User Registration Integration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 0.1 | 13-06-2015 | A Viswaradh Reddy | All | Initial draft |
| 0.2 | 28-08-2015 | A Viswaradh Reddy | 3.2.2.5 | HSDP configuration |
| 0.3 | 28-09-2015 | Adarsh Rai | All | Updated document |
| 0.4 | 04-12-2105 | A Viswaradh Reddy | All | Updated Configuration |
| 0.5 | 05-02-2016 | Adarsh Rai | 3.2 | Updated Configuration |
| 0.6 | 03-03-2016 | Harsh Hungund | 3.2 | Updated Configuration |
| 0.7 | 01-04-2016 | Adarsh Rai | 4 | Parental Access |
| 0.8 | 14-07-2016 | Harsh Hungund | 2 | Updated Dependencies and Configuration |
| 0.9 | 16-09-2016 | Harsh / Viswaradh | 3,6,7 | Implemented new Initialisation |
| 0.10 | 02-12-2016 | Viswaradh | All | Updated Configuration |
| 0.11 | 27-02-2017 | Harsh / Viswaradh | All | Updated Configuration |
| 0.12 | 28-03-2017 | Sai Pasumarthy | 3.2.24 | Updated Configuration |
| 0.13 | 02-05-2017 | Sai Pasumarthy | 4 | OptInInSeparateScreen |

|  |  |
| --- | --- |
| **Author** | A Viswaradh Reddy |
| **Approved by** |  |
| **Email Id** | Viswarad.reddy@philips.com |

Table of Contents

[1. INTRODUCTION 3](#_Toc481516447)

[2. INTEGRATION 3](#_Toc481516448)

[1.1 Pod File Integration 3](#_Toc481516449)

[1.2 Framework Integration (Sub module) 3](#_Toc481516450)

[1.3 Dependencies 4](#_Toc481516451)

[1.3.1 PhilipsUIKit 4](#_Toc481516452)

[1.3.2 AppInfra 4](#_Toc481516453)

[1.4 Demo App 5](#_Toc481516454)

[3. INITIALIZATION 5](#_Toc481516455)

[1.5 Navigation 6](#_Toc481516456)

[1.6 Configuration 6](#_Toc481516457)

[1.6.1 URLaunchInput 6](#_Toc481516458)

[1.2.2 AppConfig.json 9](#_Toc481516459)

[3.3 DIRegistrationConfigurationDelegate 14](#_Toc481516460)

[4. A/B Testing for Receive Marketing Opt in Flow 14](#_Toc481516461)

[5. Social Providers’ Configuration 16](#_Toc481516462)

[6. Parental Access 17](#_Toc481516463)

[7. Version 18](#_Toc481516464)

[8. Things to take care 18](#_Toc481516465)

[9. Legal Notice for WeChat: 18](#_Toc481516466)

[10. Known Issues 18](#_Toc481516467)

# INTRODUCTION

This document provides an overview of integration of Registration feature in Mobile applications.

Source Path: <http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/_git/usr-ios-user-registration>

Also refer to the Apple Docs that are included in the package.



# INTEGRATION

## Pod File Integration

The easiest and preferred way to use these components is using [Cocoapods](https://cocoapods.org) (version 1.0.0 or newer).

If you don’t already have cocoapods, please follow below steps to install cocoapods:

1: [How to install cocoapods](http://nshipster.com/cocoapods/#installing-cocoapods)

2: [How to add Podfile to your project](http://nshipster.com/cocoapods/#podfile)

In your podfile, you have to add a source line for the Philips internal pod spec repository.

Open you podfile using command open -a Xcode Podfile

Add following to your podfile:

source 'https://github.com/CocoaPods/Specs.git'

source 'https://bitbucket.atlas.philips.com/stash/scm/ehpmat/cocoapod-specs.git'

pod 'PhilipsRegistration', '~> 9.2.0'

Close your podfile and install the library using command pod install

## Framework Integration (Sub module)

1. Add PhilipsRegigstration as a git submodule by typing the following command on the command line:

git submodule add <http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/_git/usr-ios-user-registration> hor-registration-ios

1. Now, still on the command line, go to the hor-registration-ios folder and retrieve the list of tags that are available:

git tag –list

1. Let's say the tag that you want to use is 9.2.0 (you should probably simply use the latest), switch to that tag by typing:

git checkout tags/9.2.0

1. Check out the dependent Projects

PhilipsUIKit

LocaleMatch

UAPPFramework

AppInfra

1. Open your project's workspace in Xcode. Add the Registration.xcodeproj project as a subproject of your own project.
2. Go to the Build settings of your target and find the "User header Search Path". Add an entry to the user header search path containing the value $(SOURCE\_ROOT)/hor-registration-ios
3. Add Registration and Registration Resources Targets to the project.

