

Integration Guide QdsInappWrapperLib for iOS

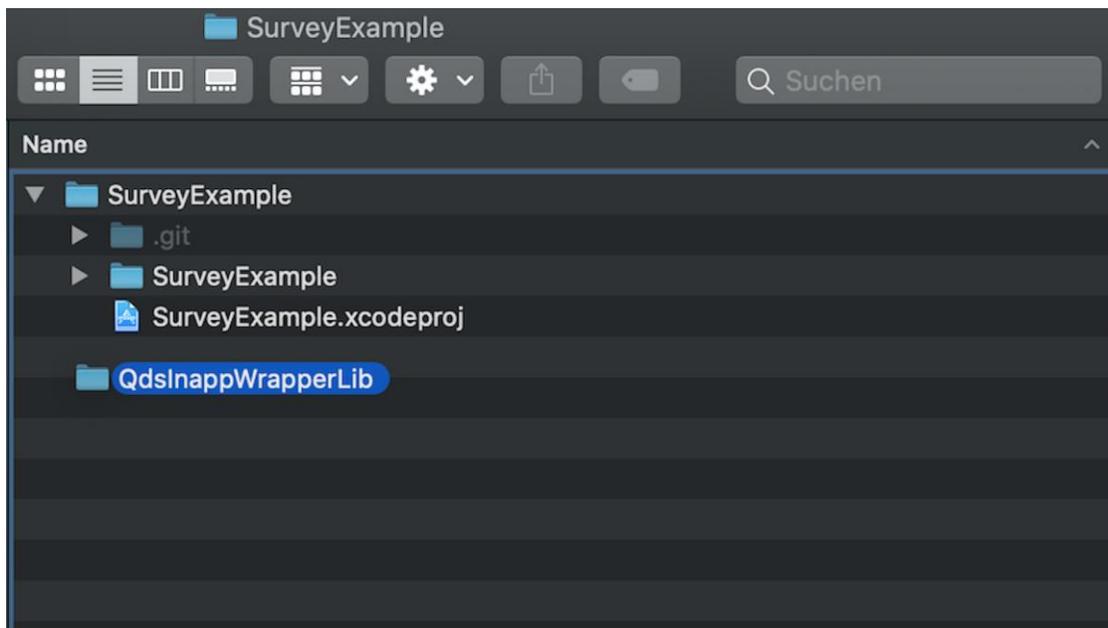
The integration of QdsInappWrapperLib takes place in the same way as IOLib of INFOnline (see INFOnline SZM Library Integration Guide iOS).

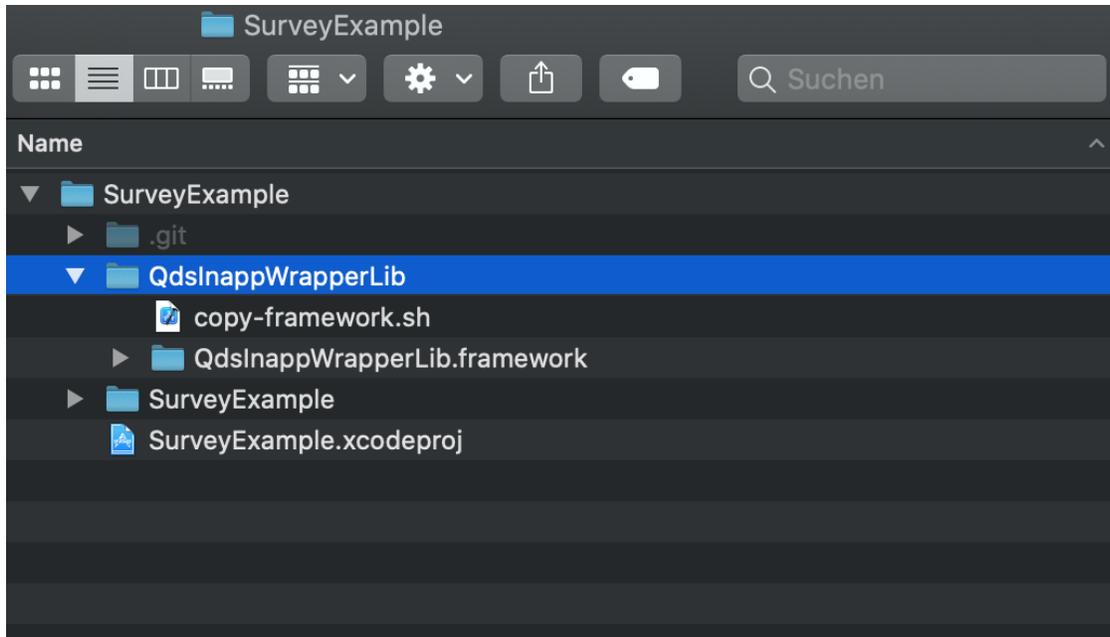
Note:

