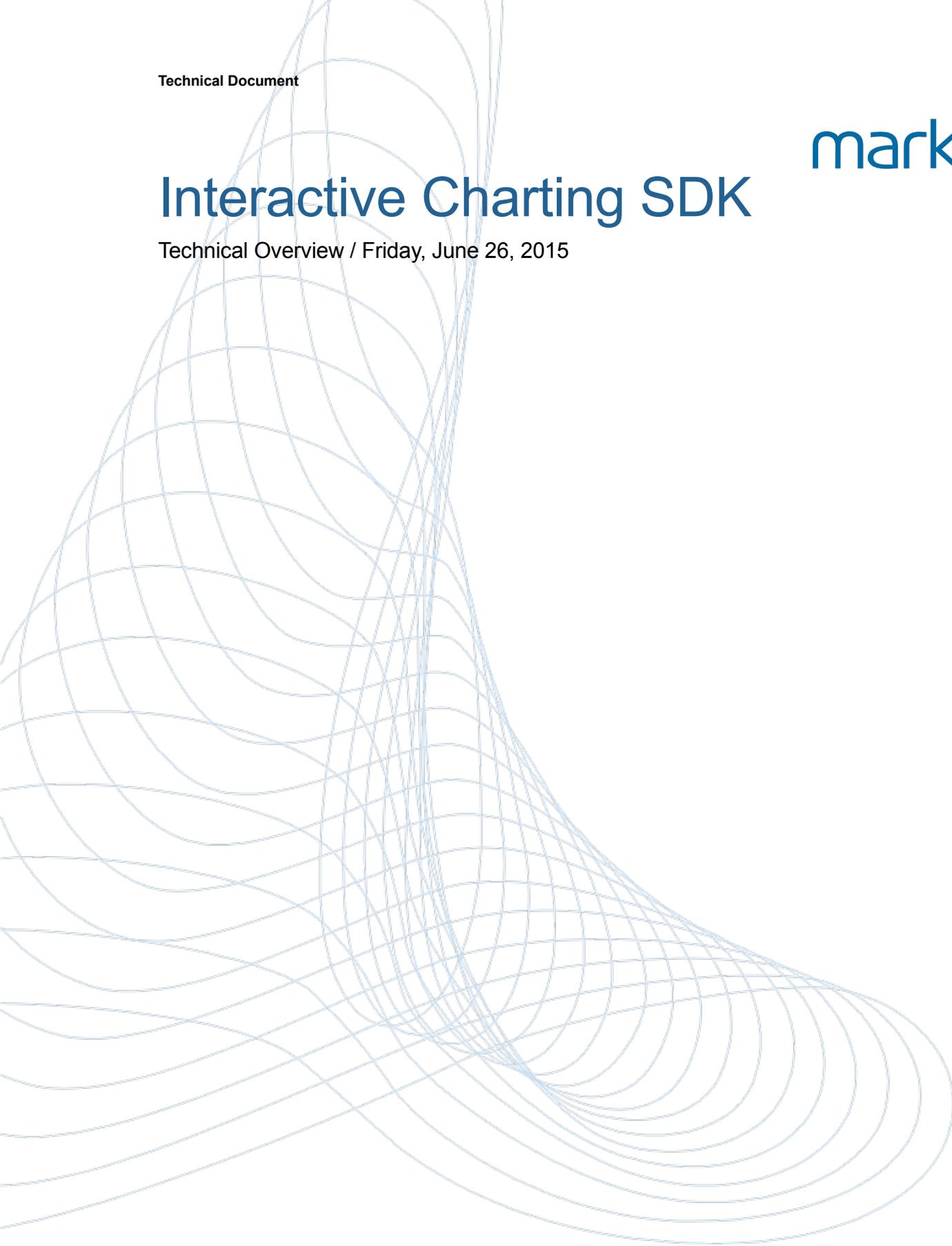


Technical Document

markit®

Interactive Charting SDK

Technical Overview / Friday, June 26, 2015



Overview	3
Features	3
Easy to Use	4
Easy to Customize	4
Data Networking	5
Supported Indicators	5
Requirements	6
Frequently Asked Questions.....	6

Overview

In general terms an SDK is a reusable software platform used to develop applications, products and solutions. SDKs contain examples, documentation, code libraries and application programming interfaces (APIs) that bring together all the different components that enable development of a project or solution. Interactive Charts is Markit On Demand's iOS SDK that allows developers to easily add and customize interactive pinch & zoom financial charting into a new or existing iOS application.

