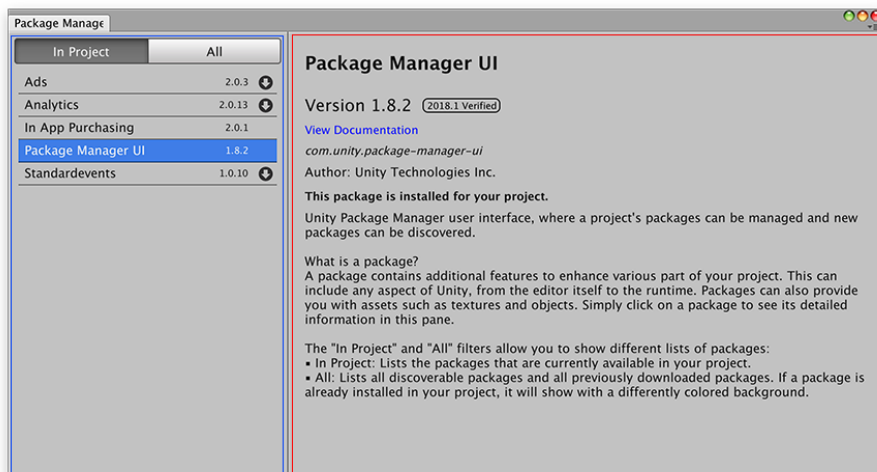


# Unity Package Manager

(注：Unity 2018.1及以后的版本才可以使用Package Manager。)

一个package是一个容器，里面放的是Assets, Shaders, Textures, plug-ins, icons和c# scripts，可以用来增强你项目的各个部分。Unity packages是Asset Store packages的更新(newer)、更紧密集成的版本(more tightly integrated versions)，能够为Unity提供更广泛的增强功能。

使用Unity的包管理器窗口(Package Manager window，在Unity的顶部菜单：**Window > Package Manager**)，可以查看有哪些当前版本的Unity可用的包(packages)，或者查看当前项目已经安装了哪些packages。另外，你也可以使用这个窗口来为项目安装，移除或更新packages。

