

Integration with the TurboScan app for iOS.

This document describes a simple way to seamlessly integrate document scanning functionality of TurboScan into other apps. The approach requires very little programming (with almost all necessary code snippets provided in this document). It uses simple custom URL calls and automatic transferring of resulting image(s) via the standard Pasteboard. (Note that TurboScan v.2.7.7 or greater has to be installed on the device.)

The user workflow is very efficient. Once the user presses the “scan” button in the controller app, he is transferred to TurboScan’s camera screen, does the “scanning”, changes some parameters (if necessary) and presses the “back” button that takes him back to the calling app. The image(s) automatically appear in the main app UI. It takes only 3-4 clicks before user is back in the original app with the scanned image.

The controller app can set the scanning mode it wants to run TurboScan in (regular or SureScan 3x), specify if it needs single or multipage scanning, and define the title to be put on the “back” button in TurboScan.

• • •

The API (Objective C is used for the sample code).

1. Displaying the “Scan” button in the controller app UI:

First approach: you only put the “Scan” button in your app if TurboScan is installed on the device. This code will check if TurboScan is installed:

```
bool installed = [[UIApplication sharedApplication] canOpenURL: [NSURL
URLWithString: @"turboscan://"]];
```

Note: with the above method you must declare the TurboScan URL scheme you want to pass to this method. Do this by using the `LSApplicationQueriesSchemes` array in your Xcode project’s Info.plist file.

Second approach: if you feel that TurboScan functionality is valuable enough for your users, you can have the “Scan” or “TurboScan” button always displayed in your app, in which case you can check the “success” flag returned from `[[UIApplication sharedApplication] openURL:...]` (in step 2 below). *False* means TurboScan is not installed and you can send user to TurboScan App Store page using this link:

```
[[UIApplication sharedApplication] openURL:[NSURL URLWithString:
@"https://itunes.apple.com/app/turboscan-quickly-scan-
multipage/id342548956?mt=8"]];
```

2. Launching TurboScan from the controller app:

Use the following code and custom URL specs to launch TurboScan upon pressing “Scan” (in this example, the calling app name is XpenseTracker):

```
bool success = [[UIApplication sharedApplication] openURL:[NSURL
URLWithString:
@"turboscan://XpenseTracker?action=camera&returnURL=xpensetracker%3A%2F%2F
&multipage=no&callerName=XpenseTracker" ]];
```

There are several query keys here (some are optional):

<i>action</i>	<i>camera</i>	- opens camera screen in single shot mode
	<i>suresscan</i>	- opens camera screen in SureScan (3x) mode
	<i>search</i>	- opens the search screen
	<i>main</i>	- opens the main menu
<i>returnURL</i>	URL to re-open the calling app. Provide any parameters you need. Please use hexadecimals for special characters (like <i>%2F</i> for slash, etc.)	
<i>multipage</i>	<i>yes</i>	- enable multipage scanning
	<i>no</i>	- enable single page scanning (default)
<i>format</i>	<i>jpeg</i>	- resulting image(s) in JPEG format (default)
	<i>pdf</i>	- resulting single document in PDF format
<i>callerName</i>	controller app name to be used as a title for the “back” button (optional)	

Tip: you can use this code to automatically substitute parameters with special characters with the escaped ones:

```
NSString* rfc3986EscapeChars = @":/?#[!$&'()*+,-;=";
NSString* escapedParameter = (NSString*)CFURLCreateStringByAddingPercentEscapes(
    NULL, (CFStringRef)myParameter, NULL,
    (CFStringRef)rfc3986EscapeChars,
    kCFStringEncodingUTF8);
```

3. Transferring back the resulting image(s):

Upon pressing the “back” button on top right in TurboScan (that should be titled with the calling app name), TurboScan copies one or more scanned pages to standard Pasteboard. The following code may be used to check for and extract the JPEG image(s) or a single (multipage) PDF file into your app:

```
// If JPEG image format is enabled:
+ (bool)pasteboardHasImages
{
    UIPasteboard *pasteboard = [UIPasteboard generalPasteboard];
    return [pasteboard containsPasteboardTypes:[NSArray
arrayWithObjects:@"public.jpeg", @"public.png", nil]];
}
```

```

}
// If PDF format is enabled:
+ (bool)pasteboardHasPDFImage
{
    UIPasteboard *pasteboard = [UIPasteboard generalPasteboard];
    return [pasteboard containsPasteboardTypes:[NSArray
        arrayWithObjects:@"com.adobe.pdf", nil]];
}

+ (void)pasteFromPasteboard
{
    UIPasteboard *pasteboard = [UIPasteboard generalPasteboard];
    NSMutableArray* imageItems = [NSMutableArray arrayWithCapacity:0];
    NSArray* arrayBoardImages = pasteboard.images;

    for(int i=0; i<arrayBoardImages.count; i++)
    {
        NSData *data = [arrayBoardImages objectAtIndex:i];
        UIImage* image = nil;
        uint8_t c = 0;

        if([data isKindOfClass:[NSData class]])
        {
            if(data == nil || data.length == 0)
                continue;

            [data getBytes:&c length:1];

            if(c != 0xFF && c != 0x89)
                continue;

            // Create thumbnail:
            image = [[UIImage alloc] initWithData:data];
        }
        else if([data isKindOfClass:[UIImage class]])
        {
            image = [[arrayBoardImages objectAtIndex:i] retain];
            data = nil;
        }
        else
            continue; // unsupported type.

        if(data == nil)
        {
            NSData* newData = [UIImageJPEGRepresentation(image, 0.4) retain];

            [imageItems addObject:newData];
            [newData release];
        }
        else
        {
            [imageItems addObject:data]; // do not release clipboard data
            [image release];
        }
    }
}

```

```

    // NOTE: at this point imageItems contains NSData with JPEG images to save
and use in your application.
    //
    // YOUR CODE HERE MAY BE SIMILAR TO THIS:

    // for(int myIndex=0; myIndex<imageItems.count; myIndex++)
    // {
    //     NSString* myFilePath = @"samplePath/myFileName.jpeg";
    //     [(NSData*)[imageItems objectAtIndex: myIndex] writeToFile:
myFilePath atomically:YES];
    // }

    // Once the images are saved or used, it is good to clear clipboard so user
does not get
    // unexpected clipboard content using Paste next time:

    // This clears pasteboard:
    [pasteboard setString:@""];
}

```

```

// PDF version:
+ (void)pastePDFfromPasteboard
{
    UIPasteboard *pasteboard = [UIPasteboard generalPasteboard];

    if( [pasteboard containsPasteboardTypes:@[ (NSString*)kUTTypePDF ] ] )
    {
        NSData *data = [pasteboard dataForPasteboardType:(NSString *)
kUTTypePDF];
        CFDataRef myPDFData = (CFDataRef)data;
        CGDataProviderRef provider = CGDataProviderCreateWithCFData(myPDFData);

        CGPDFDocumentRef pdf = CGPDFDocumentCreateWithProvider(provider);

        // Do something with PDF document here...

        // -----
    }
}

```

```

    // Once the images are saved or used, it is good to clear clipboard so
user does not get
    // unexpected clipboard content using Paste next time:

    // This clears pasteboard:
    [pasteboard setString:@""];
}

```

• • •

Please email us at support@turboscanapp.com if you have any questions or suggestions.

Thank you!