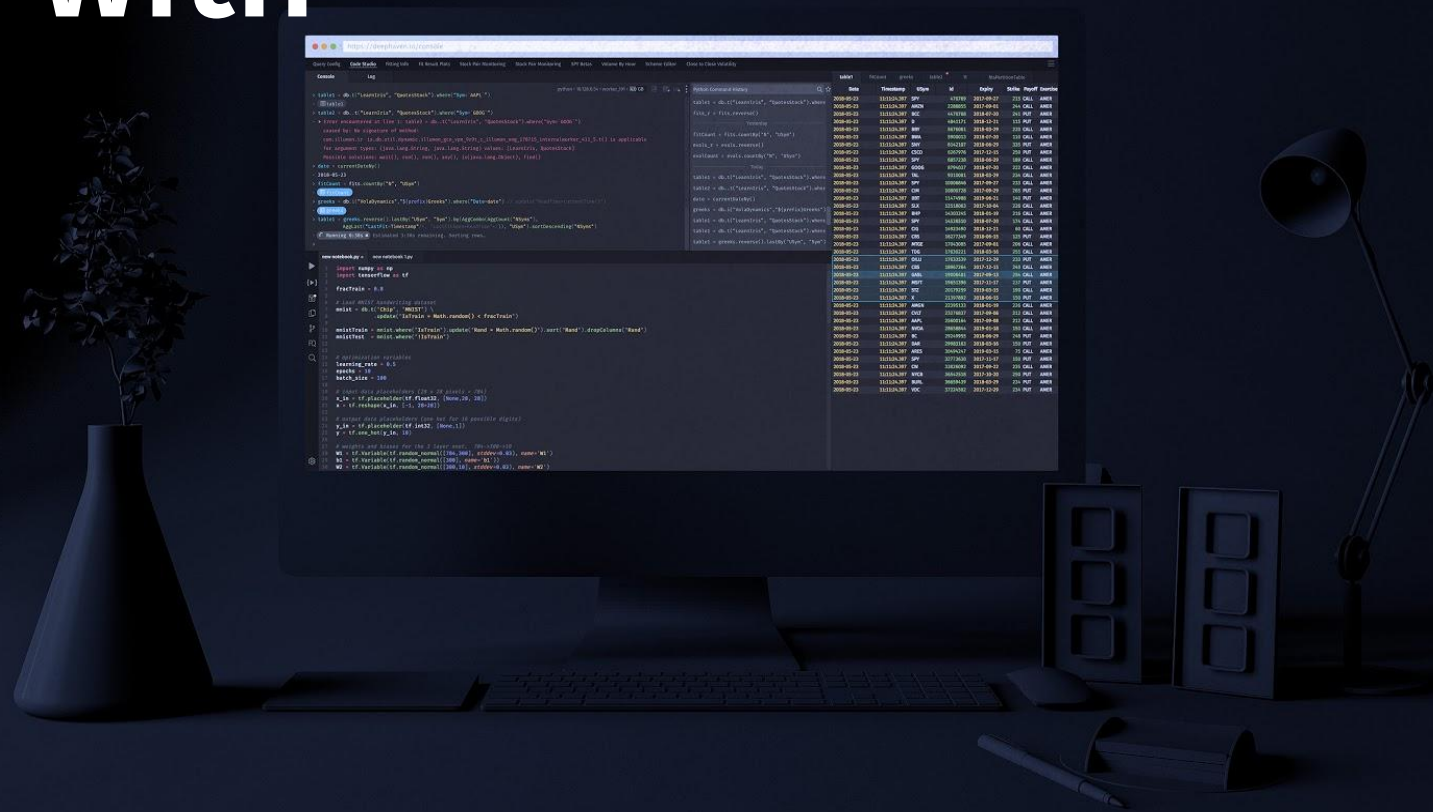


Building modern data systems with Deephaven



What is Deephaven?



Deephaven -- Why does it exist?

deephaven.io



Real-time data is hard.



People waste time with data infrastructure.



Sharing work & “getting to production” is slow.