## Dependencies



### PhilipsUIKit

The registration component uses the PhilipsUIKit for the UI components.

So the applications that are already using PhilipsUIKit need not do anything.

Applications that are not using the PhilipsUIKit have to assign the theme, else it will use the default blue theme.

Configuring the theme:



[PhilipsUIKit Documentation](http://sesvm-webserver.htce.nl.philips.com/PhilipsUIKit/ios/latest/Documentation/)

### AppInfra

The PhilipsAppTagging and Logging are included in the AppInfra library .

[AppInfra Documentation](https://atlas.natlab.research.philips.com/bitbucket/projects/MAIL/repos/app-infra_ios/browse/Documents/External/MobileAppInfra_PI16.3_v1.1.0_Integration_IOS.docx?at=refs%2Fheads%2Frelease)

## Demo App

Please find demo app in: hor-registration-ios/Demo App

# INITIALIZATION

1. Add the “hor-registration-ios/Registration Binary” to the project. It contains all the libraries which re required for registration library.
   1. PhilipsRegistration
   2. Locale match
   3. PhilipsUIKit
   4. AppInfra
   5. NHNetworkTime
   6. PhilipsIconFont
   7. UAPPFramework

Before initializing any UI or calling any other API, the URInterface class must be initialized using the initializer as shown below:

URInterface \*urInterface = [[URInterface alloc]initWithDependencies:self.urDependencies andSettings:nil];

The registration UI can be initialized using following method:



Ex:



Please refer to [URLaunchInput](#_Configuration) for the registration configuration.

## Navigation

PhilipsRegistration does not provide any navigation. The method mentioned above, when called returns the appropriate ViewController for given flow. Returned view controller must be added to a navigation controller (created by applications).

## Configuration

The configuration contains mainly two parts:

