

GoViral Social Networking Extension

Copyright © 2012-2015 Milkman Games, LLC. All rights reserved.

<http://www.milkmangames.com>

For support, contact support@milkmangames.com

Before you begin:

To View full AS3 documentation, see 'docs/as3docs/index.html'.

This extension requires the AIR 16.0 SDK or higher, which you can get at

<http://www.adobe.com/devnet/air/air-sdk-download.html> .

Review 'example/GoViralExample.as' for a sample application.

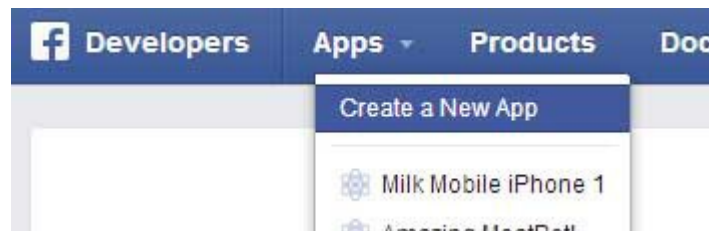
Note that GoViralExample.as is a Document Class. If you're a Flash Professional user and don't know how to use a document class, see the appendix "Using the Example Class in Flash CS6" at the end of this guide.

If you are migrating from an earlier version of the extension, please review this guide and see [releasenotes.txt](#) for details of changes.

1. Facebook Developer Site Setup

This section covers setting up a Facebook mobile app in the Developer section of the Facebook website. Before you can use the Facebook features of the extension, you'll need to create the App on the Facebook Developer site and register for a Facebook App ID.

1. Go to <http://developers.facebook.com>. If you haven't already done so, register a developer with Facebook to create your first app.
2. After logging in, select 'Apps>Create New App' from the top menu.

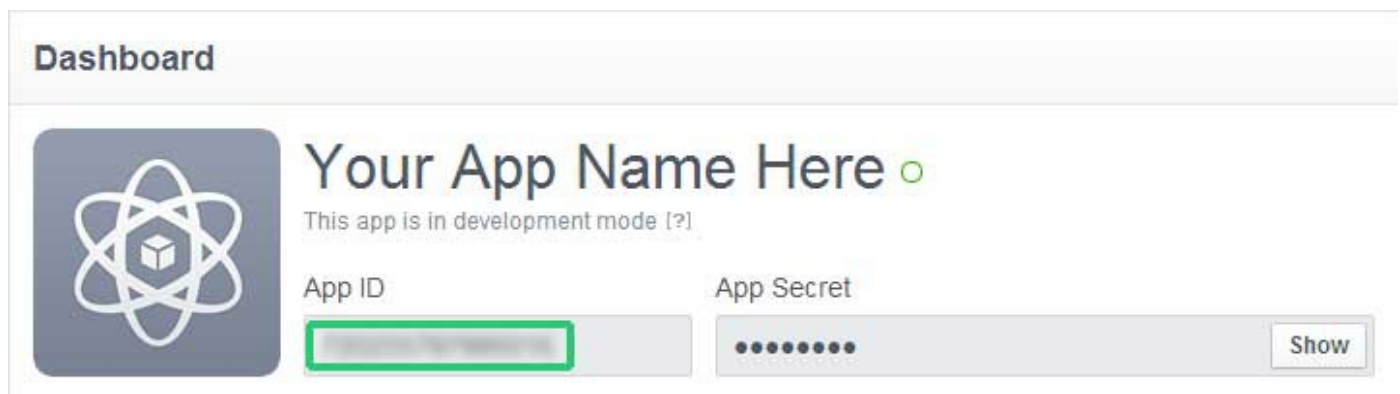


3. The 'Create New App' pop up dialog will appear.
4. Enter a display name and category information for your application, and press 'Continue'. Your app must be in the 'Games' Category to take advantage of some Game-specific Facebook features. **You**

A screenshot of the 'Create a New App' dialog box. The title is 'Create a New App' with the subtitle 'Get started integrating Facebook into your app or website'. It contains several input fields: 'Display Name' with the placeholder 'Your App Name Here', 'Namespace' with the placeholder 'A unique identifier for your app (optional)', 'Category' with a dropdown menu currently set to 'Games', and 'Sub-Category' with a dropdown menu currently set to 'Action'. At the bottom, there is a line of text: 'By proceeding, you agree to the Facebook Platform Policies'. To the right of this text are two buttons: 'Cancel' and 'Create App'.

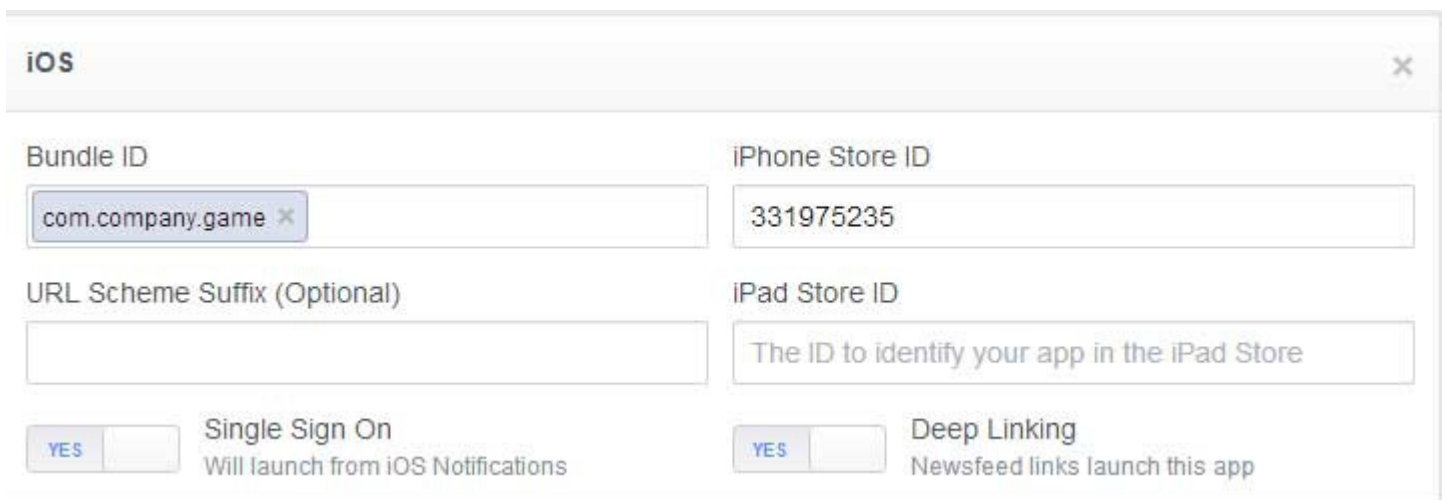
will need to use the exact display name you choose here as the name of your application in the app descriptor for AIR.

5. A Facebook App ID and App Secret have now been created for your application. Take note of the **App ID**- you'll need it later to configure your application in Adobe AIR.



