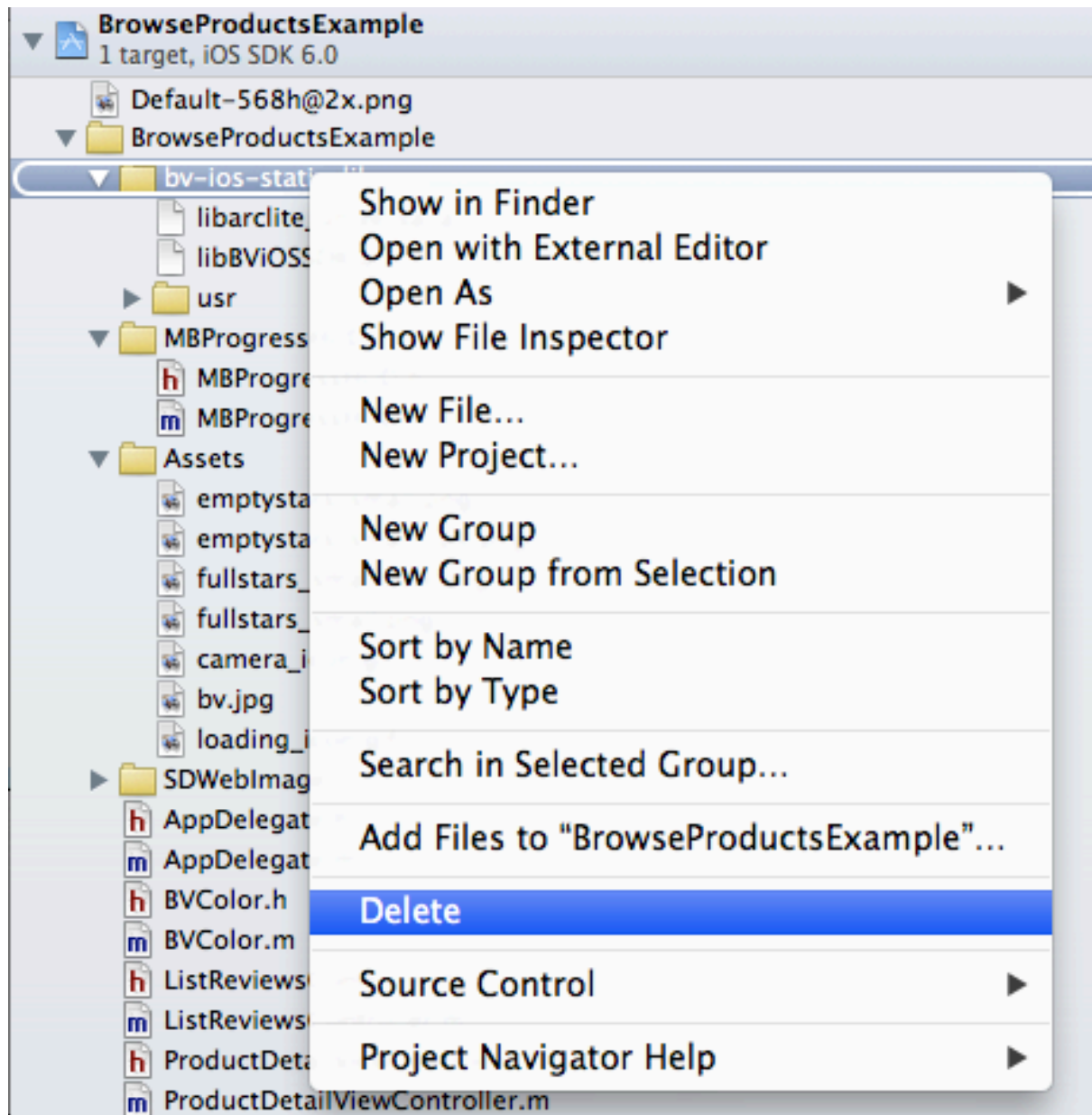


Updating from iOS SDK V1 to V2

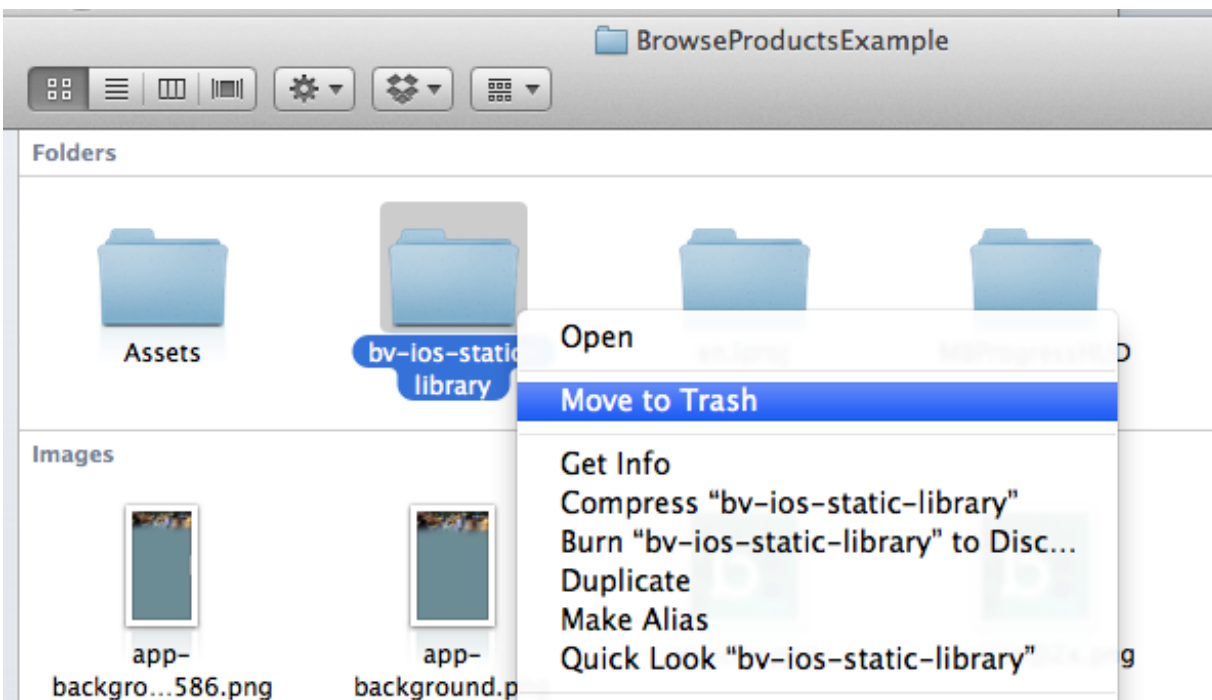
1.) Open the project that you wish to update. In the project navigator, option-click on the old bv-ios-static-library folder and click "delete."



2.) At the prompt, select "Move to Trash."