Please note that the library for in-app surveys of agof may only be initialized if the user has agreed to participate in agof studies (TCF 2.0 - Purpose 1 and 9 or identical in content) (see Integration Guide to the InApp Measurement Library).

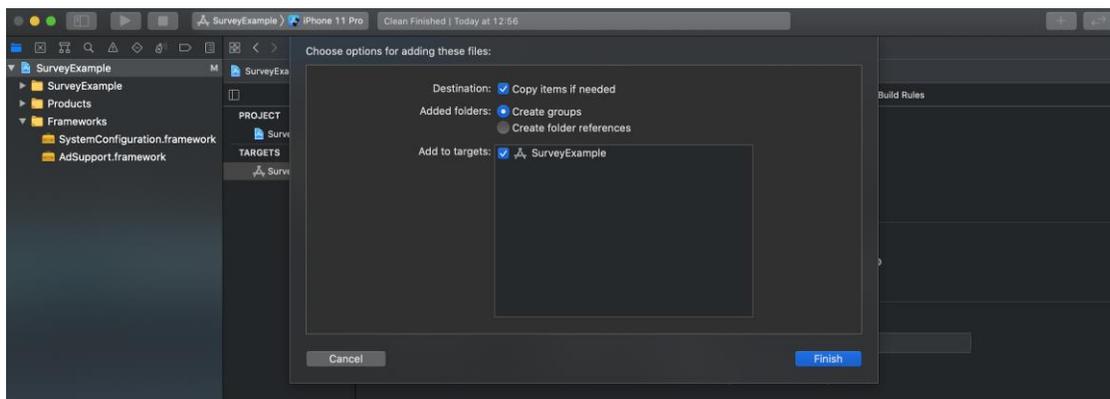
The following notes are to be additionally followed for integration:

The QdsInappWrapperLib comes as a "framework" together with a shell script for the build process. The "QdsInappWrapperLib" needs to be copied into the root directory of you project. Drag the "QdsInappWrapperLib.framework" from the file system to your Xcode project.