The screenshot shows the Facebook Developer Dashboard for an application. At the top, it says 'Dashboard'. Below that is a header with a blue square icon containing a white atomic symbol and the text 'Your App Name Here' with a green status indicator. Below the header, it says 'This app is in development mode [?]'. There are two main sections: 'App ID' and 'App Secret'. The 'App ID' field contains the value '1234567890123456' and is highlighted with a green border. The 'App Secret' field contains a series of dots and a 'Show' button.

6. Choose Settings > Basic from the menu.
7. **(iOS App Setup:)** Select 'Add Platform>iOS'. For 'iOS Bundle ID', enter your iOS Application Bundle ID. This ID will need to match both the ID you used in iTunes Connect for the App, and the ID you use in your AIR Application XML descriptor.



The screenshot shows the 'iOS' app setup dialog box. It has a title bar with 'iOS' and a close button. The dialog is divided into two columns. The left column has 'Bundle ID' with a text field containing 'com.company.game' and a dropdown arrow, and 'URL Scheme Suffix (Optional)' with an empty text field. The right column has 'iPhone Store ID' with a text field containing '331975235', and 'iPad Store ID' with a text field containing 'The ID to identify your app in the iPad Store'. At the bottom, there are two sections: 'Single Sign On' with a 'YES' button and the text 'Will launch from iOS Notifications', and 'Deep Linking' with a 'YES' button and the text 'Newsfeed links launch this app'.

8. **(iOS App Setup:)** For 'iPhone App Store ID', you'll need to enter the numeric iTunes App Store ID for your app. If your app is not yet live on the App Store, you can enter a different app's numeric ID here for now. Later, when your App has been approved by Apple, you can replace it with your own numeric ID. If you need a numeric ID to use temporarily during development, you may use 331975235, which is the ID for Adobe Photoshop Express iOS.
9. **(iOS App Setup:)** For 'Single Sign ON' and 'Deep Linking', select 'YES'.
10. **(Android App Setup:)** Select 'Add Platform' and choose Android. Facebook for Android uses a

different security method for authenticating the app as your own than Facebook does- your **Android Key Hash**. You will need to generate a Key Hash for the testing/debug version of your app, AND the release version. For now, enter any string in this box (such as "invalidkeyhash"). Later, you can use the native extension to retrieve your actual Hash. *After you complete Facebook.com setup, refer to the section, "Getting Your Android Key Hash" below.*

Android

Package Name

air.com.company.game

Class Name

air.com.company.game.AppEntry

Key Hashes

YES

Single Sign On

Will launch from Android Notifications

YES

Deep Linking

Newsfeed links launch this app

11. **(Android App Setup:)** For 'Package Name', enter your full Android app ID (such as 'air.com.yourcompany.yourappname'; for 'Class Name', enter the package name plus '.AppEntry'- such as, 'air.com.yourcompany.yourappname.AppEntry'. Enable 'Configured for Android SSO'. **You must include the 'air.' prefix!**

12. This completes the basic Facebook setup.

2. Install the AIR 16.0 (or higher) SDK in your IDE

The GoViral extension requires the AIR 16.0 SDK or higher. You can download the latest AIR SDK from <http://www.adobe.com/devnet/air/air-sdk-download.html>. If you haven't already installed the AIR 16.0 (or higher) SDK for your Flash CS6+ or Flash Builder IDE, follow the instructions below:

Enabling the AIR 16.0 SDK in Flash Professional CS6+:

1. Unzip the AIR 16.0 SDK package to a location on your hard drive.
2. Launch Flash Professional CS6+.
3. Select Help > Manage AIR SDK...
4. Press the Plus (+) Button and navigate to the location of the unzipped AIR SDK
5. Press OK
6. Select File > Publish Settings
7. Select the AIR 16.0 SDK for iOS from the 'Target' Dropdown menu

Enabling the AIR 16.0 SDK in Flash Builder 4.6+ - Windows:

1. Unzip the AIR 16.0 SDK package to a location on your hard drive.
2. Close Flash Builder.
3. Locate the Flash Builder SDK directory. On the PC, usually c:\Program Files\Adobe\Adobe Flash Builder 4.6\sdk .
4. Make a copy of the current Flex SDK directory, and give it a descriptive name. For instance, copy the "4.6.0" SDK folder inside /sdks and name the copy "4.6.0_AIR16".
5. Copy and paste the contents of the AIR 16.0 SDK on top of the 4.6.0_AIR16 directory. Accept all changes.
6. Edit the flex-sdk-description.xml file inside the new directory, and change the value of the <name> tag to 'Flex 4.6.0 (AIR 16.0)'.
7. Open Flash Builder and choose Project > Properties > Flex Compiler > Configure Flex SDKs.
8. Press 'Add' and navigate to the new folder location.

Enabling the AIR 16.0 SDK in Flash Builder 4.6- Mac:

1. Copy the contents AIR 16.0 SDK package to a location on your hard drive.
2. Close Flash Builder.
3. Locate the Flash Builder SDK directory. On the Mac, it is usually /Applications/Adobe Flash Builder 4.6/sdks/. On the PC, c:\Program Files\Adobe\Adobe Flash Builder 4.6\sdk .
4. Create a new folder inside the SDK folder, called AIR35SDK and copy the contents of the SDK package into it.
5. Open the Terminal, and merge the AIR 16.0 SDK files into your current SDK directory:

```
sudo cp -Rp /Applications/Adobe\ Flash\ Builder\ 4.6/sdks/AIR35SDK/ /Applications/Adobe\ Flash\ Builder\ 4.6/sdks/4.6.0/
```

6. Edit the flex-sdk-description.xml file inside the new directory, and change the value of the <name> tag to 'Flex 4.6.0 (AIR 16.0)'.
7. Open Flash Builder and choose Project > Properties > Flex Compiler > Configure Flex SDKs.

8. Press 'Add' and navigate to the new folder location. Press 'Add' and navigate to the new folder location.

3. Include the GoViral API Library

Add the com.milkmangames.nativeextensions.GoViral.ane library to your project. These are in the /extension folder of the extension package zip file.

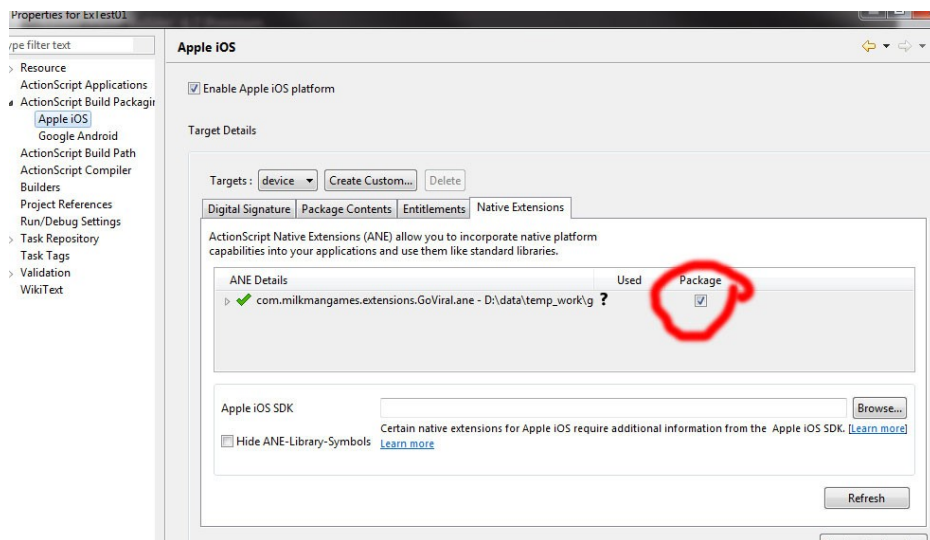
IMPORTANT: *During testing/development of an Android Facebook application, use the version of the extension in the '/extension/androiddebugane' folder that came with the extension zip package. Before public release, you should switch to the regular version.*