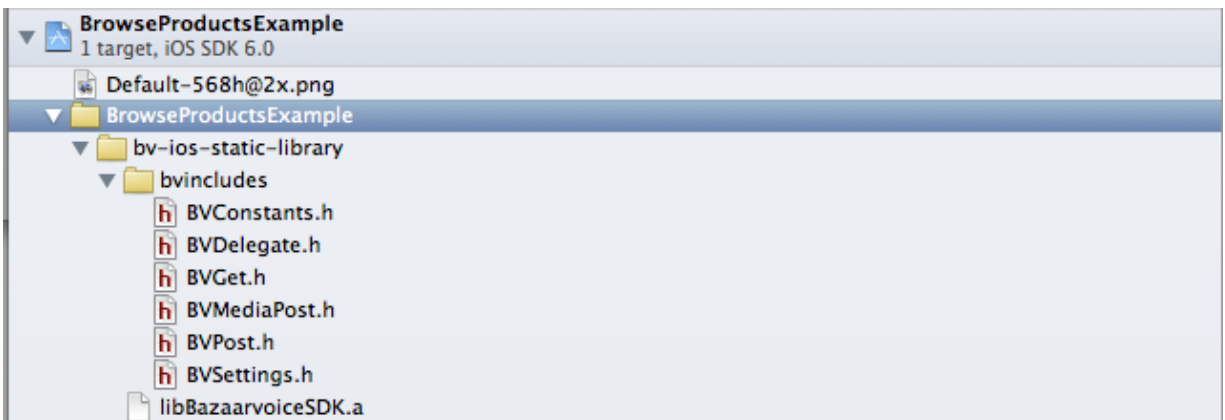
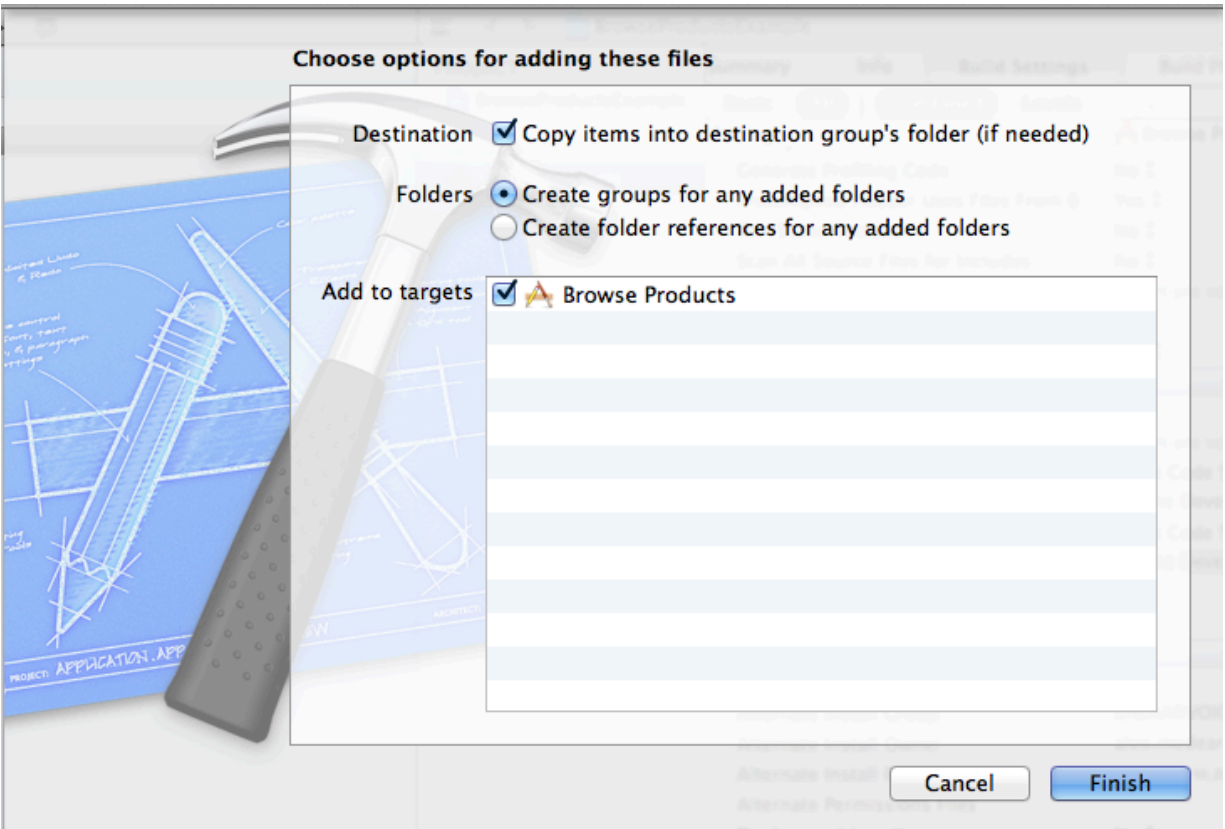


3.) In some cases, even though we selected "Move to Trash," Xcode does not properly remove the `bv-ios-static-library` folder. Open finder and navigate to your project to verify that the folder was properly removed. If not, control-click the remaining `bv-ios-static-library` folder and select "Move to Trash."



