**ABManagerAPI**

# ABManagerComponent

## 属性

### static IABManagerPool mABManagerPool

* AssetBundle资源包管理器池

### bool mInitOnAwake

* 是否在Awake方法中自动初始化

### string mABDirectoryPath

* AssetBundle资源包目录路径

### string mMainABName

* Main AssetBundle资源包的名称

### IABManager mManager

* AssetBundle资源包管理器

## 方法

### void Init()

* 初始化AssetBundle资源包管理器

# IABManager

## 属性

### bool mIsInit

* 是否有效初始化AssetBundle资源包管理器

### string mMainABName

* Main AssetBundle资源包的名称

### int mABCount

* 当前AssetBundle资源包管理器中AB包的数量

### string[] mABNames

* 当前AssetBundle资源包管理器中AB包的名称合集(不包括Main AssetBundle资源包的名称)

### string mMainABPath

* Main AssetBundle资源包的磁盘路径

## 方法

### bool Init(string dirPath, string mainABName)

* 初始化，根据指示目录路径和Main AssetBundle资源包名称完成该AssetBundle资源包管理器的初始化
* 声明：要求mainABName所指示的Main AssetBundle资源包在dirPath所指示目录下
* dirPath：AssetBundle资源包所在目录路径
* mainABName：Main AssetBundle资源包名称
* 返回值：初始化成功则返回true，否则返回false，若该AssetBundle资源包管理器已初始化，也将返回false

### bool Contains(string abName, string assetName)

* 指定的AssetBundle资源包中是否包含指定名称的资源
* abName：AssetBundle资源包的名称
* assetName：资源的名称
* 返回值：包含则返回true，否则返回false，未进行初始化也会返回false

### string[] GetAllAssetNames(string abName)

* 获取AssetBundle资源包中所有资源的名称
* abName：AssetBundle资源包的名称
* 返回值：资源名称合集，其它情况返回string类型空数组。

### IEnumerator LoadABAsync(string abName, Action<string, IABAsyncResult> listener)

* 异步加载AssetBundle资源包
* abName：待加载的AssetBundle资源包的名称
* listener：异步加载结果监听回调<
* 返回值：协程对象

### bool LoadAB(string abName)

* 同步加载AssetBundle资源包
* 声明1：该加载会同时加载所指示的AssetBundle资源包及其直接和间接依赖的AssetBundle资源包
* abName：AssetBundle资源包的名称
* 返回值：是否加载成功，若AB包已加载或加载成功则返回true，其它情况返回false

### bool UnloadAB(string abName, bool isUnloadAllLoadedObjects)

* 卸载AssetBundle资源包
* abName：待卸载的AssetBundle资源包的名称
* isUnloadAllLoadedObjects：是否卸载所指示的AssetBundle资源包已加载的资源
* 返回值：若资源卸载成功或资源未曾加载则返回true，否则返回false

### void UnloadAllABs(bool isUnloadAllLoadedObjects)

* 卸载所有AssetBundle资源包
* isUnloadAllLoadedObjects：是否卸载AssetBundle资源包已加载的资源

### IEnumerator LoadAssetAsync(string abName, string assetName, Type assetType, Action<string, string, IAssetAsyncResult> listener)

* 异步加载资源
* abName：AssetBundle资源包的名称
* assetName：待加载资源的名称
* assetType：待加载资源的类型
* listener：异步加载结果监听回调
* 返回值：协程对象

### IEnumerator LoadAssetAsync<T>(string abName, string assetName, Action<string, string, IAssetAsyncResult<T>> listener) where T : UnityEngine.Object

* 异步加载资源
* T：待加载资源的类型
* abName：AssetBundle资源包的名称
* assetName：待加载资源的名称
* listener：异步加载结果监听回调
* 返回值：协程对象

### UnityEngine.Object LoadAsset(string abName, string assetName, Type assetType)

* 同步加载资源
* abName：AssetBundle资源包的名称
* assetName：待加载资源的名称
* assetType：待加载资源的类型
* 返回值：若加载资源成功则返回Object类型的资源，否则返回null

### T LoadAsset<T>(string abName, string assetName) where T : UnityEngine.Object

* 同步加载资源
* T：待加载资源的类型
* abName：AssetBundle资源包的名称
* assetName：待加载资源的名称

### bool IsLoaded(string abName)

* 判断指定名称的AssetBundle资源包是否已加载
* abName：AssetBundle资源包的名称
* 返回值：已加载则返回true，否则返回false

# IABManagerPool

## 方法

### IABManager Get()

* 获取AssetBundle资源包管理器
* 返回值：AssetBundle资源包管理器

### void Return(IABManager manager)

* 归还AssetBundle资源包管理器
* manager：AssetBundle资源包管理器

### void Clear()

* 清空池

# ABManagerTool

## 方法

### static string[] GetABNames(string dirPath)

* 获取路径所指示目录下AssetBundle资源包的名称合集
* dirPath：AssetBundle资源包所在目录路径
* 返回值：AssetBundle资源包的名称合集，若不存在则返回长度为0的字符串数组

### static string[] GetABPaths(string dirPath)

* 获取路径所指示目录下AssetBundle资源包的路径合集
* dirPath：AssetBundle资源包所在目录路径
* 返回值：AssetBundle资源包的路径合集，若不存在则返回长度为0的字符串数组

### static string GetABName(string path)

* 获取路径所指示AssetBundle资源包的名称
* path：AssetBundle资源包的路径
* 返回值：AssetBundle资源包的名称，其它情况返回空字符串

### static string GetABPath(string dirPath, string name)

* 获取AssetBundle资源包的路径
* dirPath：AssetBundle资源包所在目录的路径
* name：AssetBundle资源包的名称
* 返回值：AssetBundle资源包的路径，其它情况返回空字符串

# IABAsyncResult

## 属性

### bool mIsDone

* AssetBundle资源包是否已加载完成

### float mProgress

* AssetBundle资源包加载进度

### int mPriority

* 加载优先级

### bool mAllowSceneActivation

* 是否允许在场景准备就绪后立即激活场景

# IAssetResult

## 属性

### bool mIsDone

* 资源是否已加载完成

### float mProgress

* 资源加载进度

### int mPriority

* 加载优先级

### bool mAllowSceneActivation

* 是否允许在场景准备就绪后立即激活场景

# IAssetAsyncResult

## 属性

### UnityEngine.Object mAsset

* 正在加载的资源对象
* 声明：加载完成前访问资源将会停止加载

### UnityEngine.Object[] mAllAssets

* 正在加载的资源对象及子资源
* 声明：加载完成前访问资源将会停止加载

# IAssetAsyncResult<T>

## 属性

### T mAsset

* 正在加载的资源对象
* 声明：加载完成前访问资源将会停止加载

### T[] mAllAssets

* 正在加载的资源对象及子资源
* 声明：加载完成前访问资源将会停止加载