In Flash Professional CS6+:

1. Create a new project of the type 'AIR for iOS' (or AIR for Android)
2. Select File > Publish Settings...
3. Select the wrench icon next to 'Script' for 'Actionscript Settings'
4. Select the Library Path tab
5. Press the 'Browse for Native Extension (ANE) File' button and select the com.milkmangames.extensions.GoViral.ane file. Press OK.
6. Select the wrench icon next to 'Target' for Player Settings
7. Select the 'Permissions' tab, and enable INTERNET and WRITE_EXTERNAL_STORAGE
8. Check the 'Manually manage permissions and manifest additions for this app' box.
9. Make sure you're using the AIR 16.0 SDK per the instructions in Section 2.
10. Select File>Publish Settings...
11. For 'Target' Select 'AIR 16.0 for iOS' (Or Android)

In Flash Builder 4.6+:

1. Go to *Project Properties*
2. Select *Native Extensions*
3. Choose *Add ANE...* and navigate to the com.milkmangames.extensions.GoViral.ane file
4. Make sure the 'package' box is checked for your target, under Actionscript Build Packaging:



5. Make sure you're using the AIR 16.0 SDK per the instructions in Section 2.

4. Update Your Application Descriptor

For extension to work, changes are required to the application XML file for your app. Modify the xml file created by your IDE with the following changes (if you're a Flash user, make sure you've followed the steps above for 'Include the Library in Flash CS6+', otherwise Flash might undo your changes as you make them):

You'll need to be using the AIR 16.0 SDK or higher, and include the extension in your Application Descriptor XML. For an example, see '**example/app.xml**'. (Remember, the <id> section must exactly match your ids from developer.facebook.com, so you can't copy that part of the example xml verbatim.)

1. Set your AIR SDK to 16.0 in the app descriptor file:

```
<application xmlns="http://ns.adobe.com/air/application/16.0">
```

2. Include a link to the extension in the descriptor:

```
<extensions>
<extensionID>com.milkmangames.extensions.GoViral</extensionID>
</extensions>
```

3. **For iOS:** Add your Facebook App ID and Display Name to the XML Application Descriptor. Refer to Section 1 for information on locating your Facebook App ID in the Facebook Developer website. Add the following section, <iphone>, to the descriptor file, and **replace the 000000 in fb000000, and in the <string> below <key>FacebookAppID</key> with your Facebook App ID. Replace 'Your App Name' with your app name-which must exactly match the one you created on developer.facebook.com, and your AIR app name. Change only the parts show in red- the fb prefix is important.**

```
<iPhone>
<InfoAdditions>
    <![CDATA[
    <key>FacebookAppID</key>
    <string>000000</string>
    <key>FacebookDisplayName</key>
    <string>Your App Name</string>
    <key>CFBundleURLTypes</key>
    <array>
        <dict>
```

```

                <key>CFBundleURLSchemes</key>
                <array>
                    <string>fb000000</string>
                </array>
            </dict>
        </array>
    ]]>
</InfoAdditions>
</iPhone>

```

4. **For Android:** Set the Android ManifestAdditions required by the extension. You'll need to add the WRITE_EXTERNAL_STORAGE and INTERNET permissions, and add the LoginActivity, and Facebook NativeAppCallContentProvider and meta-data. **Replace '000000' with your Facebook App ID. Change only the parts show in red. Replace 'com.yourcompany.example' with your application's main bundle ID.** compatibility with the newest versions of Android, you'll need to set targetSdkVersion and minSdkVersion. Note this must be done all on one line, as shown in the example below. A bug in Adobe AIR prevents Google from accepting the APK if these version elements are in separate lines of XML. Add the following section, <android>, to the descriptor file:

```

<android>
<manifestAdditions><![CDATA[
<manifest android:installLocation="auto">
    <uses-sdk android:minSdkVersion="9" android:targetSdkVersion="19" />

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application>

        <activity android:name="com.facebook.LoginActivity"
android:theme="@android:style/Theme.Translucent.NoTitleBar" android:label="Login"/>

<provider android:authorities="com.facebook.app.NativeAppCallContentProvider000000"
android:name="com.facebook.NativeAppCallContentProvider" />

<meta-data android:name="air.com.yourcompany.example.ApplicationId" android:value="0000000000000000"/>

    </application>
</manifest>
]]></manifestAdditions>
</android>

```

5. API Quick Start: Initialization

The GoViral extension can be up and running in a few simple calls. See '[example/GoViralExample.as](#)' for a full example, including Facebook methods, Twitter, and Email.

1. Import the API Classes:

```

import com.milkmangames.nativeextensions.*;
import com.milkmangames.nativeextensions.events.*;

```

2. First, initialize the API and create an instance of the GoViral object, by calling GoViral.create().

IMPORTANT: This initialization must occur in the entry point of your application. If you are using Flex/MXML, this means you must call create() in the initialize() event of the main class (not a View constructor or createComplete callback.) If you are using pure ActionScript, you must call create() inside the Constructor function of the Document Class. If you are using timeline code in Flash, call this once at the beginning of Frame 1.

Make sure the current platform is supported (as in, it's not a PC or Mac, etc.) by checking the `GoViral.isSupported()` method:

```
if(GoViral.isSupported())
{
    GoViral.create();
}
else {
    trace("GoViral only works on mobile!");
    return;
}
```

6. API Quick Start: Facebook

GoViral offers complete support for Facebook mobile APIs: Single Sign-On/Authentication, Wall Post Dialogs, Friend Invitations, and the entire Facebook Graph API.

1. Start by initializing the Facebook SDK, using your Facebook App ID. Before initializing Facebook, check it is supported with `isFacebookSupported`. (The Facebook App ID can be found in your Facebook Developer site panel, as described in Section 1, Step 5.)

```
if(GoViral.goViral.isFacebookSupported())
{
    // replace 000000 with your facebook app id!
    GoViral.goViral.initFacebook("000000","");
}
```

2. Add any event listeners you may want for Facebook related login. The following example code shows all the available authentication listeners.

```
// Listen for events.
GoViral.goViral.addEventListener(GVFacebookEvent.FB_LOGGED_IN,onFacebookEvent);
GoViral.goViral.addEventListener(GVFacebookEvent.FB_LOGGED_OUT,onFacebookEvent);
GoViral.goViral.addEventListener(GVFacebookEvent.FB_LOGIN_CANCELED,onFacebookEvent);
GoViral.goViral.addEventListener(GVFacebookEvent.FB_LOGIN_FAILED,onFacebookEvent);
```

3. Log the user in to Facebook. You'd typically want to do this in response to pressing a 'Connect with Facebook' button or similar UI.

