Digital Care Android Integration

|  |  |
| --- | --- |
| Author | Sampath kumar |
| Approved by | [Bhargavi.Upadhya@philips.com](mailto:Bhargavi.Upadhya@philips.com) |
| Reviewed by | deepthi.shivakumar@philips.com |
| Email Id | Sampath.kumar@philips.com |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 1.0 | 10-07-2015 | Deepthi Shivakumar | All | Release 2 features |
| 1.1 | 10-08-2015 | Deepthi Shivakumar | Modified Integration section | Modified gradle dependencies section and added more clarity for initialization steps. |
| 1.2 | 30-09-2015 | Deepthi Shivakumar | Modified Integration section | Modified bazaar voice related integration APIs. |
| 1.3 | 04-12-2015 | Deepthi Shivakumar | Modified Integration section | Modified changes for 15.2 PI release. |
| 1.4 | 01-03-2016 | Deepthi Shivakumar | All | Modified changes for 16.1 PI release. |
| 1.5 | 07-04-2016 | Ritesh Jha | All | Modified changes for 16.1 PI release. |
| 1.6 | 27-04-2016 | Ritesh Jha | All | HotFix on v4.1.0 (16.1 PI). URL formation logic is changed for FAQ |
| 1.7 | 16-05-2016 | Ritesh Jha | All | InApp FAQ feature |
| 1.8 | 06-07-2016 | Naveen AH | Library Integrating, Proguard Rules | Developer Mode Support |
| 1.9 | 22/06/2016 | Naveen AH | None | Bug Fixing, Category & Subcategory fix. |
| 2.0 | 20/07/2016 | Naveen AH | Library Integration & Progaurd | APPInfra Component Integration. Volley library removal. Tagging Library Removal. |
| 2.1 | 30/09/2016 | Sampath kumar | All | OneRoofChanges, bug fixes |
| 2.2 | 29/11/2016 | Sampath kumar | All | Android N changes, Memory leak fixes |
| 2.3 | 10/02/2017 | Sampath Kumar | All | Service discovery integration, China support, Memory leak fixes |
| 2.4 | 20/03/2017 | Sampath Kumar | All | FAQ Feature with Tutorials and Troubleshooting implemented |
| 2.5 | 27/04/2017 | Sampath kumar |  | FAQ Feature with Tutorials and Troubleshooting,  Defect fixes, PSRA defect fixes |
| 2.6 | 06/07/2017 | Sampath |  | DemoMicroApp Changes, bug fixes |

Table of Contents

[1. INTRODUCTION 4](#_Toc481751155)

[2. INTEGRATION 4](#_Toc481751156)

[2.1 Maven repository (Artifactory based) Integration 4](#_Toc481751157)

[2.2 Library Integration 4](#_Toc481751158)

[2.3 Library versioning 4](#_Toc481751159)

[2.4 Root gradle changes 5](#_Toc481751160)

[2.5 Gradle dependencies 5](#_Toc481751161)

[2.6 Proxy dependencies 5](#_Toc481751162)

[2.7 Configuration File 5](#_Toc481751163)

[2.8 Prerequisites 6](#_Toc481751164)

[3. INITIALIZATION 6](#_Toc481751165)

[4. How to invoke Consumer care as activity? 7](#_Toc481751166)

[5. How to invoke Consumer care as Fragment? 8](#_Toc481751167)

[6. Android Manifest Changes 9](#_Toc481751168)

[7. Consumer care configuration 10](#_Toc481751169)

[7.3 Social provider configuration 14](#_Toc481751170)

[8. Handling consumer care as fragments: 15](#_Toc481751171)

[9. Handling consumer care as activity 15](#_Toc481751172)

[10. Vertical features customization 15](#_Toc481751173)

[11. Action bar customization 18](#_Toc481751174)

[12. Supporting apps with Over 65K Methods 18](#_Toc481751175)

[13. Supported Languages 19](#_Toc481751176)

[14. Proguard Rules 19](#_Toc481751177)

[15. Frequently asked Questions 21](#_Toc481751178)

[a. How about customising fonts and sizes? 21](#_Toc481751179)

[16. Notes 23](#_Toc481751180)

# INTRODUCTION

This document provides an overview of integration procedure for consumer care library in android mobile applications.

Consumer care library provides set of standardized consumer care support features which can be used in all Philips apps.

# INTEGRATION

There are two ways to integrate “Digital Care” library with any Android app.

