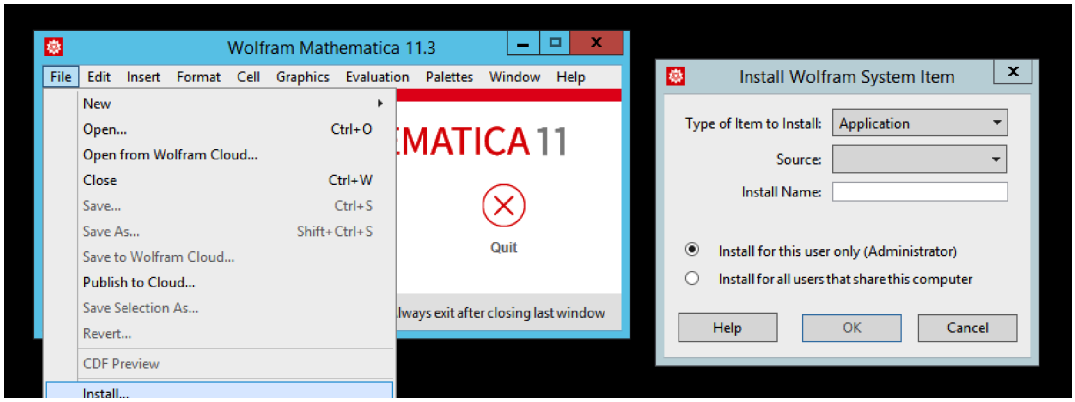
MATH-TWS: a package to connect Mathematica to Interactive Brokers Trader Workstation

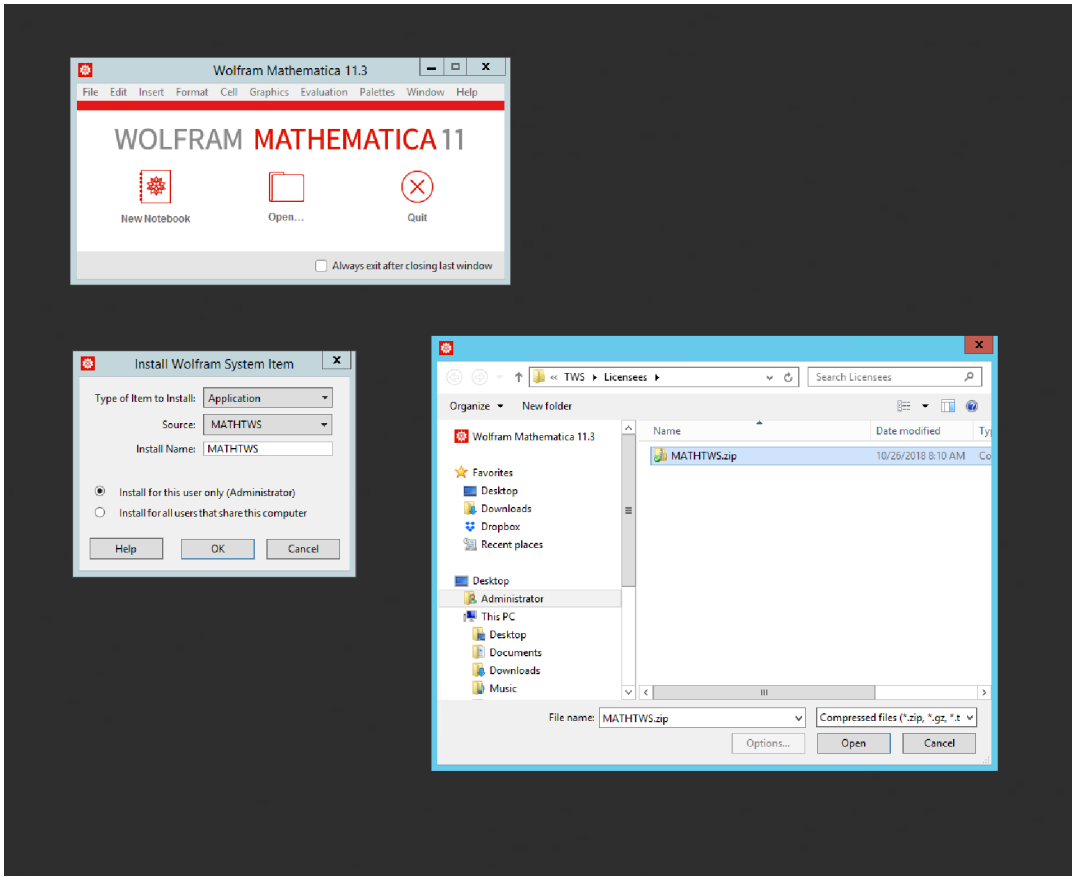
MATH-TWS is a new Mathematica package that connects Wolfram Mathematica to the Interactive Brokers TWS platform via the C++ API. It enables the user to retrieve information from TWS on accounts, portfolios and positions, as well as historical and real-time market data. MATH-TWS also enables the user to place and amend orders and obtain execution confirmations from Mathematica.

Installing MATH-TWS

From the Mathematica Menubar, choose File>Install. A new dialog box will open, “Install Wolfram System Item”.

In the field Type of Item to Install, choose Application. Set the Source field to File and browse to select the IBTWS.zip file and click the Open button. Mathematica will install the MATH-TWS package.





Load the MATH-TWS package

After you have installed the MATH-TWS application, the first step is to load the MATH-TWS package into Mathematica:

In[ ]:=
```
<< MATHTWSWin`
```

```
TWS connector version 0.8.2
```

Activation

After installing package, you will require an activation code. MATH-TWS is offered with a one-week free trial. After your trial expires you can obtain an licence for the TWS product by paying an annual license fee, which is currently $299.

