Oracle Responsys

Universal Linking Developer's Guide

Release 6.42.00

July 2019

Documentation for Oracle Responsys customers who are modifying the iOS version of their mobile apps to handle direct links from a Responsys campaign.

Oracle Responsys Universal Linking Developer’s Guide

Release 6.42.00, document revision 07

You can always find the latest version of this document here: https://community.oracle.com/docs/DOC-1006446

Contents

Overview

This document is for Responsys Customers who would like to link from a Responsys email campaign directly into their iOS mobile app (if installed), using iOS Universal Links.

Implementing this solution will enable you to achieve email-to-app deep linking in iOS 9+, while preserving click tracking and selective app deep linking (that is, you choose which links in the email will open the app).

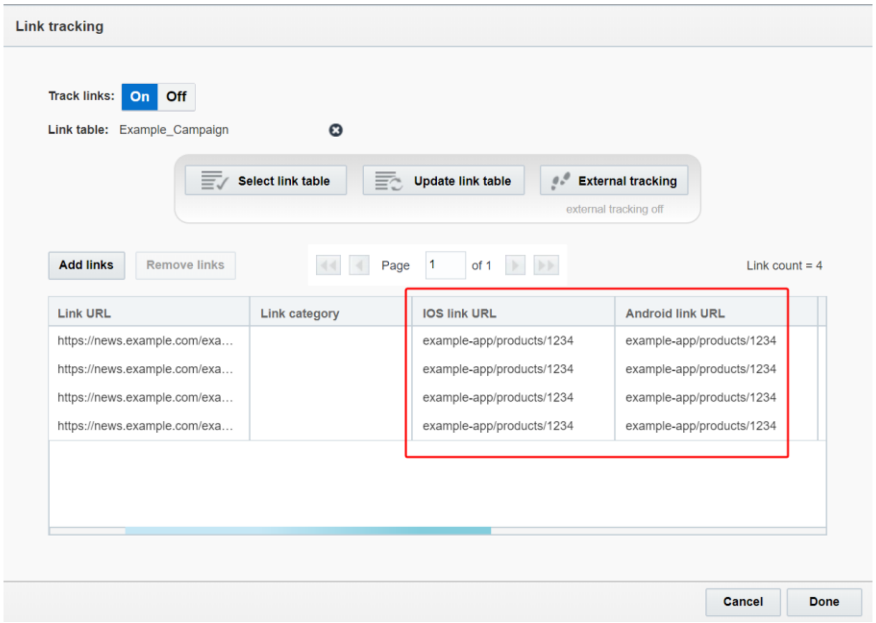
Solution Description

The solution works by declaring the Responsys link tracking URLs that are embedded in Responsys emails to be Universal Links, to be opened in your app.

To differentiate between links that should open in your app (if installed), and those that should not, Responsys will generate a slightly different link for both scenarios:

Example App Link: https://news.example.com/pub/**a**cc?_ri_=…
Example Non-App Link: http://news.example.com/pub/cc?_ri_=…

Email-to-app deep linking can be achieved on both Android and iOS devices by specifying **Android link URLs** and **iOS link URLs**. Marketers specify these URLs when creating link tables for email campaigns. The URL links must be in the format that the mobile app understands, for example, example-app://products/1234. (For more information about setting up link tables, refer to the Responsys online help.)