* + 1. **Maven repository based**: At compile time, machine must be connected with Philips network. Do follow section 2.1
    2. **Library Integration**: If unable to connect with Philips network then include libraries to your root application. Do follow section 2.2

## Maven repository (Artifactory based) Integration

The easiest and preferred way to use these components is using maven.

All dependent libraries should be downloaded from artifactory.

**Artifactory Path:**

<http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/simple/libs-release-local-android/com/philips/cdp/digitalCare/7.5.0/>

If you are inside Philips network then you can directly refer “**3.3 Gradle dependencies**” section. It will automatically download all nested dependencies from artifactory.

## Library Integration

Need to copy all aar files in libs folder; below are the libraries needed, please make gradle changes

dependencies {

compile fileTree(dir:'libs', include:['\*.jar'])

compile **'com.android.support:appcompat-v7:+'**

compile**(name:'digitalCare-v7.5.0', ext:aar)**

}

}

## Library versioning

Library version can be obtained by using below API

DigitalCareManager.getInstance().getDigitalCareLibVersion();

## Root gradle changes

buildscript {  
 repositories {  
 **maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }  
 }** dependencies {  
 classpath **'com.android.tools.build:gradle:2.3.1'** }  
}  
  
allprojects {  
 **repositories {  
 maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/libs-release-local-android' }  
 maven { url 'http://maartens-mini.ddns.htc.nl.philips.com:8081/artifactory/jcenter' }  
 }**}

## Gradle dependencies

Just by adding below gradle dependencies, digitalcare and nested possible libraries will be downloaded from artifcatory. But it has to be inside Philips network.

dependencies {  
compile(group: **'com.philips.cdp'**, name: **'digitalCare'**, version: **'7.5.0'**, ext: **'aar'**){

transitive = true

}

}

## Proxy dependencies

Gradle dependencies can get some network/proxy related issues. In order to fix this issue, we are using below proxy settings in gradle.properties of root folder.

**systemProp.https.proxyHost**=**42.99.164.34  
systemProp.https.proxyPort**=**10015**

We are using this proxy settings locally. But Eindhoven, does not use above proxy settings.

## Configuration File

1. Go to sample app’s res/values folder and copy digitalcare\_config.xml and place it in

integrating app’s res/values folder.

1. Override the values in config file and define the features as per requirement.
2. Add AppConfig.json in assets folder. It contains the information for microsite ids, service discovery environment, sectors etc.., For more details find the App Infra document in the below link :

<http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/TEAM%20Sabers/_git/ail-android-appinfra?path=%2FDocuments&version=GBdevelop&_a=contents>

## Prerequisites

**Note:** Product should be hosted at PRX server. Please read the process in doc **“ConsumerCare\_Deployment”** which is shared along with other documents**.**

* + 1. Application need to know **CTN** number for each product used in app. Ex: HD8967/01. Max CTN is allowed.
    2. Sector
    3. Catalog.

Example: **Sector**: B2C and **catalog**: CARE (Mandatory for Consumer care)

Most of the apps will have above sector and catalog.

**Note:** Above information is used for backend services and mostly uses PRX system. Hence Please provide valid and complete CTN number as input otherwise consumer care features are not functional as expected.

# INITIALIZATION

The library can be invoked both as activity and fragment.

“**DigitalCareConfigManager**” is singleton class used for communication between application and library. It exposes few public setter APIs. The application has to set few below parameters dynamically before the library is invoked. Please refer sample application.

1. Import com.philips.platform.uappframework.launcher.FragmentLauncher;
2. Import DigitalCareConfigManager from “com.philips.cdp.digitalcare”
3. Call **APPInfra** related APIs as below,

AppInfraInterface mAppInfraInterface = **new A**ppInfra.Builder().build(getApplicationContext());

**“ADBMobileConfig.json”:** This json file is tagging configuration file which has to be kept at root app asset level. Please don’t forget to add this. Otherwise by default it will consider digitalcare’s debugging tagging configuration.

**Please follow the order of initialization as per sample app.**

# How to invoke Consumer care as activity?

Please note that there is an API change for invocation. Now there is a single API to launch as activity and fragment.

Here UiLauncher is a base class which has 2 sub classes called ActivityLauncher and FragmentLauncher

ProductSelection type denotes the kind of inputs to support multiple products.