To obtain a trial license, run the twsActivate function and send the generated code to algorithmicexecution@gmail.com with the MATH-TWS Trial License in the subject line. If your trial expires and you wish to obtain or renew a full 1-year license, send an email to algorithmicexecution@gmail.com with the MATH-TWS Full License in the subject. Details will be provided by return email.

In[ ]:=
```
twsActivate[]
```

```
Send this code for getting activation
  key: O/c5KjG+zj7Tlf9NdfZ5dVQNPT6ZqSFJYOXgvwVIl+
  PrLIeM6nlqQL4kyeSxsMjXhrpNl2t2b6pTl7I2QYcT0An7qCHUbHyqUH8m9LgJNllLOmS1iz2CPrRwgk7+2
  ZfX05ubMzmCNELbx4WmVpLOoRDpRcp0ZKoqIaveFRt1J9E=
```

After you have sent your code you will receive an activation key by return email. To activate TWS, enter the activate key using the twsActivate function. For example, if your activate code is

In[ ]:=
```
twsActivate[
 "TnljS29BUnRHSmtFWjFacmI5OHJiNnVvdlZiQmYrYTN6V01DM0lHek5jN3hiaEtFa3N0aldmc1Vr\
    a3hNQW1PbTJraDArQTlwRDd0dEJySSsrRnA1YUpRc1RIUnRVZGdSM3NVbDRxUzhERmJOQjdt\
    SFNCSDg1MzRmSzdqaEFoaDZsZmpjQjJXNXQvZ3RTM0FKaXc4YXp0WjV1VzRnbExlaVJxSWtp\
    TE0vSlJFPQp2LjEKOWZkNzA4OTItMjg0OS00YzE4LTk3YWMtN2Q2MGM1MDMzZmRjCjIxNzcz\
    NjI4MDAK"]
```

```
OK
```

TWS Setup

Connection and service functions

The twsConnect function is used to connect Mathematica to a live IB TWS session. It is important to specify the correct port number and ClientId. These are set in the API settings in the TWS application.

The twsConnect function is used to connect Mathematica to a current instance of IB TWS by specifying the correct port number and client id, which are set in the TWS application:

In[ ]:=
```
twsConnect[<|"Host" → "127.0.0.1", "Port" → 7496, "ClientId" → 5|>]
```

Out[ ]=
```
True
```

The arguments in all functions in MATH-TWS can be specified explicitly, as above, or by specifying the argument values in the correct order:

In[ ]:=
```
twsConnect["127.0.0.1", 7496, 0]
```

Out[ ]=
```
True
```

We check the connection is valid using twsIsConnected:

In[ ]:=
```
twsIsConnected[]
```

Out[ ]=
```
True
```

The twsLog function keeps track of any messages logged in the interaction between Mathematica and TWS:

In[ ]:=
```
twsLogClear[]
```

```
In[ ]:= twsLog[]
Out[ ]=
```

In the following sections we will illustrate the functionality of the MATH-TWS package using the full functional form and show the abbreviated equivalent form in comments.

Account information

We begin by retrieving a list of accounts using the twsAccounts function. This works for single brokerage accounts, or multiple accounts in a Financial Advisor structure, as in this example:

```
In[ ]:= accounts = twsAccounts[]
Out[ ]=
{DF220168, DU220169, DU220170, DU220171, DU220172, DU220173}

In[ ]:= Reverse[accounts]
Out[ ]=
{DU220173, DU220172, DU220171, DU220170, DU220169, DF220168}

In[ ]:= myAccount = accounts[[-1]]
Out[ ]=
DU220173

In[ ]:= myaccountinfo = twsAccount[<|"Account" → myAccount|>]
  (*  Abbreviated form:  twsAccount[myAccount]*)
Out[ ]=
<|AccountCode → DU220173, AccountOrGroupJPY → DU220173,
 AccountOrGroupUSD → DU220173, AccountReady → true, AccountType → LLC,
 AccruedCashJPY → 0., AccruedCashUSD → 4579.06, AccruedCash_CUSD → 0.,
 AccruedCash_SUSD → 4579.06, AccruedDividendUSD → 0., AccruedDividend_CUSD → 0.,
 AccruedDividend_SUSD → 0., AvailableFundsUSD → 3.65016 × 10^6,
 AvailableFunds_CUSD → 248 842., AvailableFunds_SUSD → 3.40132 × 10^6,
 BillableUSD → 0., Billable_CUSD → 0., Billable_SUSD → 0.,
 BuyingPowerUSD → 1.46006 × 10^7, CashBalanceJPY → -3255.91,
 CashBalanceUSD → 3.64561 × 10^6, ColumnPrio_C → 2, ColumnPrio_S → 1,
 CorporateBondValueJPY → 0., CorporateBondValueUSD → 0.,
 CurrencyJPY → JPY, CurrencyUSD → USD, Cushion → 1, DayTradesRemaining → -1,
 DayTradesRemainingT_1 → -1, DayTradesRemainingT_2 → -1,
 DayTradesRemainingT_3 → -1, DayTradesRemainingT_4 → -1,
 EquityWithLoanValueUSD → 3.65016 × 10^6, EquityWithLoanValue_CUSD → 248 842.,
 EquityWithLoanValue_SUSD → 3.40132 × 10^6, ExcessLiquidityUSD → 3.65016 × 10^6,
 ExcessLiquidity_CUSD → 248 842., ExcessLiquidity_SUSD → 3.40132 × 10^6,
 ExchangeRateJPY → 0.007585, ExchangeRateUSD → 1.,
 FullAvailableFundsUSD → 3.65016 × 10^6, FullAvailableFunds_CUSD → 248 842.,
 FullAvailableFunds_SUSD → 3.40132 × 10^6, FullExcessLiquidityUSD → 3.65016 × 10^6,
 FullExcessLiquidity_CUSD → 248 842., FullExcessLiquidity_SUSD → 3.40132 × 10^6,
 FullInitMarginReqUSD → 0.74, FullInitMarginReq_CUSD → 0.,
 FullInitMarginReq_SUSD → 0.74, FullMaintMarginReqUSD → 0.74,
 FullMaintMarginReq_CUSD → 0., FullMaintMarginReq_SUSD → 0.74,
```