Note that as a feature of SSO (single sign-on), if your app is restarting after a previous session where the user has logged in, you may be automatically logged in after calling `initFacebook()` - in this case you will received the `FB_LOGGED_IN` event shortly after the app begins.

You can use the `GoViral.goViral.isFacebookAuthenticated()` method to determine if the user is logged in already. If they aren't, call `GoViral.goViral.authenticateWithFacebook()` to start a login prompt.

NOTE: Facebook only allows **READ** permissions in this initial request. **PUBLISH** permissions will cause your app to function incorrectly, or not at all. Publish permissions are handled automatically by the extension as needed.

```
// if the user is not already logged in...
if(!GoViral.goViral.isFacebookAuthenticated())
{
    // show a connect with Facebook prompt.
    // this method takes a comma separated list of facebook READ permissions as a first
```



```

        // parameter. You should have at least one read permission.

        // you can refer to the facebook documentation at
        // http://developers.facebook.com/docs/authentication/permissions/ to determine which
        // permissions your app requires.

        GoViral.goViral.authenticateWithFacebook("public_profile");
    }

```

Next, the user will be presented with the Facebook Connect prompt, and asked to grant your app permission to use Facebook on their behalf.

On Android, authentication will use an activity from the official Facebook app if it is installed, or a web view if not. For authentication to complete on Android, your Android Key Hash must be properly entered on developer.facebook.com. *After building your APK file, refer to Section 10, "Getting Your Android Key Hash".*

A `GVFacebookEvent` indicating the result will be dispatched when the login succeeds, fails, or is cancelled by the user- with the type of `GVFacebookEvent.FB_LOGGED_IN`, `GVFacebookEvent.FB_LOGIN_FAILED`, `GVFacebookEvent.FB_LOGIN_CANCELED`.

NOTE: As of April 30th, 2014, Facebook has changed the login rules such that users may deny some or all of the permissions you request. You can determine if a particular permission was granted by checking `GoViral.goViral.isFacebookPermissionGranted(permissionName)`, or get a list of denied permissions by checking `GoViral.goViral.getDeclinedFacebookPermissions()`.

Posting to the Facebook Wall

A common task your app may wish to perform is posting an update to the user's Facebook feed.

Using the Native Share Dialog

Facebook's new native sharing dialog is the ideal way to quickly post content to the Facebook Feed. It has an advantage over the deprecated "feed" dialog as it is rendered with a native UI, and works even if the user has not logged in with your app.

The Native Share Dialog is only available if the user has a recent version of the Facebook application installed on their device. If they do not, the extension will attempt to fallback to using the classic dialog (however, this requires that the user be previously authenticated.)

The following example will display a Facebook dialog prompting the user to share content, with a title, caption, description, link, and image URL. An inline callback is added that is triggered when the dialog is completed, and traces a message indicating the result:

```

// show a native sharing dialog to post the the user's wall
GoViral.goViral.showFacebookShareDialog(
    "Posting from GoViral AIR",
    "This is a caption",
    "This is a description",
    "http://www.milkmangames.com",
    "http://www.milkmangames.com/gvremoteimage.jpg").addDialogListener(function(e:GVFacebookEvent) {
    switch(e.type)
    {
        case GVFacebookEvent.FB_DIALOG_CANCELED:
            trace("The Share Dialog was canceled.");
            break;
        case GVFacebookEvent.FB_DIALOG_FAILED:
            trace("The Share Dialog has failed:"+e.errorMessage);
            break;
        case GVFacebookEvent.FB_DIALOG_FINISHED:

```

```

        trace("Successfully posted to share dialog:"+e.jsonData);
        break;
    }
});

```

Using the Deprecated Feed Dialog

If the user does not have a recent version of the Facebook application installed on their device, they will not be able to use the new Native Share Dialog. In this case, you may want to prompt them to update Facebook, or fall back to using the classic Feed Dialog.

The following example will display the classic Feed Dialog, if the new Native Share Dialog is not supported:

```

if(!GoViral.goViral.isFacebookShareDialogAvailable())
{
    // show a classic feed dialog to post the the user's wall
    GoViral.goViral.showFacebookFeedDialog(
        "Posting from AIR",
        "This is a caption",
        "This is a message!",
        "This is a description",
        "http://www.milkmangames.com",
        "http://www.milkmangames.com/gvremoteimage.jpg").addDialogListener(function(e:GVFacebookEvent) {
            switch(e.type)
            {
                case GVFacebookEvent.FB_DIALOG_CANCELED:
                    trace("The Feed Dialog was canceled.");
                    break;
                case GVFacebookEvent.FB_DIALOG_FAILED:
                    trace("The Feed Dialog has failed:"+e.errorMessage);
                    break;
                case GVFacebookEvent.FB_DIALOG_FINISHED:
                    trace("Successfully posted to feed dialog:"+e.jsonData);
                    break;
            }
        });
}

```

Using the Native Open Graph Action Dialog

The Native Open Graph Action Dialog is similar to the Native Share Dialog, but it allows you to post about a specific Open Graph Action. It also enables you to post a local `bitmapData` image directly, instead of passing a remote image URL.

Before using the Open Graph Action Dialog, you must define the Open Graph Object and Open Graph Action to use on developers.facebook.com. An example of an Action and Object you might use in your game would be to "slay" a "monster". You can visit <https://developers.facebook.com/docs/opengraph/creating-custom-stories> for detailed step-by-step instructions on how to define a custom Open Graph Action for your game.

The following example will display the Native Open Graph Action Dialog, posting a story about the user having "slain" a "monster", and attaching a `bitmapData` picture of their victory:

```

if(GoViral.goViral.isFacebookGraphDialogAvailable())
{
    // post a story about slaying the monster
    GoViral.goViral.showFacebookGraphDialog(
        "my_app:slay",
        "my_app:monster",
        "I am victorious!",

```

```

        "I destroyed a horrid beast in this game!",
        "http://www.milkmangames.com",
        bitmapData).addDialogListener(function(e:GVFacebookEvent) {
            switch(e.type)
            {
                case GVFacebookEvent.FB_DIALOG_CANCELED:
                    trace("The Graph Dialog was canceled.");
                    break;
                case GVFacebookEvent.FB_DIALOG_FAILED:
                    trace("The Graph Dialog has failed:"+e.errorMessage);
                    break;
                case GVFacebookEvent.FB_DIALOG_FINISHED:
                    trace("Successfully posted to Graph dialog:"+e.jsonData);
                    break;
            }
        });
    }
    else
    {
        trace("The device needs a newer Facebook app to use this feature.");
    }
}

```

Inviting Facebook Friends

The GoViral extension can prompt the user to invite their friends to your app, using `showFacebookRequestDialog()`.

