

Historical Product Overview

Robust historical market data and analytics enabling our clients to gain insights and make data-driven decisions

FEATURES

- Datasets support all US-listed Equities, Futures, Equity Options, and Options on Futures
- End-of-day files, intraday snapshots as well as tick data available with corresponding reference data
- Options-specific data includes implied volatility, Greeks, surfaces, risk slides, and short-term trade performance
- Enhanced analytics used by wide range of practitioners providing insights on market trends
- Underlying prices, size, volume, prints, open interest and reference data

COMPETITIVE ADVANTAGE

Clients can focus on developing strategies by leveraging our analytics. SpiderRock is a respected brand in calculating implied volatility, greeks, risk metrics, and fitting volatility surfaces.

- Fast incubation of new trading strategies
- Model market volatility and relative market movement
- Assess risk and margin requirements
- Evaluate trade cost analysis (TCA)
- Run portfolio evaluation and manage end-of-day marks

QUALITY

Our historical data is derived from the live data and analytics which powers the SpiderRock trading system and ensures a high level of accuracy and consistency.

- Represents market activity at point-in-time
- Cleaned and well documented
- Evaluated for validity and accuracy to minimize errors
- Statistical analysis on our data to verify completeness

USE CASES

CLIENT TYPES

Trading
Community

Risk &
Compliance
Administration

Regulators
& Research
Community

APPLICATIONS

Create, back-test, incubate, and optimize trading strategies
Trade cost analysis (TCA)

Portfolio management risk identification
Model market volatility and relative market movement
Replay market patterns

Compliance reporting

Identify market trends
Academic research purposes



SpiderRock Data Liberator API For Historical Data

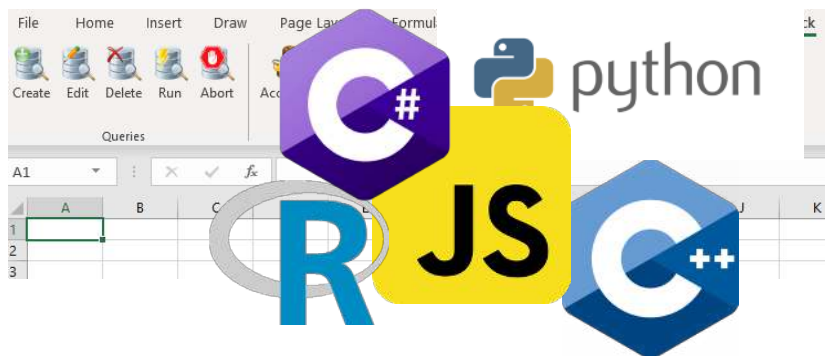
The SpiderRock Data Liberator Service provides easy access to historical datasets using a Restful API

Unlock your productivity and increase your efficiency when using historical data. Data delivery is on demand – how and when you prefer. The Liberator API allows you to select the data you need by date ranges and times, by single or groups of ticker symbols.

Using the API reduces the need to download and store large data sets during your research process and allows you to focus your efforts on performing the analysis and not managing the data. Through our Liberator API you can access SpiderRock Options, Stock and Futures data within your own applications using our cloud, compute power and pre-optimized query access for a simple monthly fee (*).

API FEATURES

- Single data access API for historical time series data; point-in-time and time series
- Datasets support all US-listed equities, options and indexes
- Historical data queries from our cloud storage
- No need to download and maintain large options market historical datasets
- Allows users to easily include data directly into algorithms
- Multiple language support / Native-language APIs and Spreadsheet plug-in



SpiderRock Liberator API comes with starter codes and notebooks for different languages that enable users to get started quickly. On-line help and security index tables allow fast look up of information.

The API is also integrated with Microsoft Excel™ for those with work processes involving downloading specific data and analyzing this data in spread sheet applications.

(*). Various data access plans; tiered based on number of queries or monthly data egress up to unlimited access. Contact SpiderRock Data Sales at gwtsales@spiderrock.net to start you free trial today.

Data Liberator functionality is provided by CloudQuant (www.cloudquant.com).