FundValueJPY → 0., FundValueUSD → 0., FutureOptionValueJPY → 0.,
FutureOptionValueUSD → 0., FuturesPNLJPY → 0., FuturesPNLUSD → 0.,
FxCashBalanceJPY → 0., FxCashBalanceUSD → 0., GrossPositionValueUSD → 0.,
GrossPositionValue_SUSD → 0., GuaranteeUSD → 0., Guarantee_CUSD → 0.,
Guarantee_SUSD → 0., IndianStockHaircutUSD → 0., IndianStockHaircut_CUSD → 0.,
IndianStockHaircut_SUSD → 0., InitMarginReqUSD → 0.74, InitMarginReq_CUSD → 0.,
InitMarginReq_SUSD → 0.74, IssuerOptionValueJPY → 0., IssuerOptionValueUSD → 0.,
Leverage_S → 0., LookAheadAvailableFundsUSD → 3.65016×10^6,
LookAheadAvailableFunds_CUSD → 248 842.,
LookAheadAvailableFunds_SUSD → 3.40132×10^6,
LookAheadExcessLiquidityUSD → 3.65016×10^6,
LookAheadExcessLiquidity_CUSD → 248 842.,
LookAheadExcessLiquidity_SUSD → 3.40132×10^6, LookAheadInitMarginReqUSD → 0.74,
LookAheadInitMarginReq_CUSD → 0., LookAheadInitMarginReq_SUSD → 0.74,
LookAheadMaintMarginReqUSD → 0.74, LookAheadMaintMarginReq_CUSD → 0.,
LookAheadMaintMarginReq_SUSD → 0.74, LookAheadNextChange → 0,
MaintMarginReqUSD → 0.74, MaintMarginReq_CUSD → 0.,
MaintMarginReq_SUSD → 0.74, MoneyMarketFundValueJPY → 0.,
MoneyMarketFundValueUSD → 0., MutualFundValueJPY → 0., MutualFundValueUSD → 0.,
NLVAndMarginInReview → false, NetDividendJPY → 0., NetDividendUSD → 0.,
NetLiquidationUSD → 3.65016×10^6, NetLiquidation_CUSD → 248 842.,
NetLiquidation_SUSD → 3.40132×10^6, NetLiquidationByCurrencyJPY → −3255.91,
NetLiquidationByCurrencyUSD → 3.65019×10^6,
NetLiquidationUncertaintyUSD → 0., OptionMarketValueJPY → 0.,
OptionMarketValueUSD → 0., PASharesValueUSD → 0., PASharesValue_CUSD → 0.,
PASharesValue_SUSD → 0., PhysicalCertificateValueUSD → 0.,
PhysicalCertificateValue_CUSD → 0., PhysicalCertificateValue_SUSD → 0.,
PostExpirationExcessUSD → 0., PostExpirationExcess_CUSD → 0.,
PostExpirationExcess_SUSD → 0., PostExpirationMarginUSD → 0.,
PostExpirationMargin_CUSD → 0., PostExpirationMargin_SUSD → 0.,
PreviousDayEquityWithLoanValueUSD → 3.40017×10^6,
PreviousDayEquityWithLoanValue_SUSD → 3.40017×10^6, RealCurrencyJPY → JPY,
RealCurrencyUSD → USD, RealizedPnLJPY → 0., RealizedPnLUSD → 0.,
RegTEquityUSD → 3.40132×10^6, RegTEquity_SUSD → 3.40132×10^6, RegTMarginUSD → 0.,
RegTMargin_SUSD → 0., SMAUSD → 3.40132×10^6, SMA_SUSD → 3.40132×10^6,
SegmentTitle_C → US Commodities, SegmentTitle_S → US Securities,
StockMarketValueJPY → 0., StockMarketValueUSD → 0., TBillValueJPY → 0.,
TBillValueUSD → 0., TBondValueJPY → 0., TBondValueUSD → 0.,
TotalCashBalanceJPY → −3255.91, TotalCashBalanceUSD → 3.64561×10^6,
TotalCashValueUSD → 3.64558×10^6, TotalCashValue_CUSD → 248 842.,
TotalCashValue_SUSD → 3.39674×10^6, TotalDebitCardPendingChargesUSD → 0.,
TotalDebitCardPendingCharges_CUSD → 0., TotalDebitCardPendingCharges_SUSD → 0.,
TradingType_S → STKNOPT, UnrealizedPnLJPY → 0., UnrealizedPnLUSD → 0.,
WarrantValueJPY → 0., WarrantValueUSD → 0., WhatIfPMEnabled → true |⟩

Where possible, MATH-TWS functions return a result formatted as an association. This enables us to convert the result into other formats such as a Dataset, for example:

In[ ]:=
```
Dataset[myaccountinfo]
```

Out[ ]=

AccountCode	DU220173
AccountOrGroupJPY	DU220173
AccountOrGroupUSD	DU220173
AccountReady	true
AccountType	LLC
AccruedCashJPY	0.0
AccruedCashUSD	2277.02
AccruedCash_CUSD	0.0
AccruedCash_SUSD	2277.02
AccruedDividendUSD	0.0
AccruedDividend_CUSD	0.0
AccruedDividend_SUSD	0.0
AvailableFundsUSD	3 647 860.
AvailableFunds_CUSD	248 842.
AvailableFunds_SUSD	3 399 018.
BillableUSD	0.0
Billable_CUSD	0.0
Billable_SUSD	0.0
BuyingPowerUSD	14 591 440.
CashBalanceJPY	–3255.91

rows 1–20 of **152**

In[ ]:=
```
twsLogClear[]
```

We can obtain account summary information across all accounts using the twsAccountSummary function:

```
In[•]:= accountSummary = twsAccountSummary[<|"Group" → "All"|>]
        (*Abbreviated form: twsAccountSummary[] *)
```

```
Out[•]=
⟨| Currency_JPY → JPY, CashBalance_JPY → -3255.91,
 TotalCashBalance_JPY → -3255.91, AccruedCash_JPY → 0., StockMarketValue_JPY → 0.,
 OptionMarketValue_JPY → 0., FutureOptionValue_JPY → 0.,
 FuturesPNL_JPY → 0., NetLiquidationByCurrency_JPY → -3255.91,
 UnrealizedPnL_JPY → 0., RealizedPnL_JPY → 0., ExchangeRate_JPY → 0.007585,
 FundValue_JPY → 0., NetDividend_JPY → 0., MutualFundValue_JPY → 0.,
 MoneyMarketFundValue_JPY → 0., CorporateBondValue_JPY → 0., TBondValue_JPY → 0.,
 TBillValue_JPY → 0., WarrantValue_JPY → 0., FxCashBalance_JPY → 0.,
 AccountOrGroup_JPY → All, RealCurrency_JPY → JPY, IssuerOptionValue_JPY → 0.,
 Cryptocurrency_JPY → , Currency_EUR → EUR, CashBalance_EUR → 44 374.,
 TotalCashBalance_EUR → 44 374., AccruedCash_EUR → 0., StockMarketValue_EUR → 0.,
 OptionMarketValue_EUR → 0., FutureOptionValue_EUR → 0.,
 FuturesPNL_EUR → 0., NetLiquidationByCurrency_EUR → 44 374.,
 UnrealizedPnL_EUR → 0., RealizedPnL_EUR → 0., ExchangeRate_EUR → 1.07383,
 FundValue_EUR → 0., NetDividend_EUR → 0., MutualFundValue_EUR → 0.,
 MoneyMarketFundValue_EUR → 0., CorporateBondValue_EUR → 0.,
 TBondValue_EUR → 0., TBillValue_EUR → 0., WarrantValue_EUR → 0.,
 FxCashBalance_EUR → 0., AccountOrGroup_EUR → All, RealCurrency_EUR → EUR,
 IssuerOptionValue_EUR → 0., Cryptocurrency_EUR → , Currency_USD → USD,
 CashBalance_USD → 2.14177 × 10^7, TotalCashBalance_USD → 2.14177 × 10^7,
 AccruedCash_USD → 27 047.8, StockMarketValue_USD → 0.,
 OptionMarketValue_USD → 0., FutureOptionValue_USD → 0., FuturesPNL_USD → 0.,
 NetLiquidationByCurrency_USD → 2.14448 × 10^7, UnrealizedPnL_USD → 0.,
 RealizedPnL_USD → 0., ExchangeRate_USD → 1., FundValue_USD → 0.,
 NetDividend_USD → 0., MutualFundValue_USD → 0., MoneyMarketFundValue_USD → 0.,
 CorporateBondValue_USD → 0., TBondValue_USD → 0., TBillValue_USD → 0.,
 WarrantValue_USD → 0., FxCashBalance_USD → 0., AccountOrGroup_USD → All,
 RealCurrency_USD → USD, IssuerOptionValue_USD → 0., Cryptocurrency_USD → |⟩
```

We can extract any specific component of the account summary information:

```
In[•]:= accountSummary["CashBalance_USD"]
```

```
Out[•]=
2.12011 × 10^7
```

Portfolio, Positions

Full information about the account portfolio is obtained using the twsPortfolio function. Again, the result is returned in the form of an association:

```
In[•]:= myportfolio = twsPortfolio[<|"Account" → myAccount|>]
        (*Abbreviated form: twsPortfolio[myAccount] *)
```

```
Out[•]=
⟨| Symbol → {}, Position → {}, MarketPrice → {}, MarketValue → {},
 AverageCost → {}, UnrealizedPNL → {}, RealizedPNL → {} |⟩
```

Information on positions held in a specific account is obtained using the twsPositions function:

```
In[ ]:= mypositions = twsPositions[<|"Account" → myAccount|>]
         (* Abbrediated form: twsPositions[myAccount]*)
```

```
Out[ ]= <|Symbol → {ESZ2}, Position → {2}, AvgCost → {190 808}|>
```

This data might be handled more conveniently as a Dataset for some applications:

```
In[ ]:= Transpose@Dataset[mypositions]
```