**Please note that we currently support hardcoded list of products.**

1. Create an instance of activity launcher and insert activity orientation type and UiKitTheme as parameters. Also create productselection type as below,

HardcodedProductList productsSelection = **new** HardcodedProductLis t(ctnList);  
productsSelection.setCatalog(Catalog.***CARE***);  
productsSelection.setSector(Sector.***B2C***);

ActivityLauncher activityLauncher =  
 **new** ActivityLauncher  
 (com.philips.platform.uappframework.  
 launcher.ActivityLauncher.  
 ActivityOrientation.SCREEN\_ORIENTATION\_UNSPECIFIED, R.style.Theme\_Philips\_BrightBlue\_Gradient\_WhiteBackground);  
activityLauncher.setCustomAnimation(R.anim.slide\_in\_bottom, R.anim.slide\_out\_bottom);

CcInterface ccInterface = **new** CcInterface();  
**if** (**ccSettings** == **null**) **ccSettings** = **new** CcSettings(**this**);  
**if** (**ccLaunchInput** == **null**) **ccLaunchInput** = **new** CcLaunchInput();  
**ccLaunchInput**.setProductModelSelectionType(productsSelection);

ccLaunchInput.setLiveChatUrl(“live chat url”); //optional

CcDependencies ccDependencies = **new**

CcDependencies(mAppInfraInterface);  
  
ccInterface.init(ccDependencies, **ccSettings**);  
ccInterface.launch(activityLauncher, **ccLaunchInput**);

Note: 1. ctnlist is string array of CTN numbers

# 2. setLiveChatUrl() is an optional API provided for those who wants to set their app own url. If it is a static url u can mention in digitalcare\_config.xml. If you want to change dynamically depends upon the country selection, you can use this API.

3. Eg. Live chat URL : "http://phchina.livecom.cn/webapp/index.html?app\_openid=ph\_6idvd4fj&token=PhilipsTest";

Note : Integrating App should append a unique value (token) to the URL as query parameter. Livecom backend receives this token inorder to maintain the unique user chat history between the platforms and the devices. Eg. Token can be HTTP UUID. If this token not set, user will see the chats of others.

# How to invoke Consumer care as Fragment?

Please note that there is an API change for invocation. Now there is a single API to launch as activity and fragment.

Here UiLauncher is a base class which has 2 sub classes called ActivityLauncher and FragmentLauncher

ProductSelection type denotes the kind of inputs to support multiple products.

**Please note that we currently support hardcoded list of products.**

1. Create an instance of activity launcher and insert activity orientation type and UiKitTheme as parameters. Also create productselection type as below,

HardcodedProductList productsSelection = **new** HardcodedProductList(ctnList);  
productsSelection.setCatalog(Catalog.CARE);  
productsSelection.setSector(Sector.B2C);

FragmentLauncher launcher = **new** FragmentLauncher(**this**, R.id.sampleMainContainer, **actionBarListener**);

launcher.setCustomAnimation(R.anim.slide\_in\_bottom, R.anim.slide\_out\_bottom);

CcInterface ccInterface = **new** CcInterface();  
**if** (**ccSettings** == **null**) **ccSettings** = **new** CcSettings(**this**);  
**if** (**ccLaunchInput** == **null**) **ccLaunchInput** = **new** CcLaunchInput();  
**ccLaunchInput**.setProductModelSelectionType(productsSelection);

**ccLaunchInput**.setConsumerCareListener(**this**);

ccLaunchInput.setLiveChatUrl(“live chat url”); //optional

CcDependencies ccDependencies = **new** CcDependencies(mAppInfraInterface);  
  
ccInterface.init(ccDependencies, **ccSettings**);  
ccInterface.launch(launcher, **ccLaunchInput**);

Note: a. ctnlist is string array of CTN numbers

1. setLiveChatUrl() is an optional API provided for those who wants to set their app own url. If it is a static url u can mention in digitalcare\_config.xml. If you want to change dynamically depends upon the country selection, you can use this API.
2. Eg. Live chat URL : "http://phchina.livecom.cn/webapp/index.html?app\_openid=ph\_6idvd4fj&token=PhilipsTest";

Note : Integrating App should append a unique value (token) to the URL as query parameter. Livecom backend receives this token inorder to maintain the unique user chat history between the platforms and the devices. Eg. Token can be HTTP UUID. If this token not set, user will see the chats of others.

ActivityOrientation is an enum to specify orientation, if do not want to fix orientation, please set as “unspecified”

Supports all kind of orientation in ActivityInfo class.

ActivityOrientation is present inside DigitalCareConfigManager class.