PRODUCT OFFERING

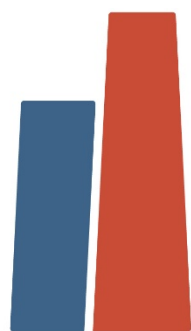
DATA TABLES	FREQ	HISTORY	PRICE	VOLUME	SIZE	GREEKS	IMPLIED VOL	VOL SURFACE
STOCK								
Stock Close Marks	EOD	Jan-10	X	X	X			
Stock Minute Bars	1 Min	Jan-10	X	X				
Stock Print Set	Trades	Jan-16	X	X	X			
Stock Imbalances Data	Every Tick	Feb-19	X	X				
EQUITY OPTIONS								
Options Close Marks	EOD	Jan-10	X	X	X	X	X	X
Options Price History ID	30 min	Jan-15	X	X	X	X	X	X
Options Price History HID	5 min	Jan-20	X	X	X	X	X	X
Options Minute Bars ATM	1 min	Jan-20	X				X	
Options Print Set	Trades	Jan-14	X	X	X	X	X	X
US FUTURES INDEXES, ETF, INDEX OPTIONS								
Futures Close Marks	EOD	Jan-19	X	X				
Futures Minute Bars	1 Min	Jan-16	X	X				
Futures Print Set	Trades	Jan-16	X	X	X			
Options (F) Close Marks	EOD	Jan-10	X	X		X	X	X
Options (F) Price History ID	30 min	Jan-15	X	X	X	X	X	X
Options (F) Price History HID	5 min	Jan-20	X	X	X	X	X	X
Options (F) Minute Bars ATM	1 min	Jan-20	X				X	
Options (F) Print Set	Trades	Jan-14	X	X	X	X	X	X
Option Pair Set ID ES								
OPTIONS VOLATILITY SURFACES								
Surface Curves EOD	EOD	Jan-10	X			X	X	X
Fixed Grid Surface EOD	EOD	Jan-10					X	X
Fixed Term Surface ATM EOD	EOD	Jan-10					X	X
Surface Curves ID	10 Min	Jan-19	X			X	X	X
Fixed Grid Surface ID	10 Min	Feb-19					X	X
Fixed Term Surfaces ATM ID	10 Min	Jan-18					X	X
VOL2G EQUITY OPTIONS VOLATILITY BUNDLE								
Stock Close Marks	EOD	Jan-10	X	X				
Options Close Marks	EOD	Jan-10	X	X	X	X	X	X
Options Fixed Grid Surfaces	EOD	Jan-10					X	X
Options Fixed Term Surfaces ATM	EOD	Jan-10					X	X
Volatility History Table by Ticker	EOD	Jan-10	X	X		X	X	X
Equity Reference Tables	EOD	Jan-10	X	X				
EQUITY REFERENCE TABLES								
Ticker Definition Map (Security ID)	Daily PIT	Jan-10						
Security Price Table (Adjusted)	Daily PIT	Jan-10						
Global Rates	Daily PIT	Jan-10						
Trading Dates	Daily PIT	Jan-10						

Samples available by request. Contact gwtsales@spiderrock.net



SpiderRock
300 S Riverside Plaza, Suite 2350
Chicago, IL 60606

Phone: 312-256-9600
gwtsales@spiderrock.net
www.spiderrock.net



SPIDERROCK

GATEWAY

SpiderRock Historical Data Archive

Hist Reference Futures

Table of Contents

Table of Contents.....	2
Overview.....	3
Available Data	3
Accessing the Files via the SFTP	5
Accessing the Files via Amazon Web Services S3.....	7
Detailed Data Explanation	11
Futures SR Closing Marks	11
Futures Minute Bars	13

Overview

Introductory Comments

This document outlines the historical data that is available from SpiderRock, what the various data tables contain, and how they can be delivered. In most cases, the data is extremely large and can take a large amount of disk space and considerable time to download.

We do offer a number of different options, including both historical and periodic updates, to deliver this data to you in the most efficient manner depending on your own individual requirements.

Available Data

Note: A version of all files also exists for major INDICES only, which include:

Post 2021-01-01, Futures data includes only CME products.

Futures

Table Name	From (Start Date)	Summary Description
Futures SR Closing Marks	1/1/2019	Futures SR Closing Mark records are created immediately after the market close, when exchanges publish official marks, and again during top of day rotation. These records contain closing quotes and prices as well as markup details for all outright futures.
Futures Minute Bars	6/1/2015	Futures Minute Bar records are created once per minute for all future markets (outright and spreads).
Futures Print Set	6/1/2015	Futures Print Set records are created for all future prints (outright and spreads) and updated at 1 minute and 10 minutes to monitor trade performance.

Support Files

Table Name	From (Start Date)	Summary Description
Ticker Defintion Hist Map	1/1/2010	Ticker Definition records exist for all SpiderRock tickers including equity tickers (stocks and ETFs) as well as index tickers and synthetic tickers for future chains and option multihedge baskets.
Ticker History	1/1/2010	Product Definition records exist for all futures and options on futures.