Out[ ]=

Symbol	Position	AvgCost
MRK	3 584 865 303 386 914 916	93.8
NFLX	12 808 237 340 241 690 724	276.994
WMT	12 808 237 340 241 690 724	134.237
QCOM	3 584 865 303 386 914 916	111
TSLA	12 808 237 340 241 690 724	219.765
CRM	12 808 237 340 241 690 724	153.516
AMZN	12 808 237 340 241 690 724	114.147
XOM	3 584 865 303 386 914 916	103.5
ORCL	12 808 237 340 241 690 724	66.8983
TGT	12 808 237 340 241 690 724	156.136
AMD	12 808 237 340 241 690 724	58.6885
PG	12 808 237 340 241 690 724	131.067
TXN	12 808 237 340 241 690 724	150.276
INTC	3 584 865 303 386 915 116	26.065
CVX	3 584 865 303 386 914 916	165.45
CAT	12 808 237 340 241 690 724	184.056
SBUX	12 808 237 340 241 690 724	88.8778
JPM	3 584 865 303 386 914 916	118.55
META	12 808 237 340 241 690 724	135.367
NVDA	12 808 237 340 241 690 724	120.237

rows 1–20 of **29**

Historical data

One of the useful applications of MATH-TWS is to obtain historical price data for securities, using the twsHistoricalData function:

In[ ]:=
```
AAPL = twsHistoricalData[<|"Symbol" → "AAPL", "Exchange" → "SMART",
   "SecType" → "STK", "BarSize" → "1 Day", "Duration" → "1 M"|>]
  (*AAPL=twsHistoricalData["AAPL", "1 Day", "1 M"]*)
```

Out[ ]=

```
⟨| Date → { Thu 16 Feb 2023 , Fri 17 Feb 2023 , Tue 21 Feb 2023 , Wed 22 Feb 2023 ,
    Thu 23 Feb 2023 , Fri 24 Feb 2023 , Mon 27 Feb 2023 , Tue 28 Feb 2023 ,
    Wed 1 Mar 2023 , Thu 2 Mar 2023 , Fri 3 Mar 2023 , Mon 6 Mar 2023 , Tue 7 Mar 2023 ,
    Wed 8 Mar 2023 , Thu 9 Mar 2023 , Fri 10 Mar 2023 , Mon 13 Mar 2023 ,
    Tue 14 Mar 2023 , Wed 15 Mar 2023 , Thu 16 Mar 2023 , Fri 17 Mar 2023 } ,
  Open → {153.43, 152.35, 150.33, 148.81, 150.09, 147.1, 147.71,
    147.05, 146.83, 144.4, 148.05, 153.77, 153.64, 152.77,
    153.49, 150.21, 147.81, 151.28, 151.19, 152.15, 156.08},
  High → {156.33, 153, 151.3, 149.95, 150.34, 147.19, 149.17,
    149.08, 147.23, 146.71, 151.11, 156.3, 154.03, 153.47,
    154.54, 150.94, 153.14, 153.4, 153.25, 156.46, 156.74},
  Low → {153.34, 150.85, 148.4, 147.16, 147.24, 145.72, 147.45,
    146.83, 145.01, 143.9, 147.33, 153.46, 151.13, 151.83,
    150.22, 147.6, 147.7, 150.1, 149.92, 151.64, 154.28},
  Close → {153.71, 152.55, 148.48, 148.91, 149.4, 146.71, 147.92,
    147.41, 145.31, 145.91, 151.03, 153.83, 151.6, 152.87,
    150.59, 148.5, 150.47, 152.59, 152.99, 155.85, 155},
  Volume → {470 699, 417 468, 374 226, 376 782, 352 661, 397 363, 320 193,
    302 050, 371 458, 366 468, 477 440, 653 775, 416 414, 371 636,
    399 626, 507 301, 635 981, 503 999, 535 385, 567 693, 602 112} |⟩
```

```
In[ ]:= twsHistoricalData[<|"Symbol" → "USD.JPY", "Exchange" → "IDEALPRO",
          "SecType" → "CASH", "BarSize" → "1 Day", "Duration" → "1 M"|>]
```

```
Out[ ]=
⟨|Date → {Thu 16 Feb 2023, Fri 17 Feb 2023, Mon 20 Feb 2023, Tue 21 Feb 2023,
    Wed 22 Feb 2023, Thu 23 Feb 2023, Fri 24 Feb 2023, Mon 27 Feb 2023,
    Tue 28 Feb 2023, Wed 1 Mar 2023, Thu 2 Mar 2023, Fri 3 Mar 2023, Mon 6 Mar 2023,
    Tue 7 Mar 2023, Wed 8 Mar 2023, Thu 9 Mar 2023, Fri 10 Mar 2023, Mon 13 Mar 2023,
    Tue 14 Mar 2023, Wed 15 Mar 2023, Thu 16 Mar 2023, Fri 17 Mar 2023},
  Open → {134.11, 133.9, 134.262, 134.262, 134.975, 134.845, 134.71,
    136.447, 136.22, 136.145, 136.183, 136.752, 136.002, 135.923,
    137.08, 137.275, 136.17, 134.395, 133.245, 134.25, 133.428, 133.722},
  High → {134.465, 135.118, 134.54, 135.23, 135.065, 135.365, 136.523,
    136.558, 136.92, 136.47, 137.097, 136.77, 136.195, 137.195, 137.915,
    137.377, 136.998, 135.06, 134.903, 135.115, 133.83, 133.722},
  Low → {133.605, 133.9, 133.923, 134.148, 134.365, 134.488, 134.055,
    135.915, 135.735, 135.257, 136.025, 135.745, 135.365, 135.542,
    136.48, 135.938, 134.115, 132.287, 133.03, 132.215, 131.718, 131.555},
  Close → {133.94, 134.148, 134.255, 135.007, 134.787, 134.71, 136.46,
    136.19, 136.185, 136.175, 136.765, 135.887, 135.925, 137.16, 137.352,
    136.148, 135.018, 133.192, 134.218, 133.46, 133.743, 131.838},
  Volume → {-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
    -1, -1, -1, -1, -1, -1, -1, -1, -1, -1}|⟩
```

```
In[ ]:= twsLog[]
```

```
Out[ ]=
IB message [162] Historical Market Data Service error
  message:No historical market data for USD/CASH@IDEALPRO Last 1d
```