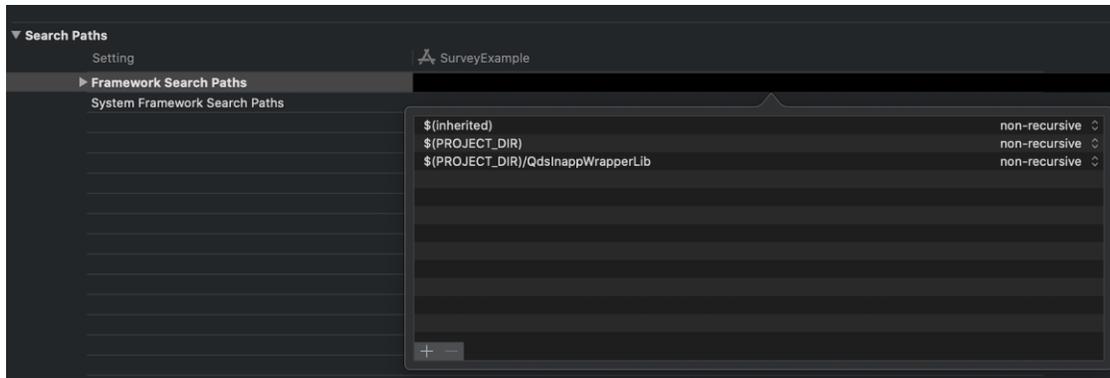




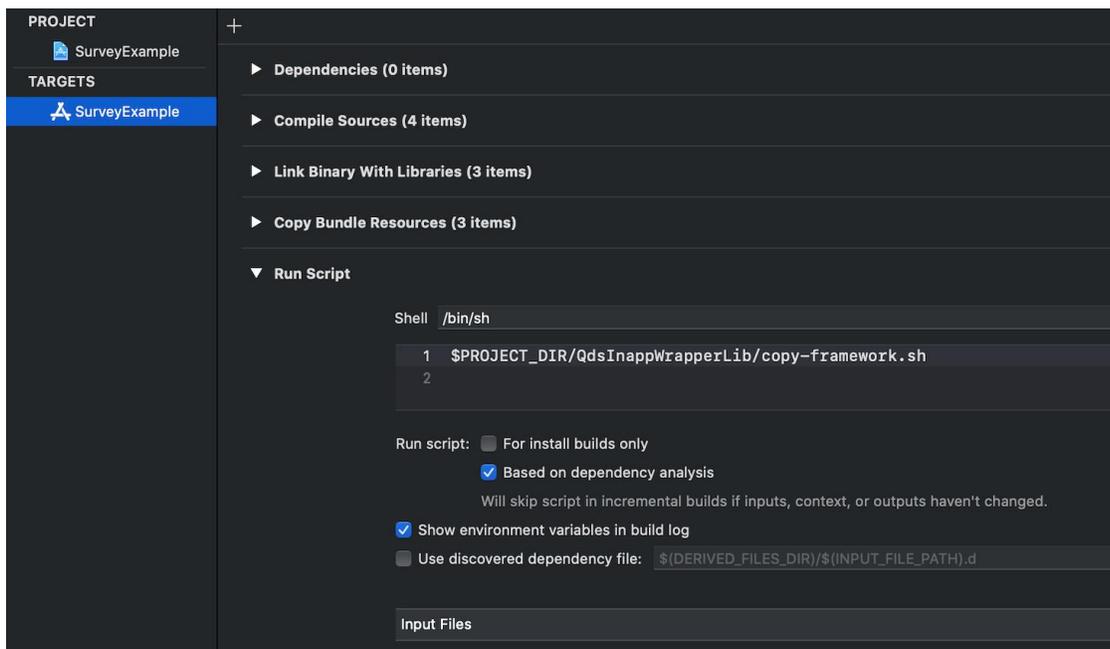
Check "Copy items if needed" in the dialogue box that appears. Also check "Create groups". If the framework hasn't appeared in the "Frameworks, Libraries and Embedded Content" section of your Xcode project, please add it manually by dragging it from the Xcode project explorer on the left to the "Frameworks, Libraries and Embedded Content" section.