Animation resource id can be passed in as zero if not relevant.

Animation resource id can be passed in as zero if not relevant.

# Android Manifest Changes

1. If apps face problems wrt label , theme etc during compilation, Please do the below,

Example from Registration sample app:

***<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.philips.cdp.registration.sample">  
  
 <application  
 android:name="com.philips.cdp.registration.sample.RegistrationApplication"  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:theme="@style/AppTheme"  
 tools:replace="android:icon,,android:label,android:theme">***

1. Add **android:configChanges="screenSize|smallestScreenSize|screenLayout|layoutDirection”** in all the activities
2. This change is required for Locale Philips Near you feature to display google maps. Please find more details below regarding debug and release keys which need to be used.

Application needs to copy below google map related meta data information in app’s manifest file and must provide new SHA key for rendering Philips service centers on google map.

<meta-data

android:name="com.google.android.gms.version"

android:value="@integer/google\_play\_services\_version" />

<meta-data  
android:name="com.google.android.maps.v2.API\_KEY"  
android:value="AIzaSyDm1M6rUwuCe\_-4pBM61QeWivE6GIu2hWM" />

<uses-feature

android:glEsVersion="0x00020000"

android:required="true" />

**AIzaSyDm1M6rUwuCe\_-4pBM61QeWivE6GIu2hWM = SHA key**

Reference for SHA key generation

<https://developers.google.com/maps/documentation/android/signup>

SHA key should be generated from the system where apk is released.

SHA key along with package name should be registered.

//Credentials For LocatePhilips Near you feature (Google Map)

<usespermission android:name="com.philips.cdp.sampledigitalcareapp.permission.MAPS\_RECEIV E" />

<permission

android:name="com.philips.cdp.sampledigitalcareapp.permission.MAPS\_RECEIVE"

android:protectionLevel="signature" />

**Please note that it works without the above permission for debug keys but we need to add it when we use release keys. Please make sure to change package name.**