4.) Now, the project is ready for the new SDK. Drag and drop the `bv-ios-static-library` into your project.

5.) Select "Copy items into destination group's folder" and click "Finish."



6.) The final step is to update your code to reflect the new iOS SDK syntax. This process will vary depending on how you were using the SDK previously. In general, however, there will be 3 things you will need to do:

6a.) Update your includes. The BV iOS SDK no longer has a "catch all" #include file. Instead, it provides headers for available constants (BVConstants.h), request types (BVGet.h, BVPost.h and BVMedia Post.h), delegate methods (BVDelegate.h) and settings (BVSettings.h).

```
#import <UIKit/UIKit.h>
#import "BVIncludes.h"
```

! 'BVIncludes.h' file not found

becomes (for example):

```
#import <UIKit/UIKit.h>
#import "BVDelegate.h"
#import "BVGet.h"
```

6b.) Update your BV requests. All functionality that was available in V1 is available in V2, though the particular syntax may have changed. In general, all BVDisplay* requests will become BVGet requests. BVSubmissionVideo and BVSubmissionPhoto requests will become BVMediaPost requests. All other BVSubmission* requests will become BVPost requests. Consult the documentation if you are having trouble locating a particular method.

```
// Create an SDK request to fetch reviews
BVDisplayReview *showDisplayRequest = [[BVDisplayReview alloc] init];
// Params to fetch reviews for a specific product and include stats
NSString *productIdString = [NSString stringWithFormat:@"ProductId:%@", productId];
showDisplayRequest.parameters.filter = productIdString;
showDisplayRequest.parameters.stats = @"Reviews";

// Set up this object as a delegate and kick off the request
showDisplayRequest.delegate = self;
self.productRequest = showDisplayRequest;
[showDisplayRequest startAsynchRequest];
```

becomes (for example):

```
// Create an SDK request to fetch reviews
BVGet *showDisplayRequest = [[BVGet alloc] initWithType:BVGetTypeReviews];
// Params to fetch reviews for a specific product and include stats
[showDisplayRequest setFilterForAttribute:@"ProductId" equality:BVEqualityEqualTo value:productId];
[showDisplayRequest addStatsOn:BVIncludeStatsTypeReviews];

// Set up this object as a delegate and kick off the request
self.productRequest = showDisplayRequest;
[showDisplayRequest sendRequestWithDelegate:self];
```

6c.) Update your delegate methods and parsing logic. The primary difference is that there is no longer a special BVResponse object. Instead, the JSON response is passed directly back to the client.

```

- (void) didReceiveResponse:(BVResponse*)response forRequest:(BVBase *)request
{
    // Extract the reviews data from this response and set up the UI to reflect this data.
    self.reviewsData = response.results;
    [self setupProductInfo];
    [self.reviewsTable reloadData];

    // Hide the loading overlay
    [MBProgressHUD hideHUDForView:self.view animated:YES];
}

```

becomes (for example):

```

- (void) didReceiveResponse:(NSDictionary *)response forRequest:(id)request;
{
    // Extract the reviews data from this response and set up the UI to reflect this data.
    self.reviewsData = [response objectForKey:@"Results"];
    [self setupProductInfo];
    [self.reviewsTable reloadData];

    // Hide the loading overlay
    [MBProgressHUD hideHUDForView:self.view animated:YES];
}

```

The new BVDelegate methods can be found in BVDelegate.h.

Hopefully you will find the new BV iOS SDK easier to use. We look forward to working with you!