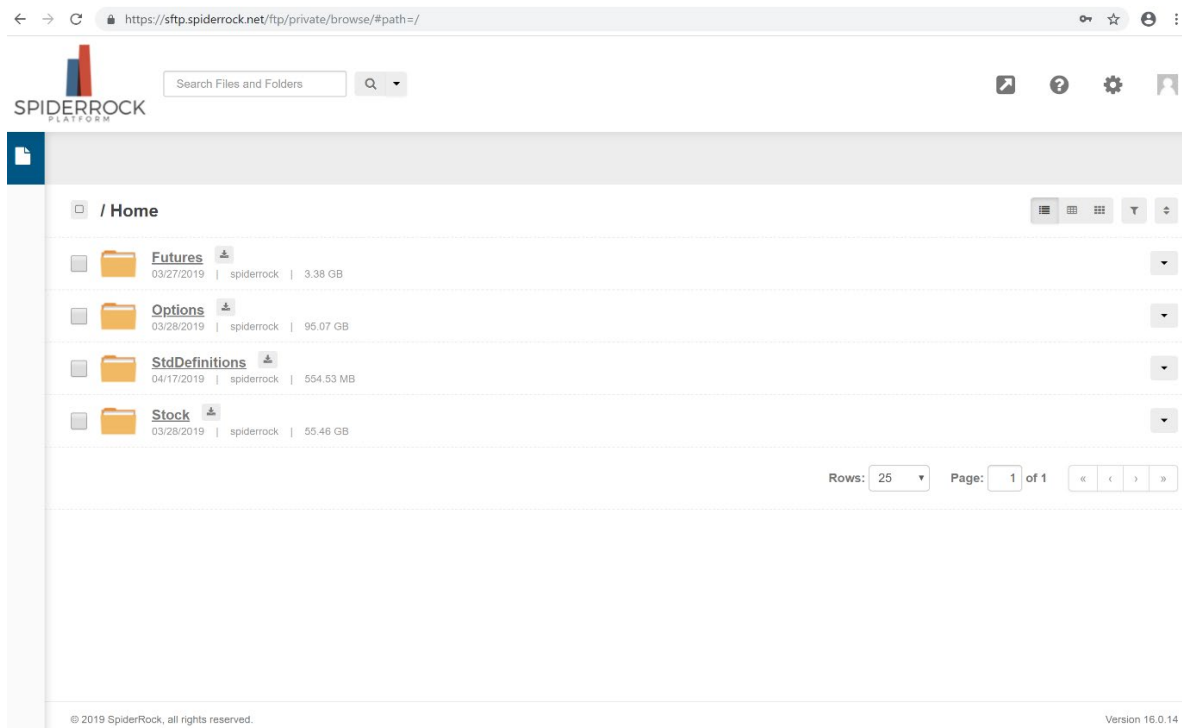
All of these files are available on a daily basis in the following format:

Table Name	Extension	Comments
Plain Text	.txt	This is a plain, uncompressed ASCII Text file. These are very large (e.g. stock minute bar for one day is around 124 megabytes). Therefore, it is not recommended that you download these but rather one of the files in a ZIP format
ZIP File	.zip	This is the plain text file compressed as a standard .ZIP file with medium compression. This should be compatible with most forms of compressed files. This is compressed to about 12% - 15% of the original size so the example given above compresses to approximately 20 Megabytes

Accessing the Files via the SFTP

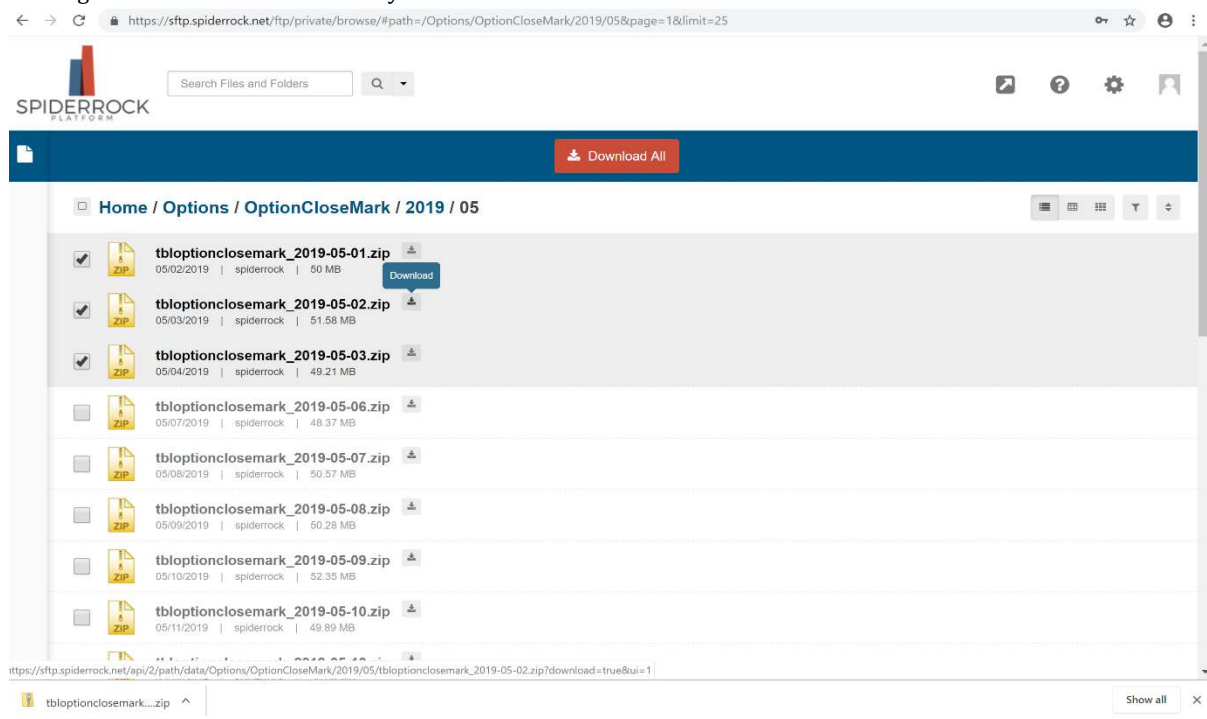
The data files can be downloaded from SpiderRock's SFTP server in compressed ZIP files. These ZIP files can then be uncompressed and transferred into a database or EXCEL workbook. To access and download the data needed, do the following:

1. On a standard browser, type "[SFTP.Spiderrock.net](https://sftp.spiderrock.net)" into the address bar
2. This goes to SpiderRock's SFTP site, where a login will be required. Type in the given SpiderRock username and corresponding password to gain access to the files.
3. The following home page should appear with folders of different assets and data:



4. Select the file you wish to open by clicking on the title (i.e. "Futures"). This should then display the different data sets available for the asset. Again, click on the title to choose which data file you wish to open based on the datatables offered.
5. Once you have opened your desired asset type and datatable, then select which date of data you wish to view. After selecting the year and month, the compressed file for each day is available.
6. The data files can be downloaded from SpiderRock's SFTP server in compressed ZIP files. These ZIP files can then be uncompressed and transferred into a database or EXCEL workbook. To access and download the data needed, do the following:

- To download, there are a few options. You can either select multiple files to download by clicking the titles of each file or the square box next to each, then selecting the red "Download All" button at the top. If you wish to download an individual file, you can either select only the file wanted and click the red "Download" button, or you can click the download icon to the right of the title, which will begin the download automatically:

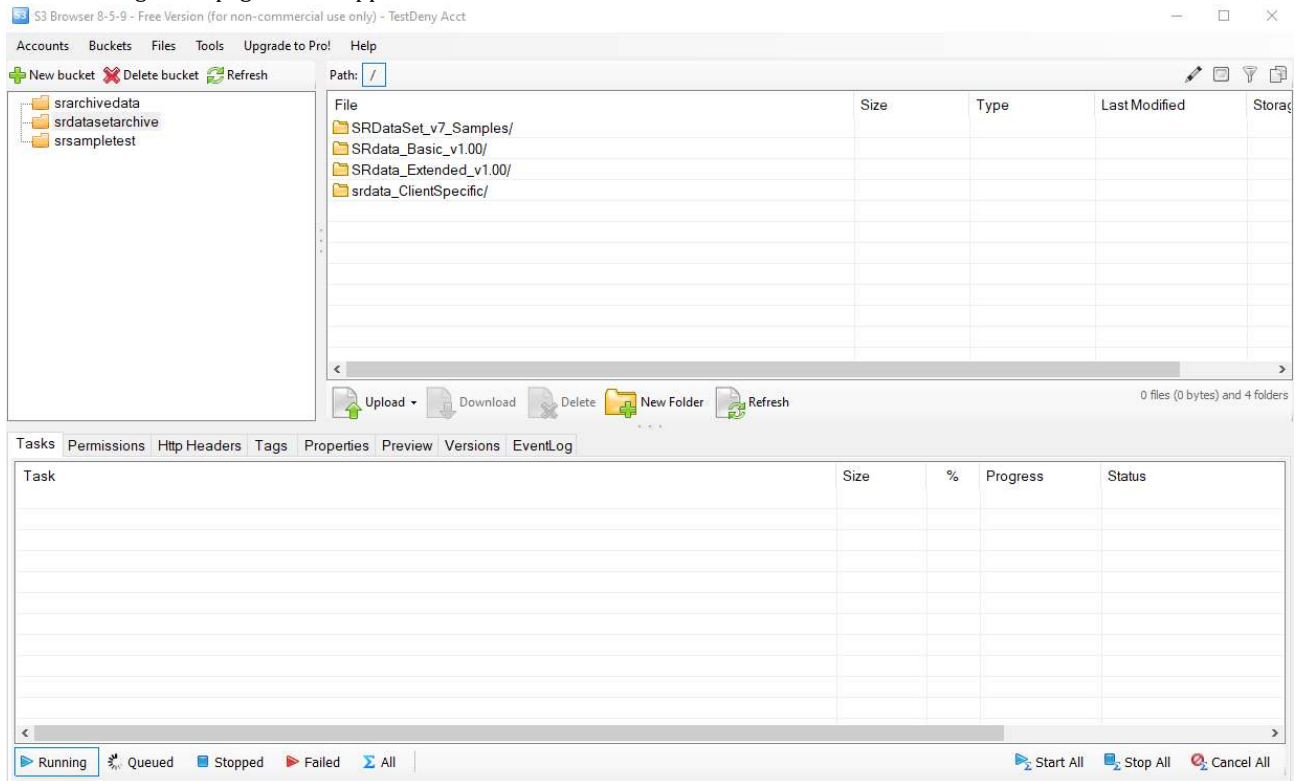


- Name the ZIP file, and subsequently the data will begin to download. Once finished downloading, move the file to where you wish to store the data.
- Double-click to open the ZIP file to see the data you downloaded. Ensure that the data is uncompressed before beginning to transfer it to another database or format.