Link tracking

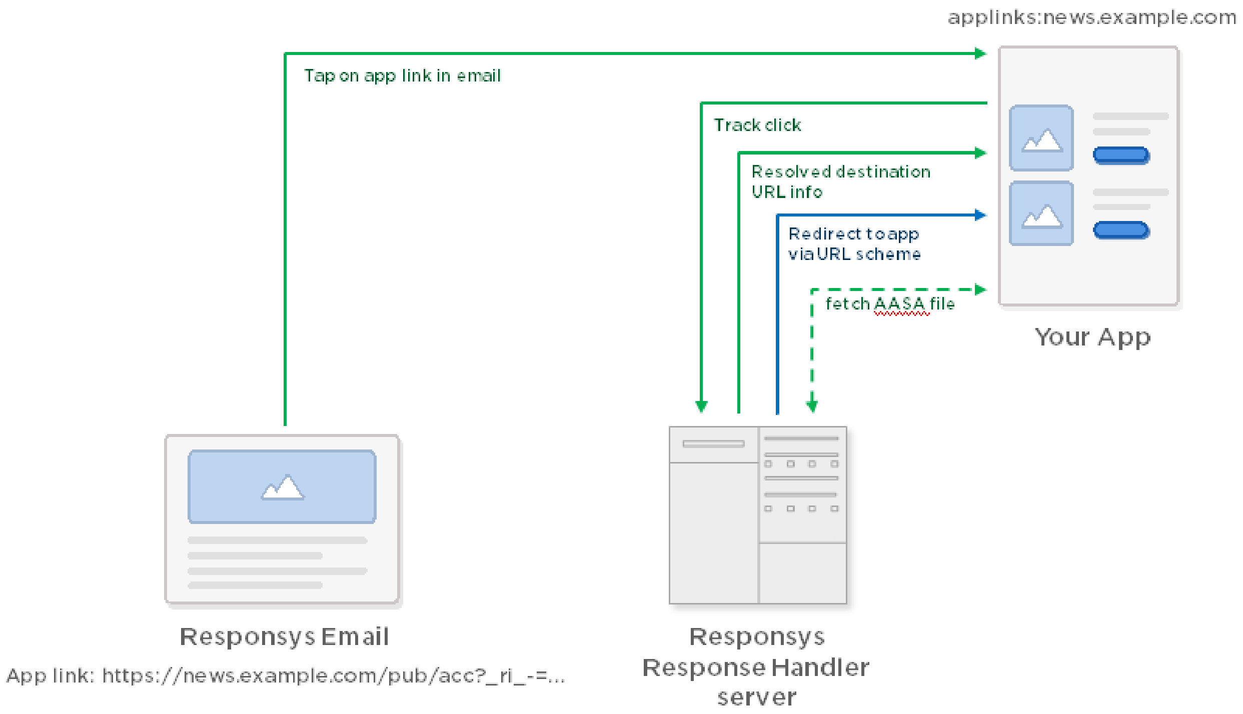
Track links: On Off

Link table: Example_Campaign

Link URL	Link category	IOS link URL	Android link URL
https://news.example.com/exa…		example-app/products/1234	example-app/products/1234
https://news.example.com/exa…		example-app/products/1234	example-app/products/1234
https://news.example.com/exa…		example-app/products/1234	example-app/products/1234
https://news.example.com/exa…		example-app/products/1234	example-app/products/1234

Note: A URL for both iOS and Android platforms should be supplied. If either the iOS link URL or Android link URL columns are blank, it is possible to receive a null response when resolving the link. Your app must be developed to handle this scenario. This is explained in more detail in the Troubleshooting section on page 14.

When a user on an iOS 9+ device with your app installed taps on an app link in a Responsys email, your app will be opened right away, without going through Mobile Safari.


applinks:news.example.com
Tap on app link in email
Track click
Resolved destination URL info
Redirect to app via URL scheme
fetch AASA file
Your App
Responsys Email
App link: https://news.example.com/pub/acc?_ri_-=...
Responsys Response Handler server

SDK Integrated flow

If your app has integrated the Responsys SDK 6.42.0 and above, the SDK will resolve the Universal Link automatically. Your app can then use that information to route the user to the appropriate screen in your app.

Standard App Flow

If the SDK has not been integrated, your app needs to then call the Responsys Response Handler server to:

1. Ensure the click is tracked
2. Resolve the Responsys Universal Link into information that your app can understand

Both of these are accomplished through an HTTPS call, which tracks the click and returns the resolved link information.

