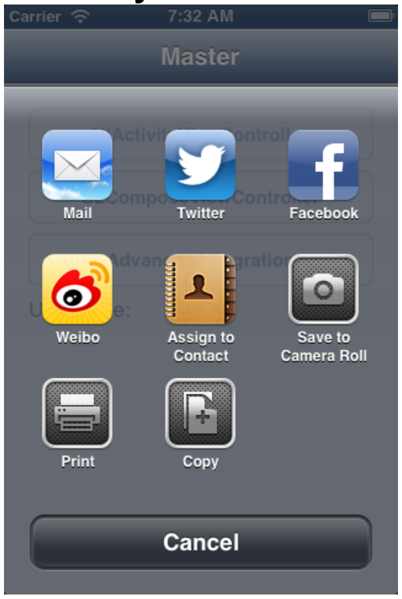
Building on the Twitter integration announced in iOS 5, Apple announced OS level integration of Facebook and SinaWeibo for iOS 6. This integration, using the new Social framework, makes incorporating sharing functionality into your applications easier and provides users with a consistent experience These APIs also make use of the single sign on (SSO) feature in iOS 6, which does credential management for you. If you are developing an app targeted for iOS 5 or higher, you can still use the Social framework but need to degrade gracefully to the Facebook SDK for devices using versions prior to iOS 6.

The Social framework supports three types of social services in iOS 6: Facebook, Twitter and SinaWeibo. You will only see the SinaWeibo in preferences if you have a Chinese keyboard enabled. For advanced integration the most important class is the SLRequest, which provides a mechanism to issue requests to Facebook's Graph API.

There are two easy ways to share with iOS6: UIActivityViewController and the SLComposeViewController.

UIActivityViewController



The first new item is the UIActivityViewController that brings up a modal view with different options to share such as email, Twitter and Facebook. You will see the UIActivityView throughout iOS6 when a user selects the share icon. No special frameworks or imports are needed here.

The code below is all you need to open a post sheet to share a text message and a picture of a cat. Feel free to put in your own picture.

```
NSString *text = @"Cat";
UIImage *image = [UIImage imageNamed:@"cat"];
NSArray *activityItems = [NSArray arrayWithObjects:text,image , nil];
UIActivityViewController *avc = [[UIActivityViewController alloc]
      initWithActivityItems: activityItems applicationActivities:nil];
[self presentViewController:avc animated:YES completion:nil];
```

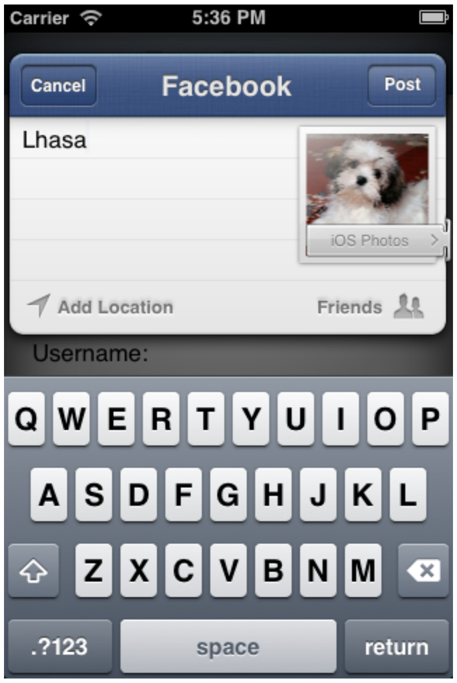
SLComposeViewController

If you are targeting one particular supported social network there is a streamlined method of creating a share sheet using SLComposeViewController (Link: https://developer.apple.com/library/ios/#documentation/NetworkingInternet/Reference/SLComposeViewController_Class/Reference/

Reference.html). This is similar to the now deprecated Twitter framework and requires you to add the Social.framework to your project. Building on the last example, the snippet below uses a share sheet to create a Facebook Post view and add an image.   One other configurable setting would be an URL. One thing to note, SLComposeViewController conforms to UIAppearanceContainer so that navigation bar and bar button items can be styled to meet your apps styles.

```
if([SLComposeViewController isAvaliableForServiceType:SLServiceTypeFacebook]) {
       SLComposeViewController*fvc = [SLComposeViewController
         composeViewControllerForServiceType:SLServiceTypeFacebook];
      [fvc setInitialText:@"Lhasa"];
      [fvc addImage:[UIImage imageNamed:@"lhasa"]];
      [self presentViewController:fvc animated:YES completion:nil];
}
```

After executing the code above you should be presented with a view controller similar to the one below. As you can see, the view controller handles all Facebook specific functionality such as assigning the album the photo should be uploaded to, adding or removing your location, and tagging any friends.



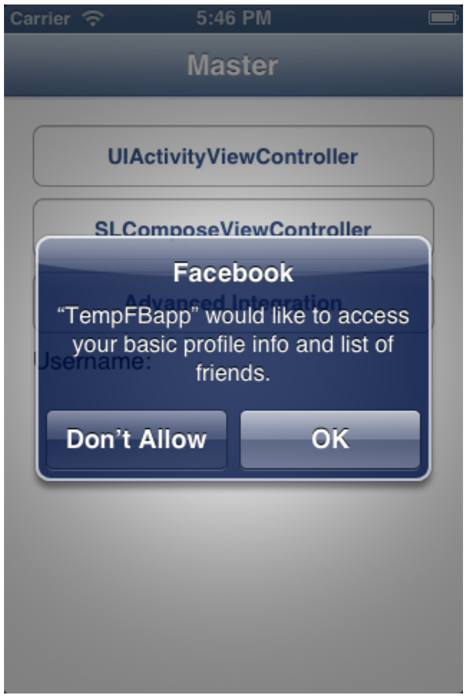
Advanced Integration

Advanced integration requires a little more work, but considerably less than the official Facebook SDK (https://github.com/downloads/facebook/facebook-ios-sdk/FacebookSDK-3.1.pkg). Advanced integration allows your app full access to the Facebook Graph API. On the Facebook developers page the Graph API Explorer tool allows you to try your URL requests before you add them to your code.