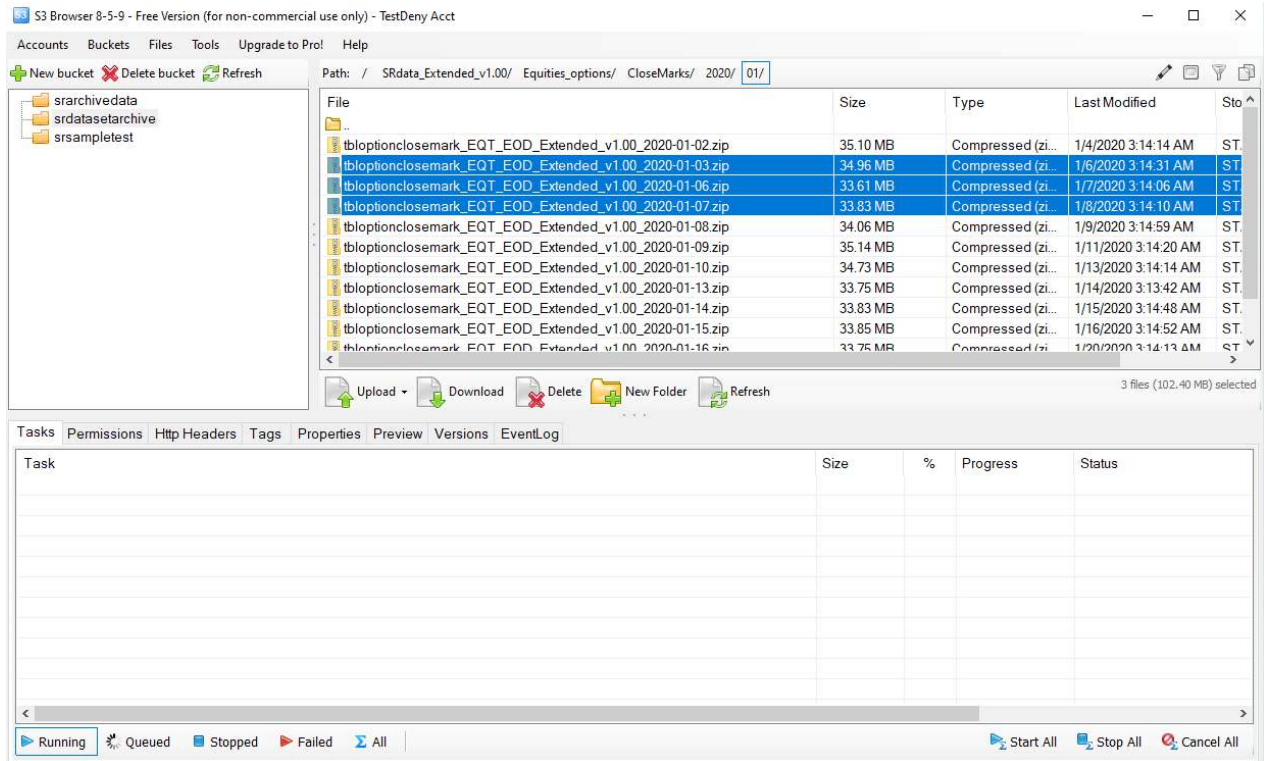
Accessing the Files via Amazon Web Services S3

The data files can be downloaded from SpiderRock's AWS S3 server in compressed ZIP files. These ZIP files can then be uncompressed and transferred into a database or EXCEL workbook. To access and download the data needed, do the following:

1. On a standard browser, type "https://s3browser.com/download.aspx" into the address bar and download the browser
2. Once downloaded, a login will be required, this consists of an ACCESS KEY and SECRET ACCESS KEY. Type in the given SpiderRock Access key and Secret Access key to gain access to the files.
3. The following home page should appear with folders of different assets and data:



4. You will be permissioned for the bucket **srdatasetarchivehist**, which includes the latest version of our data.
5. Select the file you wish to open by clicking on the title (i.e. "SRdata_hist"). This should then display the different data sets available for the dataset. Again, click on the title to choose which data file you wish to open based on the datatables offered.
6. Once you have opened your desired asset type and datatable, then select which date of data you wish to view. After selecting the year and month, the compressed file for each day is available.
7. The data files can be downloaded from SpiderRock's S3 server in compressed ZIP files. These ZIP files can then be uncompressed and transferred into a database or EXCEL workbook. To access and download the data needed, do the following:
8. To download, there are a few options. You can either select multiple files to download by clicking the titles of each file, then selecting the "Download" button at the bottom of the folder list. If you wish to download an individual file, you can either select only the file wanted and click the "Download" button. You are able to, if permissioned for the entire underlying data, download higher in the path (ie. At the monthly, yearly, or database file level).

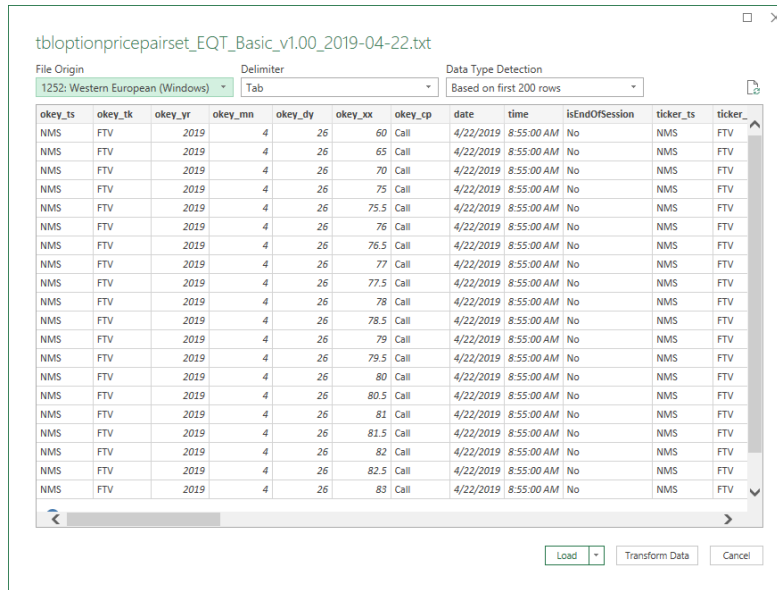


9. Choose the output folder where you want the data downloaded on your instance.
10. Name the ZIP file, and subsequently the data will begin to download. Once finished downloading, move the file to where you wish to store the data.
11. Double-click to open the ZIP file to see the data you downloaded. Ensure that the data is uncompressed before beginning to transfer it to another database or format.