Specifically, your app will make an HTTPS GET request to the Responsys Universal Link URL, with an **`Accept`** header of **`application/json`**. For example:

Request

```
GET
https://news.example.com/pub/acc?_ri_=X0Gzc2X%3DYQLgnSQGjW1zaW3I39bSee
TLTzc17JkzgM0jGLwhp37NW3DPyepOVXtpKX%3DSWRB&_ei_=Eq2tf9zs59idfPO1Sc_9B
bmHzQ63AEdx_2H8ivyA7Qye6ocZmXBu9lX1DXT3ZU_be2kmoPhtBXC5ybqhCg.

Accept: application/json
User-Agent: ResponsysPubWebResolver (CPU iPhone OS like Mac OS X)
```

Response

```
HTTP/1.1 200 OK

Content-Type: application/json

{"webLinkUrl":"https://www.example.com/products/1234","mobileDeepLinkU
rl":"example-app://products/1234"}
```

Once your app has the resolved link information, it can then use that information to route the user to the appropriate screen in your app.

On pre-iOS 9 devices, the non-Universal Link app deep linking logic will continue to work as previously: the logic routes the user through Mobile Safari and then opens the app via the URL scheme that you specified in your Responsys link table.

Implementation

Overview

There are several pre-requisites that need to be put in place before you can use Responsys Universal Linking.

Those pre-requisites are:

1. Have a branded domain set up with Responsys for link tracking (e.g. `news.yourcompany.com`)

2. The branded domain needs to be HTTPS-enabled and the certificate needs to comply with Apple's App Transport Security[1] guidelines

3. Get an apple-app-site-association file in place on your branded domain

4. Declare in your app that it will handle Universal Links for the branded domain and the pub web path (`/pub/acc`)

5. Resolve the Universal Link by either calling the Responsys SDK's helper function or implementing resolver code in your iOS app

We'll cover each of these pre-requisites in more details in the following sections.

Set up a Branded Domain

To use Responsys Universal Links, your account must be set up with a branded domain.

Domain branding gives your account the ability to use a domain/sub-domain branded for your corporate entity vs. using a standard Oracle Responsys sub-domain. The domain is the portion after the @ symbol in from addresses and reply to addresses, and is the base location in a Response Handler URL.

The branded sub-domain is used to brand your From Address, Responsys-hosted Reply To Address, and the Response Handler URL. The Response Handler URL is used for redirection links, click-tracking URLs, HTML Open Tracking URLs, Conversion Tracking URLs, and the URL of forms hosted by Oracle Responsys.

Customer with a Branded Domain

You can check that your Responsys Link Tracking branded domain is set up properly by creating a test campaign, and verifying that the links in your test email are using your branded domain.

Customer without a Branded Domain

If your Responsys account has not been converted from a non-branded to branded domain yet, complete the steps in this section.

NOTES:

- The non-branded to branded conversion is allowed only once.

- You may convert to a branded domain once after your account is created.

- The domain/sub-domain you choose to delegate to Oracle Responsys must be one that no other entity is using, as we will manage all traffic for the domain/sub-domain.

[1] https://developer.apple.com/library/ios/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html#//apple_ref/doc/uid/TP40009251-SW33

Step 1: Delegate your sub-domains to the Responsys Name Servers

To use Domain Branding, your IT system administrator must delegate your sub-domains (e.g. `news.example.com`) to Oracle Responsys nameservers, `ns1.responsys.net` or `ns2.responsys.net`.

To delegate a sub-domain to the Responsys nameservers: Add the two (NS) records below to the master zone file. Do not create a separate zone file for the sub-domain you are delegating.

```
subdomain     IN NS     ns1.responsys.net.

subdomain     IN NS     ns2.responsys.net.
```

For example, if you are delegating a sub-domain called `news.example.com`, you would add two (2) NS records to the `example.com` zone file:

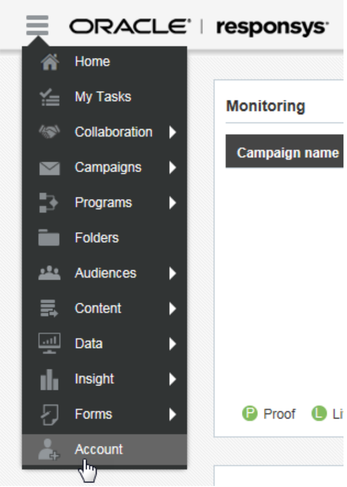
```
news      IN NS     ns1.responsys.net.

          IN NS     ns2.responsys.net.
```