Inviting Friends to Install the App

The most basic use of the Request Dialog is to invite friends to come and play your game:

```

// show a dialog inviting friends to install the game
GoViral.goViral.showFacebookRequestDialog(
    "Play my game",
    "Won't you play with me?").addDialogListener(function(e:GVFacebookEvent) {
        switch(e.type)
        {
            case GVFacebookEvent.FB_DIALOG_CANCELED:
                trace("The Request Dialog was canceled.");
                break;
            case GVFacebookEvent.FB_DIALOG_FAILED:
                trace("The Request Dialog has failed:"+e.errorMessage);
                break;
            case GVFacebookEvent.FB_DIALOG_FINISHED:
                trace("These ids were invited:"+e.data.to.join(","));
                break;
        }
    });

```

Advanced: Inviting a Friend to Take an Action

The request dialog supports a number of additional parameters for building more complex requests, shown below. You can learn more about making complex dialog requests at <https://developers.facebook.com/docs/games/requests/v2.0#params> :

```
public function showFacebookRequestDialog(message:String, title:String = null,
data:String = null, filters:String = null, to:String = null, excludeIds:String =
null, frictionless:Boolean = false, actionType:String = null, objectID:String = null,
suggestions:String = null):GVFacebookDispatcher
```

`title:string(default=null)`- *Optional*. Title to show on the request. Max length of 50 characters- this will be truncated if it's more than that.

`filters:String(default=null)`— *Optional*. **DEPRECATED**. Facebook may ignore this parameter on mobile SDKs. As an alternative, load the user's friends with a `facebookGraphRequest`, and post the necessary IDs to the `excludeIds` or to parameters. Allows filtering of recipients to `GVFacebookRequestFilter.APP_USERS`, or `GVFacebookRequestFilter.APP_NON_USERS`.

`excludeIds:String(default=null)- optional`. A comma delimited list of Facebook User IDs to exclude from the dialog picker.

actionType:String(default=null)- *Optional*. Used when defining additional context about the nature of the request. One of GVFacebookRequestActionType.ASK_FOR, GVFacebookRequestActionType.SEND, or GVFacebookRequestActionType.TURN.

suggestions:string(default=null)- *Optional*. A comma delimited list of user IDs that will be included in the dialog as the first suggested friends. Note: This parameter is can not be used together with filters. Using the two in the same dialog will result with an error.

```
// show a dialog inviting a particular friend to take their next turn
GoViral.goViral.showFacebookShareDialog(
    "Its your turn",           // message
    "Where will you move next?", // title
    "a7",                     // move data
    null,
    "1234567"                  // ID of the target player
    null,
```

```

        false,
        GVFacebookRequestActionType.TURN).addDialogListener(function(e:GVFacebookEvent) {
            switch(e.type)
            {
                case GVFacebookEvent.FB_DIALOG_CANCELED:
                    trace("The Request Dialog was canceled.");
                    break;
                case GVFacebookEvent.FB_DIALOG_FAILED:
                    trace("The Request Dialog has failed:"+e.errorMessage);
                    break;
                case GVFacebookEvent.FB_DIALOG_FINISHED:
                    trace("These ids were invited:"+e.data.to.join(",");
                    break;
            }
        });

```

If the friend accepts the request on their mobile device, they will be redirected to your app. At this time, AIR's `NativeApplication.nativeApplication` object will dispatch an `Event.INVOKE` event; you may parse its `arguments` parameter to find the data and other information that was associated with the completed request.

```

// show a dialog so the user can invite friends to app.
GoViral.goViral.showFacebookRequestDialog("Try out this app!", "My App Title");

```

Getting the User's Profile

You can retrieve the user's profile with the `requestMyFacebookProfile()` method. The `friends` property of the response is a `Vector.<GVFacebookFriend>`, that will contain the user's information.

```

// request the user's profile.
GoViral.goViral.requestMyFacebookProfile().addRequestListener(function(e:GVFacebookEvent) {
    if (e.type==GVFacebookEvent.FB_REQUEST_RESPONSE)
    {
        var myProfile:GVFacebookFriend=e.friends[0];
        trace("my Facebook ID is:"+myProfile.id);
        trace("my name is:"+myProfile.name);
        trace("my gender is:"+myProfile.gender);
        trace("My bio is:"+myProfile.bio);
        var pictureURL="http://graph.facebook.com/"+myProfile.id+"/picture?type=square";
        trace("My profile picture url is:"+pictureURL);
    }
    else
    {
        log("profile failed:"+e.errorMessage);
    }
});

```

Getting the User's Friends

NOTE: As of April 30th, 2014, Facebook has changed the rules such that you may only retrieve the IDs of friends who are already using the application. Additionally, IDs are scoped locally to your app instead of being universal IDs. To learn more about the details of these changes, visit https://developers.facebook.com/docs/apps/changelog#v2_0_graph_api.

You can retrieve the user's friends with the `requestFacebookFriends()` method. The `friends` property of the response is a `Vector.<GVFacebookFriend>`, that will contain the users' information.

```

// request the user's friends' profiles.
GoViral.goViral.requestFacebookFriends().addRequestListener(function(e:GVFacebookEvent) {
    if (e.type==GVFacebookEvent.FB_REQUEST_RESPONSE)

```

```

{
    for each(var friend:GVFacebookFriend in e.friends)
    {
        trace("i have a friend named:"+friend.name);
        trace("Their id is:"+friend.id);
        var pictureURL="http://graph.facebook.com/"+myProfile.id+"/picture?type=square";
        trace("Their profile picture url is:"+pictureURL);
    }
}
else
{
    log("friends request failed:"+e.errorMessage);
}
});

```

Posting a Photo to Facebook

You can post a `BitmapData` image to the user's Facebook photos with the `facebookPostPhoto()` method.

```

// posts a bitmapData called 'myBitmapData' to Facebook
GoViral.goViral.facebookPostPhoto(
    "a picture",
    myBitmapData).addRequestListener(function(e:GVFacebookEvent) {
        if (e.type==GVFacebookEvent.FB_REQUEST_RESPONSE)
        {
            trace("photo posted!" + e.jsonData);
        }
        else
        {
            trace("Photo post failed:" + e.errorMessage);
        }
    });

```

By default, the photo is posted to the `me/photos` graph path. The method takes an optional third parameter, `graphPath`, which you can set to another path if you want to upload the data to another album id.

Presenting a Facebook Page View (“Like us On Facebook”)

Facebook does not allow all mobile applications to programmatically “Like” a Facebook page- however you can direct the user to your Facebook Page view, where they can choose to press the “Like” button, or otherwise interact with your company.

