Mobile client quirks

The Profound UI mobile client app makes it easy for users to access their Profound UI applications from a mobile device, and makes mobile device features such as the camera available to user applications. The underlying technology used in the mobile client is Apache Cordova, an open source development framework that presents Web content on a native application via a WebView. Due to the complexities involved with Cordova, its many plugins, and the iOS and Android operating systems, sometimes issues crop up outside of Profound UI's control that need to be worked around.

Opening hyperlinks (iOS only)

Starting with Mobile Client version 2.8.4, hyperlinks are handled differently than before, due to a change in the Cordova InAppBrowser plugin. In the past, the plugin redefined *window.open* to *cordova.InAppBrowser.open*, so that links would pass through the plugin. This no longer occurs, which means that calls to *window.open* truly go to *window.open*, and window targets of *_blank* or *_system* have no effect. The result is that hyperlinks coded as *window.open(url, ...)* will now open inside the mobile client window itself, with no way to return to the previous screen. To resolve this, use *cordova.InAppBrowser.open* instead of *window.open. cordova.InAppBrowser.open(url, '_blank')* will open the URL in a window inside the mobile client, with a *Done* button to close the window and return to your screen. *cordova.InAppBrowser.open(url, '_system')* will leave your app and open the URL inside of the system browser (e.g., Safari). See https ://github.com/apache/cordova-plugin-inappbrowser for more information.

Opening hyperlinks in the system browser (iOS only)

Note: As of Mobile Client version 2.8.4, this work around is no longer required and no longer functions correctly.

With version 2.7.3 of the iOS mobile client, an internal configuration change needed to be made that had the side effect of causing hyperlinks to open inside the mobile client WebView instead of in the system web browser (Safari). Depending on the circumstances, this may be undesirable. To work around this, some JavaScript code can be used to listen for hyperlink "clicks" and, when desired, open them in the system browser. Below is some sample code that can be customized and either placed in the `profoundui/userdata/extension/mobile` directory (to affect every mobile session) or attached to individual rich displays (via the external javascript property or the display's onload event).

```
window.addEventListener('click', function (e) {
  if (e && e.target && e.target.nodeName == 'A') {
    var click = e.target;
    if (click.href && click.target == '_blank' || click.target == '_system') {
      e.preventDefault();
      window.open(click.href, '_system');
    }
  }
}, false);
//# sourceURL=clickPatch.js
```

Line	Description
1	Add a "click" event listener to intercept every click made on the screen.
2	Filter out clicks to everything except anchor tags (hyperlinks).
3	Define the `click` variable for convenience and readability.
4	Only continue if the hyperlink has a URL associated with it and the target is something that should open in a new window/tab. Note that this is sample code and you will likely want to tailor this part to fit your needs. In this case, it was determined that other targets should open inside the mobile client's WebView. Note also that the target of `_system` is used by the in-app-browser Cordova plugin to denote a URL that should open in the system browser.
5	Prevent the click event from continuing, as it is being handled here. Without this statement the URL will be opened in both the system browser and the WebView.
6	Open the desired URL in the system web browser – the "in-app-browser" plugin included with the Profound UI mobile client overrides `window.open()` and causes URLs with a target of `_system` to open in the system web browser.
10	The sourceURL directive makes it easier to debug embedded code by making it appear as if the code was loaded from a file. In this example, this event code will appear in the developer tools as if it was contained in file `clickPatch.js`. This more easily allows breakpoints to be added and code to be inspected.

Voice dictation (iOS only)

Note: As of Mobile Client version 2.8.4, voice dictation functionality has been restored and the work around listed in this section is no longer needed.

Starting with iOS 13, voice dictation in Cordova applications (like the Profound UI mobile client) has broken. When the built-in microphone icon that is part of the iOS keyboard is tapped, voice dictation only functions for 1-2 seconds before terminating. Multiple bug reports have been filed with Apple, but until Apple decides to fix the issue, dictation will remain broken. To help its customers who require voice dictation, Profound Logic is providing sample code for a drop-in work around that will provide voice dictation functionality for Profound UI applications accessed via the mobile client.

To use it, place the sample code file in the following IFS directory or any sub-directory below it (if at Profound UI version 6.3, Fix Pack 3 or earlier then each rich display will need to explicitly load this file as an external JavaScript file):

```
/www/{instance_name}/htdocs/profoundui/userdata/extension/mobile
```

Mobile applications running on iOS will then display a floating microphone icon that, when tapped, will start voice dictation on the currently-focused input field. Tapping on the icon again will end voice dictation. In addition, the icon may be dragged around the screen to a more convenient location, if desired.

Sample code: voiceDictation.js