Step 2: Convert to a Branded Domain in Responsys

To convert to a branded domain in Responsys:

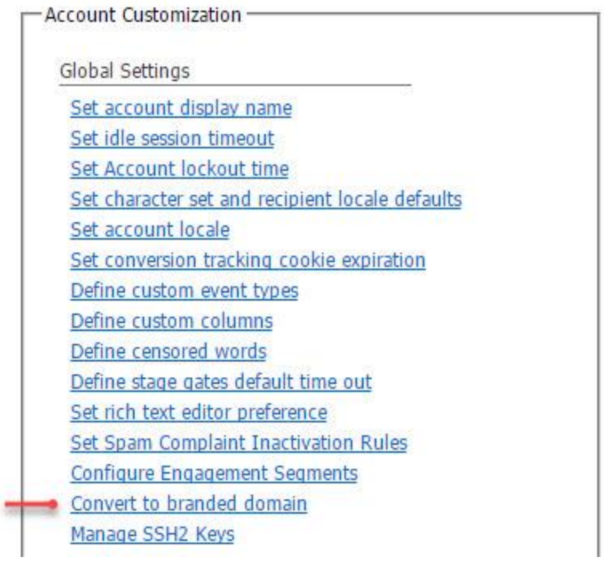
1. Log in to Responsys as a user with the **Account Admin** role.

2. From the Responsys home page, click the menu icon and then select **Account**.



3. In the **Account Customization** section's **Global Settings** list, locate and click the **Convert to branded domain** link. The system displays the **Convert to branded domain** dialog.

   NOTE: Not all customers will see this option. If you do not, please create a My Oracle Support (MOS) Service Request at https://support.oracle.com.


Account Customization
Global Settings
Set account display name
Set idle session timeout
Set Account lockout time
Set character set and recipient locale defaults
Set account locale
Set conversion tracking cookie expiration
Define custom event types
Define custom columns
Define censored words
Define stage gates default time out
Set rich text editor preference
Set Spam Complaint Inactivation Rules
Configure Engagement Segments
Convert to branded domain
Manage SSH2 Keys

4. Complete the following fields:

   Delegated branded domain: Enter the branded domain name, for example, `news.example.com`.

   From address: Enter the user name to use as the default "from" user. For example, if you enter `admin`, the From address will be `admin@news.example.com`.

   Notification email: Enter the email address that should receive the notification when the conversion is completed.

5. Click **Submit**.

After branded domain delegation is successfully completed, the account administrator and notification email address will receive a notification email. You can continue to the SSL enablement.

Enable SSL for the Branded Domain

To use Responsys Universal Links, your branded domain needs to be SSL-enabled, and the certificate used needs to comply with Apple's App Transport Security requirements[2].

SSL (Secure Socket Layer) protocol is a process where data passed between the user and server is encrypted/decrypted so that external third party cannot hijack the connection.

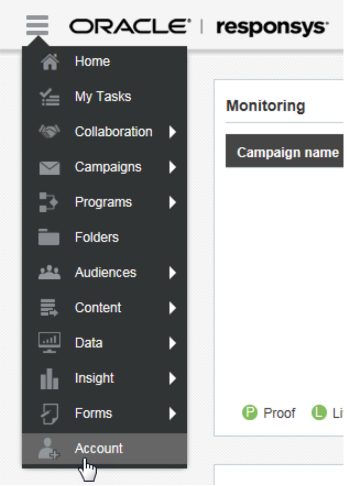
SSL behaves as a digital passport which verifies your and the end web server credentials using public and private keys. When both identities are verified, SSL grants a secured connection through HTTPS. This process is performed using SSL certificates.

Enabling SSL in Responsys allows for HTTPS support for all form, landing page, link tracking, and conversion tracking URLs generated by Oracle Responsys. SSL is possible because an SSL certificate is associated with the given domain and web server. In this case, the Response Handler URL requires SSL support so that app links are accessible over HTTPS.