Data Manipulation & Charting

We create a Dataset from the association to display the data in tabular form:

In[ ]:=
```
dsAAPL = Transpose@Dataset@AAPL
```

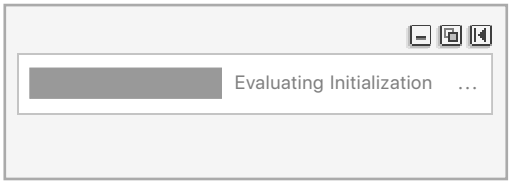
Out[ ]=

Date	Open	High	Low	Close	Volume
Tue 17 Jan 2023	134.83	137.29	134.13	135.94	468 408
Wed 18 Jan 2023	136.82	138.61	135.03	135.21	502 843
Thu 19 Jan 2023	134.09	136.25	133.77	135.27	436 645
Fri 20 Jan 2023	135.21	138.02	134.22	137.87	545 406
Mon 23 Jan 2023	138.13	143.32	137.9	141.11	625 858
Tue 24 Jan 2023	140.31	143.16	140.3	142.53	442 876
Wed 25 Jan 2023	140.89	142.43	138.81	141.86	494 583
Thu 26 Jan 2023	143.17	144.25	141.9	143.96	399 380
Fri 27 Jan 2023	143.18	147.23	143.08	145.93	504 871
Mon 30 Jan 2023	144.96	145.55	142.85	143	458 022
Tue 31 Jan 2023	142.69	144.34	142.28	144.29	377 595
Wed 1 Feb 2023	143.94	146.61	141.32	145.43	567 094
Thu 2 Feb 2023	148.82	151.18	148.17	150.82	701 740
Fri 3 Feb 2023	148.03	157.38	147.83	154.5	1 084 740.
Mon 6 Feb 2023	152.58	153.1	150.78	151.73	494 004
Tue 7 Feb 2023	150.73	155.23	150.63	154.65	595 566
Wed 8 Feb 2023	153.77	154.58	151.16	151.92	447 044
Thu 9 Feb 2023	153.78	154.33	150.42	150.87	387 271
Fri 10 Feb 2023	149.49	151.35	149.22	151.01	376 996
Mon 13 Feb 2023	150.96	154.26	150.92	153.85	430 229

Mathematica enables you to create interactive trading charts and indicators such as moving averages, Bollinger bands:

In[ ]:=
```
InteractiveTradingChart[
 Table[{AAPL[[1, i]], AAPL[[2 ;;, i]] // Values}, {i, Length[AAPL[[1, ;;]]}]]
```

Out[ ]=



With associations is is easy to extract any of the data for a specified key:

In[ ]:=
```
AAPL["Close"]
```

Out[ ]=
```
{150.77, 151.76, 149.84, 142.48, 138.2, 142.45,
 146.1, 146.4, 145.43, 140.09, 140.42, 138.98, 138.34, 142.99,
 138.38, 142.41, 143.75, 143.86, 143.39, 147.27, 149.45, 151.64}
```

It is also very straightforward to create a time series object from the association:

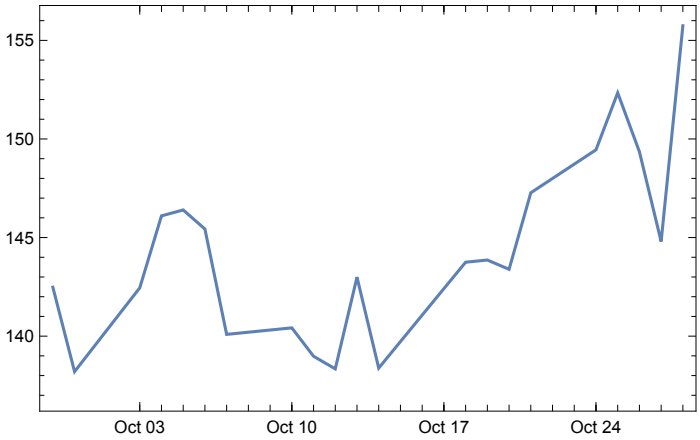
In[ ]:=
```
tsAAPL = TimeSeries[AAPL["Close"], {AAPL["Date"]}]
```

Out[ ]=

TimeSeries[Time: 12 Jan 2023 GMT−5 to 10 Feb 2023 GMT−5
Data points: 21]

In[ ]:=
```
DateListPlot[tsAAPL]
```

Out[ ]=



For multiple data series we might instead create a temporal data object with multiple paths, for example:

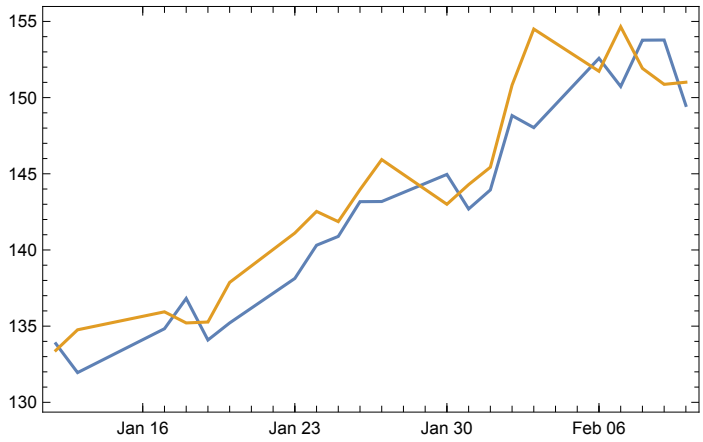
In[ ]:=
```
tdAAPL = TemporalData[{AAPL["Open"], AAPL["Close"]}, {AAPL["Date"]}]
```

Out[ ]=

TemporalData[Time: 12 Jan 2023 GMT−5 to 10 Feb 2023 GMT−5
Data points: 42 Paths: 2]

In[ ]:=
```
DateListPlot[tdAAPL]
```

Out[ ]=



Intraday data

Daily data is already available to Mathematica users using the FinancialData function. One of the advantages of using MATH-TWS is that data at higher frequencies is made available. In this example we illustrate the retrieval of 5-minute data for the S&P 500 Emini December futures contract:

In[ ]:=
```
ESM3 = twsHistoricalData[<|"Symbol" → "ESM3", "SecType" → "FUT",
    "Exchange" → "CME", "BarSize" → "1 min", "Duration" → "1 D"|>];
```

In[ ]:=
```
Transpose@Dataset@ESM3
```

Out[ ]=

Date	Open	High	Low	Close	Volume
Mon 10 Apr 2023 09:30:00	4105.75	4108	4104.5	4105.5	12 123
Mon 10 Apr 2023 09:31:00	4105.5	4106.25	4101.5	4101.5	8644
Mon 10 Apr 2023 09:32:00	4101.75	4102.25	4100	4100.5	7621
Mon 10 Apr 2023 09:33:00	4100.25	4103	4100.25	4102.5	5133
Mon 10 Apr 2023 09:34:00	4102.25	4103.25	4101	4102.5	4438
Mon 10 Apr 2023 09:35:00	4102.75	4102.75	4099	4100.25	6337
Mon 10 Apr 2023 09:36:00	4100	4100.75	4098.75	4100	4608
Mon 10 Apr 2023 09:37:00	4100	4101.75	4099.5	4100.75	4147
Mon 10 Apr 2023 09:38:00	4100.5	4101	4099	4100.25	5243
Mon 10 Apr 2023 09:39:00	4100.25	4103.25	4100	4103.25	5535
Mon 10 Apr 2023 09:40:00	4103.25	4104	4102	4103.25	4978
Mon 10 Apr 2023 09:41:00	4103.25	4105.25	4102.75	4105.25	4441
Mon 10 Apr 2023 09:42:00	4105	4107	4104.5	4106	5771
Mon 10 Apr 2023 09:43:00	4106	4107.25	4105.5	4106.25	3255
Mon 10 Apr 2023 09:44:00	4106.5	4106.75	4105.25	4106.75	3240
Mon 10 Apr 2023 09:45:00	4107	4107.75	4105.75	4107.25	3826
Mon 10 Apr 2023 09:46:00	4107.25	4107.5	4102.75	4104	8689
Mon 10 Apr 2023 09:47:00	4103.75	4105.5	4103.25	4105.25	3331
Mon 10 Apr 2023 09:48:00	4105.5	4106.75	4104.75	4105	4297
Mon 10 Apr 2023 09:49:00	4105	4105.5	4103	4103.25	3476

rows 1–20 of **450**

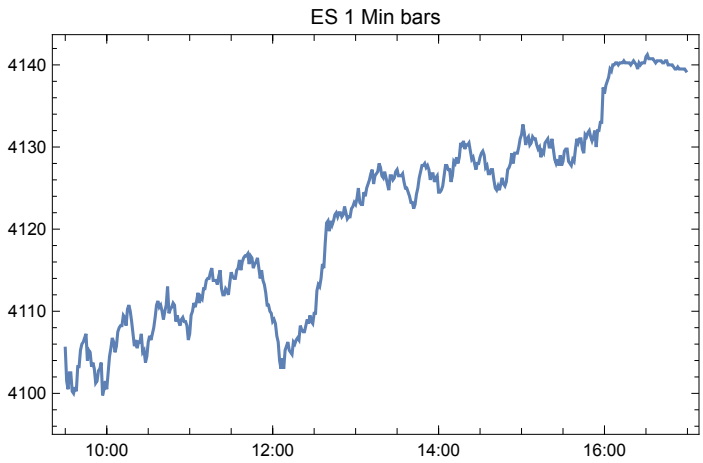
In[ ]:=
```
tsESM3 = TimeSeries[ESM3["Close"], {ESM3["Date"]}]
```

Out[ ]=

TimeSeries[ Time: 10 Apr 2023 09:30:00 GMT−4 to 10 Apr 2023 16:59:00 GMT−4
Data points: 450]

In[ ]:=
```
DateListPlot[tsESM3, PlotLabel → "ES 1 Min bars"]
```

Out[ ]=



Syntax errors and function mis-specifications are handled gracefully:

In[ ]:=
```
twsHistoricalData["Symbol" → "BADSYMBOL"]
```

```
Wolfram Library Exception
Unknown instrument STK SMART:BADSYMBOL
  USD. No security definition has been found for the request
```

Out[ ]=

```
$Failed
```

Realtime 5 second bars

IB TWS allows the user to retrieve real time data at 5-second intervals. The function twsRealTime-bars subscribes to the TWS data feed and continuously updates the specified number of 5-second data bars. The first call to the function initiates a subscription to the specified ticker symbol. Subsequent calls provide the realtime data for the symbol.

