IDC COMSTOCK MARKET DATA FEED - JAVA API

The PlusFeed Java API allows users to query the snapshot database and subscribe to streaming tick data, Intraday bar data, historical price data and data conflation to support mitigated streaming output from PlusFeed market data servers.

Key Advantages:

- No need to learn CTF wire protocol which cuts short the development time.
- No need to learn CTF commands since most of the commands are implemented as function calls.
- No need to remember CTF token numbers while specifying the request parameters. Instead use human readable CTF token names.
- The API is designed using multiple threads to minimize input buffering and hence the loss of messages.
- Supports multiple Operating Systems because it is written using Java language.
- Light-weight foot-print - 292 kilo-bytes - in jar file size.

Key Features:

- Supports Level 1 and Level 2 (depth) snapshot queries.
- Supports streaming real-time or delayed Level 1 and Level 2 quotes and trades.
- Supports options and futures chain requests.
- Supports to query tick data, time & Sales (Quotes, Trades, NBBO)
- Supports Historical Data (Daily Bars), Intraday Data (minute bars)
- Supports various asset classes including Equities, Futures, Options on Equities and Futures, Forex, and Fixed Income.
- Supports news headline and story queries and streaming updates.
- Supports both synchronous and asynchrous data requests.
- Notifies when disconnected from the PlusFeed servers.

Key Application scenarios:

- Integration the PlusFeed into third party applications or tools such as Order Management Systems, Analytical Engines
- Data feed integration into Trading tools such as Algorithmic trading, Program trading
- Web Services: The API can be extended to provide Quotes & Research services over the web. These web services can be utilized to build web applications that provide financial market news on the Internet.
- Quote Montages: The API can be used to build Desktop Applications that can be used by Traders and Brokers to query real-time stock price quotations, build time series graphs, apply analytics, review historical time and sales data and more.
- Analytical Engines: The API can be used to build engines that process real-time data to calculate value added fields on the fly e.g VWAP (Volume Weighted Average Price), to triger automated trades based on pre-configured algorithms, to build audit tools that can be used by RegNMS and much more.

Platform Support:

- Developed in JDK 1.6.0_06, the API supports all the platforms that are supported by Java SE 6 which include the popular Microsoft Windows, Solaris (SPARC & x86), and Linux Operating Systems.

The API consists of four packages - ctf, ctf.net, ctf.data, and ctf.parser.

- ***ctf package handles PlusFeed CTF commands***
- ***ctf.net package handles PlusFeed server TCP/IP socket connection.***
- ***ctf.data packages manages CTF Data Dictionary.***

- ***ctf.parser package contains Parser to parse CTF wire protocol.***

Initializing the Data Dictionary: This is the first and a mandatory step that needs to take place before working with any other functionality of the api. The init() call loads the CTF token definitions and the Enumeration definitions into the api system. The definitions are read from two files which are part of the "ctf.data" pacakge - tokens.dat and enum.dat. Working with a fully loaded data dictionary allows users to refer to tokens by names instead of numbers and enumeration descriptions instead of enumerated values. Users are encouraged to update the two files periodically to take advantage of the new data introduced in the PlusFeed.

Logging into the PlusFeed server: After initializing the Data Dictionary, the immediate next step would be to attempt to connect to the PlusFeed server. This is accomplished using the Client class in the ctf package. If the host name is reachable and the port number is accepting connections, a successful connection is established with the PlusFeed server. Otherwise, exceptions are thrown which will indicate the nature of failure. The login() method will attempt to login to the PlusFeed server. The Status class will tell whether the attempt is successful or not.

Requesting Snapshot information: The ctf.Client class contains implementations for all the CTF commands used to request data from the PlusFeed servers. To request current Level 1 snapshot information for an instrument on an exchange, the getQuote() method is used and sends it to the PlusFeed server on the current TCP/IP socket connection. It waits for a response which is matched by the presence of the same QUERY.TAG value of "1". If the exchange id and the ticker symbol are valid, a Message object is returned which contains data fields like current price, net change, etc.

The Field class contains the token and value pair. The value for token "CURRENT.PRICE" can be obtained by calling the getValue() method. Other methods are available in the Field class to fetch the value by appropriate type - by Integer, Double, Boolean and DateTime.

One can find if message is a refresh message or not. If there is an error in the request - either because an invalid exchange id has been specified or the user is not entitled for the requested exchange id - the status portion of the Message class will give the details.

Requesting for Subscriptions: The Client class allows users to subscribe to streaming updates from PlusFeed servers using the subscribe() method. The ClientListener interface allows users to register callback functions with the Client class to be notified when the updates arrive from the PlusFeed server.

Handling Connection Failures: The Client class provides addListener() method to register to events related to connection state changes. removeListener() method will unregister. The Client class notifies ClientListeners whenever the state of the connection to the PlusFeed server changes using the onStatus() method. The ctf.net.Connection class defines the available event ids for connection state changes.

DISCLAIMER: **DATA PROVIDED IS FOR DEMO PURPOSES ONLY. IT IS NOT INTENDED FOR TRADING OR DECISION MAKING PURPOSES. ALL DATA IS PAST OR DUMMY DATA, AND NOT REAL TIME. BY ACKNOWLEDGING THESE TERMS, THE VIEWER UNDERSTANDS AND AGREES THAT THE INFORMATION IN THIS PAGE, OR DERIVED FROM IT IS NOT FOR USE AS REAL DATA.**

Contact: SRIDHAR CHELIKANI, SAVEN TECHNOLOGIES, WWW.SAVEN.IN
EMAIL: SRIDHAR.CHELIKANI@SAVEN.IN, PHONE: 847-380-3069, FAX: 847-693-4123