Deephaven -- How do people use it?

deephaven.io

- 1 To build apps
- 2 As a complete data system
- 3 As a dashboard and visualization framework





Deephaven fundamentals

**Speed
matters**

**People's time
matters**

**Interoperability
matters**

Deephaven Technologies

worker_363 • dh-prod-demo-query3.int.illumon.com • Python • 12.0 GB

```
> join_Trades_Quotes = Trades.aj(Quotes, "Symbol, Timestamp", "Bid, Ask, QuoteTime = Timestamp")\
.updateView("TimeDiff = round((Timestamp - QuoteTime) / 1000)")
```

Counts

N	Source
5,508,681	Trades
5,728,804	Quotes

Quotes

Timestamp	Symbol	BidSize
11:50:16.014	TDG	100.0000
11:50:16.012	MPC	100.0000
11:50:16.012	LEN	100.0000
11:50:16.011	AEO	600.0000
11:50:16.011	EFX	200.0000
11:50:16.010	CNP	100.0000
11:50:16.009	TPR	1,500.0000

Trades

Timestamp	Symbol	Price	Size
11:50:16.011	NEE	299.08	
11:50:16.011	NEE	299.08	
11:50:16.010	CNP	20.96	
11:50:16.009	QQQ	280.70	
11:50:16.000	NCLH	17.76	1
11:50:16.000	DPZ	399.49	
11:50:15.984	PBA	21.86	1

SEC Scrape

14 Betas

15 Jup Betas

Vol and Corr

Snotel Dashboard

Big Studio

Python Studio

Market Summary

Earnings Call RSS SA

Code Editor

New

Controls

Panels

Sharing

ConsoleLog

O worker_364 - dh-prod-demo-query3.int.illumon.com • Python • 12.0 GB

< >

Command HistoryFile ExplorerPQ Explorer

Search by name

Exercising Notebooks

General Notebooks

Ema with Threads.groovy

Price Changes.py

Simple Query Basics.groovy

Simple Query Basics.py

Tensor conversion.py

Window check.groovy

Temp Demo Notebook

EOD Generator.groovy

```
> from deephaven import *  
totals2 = tradesMkt.updateView("Dollars = Size * Price")\  
.by(caf.AggCombo(\  
    caf.AggFirst("FirstPrice = Price"),\  
    caf.AggLast("LastPrice = Price"),\  
    caf.AggCount("Count"),\  
    caf.AggSum("Shares = Size", "Dollars"),\  
    caf.AggAvg("AvgPrice = Price"),\  
    caf.AggWavg("Size", "WtdAvgPrice = Price")\  
) , "Sym")\  
.updateView("IntraChange = LastPrice - FirstPrice", "IntraPerc = LastPrice / FirstPrice - 1")  
  
- totals2  
  
>
```

Simple Query Basics.py

▶ { } ⌕

PO

4 #1 Get intraday trades
5 #tradesToday = db.i("FeedOS", "EquityTradeL1").where("Date = currentDateNy()")
6 tradesToday = db.i("FeedOS", "EquityTradeL1").where("Date = lastBusinessDateNy()")
7
8 #2 Do a couple basic things to that table
9 tradesToday = tradesToday.reverse().renameColumns("Sym = LocalCodeStr")
10
11 #3 Create a dependent table -- and demonstrate Aggregation / lastBy
12 tradesLast = tradesToday.view("MarketTimestamp", "Sym", "Price", "Size").lastBy("Sym").sort("Sym")
13
14 #4 Get historical trades
15 # tradesYest = db.t("FeedOS", "EquityTradeL1").where("Date = lastBusinessDateNy()")\
16 tradesYest = db.t("FeedOS", "EquityTradeL1").where("Date = '2020-07-01'")\
17 .renameColumns("Sym = LocalCodeStr")
18
19 #5 You can of course merge them
20 trades = ttools.merge(tradesToday, tradesYest)
21
22 #6 Sort for trading time and calc totals
23 import deephaven.Calendars as Calendars
24 cal = Calendars.calendar("USNYSE")
25
26 tradesMkt = tradesToday.where("cal.isBusinessTime(MarketTimestamp)")
27 totals = tradesMkt.view("Date", "Sym", "Size", "Dollars = Size * Price").sumBy("Date", "Sym").sortDe
28
29 #7 Analyze performance
30 # One can use the following to figure out I/O versus compute time for various operations, etc.
31 qopl=db.i("DbInternal", "QueryOperationPerformanceLog")\
32 .where("Dates=currentDateNy()-1")