Features

- Easily add an interactive chart of nearly any size to your existing iPad & iPhone application
- Support up to 5 chart styles: Candlestick, OHLC, Bar Graph, Line or Mountain
- Pinch, Zoom and Pan anywhere within a chart
- Support for all of the major indicators including Bollinger Bands®.
- Comparison Support
- Support for Customized Announcements
- Crosshair support; find out the value of any point in any chart
- Returns Calculator to show change over a selected period of time by holding down with two fingers.
- Allow users to draw custom trendline annotations on their chart.
- Automatically scale the Y-Axis to the data displayed at the given time.
- Heavily customizable; create your own theme for your charts to match the rest of your application
- Replace built in indicator descriptions and button names
- Fully localized; if your language isn't already there, it is easy to add new ones
- UI Flexibility; use your own menus, buttons and watch lists
- XML configurable properties that can be used cross platform (iOS & Android)
- Dynamically load data as chart is panned
- X-axis intelligently built using market exchange data
- Customizable data requests improve network performance



Easy to Use

In just a few lines of code you can create an interactive chart in your iPad or iPhone application:

```
_chartView = [[LChartView alloc]  
initWithPosition:LChartPosMake(0, 0, 1024) upperChartHeight:500  
lowerChartHeight:200 gutterHeight:50];
```

```
[self.view addSubview:chartView];
```

This chart is fully navigable. By pinching and spreading two fingers, you can zoom out and zoom in, allowing you to see additional detail.

You can use a gesture to pan the chart left and right or up and down.

Pressing and holding a tap will show cross hairs and reveal the point value underneath.



Easy to Customize

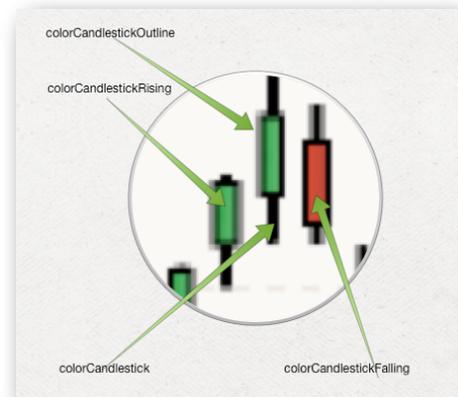
There are many ways to customize a chart including color change, fonts and line sizes to create your own theme and your own style to match your application needs.

A developer can start with either dark or light themes and change the color properties to meet their own style. Button imagery can be easily replaced as well.

The SDK contains its own definitions for indicators that can be easily replaced without getting an update from MOD. Common properties can be configured and shared between iOS and Android platforms using a unified XML file.