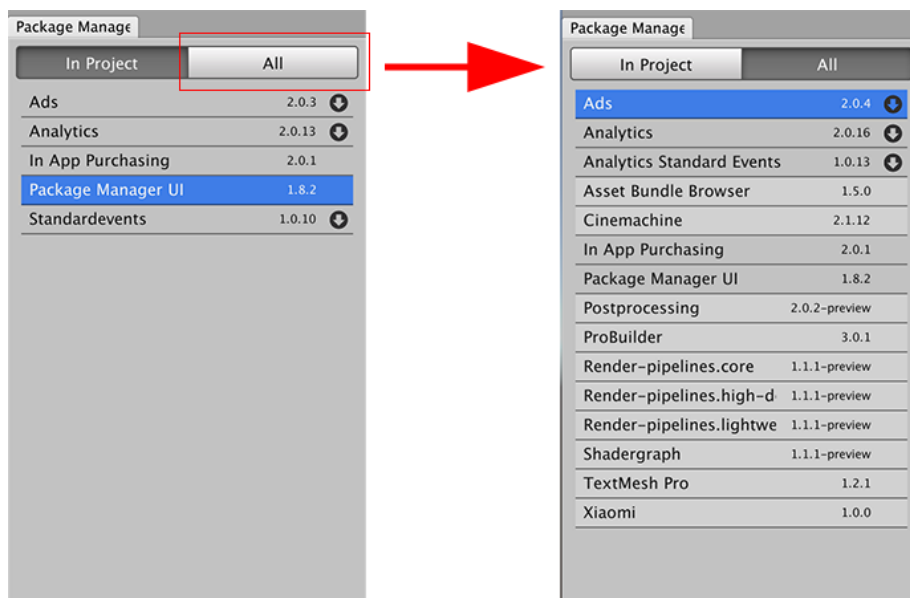


Package Manager窗口左侧显示packages列表，右则显示选中的包的详细信息。

## Finding packages 查找包

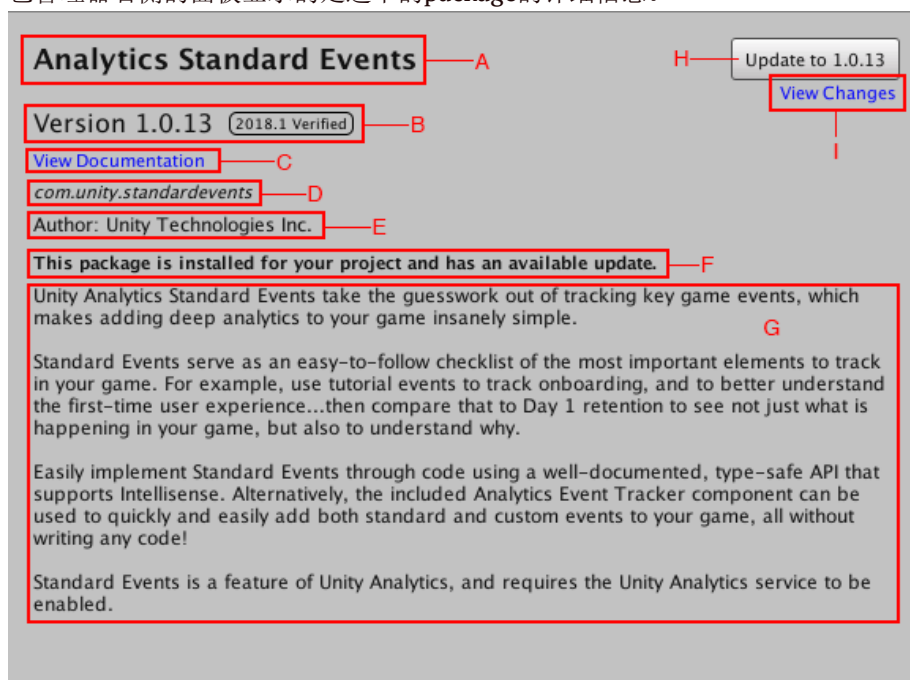
默认，包管理器窗口以**In Project**的模式显示packages列表，这样只有项目中已安装的packages才会出现在列表中。

你可以点击**All**按钮，这样为显示当前版本Unity所有可用的packages。这时左边列表里会显示所有已在包注册表(package registry，见下面)中注册的东西，无论它们是否已在当前项目中安装。



## Viewing package details 查看package详细信息

包管理器右侧的面板显示的是选中的package的详细信息。

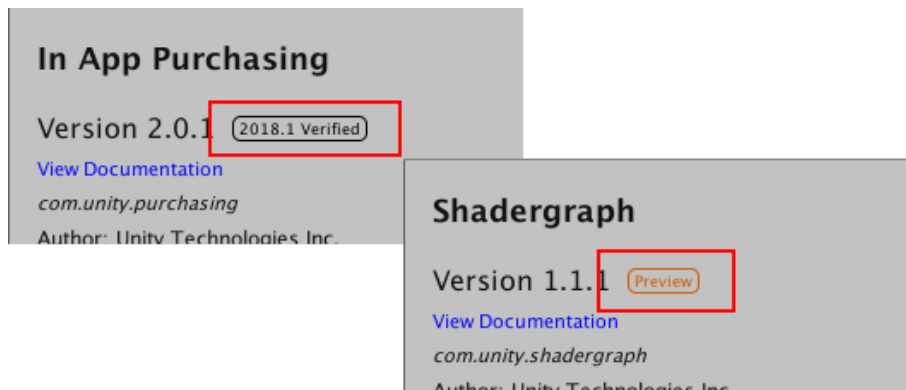


包括以下信息：

- (A) 显示名称 (the display name)
- (B) 版本号 (和 tag，如果有的话)
- (C) package的文档页的超链接
- (D) 以 `com.unity.` 开头的正式包名
- (E) 作者
- (F) 是否已安装，是否可更新
- (G) 简要说明
- (H) package的安装(install), 移除(remove), 或更新(update)按钮
- (I) 查看package变化日志的链接