totalsWithJoin

totals2

tradesYest

tradesMkt

oneStockTradesMkt

qopl

trades

totals

tradi

Date

Timestamp

InternalCode

LocalCodeMarketId

Sym

ServerTimestamp

MarketTimestamp

Price

2020-10-07 16:29:44.886

1,059,068,156

505 SPY

2020-10-07T16:14:44.886

2020-10-07T16:00:00.000

340.92

2020-10-07 16:29:16.949

1,059,068,156

505 SPY

2020-10-07T16:14:16.949

2020-10-07T16:00:00.000

340.92

2020-10-07 16:15:11.893

1,059,068,156

505 SPY

2020-10-07T15:59:59.998

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.892

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.892

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.892

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.892

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.892

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.73

2020-10-07 16:15:11.891

1,059,068,156

505 SPY

2020-10-07T15:59:59.997

2020-10-07T15:59:59.997

340.74

stockPlot

Sym: AAPL

115.5

115

114.5

11:30

12:00

12:30

13:00

13:30

14:00

14:30

15:00

15:30

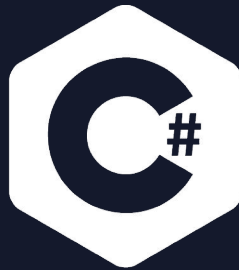
16:00

Oct 7, 2020

Sym

Ratio

Develop with historical and real-time data



User workflows supported

2017-08-24	04:07:27.812	Stock	Nasdaq	AAPL	AAPL	159.8000	200	Normal	1,067	2017-08-24T04:07:27.812	@FI
2017-08-24	04:07:28.172	Stock	Nasdaq	AAPL	AAPL	159.8000	200	Normal	1,068	2017-08-24T04:07:28.172	@FT
2017-08-24	04:07:29.301	Stock	Nasdaq	AAPL	AAPL	159.8000	79	Normal	1,069	2017-08-24T04:07:29.301	@FTI
2017-08-24	04:07:30.753	Stock	Arca	AAPL	AAPL	159.6000	1	Normal	1,070	2017-08-24T04:07:30.753	@FTI
2017-08-24	04:13:07.399	Stock	Nasdaq	AAPL	AAPL	159.8300	1	Normal	1,072	2017-08-24T04:13:07.399	@TI
2017-08-24	04:13:07.399	Stock	Nasdaq	AAPL	AAPL	160.1000	26	Normal	1,073	2017-08-24T04:13:07.399	@TI
2017-08-24	04:13:07.400	Stock	Arca	AAPL	AAPL	160.0200	20	Normal	1,074	2017-08-24T04:13:07.399	@FTI
2017-08-24	04:23:17.545	Stock	Nasdaq	AAPL	AAPL	159.7000	12	Normal	1,075	2017-08-24T04:23:17.545	@TI
2017-08-24	04:23:22.293	Stock	Arca	AAPL	AAPL	160.0200	10	Normal	1,076	2017-08-24T04:23:22.293	@TI
2017-08-24	04:23:55.645	Stock	Arca	AAPL	AAPL	160.0200	4	Normal	1,077	2017-08-24T04:23:55.645	@TI
2017-08-24	04:30:03.136	Stock	Nasdaq	AAPL	AAPL	159.8000	15	Normal	1,079	2017-08-24T04:30:03.136	@TI
2017-08-24	04:30:03.136	Stock	Nasdaq	AAPL	AAPL	159.7100	5	Normal	1,080	2017-08-24T04:30:03.136	@TI
2017-08-24	04:30:03.222	Stock	Nasdaq	AAPL	AAPL	159.8600	1	Normal	1,081	2017-08-24T04:30:03.222	@TI

An example ticking table

```
In [ ]: t2 = db.timeTable("00:00:01").update("X=i")
display(GridWidget(t2))
```

An example plot

```
In [ ]: taAPL = db.t("LearnDeePhaven", "StockTrades")\
        .where("Date=" 2017-08-21", "USym=" AAPL")
display(GridWidget(taAPL))

plotAAPL = Plot.plot("AAPL", taAPL, "Timestamp", "Last").figure
display(ChartWidget(plotAAPL))
```

An example dual chart plot

```
In [ ]: t4 = db.t("LearnDeephaven", "StockTrades").where("Date=`2017-08-24`", "USym in `PFE`,`XOM`")

plotDualYAxis = Plot.plot("PFE", t4.where("USym = `PFE`"), "Timestamp", "Last")\
    .twinx()\
    .plot("XOM", t4.where("USym = `XOM`"), "Timestamp", "Last")\
    .figure
```