1. URLaunchInput
2. AppConfig.json

### URLaunchInput

1. DIRegistrationUIConfiguration
2. DIRegistrationFlowConfiguration
3. DIRegistrationContentConfiguration

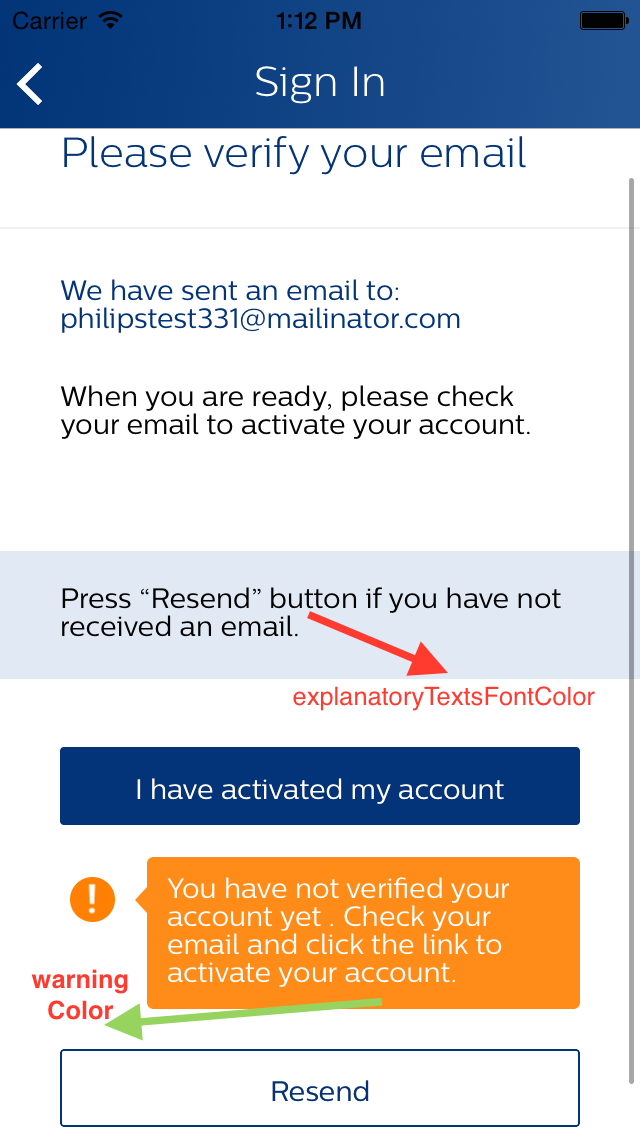
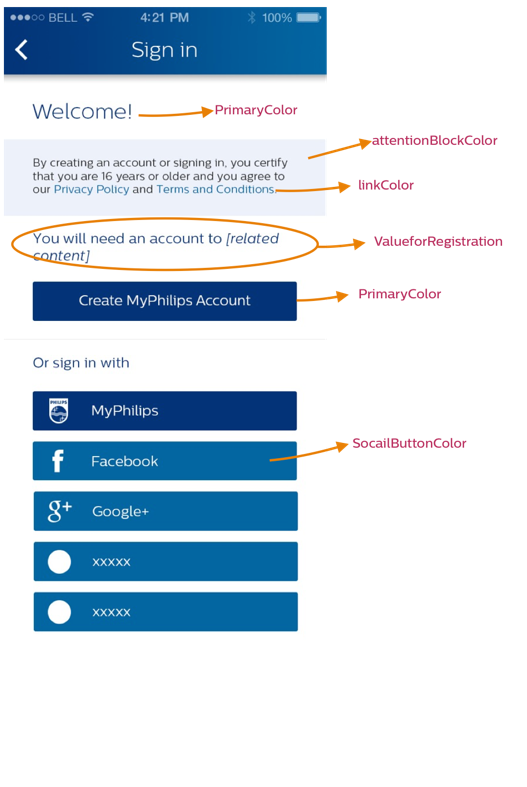
#### DIRegistrationUIConfiguration

All the UI components are from the PhilipsUIKit. Please refer to the PhilipsUIKit documentation for any customizations w.r.t UI.

Below are the additional UI components which are customizable w.r.t User Registration.







#### DIRegistrationFlowConfiguration



#### DIRegistrationContentConfiguration

This class has two attributes that can be configured as required:



### AppConfig.json

The AppConfig.json should be added to the project and json need to be updated depending on the project.

1. **JanrainConfiguration**
2. PILConfiguration
3. Flow
4. HSDPConfiguration(Runtime only)
5. SigninProviders
6. AppIdentity

**Note:** All the data required in AppConfig.json can also be configured dynamically using public library AppInfra**.** For further details on accessing the properties dynamically and updating their values, please refer APPInfra documentation. All the mandatory and optional fields remain same way. They can either be configured in json or via AppInfra APIs in code. If any field configured both way, the value provided via code will override the json provided value. For security reasons it is recommended to use dynamic configuration using AppInfra instead of AppConfig**.json**. For the dynamic use always use the set property for appinfra

Ex: [self setPropertyForKey:Property value:value]//todo

**EX: Minimum Configuration:**



**Ex: Extended configuration**

****

#### JanRainConfiguration (Mandatory)

JanRainConfiguration is required to be able to connect to janrain services. It requires following fields:

**RegistrationClientID:** Assign mandatory client id for each environment.

For e.g. “Development”: “xxxxxxxxxx”

Please contact **Digital Services <digital.services@philips.com>** for Client ID for janrain and Microsite ID

#### PILConfiguration (Mandatory)

PIL configuration requires following three elements:



Note: Please contact **Digital Services <**[**mailto:digital.services@philips.com**](mailto:digital.services@philips.com)**>** for Client ID for janrain and Microsite ID

#### Flow (Optional)



#### HSDPConfiguration (Required for the HSDP supported Apps)

HSDPConfiguration is required to use HSDP services. The elements of HSDP configuration are as below. All the elements are environment specific. If configuration were not provided for any of the environments, application would bypass any communication to HSDP server for that environment. Elements for any environment cannot be provided partially. Also HSDP configuration cannot be provided via JSON file. It is mandatory to provide via runtime Only.



To set/update the values of these params, AppInfra APIs need to be accessed.

The specific keys for each of the params is as shown in the table above. For further details, please refer AppInfra Documentation.

