Article

Adding Sticker packs and iMessage apps to the system Stickers app, Messages camera, and FaceTime

Enable your Sticker pack or iMessage app in the media context.

Overview

In iOS 12 and later, Sticker packs and iMessage apps can appear in multiple contexts.

MSMessagesAppPresentationContext.messages context
    The Sticker pack or iMessage app appears in the list of iMessage apps that appears when you press the plus button.

MSMessagesAppPresentationContext.media context
    The sticker pack or iMessage app appears in the Stickers app throughout iOS, and in effect in FaceTime and the Messages camera.

People can access stickers throughout iOS through the emoji keyboard. To access effects in the Messages camera or FaceTime, a person taps the effects button. The system then displays all the iMessage apps and Sticker packs that support the media context. A person can launch an app, and use it to add images or stickers to the camera.

iMessage apps only appear in the `messages` context. To enable support for the `media`

context, set the `MSSupportedPresentationContexts` key in your extension's `Info.plist` file. If you indicate support for both `messages` and `media`, your Stickers only present in the `media` context including in Messages.

A Sticker pack or iMessage app's context limits its available features. For example, In the [MSMessagesAppPresentationContext.messages](#) context, a person can peel off stickers and attach them to any bubbles in the current conversation. Your app can also programatically send stickers, images, text, or interactive messages.

In the [MSMessagesAppPresentationContext.media](#) context, your app appears in effects in Messages and FaceTime. Therefore, the system limits your app to features that make sense over a photo or video. For instance, a person can't play an interactive game while taking a photo; however, they can add stickers or images to the picture.

Specify the supported contexts

To specify the supported contexts:

1. Open the `Info.plist` file for your Sticker pack or iMessage app extension.

2. Add the `MSSupportedPresentationContexts` key, using an Array type.

3. Add String values for the supported contexts. For the [MSMessagesAppPresentationContext.messages](#) context, add the `MSMessagesAppPresentationContextMessages` value. For [MSMessagesAppPresentationContext.media](#), add `MSMessagesAppPresentationContextMedia`.

You must specify one of the contexts, but if you specify both contexts, they only present in the [MSMessagesAppPresentationContext.media](#) context.

For more information on setting `Info.plist` keys, see [Edit property lists](#).

Work with the media context

The [MSMessagesAppPresentationContext.media](#) context supports only a subset of the features provided by the Messages framework. People can peel stickers from a sticker browser view, and place them in the Messages camera or FaceTime. They can reposition, resize, rotate, or remove stickers from the viewfinder.

iMessage apps can also insert stickers or images into the viewfinder using the `insert(_:completionHandler:)` or `insertAttachment(_:withAlternateFilename:completionHandler:)` methods. However, if the attachment isn't an image supported by `MSSticker`, then the `insertAttachment(_:withAlternateFilename:completionHandler:)` method fails with an `MSMessageErrorCode.apiUnavailableInPresentationContext` error.

Additionally, you can't insert interactive messages or text into the media context. You also can't send items automatically. Therefore, the following methods from the `MSConversation` class aren't available:

- `insert(_:completionHandler:)`

- `insertText(_:completionHandler:)`

- `sendAttachment(_:withAlternateFilename:completionHandler:)`

- `send(_:completionHandler:)`

- `send(_:completionHandler:)`

- `sendText(_:completionHandler:)`

If you call these methods, they fail with an `MSMessageErrorCode.apiUnavailableInPresentationContext` error.

Additionally, because you can't send or receive interactive messages, the system never calls the following methods in the `MSMessagesAppPresentationContext.media` context:

- `willSelect(_:conversation:)`

- `didSelect(_:conversation:)`

- `didReceive(_:conversation:)`

- `didStartSending(_:conversation:)`

- `didCancelSending(_:conversation:)`

Finally, the `MSMessagesAppPresentationContext.media` context is already inside the Messages camera or FaceTime; therefore, you can't display another camera inside the current viewfinder.

Detect the current context

Use the <u>MSMessagesAppViewController</u> object's <u>presentationContext</u> property to determine your iMessage app's current context. Enable only the features that make sense in that context.

```swift
if presentationContext == .messages {
    // The system is presenting your iMessage app inside the Messages app.
    // You have full access to the Messages framework.
    // You can insert or send stickers, text, attachments, or interactive messag
} else if presentationContext == .media {
    // The system is presenting your iMessage app in effects.
    // You can only insert stickers and images.
}
```

See Also

Custom sticker packs

📄 Adding your sticker packs to Messages

Drag and drop your sticker pack into the Stickers asset catalog to let people access your stickers from Messages.

class **MSStickerBrowserViewController**

A view controller that provides dynamic content to the standard sticker browser.

class **MSStickerBrowserView**

A browser view that displays a dynamically generated list of stickers.

class **MSStickerView**

A view for displaying a sticker.

enum MSStickerSize

The size of the stickers in the browser view.