Field Delimiters

These files have all been created as data exported directly from our underlying MySQL database tables. These are exported in ASCII formatted files with a TAB (\t or HEX 09) as the field delimiter. Therefore, these uncompressed files (subject to size limitations of your installation of EXCEL) can be opened by EXCEL as follows:

1. Start EXCEL and open a new workbook
2. Select "Data" on the ribbon, and then select "From Text/CSV"
3. Make sure that the filter is set to "All Files"
4. Navigate to where you downloaded and "unzipped" the file (**note you MUST open the uncompressed** version of the file, so if you downloaded the compressed version you must first uncompress it into an uncompressed format using either WINZIP or a similar tool).
5. Select the text file you want to open and press import. You will then be presented with a window which is similar to the following:



6. On this screen make sure you select the Delimiter as “Tab” and start import at Row 1 (as this data **DOES HAVE column headers – so make sure you check the option ‘Use First Row as Headers’ in the ‘Transfrom Data’ window**) and then select “Load”. The rows will then start loading, which may take some time.

If desired, select “Transform Data” and on the next window you can apply any specific formatting required.

Please note that EXCEL does have limitations on the number of rows and columns. Depending on your installation, it may not be possible for EXCEL to load the entire file. For example, you will not be able to load all the data in excel for Live Surface Grids, Option Price Pair sets, Stock Minute Bars, and all Print sets. EXCEL will notify you if it cannot fit all of the data into a workbook.

File Naming conventions and Updates

All data is kept in separate files (in each of the above named formats) for every trading day from the starting date shown. For example, the stock Minute bar for the trading day of March 31st 2019 can be found at:

Sub Folder	File Name	Size	File Type
/StockMinuteBars	/tblstockminutebarhist_2019-03-31.txt	125,117KB	Uncompressed text version * see note below – re text files
/StockMinuteBars	/tblstockminutebarhist_2019-03-31.zip	20,649KB	Compressed ZIP file

All Data is updated in batch processes that typically run overnight. This means that another “days” activity is added to these tables once a day and at the completion of a month and the data is not changed again.

The exceptions to this rule are

- tblTradingDate which rarely changes and the ZIP file remains static.

By 7:00am on each trading day, the previous trading day’s data should be completely loaded and ready for downloading.

Plain Text Files

Please note at the time of publishing this document the plain text files are not available (simply due to disk space considerations) and only the ZIP files are available which are an exact representation of the text files but compressed using standard ZIP formatting with the “optimal” flag set.

File headers

All of these text files contain a header (each column is delimited by a TAB – i.e. exactly the same as the data). Therefore, the very first record for each file contains the data name for each column. For example, if you were to load the file into EXCEL, row 1 would contain a header line.

Loading these files in a database

Loading files into a database is often the most practical way to import this data and most relational databases contain some type of functionality that will allow you to import external file directly into a database table. For example, MySQL has a standard import function and the command to import the stock minute bar would be:

```
load data local infile 'c:/ExternalData/tblstockminutebarhist_2019-06-29.txt' into table
tblstockminutebar_2019_06
fields terminated by '\t'
lines terminated by '\r\n'
ignore 1 lines
(date,time,skey_tk,skey_ts,skey_at,pOpen,pHigh,pLow,pLast,pVwap,pVlm,pCount,qOpen,qHigh,qLow,qLast,qCount,bid
,ask,qTwap,bsz,asz,width);
```

Notes:

1. This assumes you have unzipped the file to a folder on your C: drive called ExternalData. The convention for MySQL is to use the forward slash (/) as the directory separator as opposed to the backslash.
2. This assumes you have created a table in the target schema called tblStockMinuteBar_2019_06 and its columns are identical to that of the input file.
3. Ignore 1 lines is the instruction to ignore the 1st line of the data (i.e. the header line).
4. This example has explicitly laid out the columns to be imported – this specifies the “target” columns and NOT the source columns, but the order of these columns shown in this statement MUST be the order of the data in the source text file.

Concatenating Files

Should you wish to combine multiple input files into a single output text file you can achieve this with your operating systems concatenate functionality.

For example, in Windows the procedure would be:

1. Download the multiple files into a single folder (e.g. **c:\Downloaded**) and giving them different names
2. Create an output folder (e.g. **c:\Output**)
3. Once that is completed then issue the following command line command:
Type c:\Downloaded*.txt >> c:\Output\MergedData.txt

This will then create a single merged file of the multiple input files. Please note the following:

1. Make sure that the output folder exists and is EMPTY
2. Make sure you only merge together “like” data – in other words do not merge together different types of files
3. All of these files **have headers** so you should try and exclude the very 1st record of each data file if you want data only.
4. Over a period of time these files change (i.e. columns get added and/or removed). Therefore, if combining files over a period of time, you must be aware of this fact and ensure you are merging like data.