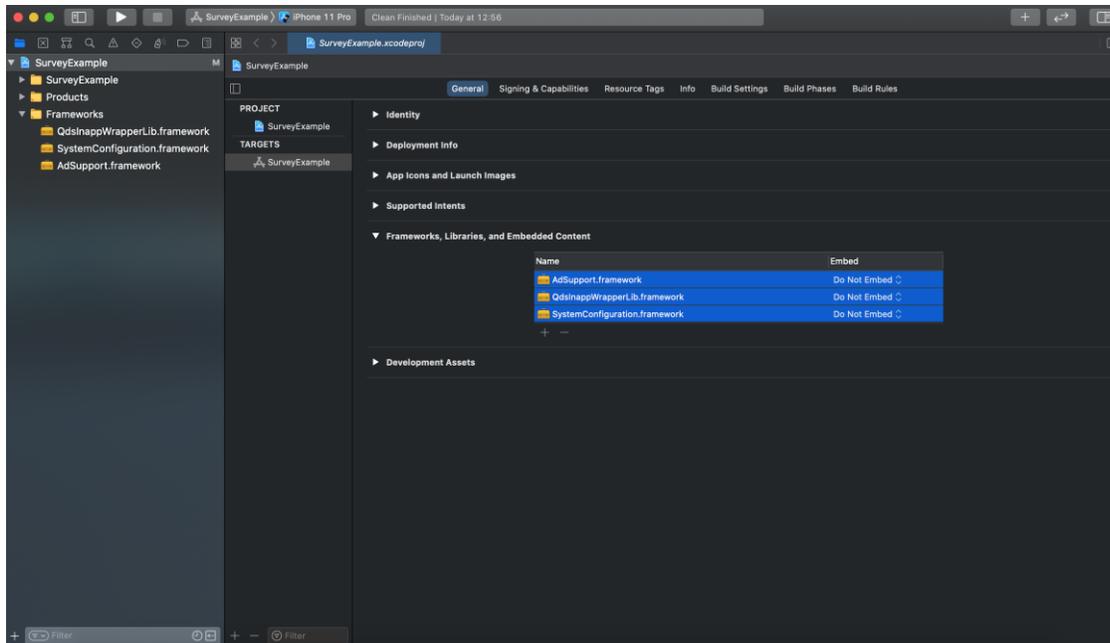
Furthermore Xcode should have added the path to the framework to the "Framework Search Paths" in the "Build Settings" tab. If the path is missing, add the path to the framework directory accordingly.



Now you just need to include the shell script in the "Build Phases". Switch over to the "Build Phases" tab and create a new "Run Script" configuration via the "New Run Script Phase" action. If you copied the "QdsInappWrapperLib" directory to the root directory of the Xcode project, you may include the script within the newly created "Run Script" phase as follows:
`$PROJECT_DIR/QdsInappWrapperLib/copy-framework.sh.`



"SystemConfiguration.framework" and "AdSupport.framework" must be integrated as additional "Frameworks, Libraries and Embedded Content" (see IOLib).



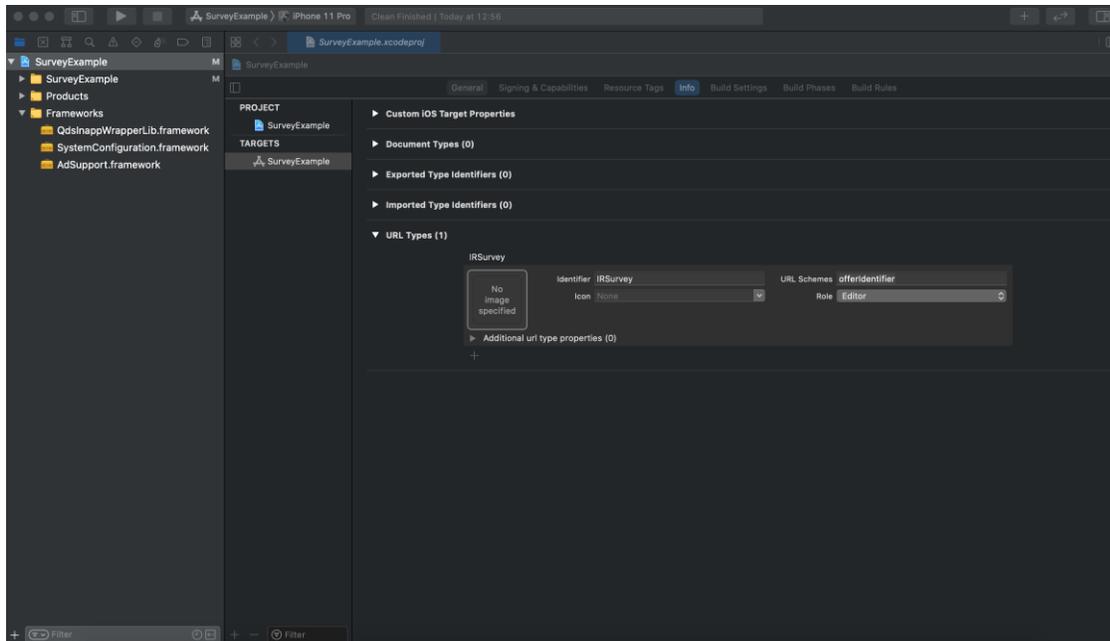
To obtain access to all required interfaces, the corresponding source file must import the required header file as follows:

```
#import <QdsInappWrapperLib/QdsInappWrapperLib.h>
```

