0}
9   m_GameObject: {fileID: 0}
10  m_Enabled: 1
11  m_EditorHideFlags: 0
12  m_Script: {fileID: -182810398, guid: a8701fc72a91f814ea6a07ebb465d2b8, type: 3}
13  m_Name: Monster_Alvida_attack01
14  m_EditorClassIdentifier: 
15  clip: {fileID: 7781131826771658959, guid: 738e8981a4141414a9e9edc602359797, type: 3}
16  loadFromClip: 0
```

```
Monster_Alvida_attack01.asset x Monster_Alvida_attack01.FBX.meta x
C:\Users\user\Desktop\Dev\OPProject\Assets\Creatures\Monster_Alvida\Animation\Monster_Alvida
wenghaowen, 7天前 | 3 authors (mengmingheng and others)
1 fileFormatVersion: 2
2 guid: 738e8981a4141414a9e9edc602359797
3 ModelImporter:
4   serializedVersion: 20200
5   internalIDToNameTable:
6     - first:
7       74: 7781131826771658959
8       second: Monster_Alvida_attack01
9   externalObjects: {}
10  materials:
11    materialImportMode: 2
12    materialName: 0
13    materialSearch: 1
14    materialLocation: 1
15  animations:
```

```
if (AssetDatabase.TryGetGUIDAndLocalFileIdentifier(t_clip, out string t_guid, out long t_localid))
{
    string fbxName = AssetDatabase.GUIDToAssetPath(t_guid).Split('/').Last();
    return fbxName;
}
else return "";
```

# 参考文献

- <https://zhuanlan.zhihu.com/p/96709802>
- [https://blog.uwa4d.com/archives/USparkle\\_inf\\_UnityEngine.html](https://blog.uwa4d.com/archives/USparkle_inf_UnityEngine.html)
- <https://docs.unity3d.com/Manual/AssetMetadata.html>
- <https://learn.unity.com/tutorial/assets-resources-and-assetbundles#>
- <https://docs.unity3d.com/Packages/com.unity.addressables@0.3/manual/AddressableAssetsGettingStarted.html>