Here is just a partial list of customization properties from the LChartConfiguration class header.

```
// Chart(s) upper & lower  
@property (nonatomic, strong) UIColor *colorUpperChartBackground;  
@property (nonatomic, strong) UIColor *colorLowerChartBackground;  
@property (nonatomic, strong) UIColor *colorNetworkActivityBackground;  
  
// Chart miscellaneous items  
@property (nonatomic, strong) UIColor *colorCrosshair;  
  
// Popover Background  
@property (nonatomic, strong) UIColor *colorIndicatorPopOverBackground;  
  
// Axis - Upper  
@property (nonatomic, strong) UIColor *colorXAxisUpperBackground;  
@property (nonatomic, strong) UIColor *colorYAxisUpperBackground;  
@property (nonatomic, strong) UIColor *colorXAxisUpperLine;  
@property (nonatomic, strong) UIColor *colorYAxisUpperLine;
```



```
@property (nonatomic, strong) UIColor *colorUpperMinorTickStyleLabel;  
@property (nonatomic, strong) UIColor *colorUpperMinorTickStyleLine;  
@property (nonatomic, strong) UIColor *colorUpperMajorTickStyleLabel;  
@property (nonatomic, strong) UIColor *colorUpperMajorTickStyleLine;  
@property (nonatomic, strong) UIColor *colorUpperMajorGridStyleLine;  
@property (nonatomic, strong) UIColor *colorUpperTickLabel;  
@property (nonatomic, strong) UIFont *fontUpperXAxisLabel;  
@property (nonatomic, strong) UIFont *fontUpperYAxisLabel;  
@property (nonatomic, strong) NSNumber *widthXAxisUpperLine;  
@property (nonatomic, strong) NSNumber *widthYAxisUpperLine;
```

Data Networking

To avoid conflicts with your own application networking, and to support those clients who wish to log all network traffic, the Interactive Charts SDK delegates network calls to your application. MarkitCharts uses the following delegate method to request data:

```
-(void) chartView:(LChartView *)chartView performNetworkOperationForURL:(NSURLRequest *)request  
onCompletion:(void (^)(NSData *, NSError *))onCompletion
```

The request will include the URL which contains all necessary information to communicate with the Markit Chart Server. Developers can submit this URL to their own network code and once a return is received simply pass the response back.

Supported Indicators

The SDK currently has the following indicators fully implemented with more coming soon.

Upper Indicators

- *Bollinger Bands®*
- *Dividends, Earnings, Splits, Custom Announcements*
- *Exponential Moving Averages (EMA)*
- *High / Low*
- *Latest Price*
- *Linear Regression*
- *Moving Average Envelope (MAE)*
- *Parabolic Stop & Reverse (PSAR)*
- *Price*
- *Price Channel*
- *Simple Moving Average (SMA)*
- *Time Series Forecast (TSF)*
- *Weighted Moving Average (WMA)*

Lower Indicators

- *Accumulation Distribution*
- *Average True Range*
- *Chaikin's Volatility*
- *Directional Moving Index (ADX / ADX-R)*
- *Dividend Yield*
- *Fast / Slow Stochastics*
- *Mass Index*
- *Momentum*
- *Money Flow*
- *Money Flow Index*
- *Moving Average Convergence Divergence (MACD)*
- *On Balance Volume*
- *PE Range*
- *PE Ratio*
- *Price Rate of Change (PROC)*
- *Relative Strength Index*
- *Revenues*
- *Rolling Dividend*
- *Rolling EPS*
- *Ultimate Oscillator*
- *Up/Down*
- *Volume*
- *Volume Rate Of Change*
- *Williams Percent R*

Requirements

- Software: iOS 7.x or later

- Hardware: iPad 2 or later, iPhone 4 or later

Frequently Asked Questions

Q: Where is the separation between what the SDK framework provides and what a client’s development team must provide?

A: The SDK will provide the complete workflow for everything that is inside of the chart. The buttons around the chart that change the scale or allow the entry of a symbol is treated separately from the SDK and thus can be customized by the client for their own needs.

Q: Does that mean that a client developer cannot change the underlying button images that is within the chart?

A: No. The LChartConfiguration class provides a UIImage property that can be set.

Q: Is there any third party technology that MarkitCharts depends on?

A: Yes. The SDK currently leverages both open source and proprietary technologies.

Q: Is the source code available for the Interactive Charting framework?

A: Markit will provide the framework headers, library, sample library and documentation that can be used to add interactive charting to a client application. At this time, source code for the actual framework is not provided.

Q: Who handles the support should an issue arise while using the Interactive Charting framework or any of its underlying technologies?

A: Markit will provide support. It is not necessary to work directly with any third party vendor.