**Secret Key**

@{@"CN":@"<SecretKey>",@"default":@"<SecretKey>"}

**SecretKey** : It should be Whitebox signer API secret key and it will contain 128 characters.

**How to get White Box signer API Secret Key:**

With respect to the operational aspects the workflow is as follows:

1.       The business in question obtains the required HSDP API signing key material (\*)

(production, staging, test) from HSDP DevOps.

2.       The business uses the [Philips Secure Data Transfer](https://www.sdt.philips.com/) website to send the HSDP API signing key material (\*)

to CDP2 ([Maarten Muijen](mailto:maarten.muijen@philips.com)).

3.       Within 5 working days the transformed HSDP API signing key material (\*\*)

will be returned to the original sender again using [Philips Secure Data Transfer](https://www.sdt.philips.com/).

(\*) The HSDP API signing key material shall be provided as a text encoded file: as provided by HSDP (text file).

In addition, the request shall include meta-data for: 1) the business requesting and 2) type of key (testing/staging/production).

(\*\*) The Transformed HSDP API signing key material: 64-byte binary encoded file.

#### SigninProviders

All the sign-in providers should be added to the default list of sign-in providers. If country specific providers are required, a separate list of providers with country code should be added. Please refer to the example for more details.

Sign-in Providers supported by registration currently:

****

Note: Please refer to the demo app for more configurations

## DIRegistrationConfigurationDelegate

Below mentioned delegate methods should be implemented to load Terms & Conditions and Privacy Policy when respective one is tapped in registration component:



# A/B Testing for Receive Marketing Opt in Flow

With aim of increasing number of user opt-ins, we are doing A/B testing of marketing opt-in feature. For that reason, we provide different user experiences of marketing opt-in to different group of users to find out which one is more engaging for the users.

For ex. For one group of users we display marketing opt-in option on create account screen while to another group of users we display option of marketing opt-in immediately after account creation. Similarly, we provide three different experiences for marketing opt-in via A/B testing.

Currently, selection of user group for a particular experience is random and is decided by mbox server managed by Analytics team.

**Configure A/B testing for ReceiveMarketingOptIn:**

Proposition need to add the **“DOT-ReceiveMarketingOptIn**” Experience to the appconfig under app infra for "abtest.precache" array.

Ex:

"appinfra": {

"abtest.precache":["DOT-ReceiveMarketingOptIn"]

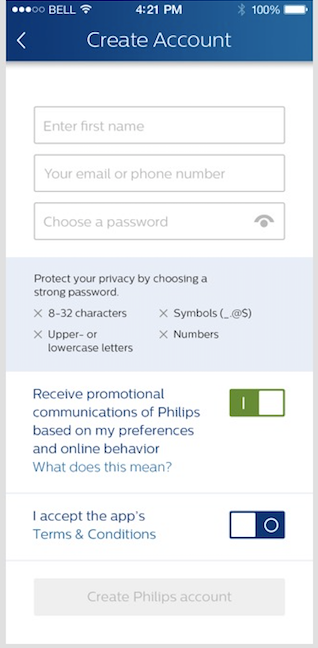
}

Application need to update the cache for the appinfra



**There are three flows:** Default flow is OriginalOptInText

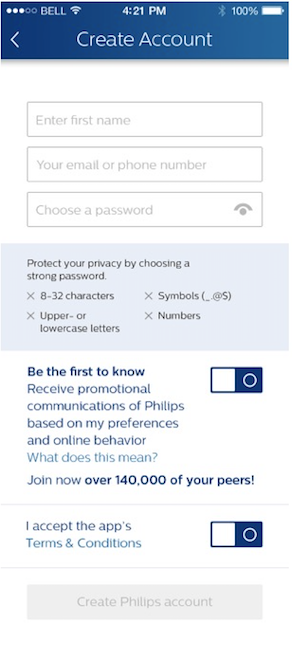
1. OriginalOptInText: Its same a old flow



1. OptInInSeparateScreen : This option will show after create account screen where the user can read more details and optin



1. AddBenefitsToOptInInRegistrationScreen : Social proof has been tested as a strong persuasion element , so this contain number of user has optedin (currently this value is hardcoded)



# Social Providers’ Configuration

**WeChat** : [Confluence page](https://confluence.atlas.philips.com/display/UR/How+to+integrate+the+WeChat+in+the+Mobile+apps)

1) Request the WeChat client id from DS team, by raising a ticket.