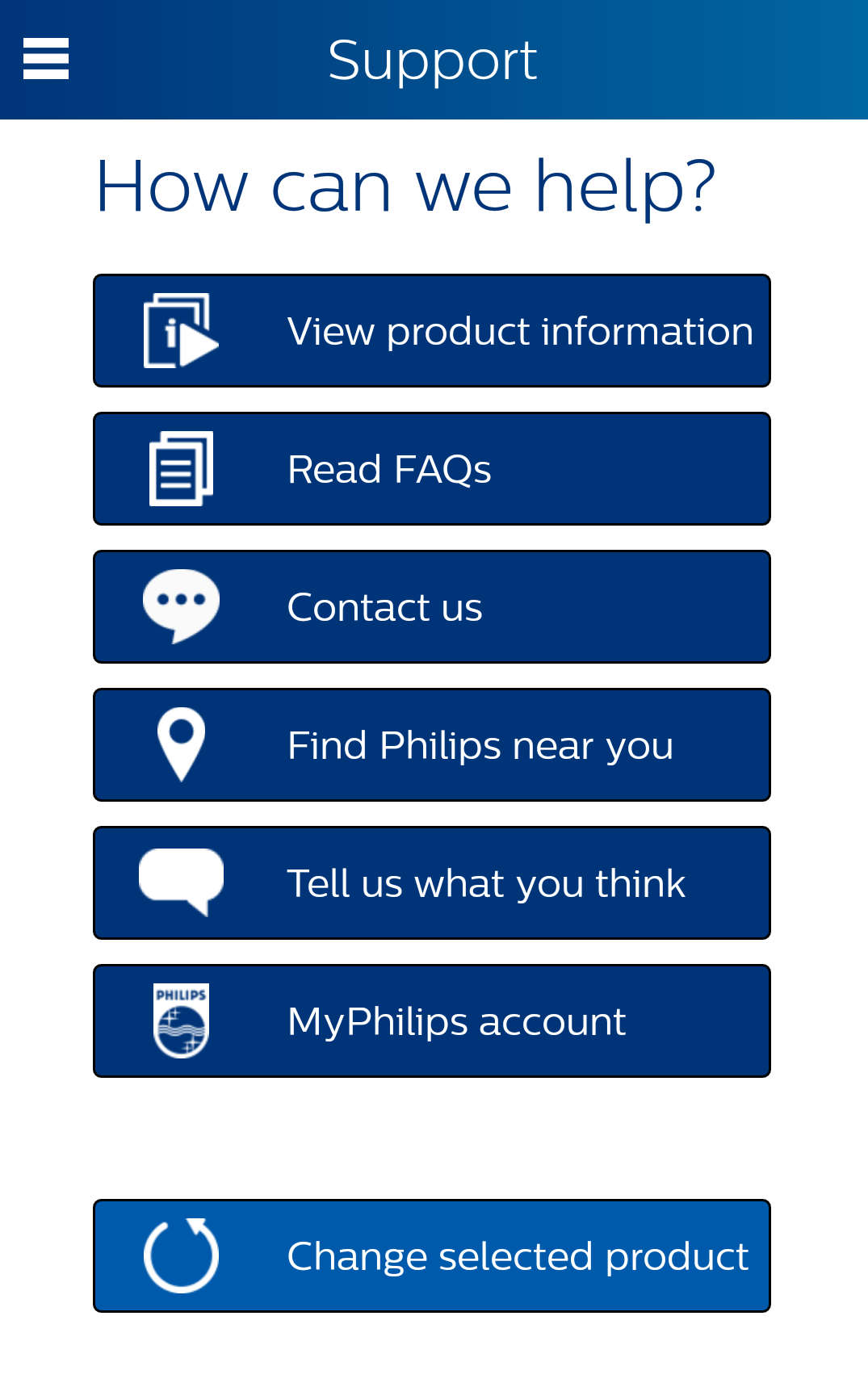
# Consumer care configuration

The application has to copy digitalcare\_config.xml from sample app resource folder and create one on app side by overriding the parameters present in it. The file contains the following,

1. UI resources like themes, styles, background colors and action bar customization.
2. Facebook &Twitter Philips product page url which is app/product specific.
3. An array is defined for customizing main menu title, main menu icons, product menu title and social provider list. Each menu contains default features supported by Consumer Care library. App can include only the required features by adding/deleting/modifying the array items.
4. Each title string in above arrays is expected to be a resource key and not the value.

The digitalcare\_config.xml on the library side will have default values. In case the app does not override, all the resources would be picked from library. Main menu configuration.

#### 7.1 Main menu configuration



**main\_menu\_resources**

**activity\_background\_color**

**Product Selection Feature**

**button\_register\_background**

**actionbar\_title\_Color**

**text\_header\_color**

**main\_menu\_title**

**button\_background**

**actionbar\_background\_start\_color**

**actionbar\_background\_end\_color**

**Actionbar\_hamburger\_menu**

**button\_text\_color**

#### 7.2 Product menu configuration



**product\_menu\_title**

**actionbar\_back\_button**

## Social provider configuration



**text\_sub\_header\_color**

**social\_service\_provider\_menu\_resources**

**social\_service\_provider\_menu\_title**

**actionbar\_back\_button**

# Handling consumer care as fragments:

If it is invoked as fragment, action bar has to be handled by verticals because they have control.

1. Adding cc fragments will be taken care by library.We define one interface called “ActionBarListener” and it will have method called updateActionBar (String titleActionbar, Boolean actionBarLeftArrow).
2. We invoke this callback and verticals have to implement this listener and update the action bar accordingly.
3. On click of back arrow app has the control again and it is responsible for popping out the fragment from the stack. Consumer care has no action over it.
4. Handling hamburger menu icon and anything related to action bar is the responsibility of the app.

# Handling consumer care as activity

If it is invoked as activity, then CC will be launched as full screen and also responsible for updating action bar. Action bar customization happens depending on the configuration set by app in config xml file

# Vertical features customization

The consumer care features are highly customizable; each vertical can add any specific button inside support menu if it is really required.

1. App can add its own button by passing valid string for button title and icon resource name in configuration file.
2. Application has to implement interfaces called CcListener whers it contains the following override methods
3. onMainMenuItemClicked

Invoked on click of each button on support home page. This helps in configuring features on main/home page.