You can present the page view by passing the page's ID to the `presentFacebookPageOrProfile()` method. The Page ID can easily be determined by visiting the page in the browser, and copying the numeric ID from the end of the URL. For instance, Milkman Games' Facebook page URL is <https://www.facebook.com/pages/Milkman-Games-LLC/215322531827565>, so the ID is 215322531827565:

```

// send the user to your Facebook page to complete other actions
GoViral.goViral.presentFacebookPageOrProfile("215322531827565");

```

Posting a Game High Score to Facebook Leaderboards

You can post a high score to the Facebook Leaderboards (if your app is a game) with the

`postFacebookHighScore()` method. Facebook automatically handles game actions by displaying them in the timeline, with special messages such as "Alice beat Bob's High Score" (display will depend on the popularity of your game and other factors):

```
// posts a facebook high score of 3500 points
GoViral.goViral.postFacebookHighScore(3500).addRequestListener(function(e:GVFacebookEvent) {
    if (e.type==GVFacebookEvent.FB_REQUEST_RESPONSE)
    {
        trace("Score posted!");
    }
    else
    {
        trace("Score post failed:"+e.errorMessage);
    }
});
```

Retrieving the Leaderboards

You can retrieve the leaderboards for the game by making a graph request to the `YOUR_FACEBOOK_APP_ID/scores` path. The following example traces the json Score data retrieved. (You can inspect the `e.data` property to iterate through the score data in Object form):

```
// get the leaderboards
GoViral.goViral.facebookGraphRequest(FACEBOOK_APP_ID+"/scores").addRequestListener(function(e:GVFacebook
Event) {
    if (e.type==GVFacebookEvent.FB_REQUEST_RESPONSE)
    {
        trace("Score data:"+e.jsonData);
    }
    else
    {
        trace("Score request failed:"+e.errorMessage);
    }
});
```

When the request is finished, a `GVFacebookEvent.FB_REQUEST_RESPONSE` will be dispatched, with a `graphPath` property of `me/scores`. If an error occurs, `GVFacebookEvent.FB_REQUEST_FAILED` will be dispatched instead.

Posting a Game Achievement to Facebook

You can post an achievement to the Facebook (if your app is a game) with the `postFacebookAchievement()` method. Before this will work, you need to post the achievement definition xml to the web and register it with Facebook. For more information on Facebook Achievements, see <https://developers.facebook.com/docs/games/achievements>.

```
// posts an achievement to facebook
GoViral.goViral.postFacebookAchievement("http://www.friendsmash.com/opengraph/achievement_5.html"
).addRequestListener(function(e:GVFacebookEvent) {
    if (e.type==GVFacebookEvent.FB_REQUEST_RESPONSE)
    {
        trace("Achievement posted!");
    }
    else
    {
        trace("Achievement request failed:"+e.errorMessage);
    }
});
```

```
    }  
});
```

Handling Mobile App Ad Installs

The extension automatically handles Facebook install tracking.

Getting the Custom Audience Advertising ID

You can retrieve the user's Custom Audience Advertising ID with the `requestFacebookMobileAdID()` method. This will cause a `GVFacebookEvent.FB_AD_ID_RESPONSE` event to be dispatched; its property `facebookMobileAdId` will contain the user's custom ad id if available, or null if is not.

For additional information on Custom Audience Advertising ID tracking, see <https://developers.facebook.com/docs/ads-for-apps/custom-audiences-for-mobile-apps/>.

```
// request the user's custom audience ad id.  
GoViral.goViral.requestFacebookMobileAdID().addRequestListener(function(e:GVFacebookEvent) {  
    trace("Ad id is:"+e.facebookMobileAdId);  
});
```

Using Facebook App Events for Ad Tracking and Analytics

Facebook App Events allow you to have real time, cross-platform analytics in your game or application and view them on the Facebook developer website. The following code shows examples of different types of App Event tracking calls. You can explore the Facebook documentation, or the inline documentation of the `GVFacebookAppEvent` class to see more possible event names and parameters.

```
// posts a bitmapData called 'myBitmapData' to Facebook  
GoViral.goViral.facebookPostPhoto("posted from ios sdk",myBitmapData);  
  
// example of logging a level achieved event  
var levelAppEvent=new GVFacebookAppEvent(GVFacebookAppEvent.EVENT_NAME_ACHIEVED_LEVEL);  
levelAppEvent.setParameter(GVFacebookAppEvent.EVENT_PARAM_LEVEL, "32");  
GoViral.goViral.logFacebookAppEvent(levelAppEvent);  
  
// example of logging a spent credits event  
var creditsEvent=new GVFacebookAppEvent(GVFacebookAppEvent.EVENT_NAME_SPENT_CREDITS);  
creditsEvent.setValueToSum(15);  
creditsEvent.setParameter(GVFacebookAppEvent.EVENT_PARAM_CONTENT_TYPE, "music");  
creditsEvent.setParameter(GVFacebookAppEvent.EVENT_PARAM_CONTENT_ID, "somesong");  
GoViral.goViral.logFacebookAppEvent(creditsEvent);  
  
// example of sending a custom event  
var customEvent=new GVFacebookAppEvent("customEventName");  
GoViral.goViral.logFacebookAppEvent(customEvent);
```

For more information on Facebook App Event tracking, visit <https://developers.facebook.com/docs/ios/app-events>.

Calling the Facebook Graph API

GoViral gives you access to the entirety of the Facebook Graph API- so you can use any method Facebook supports- scores, achievements, videos, checkins, events, groups and more – directly from Actionsript 3.

The following code shows the method signature for `GoViral.goViral.facebookGraphRequest()`. For more information on the Graph API, refer to <https://developers.facebook.com/docs/reference/api/>.

facebookGraphRequest() METHOD

```
public function facebookGraphRequest(graphPath:String, httpMethod:String = GET,
params:Object = null, publishPermissions:String = null):void
```

Post a raw request to the Facebook Graph API. Results are returned as JSON. This gives you native access to the entire Facebook platform. Refer to the Facebook developer documentation for details on the many accessible API methods. Facebook now requires that any 'PUBLISH' permissions are requested separately from initial sign on. If the graph request requires any PUBLISH permissions, include them in the `publishPermissions` parameter. If the user has not yet accepted the permissions, they will be prompted to do so before the request is processed. After the request, a `GVFacebookEvent` will be dispatched with the result- `GVFacebookEvent.FB_REQUEST_RESPONSE` will be dispatched if the request succeeds, and the string value of its property `'jsonData'` will contain the raw JSON response. The property `'data'` will contain an Object of the parsed JSON for convenience. If the request fails, a `GVFacebookEvent.FB_REQUEST_FAILED` event will be dispatched.

Parameters:

graphPath:String-The graph path command you wish to execute, such as "me/friends" to get user's friends. See the Facebook developer documentation for more details.

httpMethod:String (default =GET)- Optional. Default value is `GVHttpMethod.GET`. The `httpMethod` to use for the request, either `GVHttpMethod.GET`, `GVHttpMethod.POST`, or `GVHttpMethod.DELETE`. Refer to the Facebook developer documentation for details on the appropriate method.

params:Object(default =null)-Optional. An Object with string keys and values, for each additional parameter you want to send with the Graph request. See the Facebook developer documentation for more details.

publishPermissions:String- optional. A comma-separated list of any PUBLISH permissions required for the request.