To use advanced integration you first you need to register your app in iTunes Connect. It can be in the "Prepare for Upload" stage, but you need the Apple ID of the app for the next step. Second, you need to register your app at http://developers.facebook.com on the Facebook App Dashboard, accessible by selecting Apps in the navigation bar and then selecting Create New App. The tabs for Basic, Permissions and Advanced need to be updated. Be sure to have your BundleID and iPhone/iPad App Store ID.

Using the Accounts framework as well as the Social framework you create an ACAccountStore object so that you can request access the users Facebook information. You will need to create an NSDictionary that contains a value for the key ACFacebookAppIdKey. This value is the retrieved from Facebook after registering your application. Although the Apple class reference for ACAcountStore, as of Sept 22,2012, says that the ACFacebookPermisionsKey is optional, it is not. You must include at least one ACFacebookPermissionsKey in the options dictionary.. According to Facebook, it must be one of the following read permissions: email, user_birthday, or user_location. One other key is the ACFacebokAudienceKey which chooses who is allowed to see the post. Other read and publish permissions can be found at https://developers.facebook.com/docs/howtos/ios-6/. There are two items that you cannot get permission for with your app: the user's timeline and managing pages.

When the requestAccessToAccountsWithType is invoked, the user will receive a control that resembles the image below asking them if they want to give permission to your app to access their basic account information. If you request other permissions beyond the basic profile, for instance friends_videos, another similar message will be displayed asking to access that information as well.



Facebook

"TempFBapp" would like to access your basic profile info and list of friends.

The sample code below will get you started;

```
self.accountStore = [[ACAccountStore alloc]init];
ACAccountType *FBaccountType= [self.accountStore
accountTypeWithAccountTypeIdentifier:
     ACAccountTypeIdentifierFacebook];
NSString *key = @"987654"; //put your own key from FB here
NSDictionary *dictFB = //use ACAccountStore to help create your dictionary
[self.accountStore requestAccessToAccountsWithType:FBaccountType options:dictFB
     completion: ^(BOOL granted, NSError *e) {
         if (granted) {
              NSArray *accounts = [self.accountStore
accountsWithAccountType:FBaccountType];
              //it will always be the last object with SSO
              self.facebookAccount = [accounts lastObject];
         } else {
              //Fail gracefully...
              NSLog(@"error getting permission %@",e);
     } }];
```

Not only did requestAccessToAccountsWithType get the account but it also retrieved a token for the account. Apple has encapsulated the OAuth request with this method so the developer no longer needs to worry about managing the OAuth request themselves. The token received is necessary for subsequent calls. SLRequest below contains a property, *account,* from the ACAccount framework which holds that token. The code below shows how to use that account to grab the users information from the Facebook Graph API and convert it to an NSDictionary. Other URLs calls can be found on the Facebook developer page.

```
NSURL *requestURL = [NSURL URLWithString:@"https://graph.facebook.com/me"];
SLRequest *request = [SLRequest requestForServiceType:SLServiceTypeFacebook
                               requestMethod:SLRequestMethodGET
                               URL:requestURL
                               parameters:nil];
                        request.account = self.facebookAccount;
                        [request performRequestWithHandler:^(NSData *data,
                                                             NSHTTPURLResponse
*response,
                                                             NSError *error) {

     if(!error){
          NSDictionary *list =[NSJSONSerialization JSONObjectWithData:data
                                   options:kNilOptions error:&error];
          NSLog(@"Dictionary contains: %@", list );
              }
              else{
                  //handle error gracefully
              }

         }];
```

Remember that the user may change their Facebook settings or the app token may time out. If you listen for an ACAccountStoreDidChangeNotification you can then ask the user to renewCredentialsForAccount:

```
 -(void)accountChanged:(NSNotification *)notif//no user info associated with this
notif
{
    [self attemptRenewCredentials];
}
-(void)attemptRenewCredentials{
    [self.accountStore renewCredentialsForAccount:(ACAccount
*)self.facebookAccount completion:^(ACAccountCredentialRenewResult renewResult,
NSError *error){
        if(!error)
        {
            switch (renewResult) {
                case ACAccountCredentialRenewResultRenewed:
                    NSLog(@"Good to go");
                    [self get];
                    break;
                case ACAccountCredentialRenewResultRejected:
                    NSLog(@"User declined permission");
                    break;
                case ACAccountCredentialRenewResultFailed:
                    NSLog(@"non-user-initiated cancel, you may attempt to retry");
                    break;
                default:
                    break;
            }

        }
        else{
            //handle error gracefully
            NSLog(@"error from renew credentials%@",error);
        }
    }];


}
```

Closing

This post presents ideas on the new Accounts and Social frameworks for using Facebook in your enterprise iOS6 applications. It should give you the foundation needed to begin implementing Facebook integration in your applications. I've attached the source for TempFBapp. If you've got questions you may contact me at strohtennis @ gmail.