1. onProductMenuItemClicked

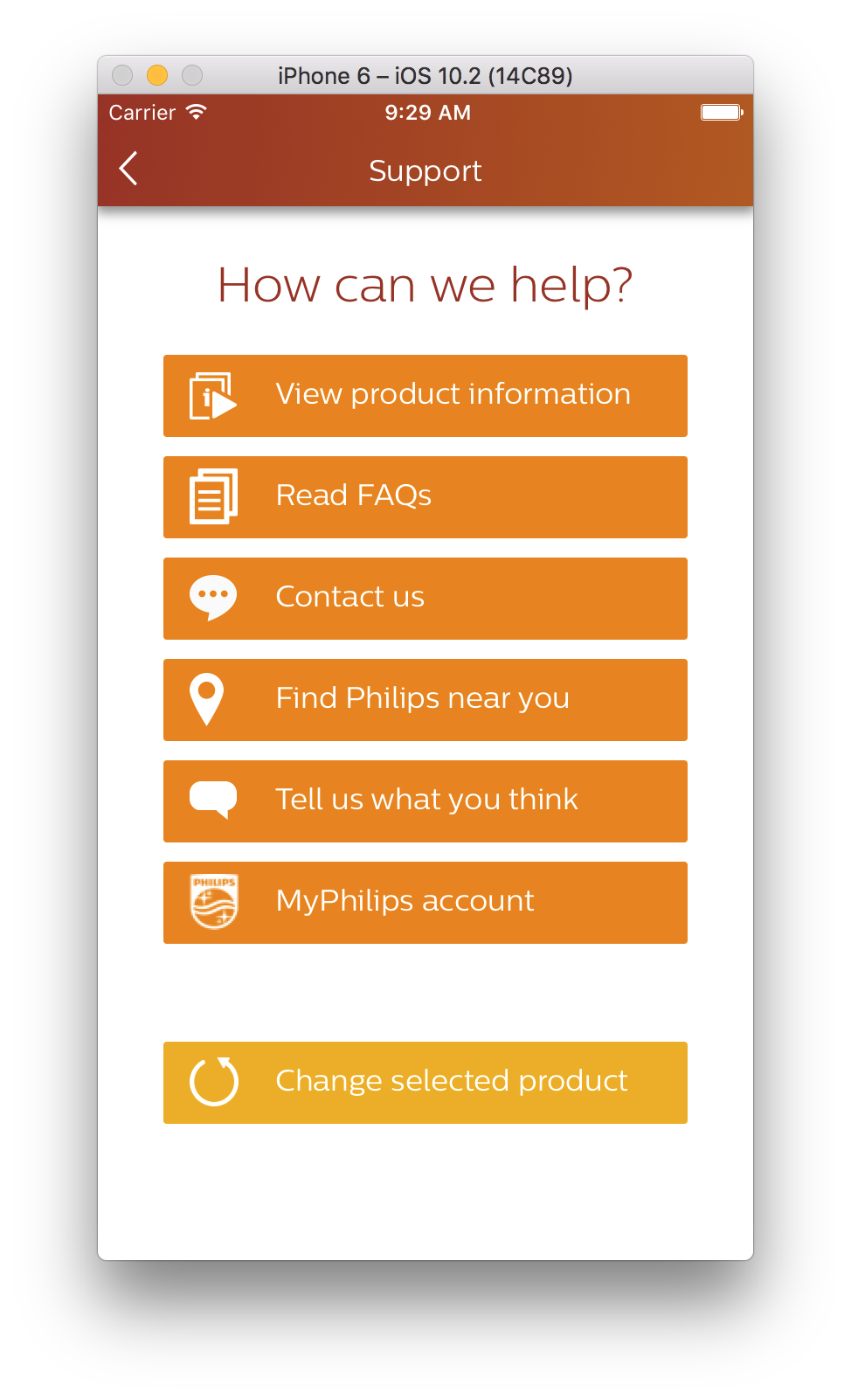
Invoked on click of each button on product info page. This helps in configuring features on product info page.

1. onSocialProviderItemClicked

Invoked on click of each social provider buttons in Contact us page. This helps in configuring list of social list providers. CC by default supports Facebook and Twitter.

1. App is required to implement this to take control on app side for the menu items which is added by verticals. For consumer care menu item, library is supposed to take action and app should not ideally override this. If app returns “true” then it means that app will take action on click of that button and library will not perform any operation in that case otherwise library will take action if app returns “false”.

Example: Air Fryer customization



# Action bar customization

Here is how we customize navigation/action bar in consumer care module,

1. We allow customization of both Hamburger menu icon and back arrow button in navigation bar.
2. Projects which uses navigation bar will go for above option.
3. Projects where in navigation bar is not used, will  have to do the following
4. Replace hamburger menu icon with back arrow icon.
5. Set navigation bar background color same as screen background color.
6. Sets a Boolean in config file which says hamburger menu title required as “False”, in this case navigation bar title will not be shown in consumer care.
7. Projects which uses spring board, replaces hamburger menu with spring board icon.