Detailed Data Explanation

Overview

This section explains in detail the columns that are available in each table. In the following tables, we show the columns that are included in these data fields and are shown in “ordinal order” i.e. the order in which the data appears in the file from left to right as indicated by the “Order” column in the following tables.

These files are all plain ASCII text files so the data types we have provided are the same as the source data type and the shown data would be compatible with that data type.

(*) indicates data included in the basic package.

Futures Data Files - Overview

Our datasets provide individual prices, prints, volume and other price related data for all exchange traded products listed on the CME, CBOT, NYMEX, and COMEX prior to 2021. After 2021, the datasets include data solely from CME. Futures Data Files include Closing Marks, Minute Bars, and Print Set.

Futures SR Closing Marks

Futures SR Closing Mark records are created immediately after the market close, when exchanges publish official marks, and again during top of day rotation. These records contain closing quotes and prices as well as markup details for all outright futures.

Order	Field	Data Type	Comment
1	fkey_at	enum('None','EQT','IDX','BND','CUR','COM','FUT','SYN','WAR','FLX','MUT','SPD','MM','MF','COIN','TOKEN')	
2	fkey_ts	enum('None','SR','NMS','CME','ICE','CFE','CBOT','TD','NYMEX','COMEX','RUT','CBOE','ISE','ARCA','NYSE','OTC','GDAX','BSTAMP','KRAKEN','TST','USR1','USR2','USR3','NSDQ','MFQS','PHLX','MIAX','TSE')	
3	fkey_ts	enum('None','SR','NMS','CME','ICE','CFE','CBOT','TD','NYMEX','COMEX','RUT','CBOE','ISE','ARCA','NYSE','OTC','GDAX','BSTAMP','KRAKEN','TST','USR1','USR2','USR3','NSDQ','MFQS','PHLX')	
4	fkey_tk	varchar(12)	
5	fkey_yr	smallint(5) unsigned	
6	fkey_mn	tinyint(3) unsigned	
7	fkey_dy	tinyint(3) unsigned	
8	tradingDate	enum	
9	tradingSession	enum('None','RegularMkt','PreMkt','PostMkt','PostMktETF','NextDay')	Trading Session: ('None','RegularMkt','PreMkt','PostMkt','PostMktETF','NextDay')
10	srCloseTime	timespan	SR Closing mark Time
11	clsMarkState	enum('None','LastPrt','SRClose','ExchClose','Final')	Close mark state: None; LastPrt; SRClose; ExchClose; Final
12	opnPrc	double	Opening price
13	minPrc	double	Low price
14	maxPrc	double	High Price
15	openInterest	int(11)	Open interest
16	prtCount	int(11)	print count
17	prtVolume	int(11)	print volume

18	realized Cnt	int(11)	number of minute bar segments used in realizedVar calc
19	realized Var	float	realizedVar = SUM[LOG(qteTwap(T) / qteTwap(T+1)) ^ 2] / realizedCnt @ 1 minute intervals during market hours
20	avgMktSize	float	MEAN[0.5 * (bidSize + askSize)] @ 1 minute intervals during market hours
21	avgMktWidth	float	MEAN[askPrice - bidPrice] @ 1 minute intervals during market hours
22	bidPrc	double	bid price (close - 1min)
23	askPrc	double	ask price (close - 1min)
24	srClsPrc	double	sr close mark (close - 1min)
25	closePrc	double	official exchange closing mark (last print;then official close)
26	priorDate	datetime	
27	priorSRClsPrc	float	values archive in the previous trading period
28	priorClosePrc	float	Prior Close Price
29	timestamp	datetime	YYYY-MM-DD HH:mm:ss

Data Updates/Fixes/Problems

Futures Minute Bars

Futures Minute Bar records are created once per minute for all future markets including outright and spreads. Data included in this set includes high/low values, spread data, print information, and bid/ask data.

Order	Field	Data Type	Comment
1	fkey_at	enum('None','EQT','IDX','BND','CUR','COM','FUT','SYN','WAR','FLX','MUT','SPD','MM','MF','COIN','TOKEN')	
2	fkey_ts	enum('None','SR','NMS','CME','ICE','CFE','CBOT','TD','NYMEX','COMEX','RUT','CBOE','ISE','ARCA','NYSE','OTC','GDAX','BSTAMP','KRAKEN','TST','USR1','USR2','USR3','NSDQ','MFQS','PHLX','MIAX','TSE')	
3	fkey_tk	varchar(12)	
4	fkey_ts	enum('None','SR','NMS','CME','ICE','CFE','CBOT','TD','NYMEX','COMEX','RUT','CBOE','ISE','ARCA','NYSE','OTC','GDAX','BSTAMP','KRAKEN','TST','USR1','USR2','USR3','NSDQ','MFQS','PHLX')	
5	fkey_yr	smallint(5)	
6	fkey_mn	tinyint(3)	
7	fkey_dy	tinyint(3)	
8	date	varchar(10)	End of Minte Bar
9	tradingDate	DateKey	
10	tradingSession	enum('None','RegularMkt','PreMkt','PostMkt','PostMktETF','NextDay')	Trading Session: ('None','RegularMkt','PreMkt','PostMkt','PostMktETF','NextDay')
11	prtOpen	double	Print at open of period (Bar)
12	prtHigh	double	High price during period (Bar)
13	prtLow	double	Low price during period (Bar)
14	prtLast	double	Print at close of period (Bar)
15	prtVWap	double	Period (Bar) VWap
16	prtVolume	int(11)	Period (Bar) number of contracts
17	prtCount	int(11)	Number of prints during the period
18	qteHiBid	double	Quote bid high
19	qteLoAsk	double	Quote ask low
20	qteTwap	double	Quote Twap
21	qteCount	int(11)	Quote number of contracts
22	bid	double	Bid at the time of the print
23	ask	double	Ask at the time of the print
24	bidSz	int(11)	Bid size at the time of print
25	askSz	int(11)	Ask size at the time of print
26	width	float	reference point for expWidth (SR debug use)
27	isEOB	enum('None','Yes','No')	is end-of-bar (every 10 minutes)
28	isEOH	enum('None','Yes','No')	is end-of-hour
29	timestamp	datetime(3)	same as date + time