Set up SSL for a Branded Domain

To set up SSL in Responsys for your branded domain(s):

1. Log in to Responsys as a user with the **Account Admin** role.
2. From the Responsys home page, click the menu icon and then select Account.

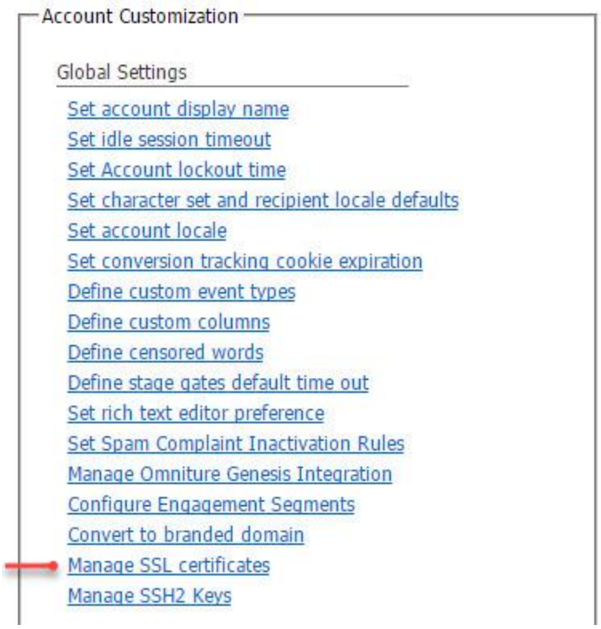


3. In the **Account Customization** section's **Global Settings** list, locate and click the **Manage SSL certificates** link. The system displays the **Manage SSL Certificates** dialog.

[2] https://developer.apple.com/library/ios/documentation/General/Reference/InfoPlistKeyReference/Articles/CocoaKeys.html#//apple_ref/doc/uid/TP40009251-SW33

NOTES:

- The **Manage SSL certificates** link will not be visible until the branded domain setup is completed.
- **Not all customers will see this option.** If you do not, please create a My Oracle Support (MOS) Service Request at https://support.oracle.com.

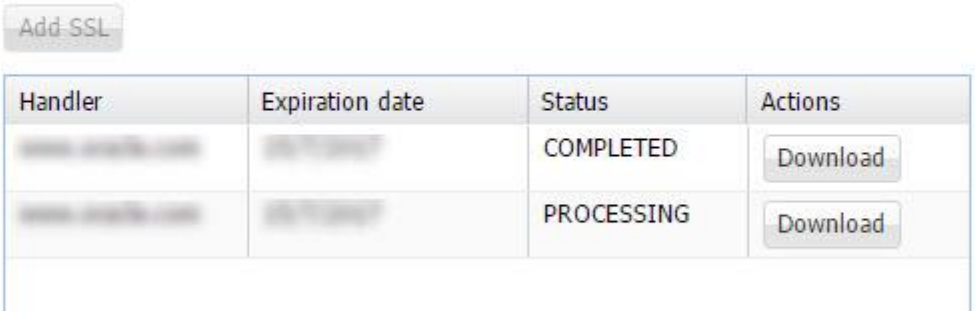


Account Customization

Global Settings

- Set account display name
- Set idle session timeout
- Set Account lockout time
- Set character set and recipient locale defaults
- Set account locale
- Set conversion tracking cookie expiration
- Define custom event types
- Define custom columns
- Define censored words
- Define stage gates default time out
- Set rich text editor preference
- Set Spam Complaint Inactivation Rules
- Manage Omniture Genesis Integration
- Configure Engagement Segments
- Convert to branded domain
- Manage SSL certificates
- Manage SSH2 Keys

4. Click **Add SSL**. A new row is inserted into the table.



Handler	Expiration date	Status	Actions
[redacted]	[redacted]	COMPLETED	Download
[redacted]	[redacted]	PROCESSING	Download

5. In the **Handler** column, select the response handler for which the SSL needs to be set up. Status column is set to “Processing”.