7. API Quick Start: Twitter

Post a Status Update to Twitter

Posting a status update to Twitter is simple – just call `showTweetSheet()` with the message you'd like to post. (Twitter may not be available on iOS devices with a version below 5, or Android devices without Twitter. You can determine this before hand with the function `isTweetSheetAvailable()`):

```
if(GoViral.goViral.isTweetSheetAvailable())
{
    // show a twitter status post dialog
    GoViral.goViral.showTweetSheet("twitter post!").addDialogListener(function(e:GVTwitterEvent) {
```

```

        switch(e.type)
        {
            case GVTwitterEvent.TW_DIALOG_FINISHED:
                trace("Twitter dialog finished.");
                break;

            case GVTwitterEvent.TW_DIALOG_FAILED:
                trace("Twitter dialog failed.");
                break;

            case GVTwitterEvent.TW_DIALOG_CANCELED:
                trace("Twitter dialog canceled.");
                break;
        }
    });
}

```

Android User Note: Due to the limitations of interprocess communication on Android, the native Twitter app on the Android device may dispatch a `DIALOG_FINISHED` event, even if the dialog was canceled by the user.

Post an Image to Twitter

You may post a `bitmapData` image to Twitter using the `showTweetSheetWithImage()` method.

```

if (GoViral.goViral.isTweetSheetAvailable())
{
    // show a twitter status post dialog with bitmapData image
    GoViral.goViral.showTweetSheetWithImage(
        "twitter post!",
        myBitmapData).addDialogListener(function(e:GVTwitterEvent) {
            switch(e.type)
            {
                case GVTwitterEvent.TW_DIALOG_FINISHED:
                    trace("Twitter dialog finished.");
                    break;

                case GVTwitterEvent.TW_DIALOG_FAILED:
                    trace("Twitter dialog failed.");
                    break;

                case GVTwitterEvent.TW_DIALOG_CANCELED:
                    trace("Twitter dialog canceled.");
                    break;
            }
        });
}

```

Android User Note: Due to the limitations of interprocess communication on Android, the native Twitter app on the Android device may dispatch a `DIALOG_FINISHED` event, even if the dialog was canceled by the user.

On Android, the extension will locate an app on that phone that responds to requests to send Tweets. Some Twitter Android apps from some Android manufacturers do not accept messages that have both custom status messages and images; in these cases, the image will be attached but a default status message may be displayed for the user to edit. If you are not seeing an image and message attachment while using the official Twitter app from Google Play, update to the latest version, as Twitter has recently fixed a bug affecting this behavior.

Presenting a Twitter Profile View (“Like us On Twitter”)

You can present a Twitter profile page view by passing the Twitter handle/ user ID to the `presentTwitterProfile()` method. For instance, the following code will show milkmangames' Twitter page:

```
// send the user to your Twitter page
GoViral.goViral.presentTwitterProfile("milkmandgames");
```

8. API Quick Start: Email

Email is a great simple way to invite your users to share information from your app with their friends. GoViral supports plain emails, html emails, and emails with image attachments, which you may pass in as `BitmapData`.

Send an Email

The `showEmailComposer()` method will display a native iOS or Android Email Composer window to the user, prepopulated with the information of your choice. It takes a Subject, comma-separated list of 'to' addresses, and email body as its parameters. The last parameter is a Boolean value- set it to false if your email body is formatted as plain text; set it as true if your email body is formatted as html. (If you want to avoid making the request on devices where email is not set up, you may check `isEmailAvailable()`):

```
if(GoViral.goViral.isEmailAvailable())
{
    // show an email to who@where.com and john@doe.com, with subject 'this is a subject!',
    // and a plain text body of 'This is the body of the message.'
    GoViral.goViral.showEmailComposer(
        "This is a subject!",
        "who@where.com,john@doe.com",
        "This is the body of the message.",
        false
    ).addDialogListener(function(e:GVMailEvent):void {
        switch(e.type)
        {
            case GVMailEvent.MAIL_SENT:
                trace("Mail sent!");
                break;

            case GVMailEvent.MAIL_SAVED:
                trace("Mail saved.");
                break;

            case GVMailEvent.MAIL_CANCELED:
                trace("Mail canceled.");
                break;

            case GVMailEvent.MAIL_FAILED:
                trace("Failed sending mail.");
                break;

        }
    });
}
```

Android User Note: Due to the limitations of interprocess communication on Android, the native email app on the Android device may dispatch a `MAIL_SENT` event, even if the dialog was canceled by the user.

Send an Email with Image Attachment

The `showEmailComposerWithBitmap()` method will display a native iOS or Android Email Composer window to the user, prepopulated with the information of your choice. It uses the same parameters as `showEmailComposer()`, with one additional parameter- the `BitmapData` image you'd like to attach to the mail. (If you want to avoid making the request on devices where email is not set up, you may check `isEmailAvailable()`):

```
if(GoViral.goViral.isEmailAvailable())
{
    // show an email to who@where.com and john@doe.com, with subject 'this is a subject!', a
    // and a plain text body of 'This has a pic attached.' Attaches a bitmapData image called
    // myBitmapData.
    GoViral.goViral.showEmailComposerWithBitmap(
        "This is a subject!",
        "who@where.com,john@doe.com",
        "This has a pic attached.",
        false,
        myBitmapData
    ).addDialogListener(function(e:GVMailEvent):void {
        switch(e.type)
        {
            case GVMailEvent.MAIL_SENT:
                trace("Mail sent!");
                break;

            case GVMailEvent.MAIL_SAVED:
                trace("Mail saved.");
                break;

            case GVMailEvent.MAIL_CANCELED:
                trace("Mail canceled.");
                break;

            case GVMailEvent.MAIL_FAILED:
                trace("Failed sending mail.");
                break;
        }
    });
}
```

Android User Note: Due to the limitations of interprocess communication on Android, the native email app on the Android device may dispatch a `MAIL_SENT` event, even if the dialog was canceled by the user.

Send an Email with a ByteArray Attachment

The `showEmailComposerWithByteArray()` method will display a native iOS or Android Email Composer window to the user, prepopulated with the information of your choice. It lets you specify any type of attachment you wish, in the form of a `ByteArray`:

```
if(GoViral.goViral.isEmailAvailable())
{
    // show an email to who@where.com and john@doe.com, with subject 'this is a subject!', a
    // and a plain text body of 'This has a pic attached.' Attaches a ByteArray image called
    // myByteArray, with a mimetype of text/xml and a file name of myfile.txt.
    GoViral.goViral.showEmailComposerWithByteArray(
        "This is a subject!",
        "who@where.com,john@doe.com",
        "This has an attachment.",
        false,
```

```

        myByteArray,
        "text/xml",
        "myfile.txt"
    ).addDialogListener(function(e:GVMailEvent):void {
        switch(e.type)
        {
            case GVMailEvent.MAIL_SENT:
                trace("Mail sent!");
                break;

            case GVMailEvent.MAIL_SAVED:
                trace("Mail saved.");
                break;

            case GVMailEvent.MAIL_CANCELED:
                trace("Mail canceled.");
                break;

            case GVMailEvent.MAIL_FAILED:
                trace("Failed sending mail.");
                break;
        }
    });
}

```

Android User Note: Due to the limitations of interprocess communication on Android, the native email app on the Android device may dispatch a MAIL_SENT event, even if the dialog was canceled by the user.

9. API Quick Start: Generic Share a Message or Image

All supported versions of Android can be used to post a message and/or image with no particular target, and allow the user to select an app from a list of installed apps to handle the post. Similar functionality is available in iOS 6 and higher.