# Supporting apps with Over 65K Methods

This is special case if app which has more than 65K methods do follow below link for reference.

<https://developer.android.com/tools/building/multidex.html>

According to this do following changes in gradle and application class

In Gradle file:

android {  
    compileSdkVersion 25  
    buildToolsVersion "25.0.2"  
  
    defaultConfig {  
        ...  
        minSdkVersion 21  
        targetSdkVersion 25  
        ...  
  
        // Enabling multidex support.  
        multiDexEnabled true  
    }  
    ...  
}

In Application Class:

@Override  
public void onCreate() {  
   MultiDex.install(this);

  Super.onCreate();

}

# Supported Languages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| English | German | French (FR) | Simplified Chinese | Traditional Chinese | Portuguese (EU) |
| Russian | Arabic | Japanese | Dutch | Malay | Montenegrin |
| French-CA | Romanian | Spanish Mexican | Ukrainian | Portuguese (BR) | Serbian |
| Gaelic | Greek | Hebrew | Hindi | Finnish | Swedish |
| Italian | Polish | Spanish | Korean | Czech | English UK |
| Norwegian | Albanian | Bosnian | Bulgarian | Croatian | Estonian |
| Hungarian | Indonesian | Kazakh | Latvian | Lithuanian | Macedonian |
| Slovak | Slovenian | Spanish (AR) | Thai | Vietnamese | Danish |
| Farsi | Turkish | Hongkong-China |  |  |  |

# Proguard Rules

**-dontusemixedcaseclassnames  
-dontskipnonpubliclibraryclasses  
-dontskipnonpubliclibraryclassmembers  
-dontpreverify  
-keepattributes** \*Annotation\*  
**-keepattributes** Signature  
  
  
*# Gson specific classes***-keep** class sun.misc.Unsafe { **\*;** }  
**-keep** class com.google.gson.stream.\*\* { **\*;** }  
**-keep** class com.philips.cdp.prxclient.\*\* {**\*;**}  
**-keep** class com.philips.cdp.prxclient.prxdatamodels.\*\* { **\*;** }  
**-keep** class com.google.gson.examples.android.model.\*\* { **\*;** }  
  
  
  
  
*#Tagging***-keep** public class com.adobe.mobile.\*\* {**\*;**}  
**-keep** public class com.philips.cdp.tagging.\*\* {**\*;**}  
  
  
*#LocaleMatch***-keep** public class com.philips.cdp.localematch.\*\* {**\*;**}  
  
  
*#Network***-keep** class org.apache.http.\*\* { **\*;** }  
**-keep** class android.net.http.\*\* { **\*;** }  
  
*#UIKit***-keep** class com.shamanland.\*\* {**\*;**}  
**-keep** class uk.co.chrisjenx.\*\* {**\*;**}  
  
  
*#ConsumerCare***-keep** class com.philips.cdp.digitalcare.\*\* {**\*;**}  
  
**-keep** public class \* extends android.app.Activity  
**-keep** public class \* extends android.app.Application  
**-keep** public class \* extends android.content.BroadcastReceiver  
  
  
**-keep** public class \* extends android.app.Fragment {  
 **<init>(...);**}  
**-keep** public class \* extends android.support.v4.app.Fragment {  
 **<init>(...);**}  
  
 **-keepclassmembers** enum \* {  
 **public static \*\*[] values();  
 public static \*\* valueOf(java.lang.String);** }  
  
**-keep** public class \* extends android.view.View {  
 **public <init>(android.content.Context);  
 public <init>(android.content.Context, android.util.AttributeSet);  
 public <init>(android.content.Context, android.util.AttributeSet, int);  
 public void set\*(...);**}  
  
**-keepclasseswithmembers** class \* {  
 **public <init>(android.content.Context, android.util.AttributeSet);**}  
  
**-keepclasseswithmembers** class \* {  
 **public <init>(android.content.Context, android.util.AttributeSet, int);**}  
  
**-keepclassmembers** class \* extends android.content.Context {  
 **public void \*(android.view.View);  
 public void \*(android.view.MenuItem);**}  
  
**-keepclassmembers** class \* implements android.os.Parcelable {  
 **static android.os.Parcelable$Creator CREATOR;**}  
  
**-keepclassmembers** class \*\*.R$\* {  
 **public static <fields>;**}  
  
**-keepclassmembers** class \*{  
 **@android.webkit.JavascriptInterface <methods>;**}  
  
  
*#GooglePLayServices***-keep** class android.support.\*\* {**\*;**}  
**-keep** class android.view.\*\* {**\*;**}  
  