## Version tags 版本标签

有些package会成版本号(version number)旁边显示版本标签。这些标签(tags)传递关于这个package当前版本的信息。



包管理器使用下面这些值：

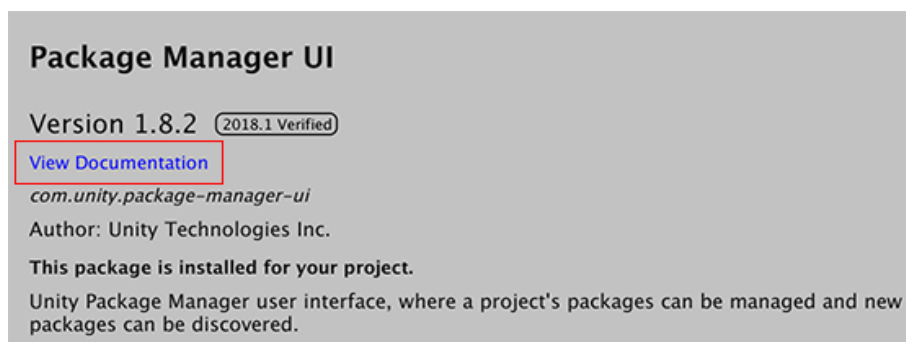
标签 (TAG)	含义(MEANING)
verified	已验证版。Unity的质量保证团队已经正式验证过这个package可以和一个具体版本号的Editor一起工作
preview	预览版。这个package正处于发布周期的早期阶段，可能没有文档，可能也没有被完全验证过。

一个没有package的包被认为是生成安全的(A Package without a tag is considered production quality)。

## Finding more information

通过查看package的文档，你可 查看更多关于一个package的信息。

你可以点击**View Documentation**来查看package的文档。



Package文档会成默认浏览器中打开。

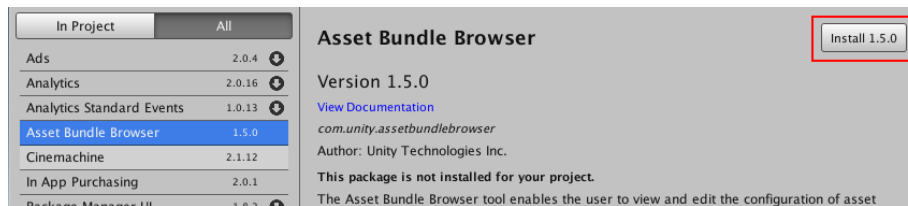
## Accessing the Package Manager window 打开包管理器窗口

通过Package Manager窗口，你可以完成各种各样的任务：

- 安装一个新的package
- 移除一个已安装的package
- 更新一个已安装的package

打开包管理器窗口的方法：在Unity的主菜单栏中依次点**Window > Package Manager**，如下图：

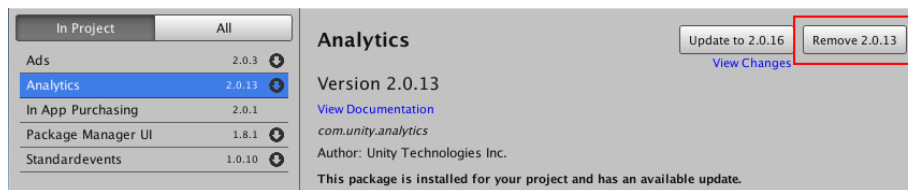
## Installing a new package 安装一个新包



安装一个新包的方法：

1. 打开包管理器窗口，点击**All**按钮。
2. 选择你想安装的**Packages**。对应的package信息会出现在右边的详细信息面板中。
3. 在右侧的详细信息面板中点击**Install X.X.X**按钮。当进度条走完了，新包也就可以使用了。

## Removing an installed package 移除已安装的包



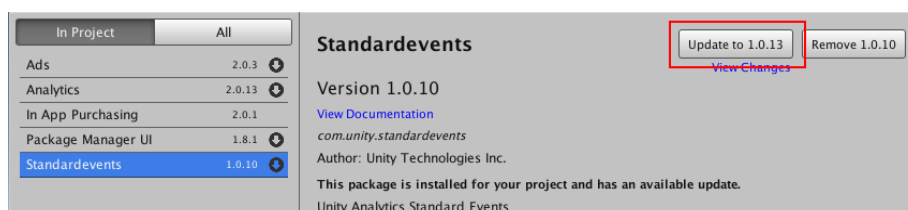
移除一个已安装包的方法：

1. 打开包管理器窗口。
2. 点击**In Project**按钮。
3. 选中你想移除的包。
4. 在右侧详细信息面板中点击**Remove X.X.X**按钮。当进度条走完时，package也就从列表中消失了。