Query Monitor

MainMkt Summary02 Simulator13 SEC Scrape14 Betas15 Jup BetasVol CorrSnotel DashboardsBig StudioPython >|+ NewControlsPanelsSharing

System: Deephaven - dh-prod-demo.int.illumon.comPROD

75 Running, 30 Completed, 2 Stopped, 18 Failed, 1 Error, 1 Disconnected, 302 None

NewStartStopRestart

Owner/NameSearch FilterMy QueriesTypeType Worker HostDB Server NameDraftsEnabled Only

Owner	Name	Enabled	Status	Query Type	InitStartTime	
adityapethe	AdverseSelection with decile	true	Running	Live Query (Script)	2020-10-08T07:56:14	202
adityapethe	MomentumTradeModel	true	Running	Live Query (Script)	2020-10-08T07:57:17	202
alexpeters	Volatility Study	true	Running	Live Query (Script)	2020-10-08T07:00:05	202
bender	IDS-6840 Tests	true	Running	Live Query (Script)	2020-10-08T07:55:17	202
bender	stringstuff	true	Running	Live Query (Script)	2020-10-08T07:55:50	202
ckent	COVID-19	true	Running	Live Query (Script)	2020-10-08T03:15:06	202
david	dg_Graph	true	Running	Live Query (Script)	2020-10-08T07:57:15	202
demoUser	Draft, Demo Plots	true	Stopped	Live Query (Script)		
demoUser	EOD Program with Plot	true	Running	Live Query (Script)	2020-10-08T07:55:22	202
harrisonspisak	Correlation	true	Running	Live Query (Script)	2020-10-08T07:55:15	202
harrisonspisak	Poker	true	Running	Live Query (Script)	2020-10-08T07:55:14	202
harrisonspisak	SnotelData	true	Running	Live Query (Script)	2020-10-08T07:55:14	202
iris	Accenture	true	Running	Live Query (Script)	2020-10-08T07:56:48	202
iris	Adam Kreysar	true	Running	Live Query (Script)	2020-10-08T07:56:06	202
iris	All Fitter Outputs	false		Live Query (Script)		
iris	Arman's Demo (Mean Reversion Simulation)	false		Live Query (Script)		
iris	Axpo	true	Running	Live Query (Script)	2020-10-08T07:57:09	202
iris	Axpo 2	true	Running	Live Query (Script)	2020-10-08T07:55:16	202
iris	Basic Andy	true	Running	Live Query (Script)	2020-10-08T07:56:39	202
iris	Basic Example	true	Running	Live Query (Script)	2020-10-08T07:56:55	202
iris	Bender Plotting	true	Running	Live Query (Script)	2020-10-08T07:56:49	202
iris	Beta Calculator	true	Running	Live Query (Script)	2020-10-08T07:55:15	202
iris	Beta Calculator 2	true	Running	Live Query (Script)	2020-10-08T07:55:15	202
iris	CME Futures	true	Running	Live Query (Script)	2020-10-08T07:55:14	202
iris	CME Quotes Downsampling Script	false		Batch Query (RunAndDone)		
iris	CME Quoting Data (Genesis)	false		Live Query (Script)		
iris	CME Slippage and Impact	false		Live Query (Script)		
iris	CharlestTest	false		Live Query (Script)		
iris	Clear Betas	true	Completed	Batch Query (RunAndDone)	2020-09-23T08:49:44	202
iris	CloseToCloseVolatility	false		Live Query (Script)		
iris	Controller Test	false		Live Query (Script)		
iris	Copying Bug Test	false		Live Query (Script)		
iris	Correlation Matrix	false		Live Query (Script)		
iris	DISStats	false		Live Query (Script)		
iris	DIS_LastBy	true	Running	Live Query - Merge Server	2020-10-08T00:05:07	202
iris	Dash Big Names 1mo Returns	true	Running	Live Query (Script)	2020-10-08T07:55:55	202
iris	Dash Price and Volume	true	Running	Live Query (Script)	2020-10-08T07:55:08	202
iris	Dash Real-time Monitor	true	Running	Live Query (Script)	2020-10-08T07:56:51	202
iris	DeleteMayStreetOptionsFiles	true	Completed	Batch Query - Import Server	2020-10-08T09:22:29	202

SummarySettingsPermissionsSchedulingScript

Persistent Query ConfigurationEnabledON

NameD

TypeLive Query (Script)