2) Download WeChat SDK at the following site :

[http://open.wechat.com/cgi-bin/newreadtemplate?t=overseas\_open/docs/mobile/getting-started/ios - getting-started\_ios](http://open.wechat.com/cgi-bin/newreadtemplate?t=overseas_open/docs/mobile/getting-started/ios#getting-started_ios)

Reference :<http://open.wechat.com/cgi-bin/newreadtemplate?t=overseas_open/docs/mobile/getting-started/ios#getting-started_ios>

Once downloaded, add this SDK to your project.

3) Following parameters for WeChat configuration should be provided at run time only and not via AppConfig.json:

"WeChatAppId":"<wechat\_app\_id>",

"WeChatAppSecret":"<wehchat\_app\_secret>"

4) Initialize the WeChat component before launching the registration as shown below:

[WXApi registerApp:@"<wechat\_app\_id>"];

5) Add the following param to application’s info plist:

LSApplicationQueriesSchemes : <array><string>weixin</string>/<array>

6) Add to URL types the following:

Identifier : weixin

URL Schemes : <wechat\_app\_id>

7) Implement the following callback in appdelegate of your app:



**Google+**

1) Request the Google+ client id from DS team, by raising a ticket per environment.

2) Add the following params for Google+ configuration in appconfig.json:

"GooglePlusClientId":"<googleplus\_client\_id>",

        "GooglePlusRedirectUri":"<googleplus\_redirect\_url>”

3) Implement the following callback in appdelegate of your app:

4)Add to URL types the following:

URL Schemes : <googleplus\_redirect\_url>

# Countries List Configuration

* 1. Propositions can optionally configure the countries list to be displayed before registering/logging-in by adding below key into AppConfig.json:

"supportedHomeCountries": ["RW","BG","CZ","US","CN"]

* 1. If propositions don’t configure this value, all countries supported by platform will be displayed.
  2. Propositions can also optionally configure the fallback country (for cases where Service Discovery returns country that is not supported) by setting the below key into AppConfig.json.

"fallbackHomeCountry":"US"

* 1. If fallbackHomeCountry not provided, **US** is taken as default fallback country.

**Note :**

* The fallback country should be there in the supportedHomeCountries if provided or in platform supported countries if supportedHomeCountries not provided.
* Only countries that are supported by platform from supportedHomeCountries will be displayed.

# Parental Access

The parental access and Coppa flow have now been entirely implemented inside the User Registration component (pushing out mail based consent system). For more details on what it is and how it works, please check Coppa Flow document, Release document and apple docs.

We have added a configuration parameter laucnhFlow in DIRegistrationFlowConfiguration for verticals to request display Parental Consent flow to user in case of US locale and its been 24hrs since first consent is given or if user has denied the consent and vertical wants to request the consent again.

# Version

To read the version number of the library the API is:

[DIRegistrationVersion version]

# Things to take care

* User will be logged Server is throwing the invalid refresh token error due to more number of user session or time stamp issue(fixed but in case the NTP server is not updated). It will trigger the -(void)loginSessionRefreshFailedAndLoggedout delegate.
  + When its triggered , Request the user to login again.
* As we are storing the few strings in the key chain , from simulator in IOS 10 onwards you need to keyChain sharing capability to the app.

# Legal Notice for WeChat:

As few features require legal notice inside apps under the Legal notice.

* For Wechat please add :
  + “Powered by 微信开放平台”或“技术由微信开放平台提供”

# Known Issues

* Reset password requires user to enter email in mobile and website.
* Merge is not possible If the social provider (like facebook or googleplus) not providing the email address after authentication.
* One Click Registration is not Implemented.
* Email Verification some times leads to invalid parameters or will show blank response(Currently tracking with IT to fix).
* Only Facebook and Google+ only tested E2E.
* Currently only Login via social login API, Logout and Refresh Session API’s only implemented in HSDP.
* User can’t set the password if he created the account via Social.
* User with corrupted data can’t login (same as web)
* User can’t know which provider is registered previously . when he try to create account again.
* Acceptance Environment will not work for first 3days (in every alternate week ) due to Load and performance tests.
* While refreshing if there is bad network , user may need to sign in again.
* The Terms and Conditions are stored in the app. Once the user login to different device or uninstall the app, User need to accept the T&C again.