6. Below the table, enter the notification email address and click **Save**.

7. From the **Actions** column, select **Generate CSR**. The Generate CSR dialog is displayed.

8. From the **Generate CSR** dialog, complete the fields as follows:

   Generate CSR For: Domain(s) for which to generate the CSR. If you wish to use multiple domains instead of the response handler domain, select either the wild card or SAN option. Wild card domains are set up with leading *. Separate SAN domain names with a comma (,).

   Country code: ISO-2 country code of the country where your organization is legally registered.

   State or Province Name: Name of the state or province where your organization is located. Do not abbreviate the name.

   Locality or City Name: Name of the city where your organization is registered/located. Do not abbreviate the name.

   Organization Name: Legal name of your organization. Do not abbreviate the name. Include suffixes such as Inc, Corp, or LLC.

   Organization unit name: If applicable. Use the DBA name of your organization.

   SSL contact email address: Optional.

9. Click **Submit**.

   The system generates the CSR.

10. From the **Actions** column, choose **Download CSR**.

11. Using the CSR that you downloaded, purchase the SSL certificate from your preferred SSL certificate vendor.

12. After you have the SSL certificate, return to the **Manage SSL Certificates** page in Responsys.

13. From the **Actions** column, choose **Upload SSL** for the row you added in the previous steps.

14. Browse to the SSL files on your system, and click **Upload**. All of the uploaded files are displayed as a list. Select one as the main SSL certificate. The others are treated as intermediate CA certificates.

    After you add the SSL certificate and it is successfully installed, the status changes to “Active”, and the certificate expiration date is shown in the list.

Verify SSL Compliance with AATS

You can check that your branded domain is set up with SSL and complies with Apple’s App Transport Security by using **`nscurl`** on a Mac:

```
nscurl --ats-diagnostics https://news.example.com/pub/cc
```

Set up the AASA File

To associate your app with your branded domain, Oracle Responsys must put an **apple-app-site-association** (AASA) file in place on each of the branded domains that will be included in the links in the email campaign content. Contact your Customer Success Manager (CSM) or Oracle Responsys Support to complete this task.

When your app is installed on the user's device, iOS will try to fetch the AASA file on your branded domain in order to verify the association. iOS makes a request to:

```
https://news.example.com/apple-app-site-association
```

Here is an example apple-app-site-association file, set up for Responsys Universal Links:

```
{
  "applinks": {
    "apps": [],
    "details": [
      {
        "appID": "G3G853V3SJ.com.example.exampleapp",
        "paths": [
          "/pub/acc"
        ]
      }
    ]
  }
}
```

The AASA file for your app must reference your Apple team identifier and bundle ID, instead of the **`appID`** shown in the example above.

Note that the path in the file above specifies **`/pub/acc`**, so that only app links will be opened in your app, while others will always go to your website, even if the app is installed.

Declare Associated Domain in your iOS App

In this step, you'll associated your branded domain with your iOS app. Apple has detailed documentation on this available at:

https://developer.apple.com/library/ios/documentation/General/Conceptual/AppSearch/UniversalLinks.html#//apple_ref/doc/uid/TP40016308-CH12-SW1

In particular, you must include your branded domain in your app's **`com.apple.developer.associated-domains`** entitlement. For example:

```
applinks:news.yourcompany.com
```

To do this in Xcode, go to the **Capabilities** tab, open the **Associated Domains**, and then enter the above entry.

Resolve the Universal Link

If your app is integrated with the Responsys SDK, you must call the SDK's Helper function to resolve the Universal Link (see Call the SDK's Helper Function below for instructions). Otherwise, you must implement resolver code in your app to resolve the link.

Call the SDK's Helper function (SDK Integrated Flow)

This step is required when your app is integrated with the Responsys SDK.

<placeholder>

Implement Resolver Code in your iOS App (Standard App Flow)

The final setup step is to resolve the Responsys Universal Link into information that your app can understand. This step is only required if your app has not integrated the Responsys SDK.