Data Updates/Fixes/Problems

Futures Print Set

Futures Print Set records are created for all future prints (outrights and spreads) and updated at 1 minute and 10 minutes to monitor trade performance.

Order	Field	Data Type	Comment
1	fkey_at	enum('None','EQT','IDX','BND','CUR','COM','FUT','SYN','WAR','FLX','MUT','SPD','MM','MF','COIN','TOKEN')	
2	fkey_ts	enum('None','SR','NMS','CME','ICE','CFE','CBOT','TD','NYMEX','COMEX','RUT','CBOE','ISE','ARCA','NYSE','OTC','GDAX','BSTAMP','KRAKEN','TST','USR1','USR2','USR3','NSDQ','MFQS','PHLX','MIAX','TSE')	
3	fkey_tk	varchar(12)	
4	fkey_ts	enum('None','SR','NMS','CME','ICE','CFE','CBOT','TD','NYMEX','COMEX','RUT','CBOE','ISE','ARCA','NYSE','OTC','GDAX','BSTAMP','KRAKEN','TST','USR1','USR2','USR3','NSDQ','MFQS','PHLX')	
5	fkey_yr	smallint(5)	
6	fkey_mn	tinyint(3)	
7	fkey_dy	tinyint(3)	
8	date	date	
9	prtNumber	bigint(20)	Unique print set identifier, will increment but not guaranteed to be sequential
10	ticker_tk	varchar(12)	underlying stock key (ticker symbol)
11	tradingSession	enum('None','RegularMkt','PreMkt','PostMkt','PostMktETF','NextDay')	Trading Session: ('None','RegularMkt','PreMkt','PostMkt','PostMktETF','NextDay')
12	tradingDate	DateKey	
13	prtExch	varchar(15)	print exchange
14	prtSize	int(11)	print size [contracts]
15	prtPrice	double	print price
16	prtOrders	tinyint(5)	number of orders participating in this print
17	prtClusterNum	int(11)	incremental print cluster counter (one counter per fkey; used to group prints into clusters)
18	prtClusterSize	int(11)	cumulative size of prints in this sequence (prints @ same or more aggressive price with less than 25 ms elapsing since first print; can span exchanges)
19	prtVolume	int(11)	cumulative day (electronic) print volume in contracts
20	prtSide	enum('None','Mid','Bid','Ask')	Print side: None; Mid; Bid; Ask
21	prtTimestamp	bigint(11)	exchange high precision timestamp (if available)
22	netTimestamp	bigint(11)	inbound packet PTP timestamp from SR gateway switch; usually synchronized

			with facility grandfather clock
23	timestamp	datetime(3)	YYYY-MM-DD HH:mm:ss
24	bidPrice	double	exch best bid @ print arrival time
25	askPrice	double	exch best ask @ print arrival time
26	bidSize	int(11)	bid size @ print arrival time
27	askSize	int(11)	ask size @ print arrival time
28	bidPrice2	double	exch 2nd best bid @ print arrival time
29	askPrice2	double	exch 2nd best ask @ print arrival time
30	bidSize2	int(11)	2nd best bid @ print arrival time
31	askSize2	int(11)	2nd best ask @ print arrival time
32	prtProbability	float	[M1] probability that buying prtQuan contracts @ prtPrice will have positive m1 pnl (prtPriceM1 >= prtPrice) [recorded at time of print]
33	prtProbability M2	float	alternate probability model
34	prtProbability M3	float	alternate probability model
35	bidPriceM1	double	bid price +1 minute
36	askPriceM1	double	ask price +1 minute
37	prtPriceM1	double	market price +1 minute [mid-quote if not intervening prints; most recent print otherwise]
38	pnlM1	float	pnl after 1 minute
39	pnlM1Err	enum('None','Yes','No')	M1PnL Error indicator ('None','Yes','No')
40	bidPriceM10	double	bid price +10 minutes
41	askPriceM10	double	bid price +10 minutes
42	prtPriceM10	double	market price +10 minutes [most recent print (if any) otherwise mid-quote]
43	pnlM10	float	pnl after 10 minutes
44	pnlM10Err	enum('None','Yes','No')	M10PnL Error indicator ('None','Yes','No')

Data Updates/Fixes/Problems