

# Core Text Objective-C Wrapper

Adrian Kosmaczewski

2010-07-08

One of the most promising and mysterious new frameworks introduced in iOS 3.2 is Core Text. Apple defines Core Text as a “text drawing engine”, which allows Mac (and now iPad) apps to render rich text on any graphics context. Strings drawn with Core Text feature lots of custom settings such as detailed font information, columns, variable line and paragraph heights and several different attributes, which designers and font aficionados surely understand much better than I do. Many new apps have been using this framework since the release of the iPad, particularly newspapers and ebook reader applications, rendering gorgeous text in custom fonts, many of them not available natively in iOS. This framework is also used in lifestyle and corporate applications, too, where using a custom font is sometimes required to match the specifications of brands and trademarks.



It is very important to understand that Core Text is really just a text drawing engine to be used on top of Quartz (Core Graphics), to render rich text on any graphics context. Core Text cannot be used to create a rich text editor, for example, so don't expect to extend UITextView with it. But you can use it to draw any kind of rich text on screen, which can make you avoid using

UIWebView instances for that.

One of the most interesting capabilities of Core Text is being able to render text in several columns. However, Core Text is a C-based framework, and I think that understanding and using the concepts and structures required to render text in columns can be particularly tricky. To make the my life and that of my fellow developers easier, I've created a small set of Objective-C classes that encapsulate the creation of framesetters, attributed strings and other constructions, and takes care of the creation of several columns, as well as the division of the text in several pages if required.

The API interface is very simple (in purpose) and I'm pretty sure you'll be able to integrate it very easily in your own projects, particularly if you look at the sample project where the classes are used. I've also added a category for UIFont, that returns the associated CTFontRef pointer, in a similar fashion to UIImage, which is able to return a pointer to the underlying CGImageRef pointer. It also allows to create a CTFontRef from any font embedded in the application bundle. I am puzzled that the framework designers haven't included this by default in UIKit.

A future extension I'd like to add would be a couple of categories to parse simple RTF or HTML strings (to start with, probably just with bold and italic text) and create the appropriate attributed string from it; there's a couple of AppKit extensions to NSAttributedString that do exactly that, but for the moment they are only available in the Mac version of Cocoa, and I haven't found anything similar for iOS yet (feel free to leave a comment below if you know about some implementation for generating NSAttributedStrings from HTML or RTF text!)

The code, as usual, is available through Github<sup>1</sup> (BSD license), so feel free to use it and even contribute bug fixes or enhancements if you want. The font embedded in the project is Polsku Regula, available through the Open Font Library.org<sup>2</sup>. The project requires Xcode 3.2.3 and the iOS SDK 4.0, and it creates a simple application that works on the iPad (iOS 3.2). As usual, use it at your own risk. Enjoy!

---

<sup>1</sup><http://github.com/akosma/CoreTextWrapper>

<sup>2</sup><http://openfontlibrary.org/files/ospublish/140>