Share a message

The `shareGenericMessage()` method allows you to send a message, and have the user select an app from their installed apps to handle the post. This feature is supported on all Android devices, and iOS devices running iOS 6 or higher. To ensure the feature is available on the current device, you should check `isGenericShareAvailable()` before calling `shareGenericMessage()` :

```

// show a dialog for the user to choose how to share a social message
if(GoViral.goViral.isGenericShareAvailable())
{
    GoViral.goViral.shareGenericMessage(
        "This is a subject!",           // not all targets will use the 'subject'
        "I'm sharing this!",           // this is the message to share
        false,                          // set to 'true' if the message is html-formatted
        800,                           // on iOS 6 ipads, show the popup at 800 pts x
        800                            // on iOS 6 ipads, show the popup at 800 pts y
    ).addDialogListener(function(e:GVShareEvent):void {
        trace("Generic share completed.");
    });
}

```

Share a message and image

The `shareGenericMessageWithImage()` method behaves exactly like , but with an extra `BitmapData` paramter for sharing images :

```

// allows the user to choose
// plain text body of 'This is the body of the message.'

// show a dialog for the user to choose how to share a social message
if(GoViral.goViral.isGenericShareAvailable())
{
    GoViral.goViral.shareGenericMessageWithImage(
        "This is a subject!",           // not all targets will use the 'subject'
        "I'm sharing this!",           // this is the message to share
        false,                          // set to 'true' if the message is html-formatted
        myBitmapData,                  // the bitmapData image to share
        800,                           // on iOS 6 ipads, show the popup at 800 pts x
        800,                           // on iOS 6 ipads, show the popup at 800 pts y
    ).addDialogListener(function(e:GVShareEvent):void {
        trace("Generic share completed.");
    });
}

```

Fast Social Sharing with Sina Weibo, Twitter, SMS, or Facebook

iOS 6 introduces a new feature, the Social Composer View- which can be used for quickly sharing messages and images to built-in social services as Sina Weibo, Twitter, and Facebook. This feature is accessible to the extension via the `displaySocialComposerView()` method. You may specify a social service type from one of the constants `GVSocialServiceType.FACEBOOK`, `GVSocialServiceType.TWITTER`, `GVSocialServiceType.SMS` or `GVSocialServiceType.SINAWEBIBO`, as well as a message- and optionally, an image and URL. Before calling it, you should check `isSocialServiceTypeAvailable()` to make sure the service is enabled and the OS supports it.

Although Android does not support the same built-in composer view feature as iOS, the extension will attempt to present a similar dialog on Android using the related application, if it is installed and it supports the type of messaging.

```

// share with Sina Weibo in iOS 6+:
if(GoViral.goViral.isSocialServiceAvailable(GVSocialServiceType.SINAWEBIBO))
{
    GoViral.goViral.displaySocialComposerView(
        GVSocialServiceType.SINAWEBIBO, // GVSocialServiceType.FACEBOOK or .TWITTER also available
        "I'm sharing this!",             // this is the message to share
        myBitmapData,                    // the (optional) bitmapData image to share
        "http://www.milkmangames.com"    // the (optional) url to share
    ).addDialogListener(function(e:GVShareEvent):void {
        switch(e.type)
        {
            case GVShareEvent.SOCIAL_COMPOSER_FINISHED:
                trace("composer finished!");
                break;
            case GVShareEvent.SOCIAL_COMPOSER_CANCELED:
                trace("Facebook composer canceled.");
                break;
        }
    });
}

```

NOTE: On Android, the `GVShareEvent.SOCIAL_COMPOSER_FINISHED` event is always dispatched due to limitations of the Android OS.

10. Getting Your Android Key Hash

Facebook uses a different method of authenticating your Application as yours on Android than it does on iOS; the *Android Key Hash*. Your Android Key Hash is a special key string that is unique to the keystore (.p12 file) you use to sign your application.

To get your Android Key Hash:

1. Build your application with the version of the .ane found in the '/extension/androiddebugane' folder.
2. Run your app on your Android phone. When `initFacebook()` is called, a pop-up window will display your Android Key Hash.
3. You may select the 'Mail' button the pop-up to email the Key Hash to yourself.
4. Take the Key Hash and enter it under 'Android Key Hash' on the developer.facebook.com page for your app.
5. **Facebook authentication will not work on Android until your Key Hash has been entered on developer.facebook.com.**
6. **The Key Hash may be different between your testing/debug app and your release/Google Play app. Be sure to use the correct Key Hash before releasing your app publicly.**
7. **Be sure to switch back to the release version of the ANE before releasing your app to Google Play- otherwise your Key Hash will be compromised and visible to all users.**

11. Troubleshooting Common Problems

"Why does login not work / the app crash / Twitter not work / email not work?"

- The settings in `application.xml` are not correct; refer to *Update Your Application Descriptor*. A common mistake is to forget the 'fb' prefix before your Facebook App ID inside the URL Scheme. Read the example XML settings carefully- 90% of tech support requests can be solved by fixing missing or incorrect xml settings. **Your Facebook 'DisplayName' in XML must also match exactly the one you set on developer.facebook.com.**

If you're using Flash, make sure that you've ticked the box to Manage Permissions and Manifest Additions Manually.

- The settings on developer.facebook.com are not correct; refer to Section 1, *Update Facebook Developer Site Setup*.
- If you have an event listener for `Event.DEACTIVATE`, in which you call `nativeApplication.exit()`, you may be causing your application to close itself when a new window appears (and thinking that it's 'crashing'.)
- Before you can call Facebook specific functions successfully, you must log the user into facebook using the `authenticateWithFacebook()` method. You can check if the user's currently logged in with the `isFacebookAuthenticated()` method.
- *For Facebook, if you have sandbox mode enabled on developer.facebook.com, only developers registered with your app will be able to login.*

"How do I use the GoViralExample.as file in Flash Professional CS6 or Higher?"

- First, create the application, add the extension, and update developer.facebook.com, by following this guide, Sections 1-4.
- Copy and paste `GoViralExample.as` into the same folder as your .fla. Do not copy and paste its contents on

to the timeline. That will not work.

- In Flash properties, under 'Document Class', type 'GoViralExample' (no quotes) and press OK.
- Edit the GoViralExample.as file, and change the YOUR::APP_ID value to your own facebook id.
- Build and install the application.

“Why doesn't the extension work when I run my swf on my Mac / iPhone / Windows Computer?”

- *The extension uses features that are built into the iOS and Android operating systems. The extension will only work when you run it on a phone or tablet.*

“I still need help! How can I get technical support?”

- Your purchase comes with free technical support by email – just send us a message at support@milkmandgames.com . We're here to help! Please remember that...:
- We're open during United States business hours, excluding U.S. Holidays, Monday-Friday, Pacific Standard Time. We strive to answer all support email within 24 hours (not including weekends and holidays) but usually do so much faster. Remember that we may not be in the same time zone.
- Please remember to mention: which extension you're having a problem with, what IDE you're using (such as 'Flash CS6' or 'Flash Builder 4.6'), and what device you're targeting. If you're experiencing an error message, please specify what that message is.
- *We don't provide tech support through blog comments, Facebook, or Twitter. Please email us and we'll be happy to help you out!*