Integration of the return address (optional)

The return to the app takes place by registration of a custom URL scheme.

This takes place in the app target under "Info":



The term offerIdentifier must be replaced with the identifier assigned by INFOnline (see: Functionality).

Important:

With the release of iOS 9, the operating system displays a "Back to App" link in the upper left corner of the display by default. For this reason, the "Back to App" link is disabled in the survey on iOS 9 or later, and is thus not shown.

Functionality

The QdsInappWrapperLib is represented by the class QdsInappWrapper. A concrete instance of this class is created with:

```
[[QdsInappWrapper alloc] init];
```

The QdsInappWrapperLib must be initialized when calling the app. This takes place by way of the above-mentioned call.

To enable display of a survey invitation, call the method

```
showSurveyInvitation:offerIdentifier  
withCountryCode:countryCode  
userHasAcceptedPermissionForTracking:userHasAcceptedPermissionForTracking
```

The offerIdentifier, country code (with "de") and the flag whether the user gave permission for tracking must be passed on in this context. The parameter offerIdentifier is the unique identifier of the offer for the respective app. The offerIdentifier is assigned by INFOnline for each app and each operating system. The parameter userHasAcceptedPermissionForTracking is the flag whether the user gave permission for tracking.

Important:

Any initial loading processes should be completed, since a poll invitation may be displayed immediately after calling `showSurveyInvitation`.

```
[[[QdsInappWrapper alloc] init]
    showSurveyInvitation:@"offerIdentifier"
    withCountryCode:@"de"
    userHasAcceptedPermissionForTracking:true];
```

If loading dialogues are shown when starting the app, `showSurveyInvitation` should only be called subsequently, since the invitation dialogue is displayed prematurely otherwise. Generally, calling `showSurveyInvitation` means that a dialogue with a survey invitation may be displayed after calling of the method. `showSurveyInvitation` should be called once the app is ready to display a questionnaire if needed. Generally, `showSurveyInvitation` can be called repeatedly.

Notes:

After successful integration of `QdsInappWrapperLib`, the **survey invitation should be displayed** immediately after the initial launch of the app, and every time it is switched to the foreground. If this is not possible for technical reasons, a suitable alternative entry point is to be selected by the developer. The important factor is that the user must get the opportunity to interact with the potential survey dialogue.

The **opt-out function** is controlled by `QdsInappWrapperLib` via the survey invitation, so no further action is necessary for this. Survey and data protection information on the agof study are optionally shown to the survey participant within the survey.

Test - Display dialogue window:

Please use the INFOnline SZM checker to mark the device as a test device. The documentation for the INFOnline SZM checker is available at https://www.infonline.de/downloads/9-1356-942/INFOnline_Manual_SZM-Checker_InAPP-Befragung.pdf.

Please check correct display of the survey window, clickability of all buttons, and opening of the questionnaire by clicking on the participate button.

The customer is responsible for installation of `QdsInappWrapperLib` in accordance with the valid guidelines of the testing organization. The customer is responsible for ensuring that the implementation of the library is correct even if the app is changed.

Support:

agof Service Center

Phone: 0800 / 410 29 77 (free of charge)

From abroad: 0049 -- (0)228 / 410 29 47

E-mail: servicecenter@agof.de