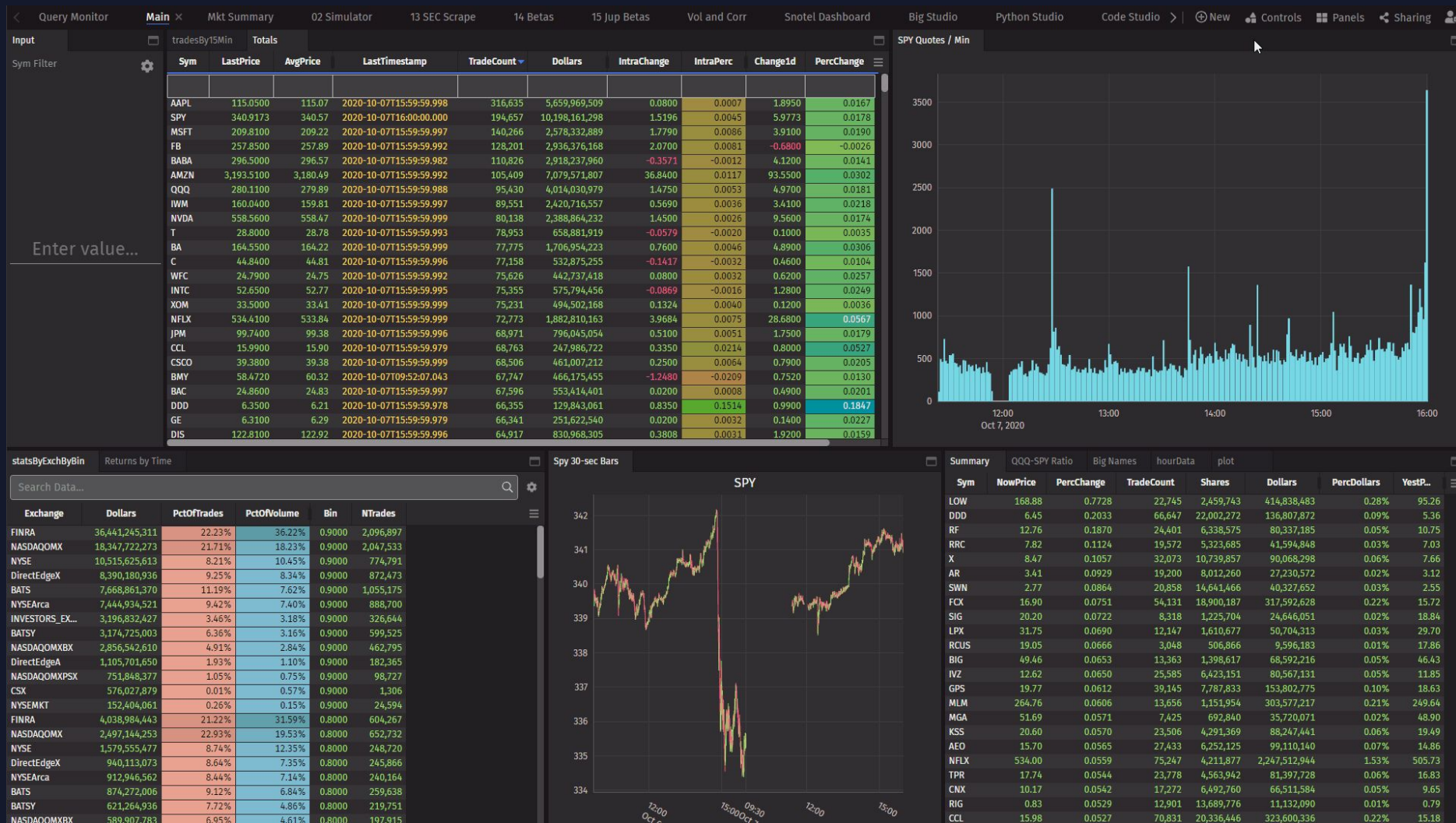
DB ServerQuery_1 (dh-prod-demo-query1.int.illumon.com:2)

Heap Size4Max memory (GB)

Show Advanced

DeleteCopyRevertSave

Go to production with ease



deephaven.io

Share user-created dashboards with colleagues



Deephaven pursuits

deephaven.io

Opportunities for STAC community

Platforms

Turnkey data software solutions for high-end use cases. A capital markets ecosystem of ready data and software.

Compatible wire formats

Project Barrage for real-time data

Pete Goddard
pgoddard@deephaven.io