In order to track clicks and to direct the user to the appropriate screen in your app, you'll make an HTTPS request from the following method in your app delegate:

```
- (BOOL)application:(UIApplication *)application
continueUserActivity:(NSUserActivity *)userActivity
restorationHandler:(void (^)(NSArray *))restorationHandler {
```

The details about the request that your app should make to resolve the link are explained above, and you can make this call in any way you see fit. Here is an example implementation:

```
- (BOOL)application:(UIApplication *)application
continueUserActivity:(NSUserActivity *)userActivity restorationHandler:(void
(^)(NSArray *))restorationHandler {

    // check for Responsys Response Handler Universal Link that needs to be
resolved
    if ([userActivity.activityType
isEqualToString:NSUserActivityTypeBrowsingWeb] &&
[userActivity.webpageURL.path isEqualToString:@"/pub/acc"]) {

        NSURL *url = userActivity.webpageURL;
        NSLog(@"Resolving Responsys Universal Link %@", url.absoluteString);

        // TODO: resolving the Universal Link will involve a network call
which may take a few seconds, time out, fail, etc, so you might want to show
a spinner
```

```
        // create a new ephemeral session configuration
        NSURLSessionConfiguration* sessionConfig = [NSURLSessionConfiguration
ephemeralSessionConfiguration];

        // set request timeout (in seconds)
        sessionConfig.timeoutIntervalForRequest = 10;

        //set user agent
        sessionConfig.HTTPAdditionalHeaders = @{@"User-Agent":
@"ResponsysPubWebResolver (CPU iPhone OS like Mac OS X)"};

        NSURLSession* session = [NSURLSession
sessionWithConfiguration:sessionConfig delegate:nil delegateQueue:nil];
        NSMutableURLRequest* request = [NSMutableURLRequest
requestWithURL:url];
        request.HTTPMethod = @"GET";

        // set accept header to get a JSON response
        [request addValue:@"application/json" forHTTPHeaderField:@"Accept"];

        // make request to Responsys to track click & resolve the Universal
Link
        NSURLSessionDataTask* task = [session dataTaskWithRequest:request
completionHandler:^(NSData *data, NSURLResponse *response, NSError *error) {
            if (error == nil) {

                if ((long)((NSHTTPURLResponse*)response).statusCode >= 200 &&
(long)((NSHTTPURLResponse*)response).statusCode <= 299) {

                    // parse resolved JSON
                    NSError *parseError = nil;
                    NSDictionary *resolvedResponsysUniversalLinkInfo =
[NSJSONSerialization JSONObjectWithData:data options:0 error:&parseError];
                    if (!resolvedResponsysUniversalLinkInfo) {
                        NSLog(@"Couldn't parse resolved Responsys Universal
Link info: %@; data = %@", parseError, [[NSString alloc] initWithData:data
encoding:NSUTF8StringEncoding]);
                    } else {
                        NSLog(@"Resolved Responsys Universal Link Info: %@",
resolvedResponsysUniversalLinkInfo);

                        // TODO: add your code that would use the resolved
Responsys Universal Link info
                    }
                } else {
                    // resolve request returned error
                    NSLog(@"Request to resolve Responsys Universal Link
returned error: %ld; data = %@",
(long)((NSHTTPURLResponse*)response).statusCode, [[NSString alloc]
initWithData:data encoding:NSUTF8StringEncoding]);
                }
            } else {
                // resolve request failure
                NSLog(@"Request to resolve Responsys Universal Link failed:
%@", [error localizedDescription]);
            }
        }];
        [task resume];
        [session finishTasksAndInvalidate];
    }

    return YES;
}
```

Troubleshooting

Handling null values (mobileDeepLinkUrl is null)

When your app is resolving the link information, it is possible to receive a `null` value for `mobileDeepLinkUrl`.

```
{
  "webLinkUrl": "https://www.example.com/products/1234",
  "mobileDeepLinkUrl": null
}
```

This can happen if you have implemented the Android App Links feature and have not provided a value for Android link URL, but have provided a value for an iOS link URL within a Link Table in your Email Campaign.

Apps must be developed to handle these `null` values to ensure an optimal user experience.