RKReplayKit屏幕共享集成

实现屏幕共享的主要步骤如下：

- 创建一个 Broadcast Upload Extension 用于开启屏幕共享的进程。
- 将录屏数据作为自定义视频源发送给 SDK，并使用 SDK 进行视频流的传输。

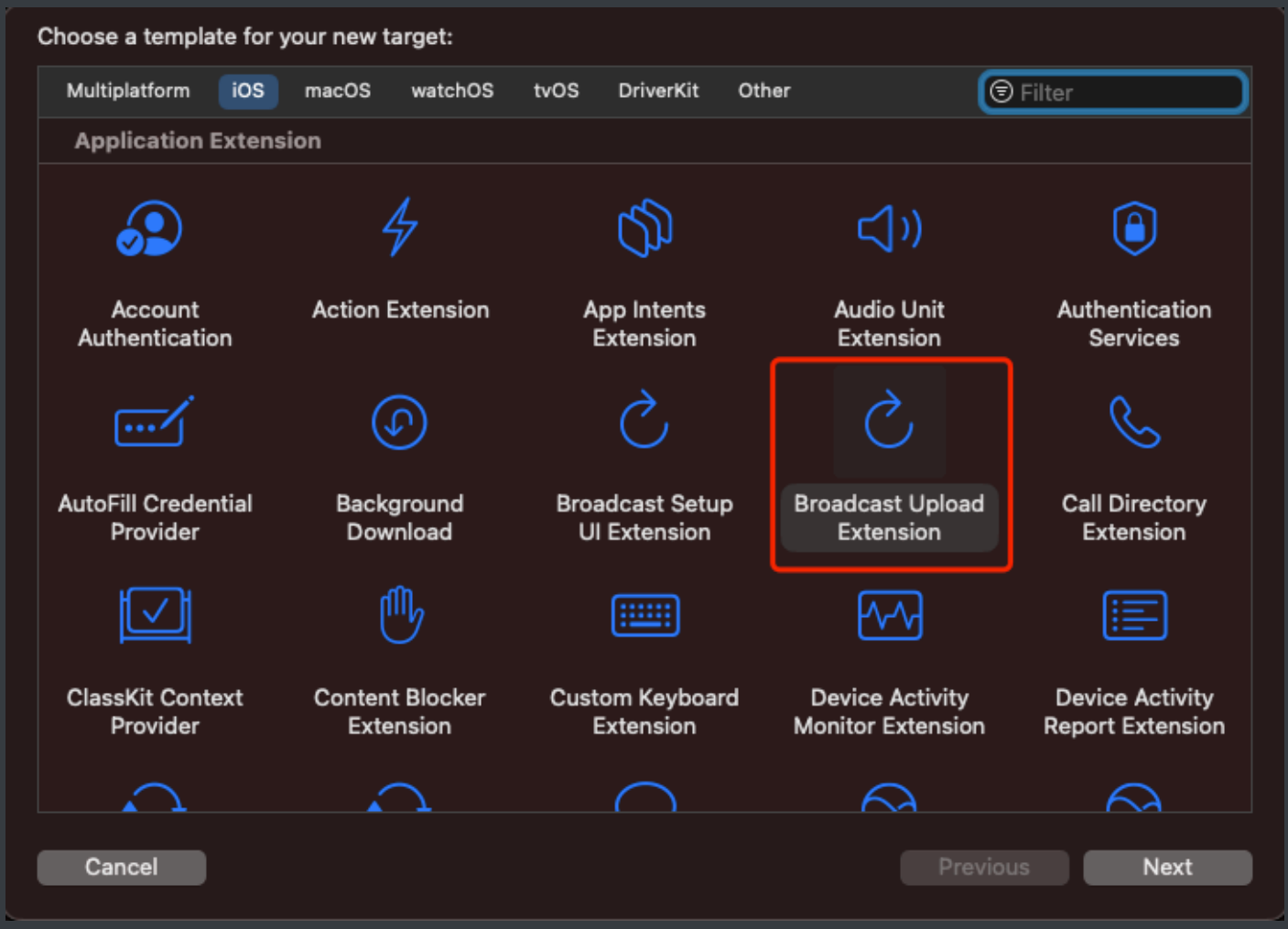
1. 创建屏幕共享 Extension

参考以下步骤：

a. 创建一个 Broadcast Upload Extension。

用 Xcode 打开项目的工程文件，在菜单栏中选择 Editor > Add Target…。

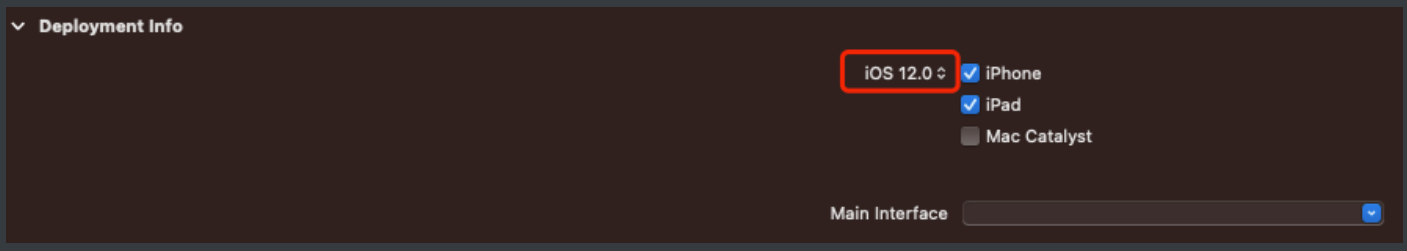
在弹出窗口中，选择 iOS 页的 Broadcast Upload Extension，点击 Next。如下图：



b. 在 Product Name 一栏输入 Extension 的名字，如 RKScreenShare，可以不勾选**Include UI Extension**，然后点击 Finish。

创建完成后，你会在项目中看到该 Extension 的文件夹RKScreenShare，用于存放屏幕共享功能的实现代码

Extension的**deployment Info**中的**target**需要设置适当的版本



c. 打开项目中的 Podfile，为 Extension 添加依赖项：

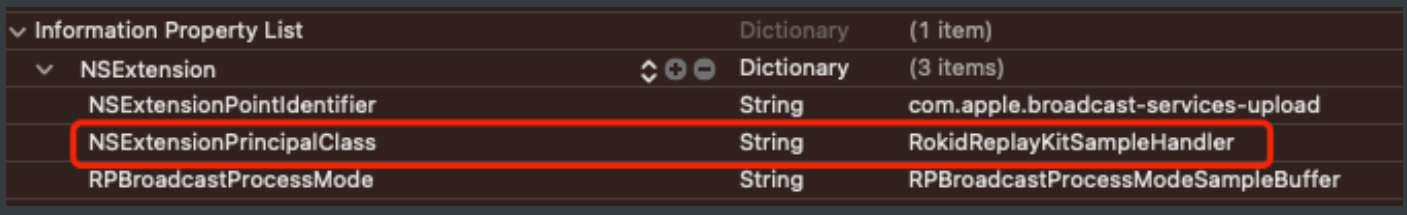
```
target 'RKScreenShare' do
  use_frameworks!
  # 集成
  pod 'RKBaseModule/RKReplayKit', :git => "https://github.com/rokid-cd/RKBaseModule.git"
end
```

在项目根目录下运行 pod install 命令，安装依赖项。

2. 屏幕共享功能实现

a.方式一： info.plsit配置（推荐）

直接修改 info.plsit下的 NSExtensionPrincipalClass 为 RokidReplayKitSampleHandler



b.方式二：代码配置

在Extensions 的**SampleHandler.m**文件中添加如下代码：

```
#import "SampleHandler.h"
#import <RokidReplayKit/RokidReplayKit.h>

@implementation SampleHandler

- (void)broadcastStartedWithSetupInfo:(NSDictionary<NSString *,NSObject
*> *)setupInfo {
    // User has requested to start the broadcast. Setup info from the UI
extension can be supplied but optional.
    [RokidReplayKitExtension extensionStartSampleHandler];
}

- (void)broadcastPaused {
    // User has requested to pause the broadcast. Samples will stop
being delivered.
}

- (void)broadcastResumed {
    // User has requested to resume the broadcast. Samples delivery will
resume.
}

- (void)broadcastFinished {
    // User has requested to finish the broadcast.
    [RokidReplayKitExtension extensionFinished];
}

- (void)processSampleBuffer:(CMSampleBufferRef)sampleBuffer withType:
(RPSampleBufferType)sampleBufferType {

    switch (sampleBufferType) {
        case RPSampleBufferTypeVideo:
            // Handle video sample buffer
```

```
            [RokidReplayKitExtension
extensionProcessSampleBuffer:sampleBuffer withType:sampleBufferType];
            break;
        case RPSampleBufferTypeAudioApp:
            // Handle audio sample buffer for app audio
            break;
        case RPSampleBufferTypeAudioMic:
            // Handle audio sample buffer for mic audio
            break;

        default:
            break;
    }
}

@end
```