**-keep** interface android.support.v13.app.\*\* { **\*;** }  
**-keep** public class \* extends android.support.v13.\*\*  
**-keep** public class \* extends android.app.Fragment  
**-keep** class com.philips.cdp.uikit.customviews.\*\*  
  
*#-keep class com.google.android.gms.\*\* { \*; }***-dontwarn** com.google.android.gms.\*\*  
  
**-dontwarn** org.w3c.dom.bootstrap.DOMImplementationRegistry  
**-dontwarn** com.philips.cdp.productselection.R$id  
**-dontwarn** android.view.\*\*  
**-dontwarn** android.media.session  
**-dontwarn** android.app.\*\*  
  
**-dontwarn** com.philips.cdp.digitalcare.\*\*  
**-dontwarn** com.philips.cdp.productselection.\*\*  
**-dontwarn** android.support.\*\*  
**-dontwarn** com.adobe.mobile.\*\*  
**-dontwarn** org.apache.\*\*  
  
  
**-dontwarn** com.shamanland.\*\*  
**-dontwarn** uk.co.chrisjenx.\*\*

# Frequently asked Questions

# How about customising fonts and sizes?

We use Central Sans Font as per Philips design guidelines. Both fonts and sizes cannot be configured as of now.

* 1. **Can we add our own buttons inside consumer care?**

Yes. However we have restricted this to main menu screen, product menu and list of social service providers list.

* 1. **Does Consumer care support App Tagging?**

Yes. Please make sure you call enableTagging and provide apps ID. Please check sample app for more details. Unless App calls enableTagging () API tagging will not be enabled inside consumer care.

* 1. **Where do I find my product information?**

Each business has to provide complete ctn number including slash

Ex: HD9240/90

With this please check below link which helps in finding out product tree information like category, sub category etc. please add relevant ctn number

1. <http://www.philips.co.uk/prx/product/B2C/en_GB/CONSUMER/products/HD9240/90.summary>
2. <http://nlvu077.gdc1.ce.philips.com:9080/repobrowser/catalogBrowser.jsp?catalogid=catalog_CL_CONSUMER&catalogType=CONSUMER&country=CL&language=es>

Please replace locale to check whether product data is available in corresponding countries.

* 1. **Does Consumer care support multiple products?**

Yes it supports multiple products from 16.1 release which has version 4.0.0.

Currently app can send hardcoded list of products as input after which consumer care will take care of selection of product.

* 1. **How to fix below fatal exception?**

**com.android.build.api.transform.TransformException: com.android.builder.packaging.DuplicateFileException: Duplicate files copied in APK lib/mips/libblasV8.so**

**Or any duplication related exception, the duplication library can be added to packageOption.**

android {

           packagingOptions {

             exclude ‘**lib/armeabi-v7a/libblasV8.so**’

           }

         }

android {

packagingOptions {

exclude 'lib/arm64-v8a/librsjni.so'

pickFirst 'lib/arm64-v8a/libRSSupport.so'

}

}

# Notes

1. Registration is developed as separate library project. App is expected to invoke registration library on click of registration button inside consumer care support page.
2. Library should be initialized as per document or sample app otherwise library would throw runtime exception.
3. Developer option “don’t keep activities” should be disabled.
4. Please refer sample application for more details.
5. GooglePlayServices/GoogleMaps added in the ConsumerCare Component is 10.2.0, If the vertical apps needs the latest they can exclude in gradle level by adding there required GooglePlayServices/GoogleMaps version.
6. For more info on AppInfra properties find the below link :

<http://tfsemea1.ta.philips.com:8080/tfs/TPC_Region24/CDP2/TEAM%20Sabers/_git/ail-android-appinfra?path=%2FDocuments&version=GBdevelop&_a=contents>

1. China specific implementation is attached in the below document (CC\_China\_Support.docx)



1. For china livechat url we exposed the API. Integrating App should append a unique value (token) to the URL as query parameter. Livecom backend receives this token inorder to maintain the unique user chat history between the platforms and the devices. Eg. Token can be HTTP UUID. If this token not set, user will see the chats of others.If the token is matched in the database (in other words, if the user has already used the live chat inside the app), Livecom will identify this as an existing "contact" and make his history accessible to the agent. On the APP side, the user will be able to go back through his previous history