注意：

- 你仅可以移除不被别的packages引用的packages。
- 如果你移除一个包，则任何编辑器下或运行时对应的功能将不再可用。

## Updating an installed package 更新一个已安装的包



你可以在**In Project**或**All**模式下，更新已一个packge：

1. 打开包管理器窗口。任何名字后面有下载箭头的包都有可用的更新。

2. 选中你想要更新的packages。
3. 在详细信息面板中点击Update to X.X.X按钮。

## Advanced package topics 关于package的高级话题

下面讲的是关于package manifest文件的更高级的信息。如果你中是安装、移除或更新packages的话，不需要往下看了。

一个Unity package除了它的内容(Assets, Shaders, Textures, plug-ins, icons和scripts)外，还包含一个包清单文件(package manifest file)。这个manifest文件告诉Unity怎样在包管理器窗口中显示它的信息页 和 怎样去在项目中安装这个package。

此外，它也包含另外几个文件，帮助包的部署，包括: the tests, the samples, the license, the changelog，还有文档。

## Manifests

有两类manifest文件：project manifests (`manifest.json`)和package manifests (`package.json`)。两者都使用JSON语法和包管理器(Package Manager)沟通，前者描述当前项目有哪此可用的包，后者描述当前包里面有什么。

## Project manifests

Project manifests (`manifest.json`)告诉包管理器当前项目可以使哪些包的哪些版本。

Project manifests 支持下面这些值：

KEY	JSON TYPE	DESCRIPTION
dependencies	Object	Package Manger要加载的包列表。

下面是一个`manifest.json`文件的例子：

```
1 {
2     "dependencies": {
3         "com.unity.package-1": "1.0.0",
4         "com.unity.package-2": "2.0.0",
5         "com.unity.package-3": "3.0.0"
6     }
7 }
```

Unity把project manifest文件存放在

`[your_project_root_folder]/Packages/manifest.json`。

## Package manifests

Package manifests (`package.json`)用来决定加载哪个版本的package，在Package Manager Window中显示什么信息。

Package manifests支持以下值:

KEY	JSON TYPE	DESCRIPTION
<code>name</code>	String	正式注册的包名, 遵守以下命名规则: <code>"com.unity.[your package name]"</code> 。例如, <code>"com.unity.resourcemanager"</code>
<code>displayName</code>	String	在包管理器中显示的包名。例如, <code>"ResourceManager"</code>
<code>version</code>	String	包的版本 <code>"MAJOR.MINOR.PATCH"</code> 。Unity的packages遵循 <a href="#">Semantic Versioning</a> 系统。例如, <code>"1.3.0"</code> 。
<code>unity</code>	String	支持这个包的Unity版本。例如, <code>"2018.1"</code> 表明 Unity 2018.1以上的版本才能使用
<code>description</code>	String	包的简要说明
<code>keywords</code>	Array of Strings	用于在包管理器中搜索的关键词, 用JSON array表示。例如, <code>["Physics", "RigidBody", "Component"]</code> 。
<code>category</code>	String	包所属的分类(category)。例如, <code>"Forces"</code> 。
<code>dependencies</code>	Object	此包依赖的包的列表。

下面是一个 `package.json` 文件的例子:

```
1  {
2      "name": "com.unity.package-4",
3      "displayName": "Package Number 4",
4      "version": "2.5.1",
5      "unity": "2018.1",
6      "description": "This package provides X, Y, and Z.
7      \n\nTo find out more, click the \"View Documentation\"
8      link.",
9      "keywords": ["key X", "key Y", "key Z"],
10     "category": "Controllers",
11     "dependencies": {
12         "com.unity.package-1": "1.0.0",
13         "com.unity.package-2": "2.0.0",
14         "com.unity.package-3": "3.0.0"
15     }
16 }
```

Unity把package manifest存储在

`[your_package_root_folder]/package.json`。

## The Package Registry 包注册表

Unity维护着一个可供分发的正式包的中央注册表(central registry)。当Unity加载时, 包管理器和注册表通信, 检查project manifest 文件, 在包管理器窗口中显示每个可用包的状态。

当你从项目中移除一个package时，包管理器更新project manifest以在**In Project**模式的列表中排除它。

当你安装或更新一个package时，包管理器为从登记处下载这个package。

## Technical details 技术细节

### Requirements 要求

要求Unity 2018.1以后的版本。

### Known limitations 已知的限制

如果你手动编辑manifest.json文件，包管理器窗口不会刷新packages列表。你需要[重新打开窗口](#)或者在**In Project**和**All**模式间进行切换，以强制更新。

参考:

#### 1. [Unity Package Manager](#)