In[ ]:=
```
twsRealTimeBars["Symbol" → "ESH3",
 "SecType" → "FUT", "Exchange" → "CME", "Count" → 5]
```

Out[ ]=

```
⟨|Date → {Tue 14 Feb 2023 08:31:00 GMT−5, Tue 14 Feb 2023 08:31:05 GMT−5},
Open → {4138.75, 4141}, High → {4143.75, 4145},
Low → {4138.75, 4140.25}, Close → {4141, 4142.75}, Volume → {811, 727}|⟩
```

In[ ]:=
```
twsRealTimeBars["AAPL"]
```

Out[ ]=

```
⟨|Date → {Tue 14 Feb 2023 08:31:10 GMT−5, Tue 14 Feb 2023 08:31:15 GMT−5},
Open → {153.41, 153.32}, High → {153.41, 153.32},
Low → {153.37, 153.32}, Close → {153.41, 153.32}, Volume → {8, 1}|⟩
```

```
In[•]:= twsRealTimeBars["BADSYMBOL"]
```

```
Wolfram Library Exception
Unknown instrument STK SMART:BADSYMBOL
  USD. No security definition has been found for the request
```

```
Out[•]=
$Failed
```

Current market data

The function twsMktData creates a subscription to TWS realtime data and returns the latest update.
Note that the first call first call will delay for about 2 seconds, which is the time required for the data-feed subscription to be registered with TWS and for the data data to be received. Subsequent calls will take data from the data-feed, without any delay.

```
In[•]:= twsMktData["AAPL"]
```

```
Out[•]=
⟨|LastPrice → 153.35, LastSize → 1, Bid → 153.11,
 Ask → 153.3, BidSize → 7, AskSize → 10, High → 0, Low → 0|⟩
```

```
In[•]:= twsMktData["Symbol" → "ESH3", "SecType" → "FUT", "Exchange" → "CME"]
```

```
Out[•]=
⟨|LastPrice → 4161, LastSize → 25, Bid → 4161, Ask → 4161.25,
 BidSize → 28, AskSize → 34, High → 4168.5, Low → 4155.75|⟩
```

```
In[•]:= twsMktData["BADSYMBOL"]
```

```
Wolfram Library Exception
Unknown instrument STK SMART:BADSYMBOL
  USD. No security definition has been found for the request
```

```
Out[•]=
$Failed
```

```
In[•]:= twsMktData[<|"Symbol" → "EUR.USD", "Exchange" → "IDEALPRO", "SecType" → "CASH"|>]
```

```
Out[•]=
⟨|LastPrice → 0, LastSize → 0, Bid → -1,
 Ask → -1, BidSize → 0, AskSize → 0, High → 0, Low → 0|⟩
```

Snapshot

For one-off retrieval of current market data the twsSnapshot function provides the information as twsMktData, but without the data-feed subscription:

```
In[•]:= twsSnapshot["SPY"]
```

```
Out[•]=
⟨|Symbol → SPY, lastTimestamp → 1 676 381 512, lastPrice → 413.6,
 lastSize → 1, Halted → 0, Volume → 12 664, Close → 412.83,
 bidPrice → 413.55, bidSize → 1, askPrice → 413.6, askSize → 5|⟩
```

```
In[•]:= twsSnapshot["Symbol" → "ESH3", "SecType" → "FUT", "Exchange" → "CME"]
```

```
Out[•]=
⟨| Symbol → ESH3, bidPrice → 4155, bidSize → 2, askPrice → 4155.25, askSize → 13,
 lastPrice → 4155, lastSize → 1, Volume → 183 764, High → 4167, Low → 4132,
 Close → 4147.25, Open → 4149.75, lastTimestamp → 1 676 381 526, Halted → 0 |⟩
```

```
In[•]:= twsSnapshot["BADSYMBOL"]
```

```
Wolfram Library Exception
Unknown instrument STK SMART:BADSYMBOL
  USD. No security definition has been found for the request
```

```
Out[•]=
$Failed
```

```
In[•]:= twsSnapshot[<|"Symbol" → "EUR.USD",
   "Exchange" → "IDEALPRO", "SecType" → "CASH"|>]
```

```
Out[•]=
⟨| Symbol → EUR.USD, bidPrice → -1, bidSize → 0,
 askPrice → -1, askSize → 0, Close → 1.0612, Halted → 0 |⟩
```

Market Depth

```
In[•]:= dsMktDepth = Transpose@Dataset@twsMarketDepth["SecType" → "FUT",
          "Symbol" → "ESM3", "Exchange" → "CME", "RowsCount" → 5]
```

Out[•]=

Side	Price	Size
bid	4136.75	145
bid	4137	188
bid	4137.25	156
bid	4137.5	144
bid	4137.75	155
bid	4138	131
bid	4138.25	144
bid	4138.5	119
bid	4138.75	70
bid	4139	39
ask	4139.25	92
ask	4139.5	120
ask	4139.75	124
ask	4140	132
ask	4140.25	134
ask	4140.5	142
ask	4140.75	149
ask	4141	202
ask	4141.25	139
ask	4141.5	164

News

Beginning in TWS v966, three API news services are enabled in accounts by default and available from the API.
News headlines can be retrieved from TWS using the twsNews function. The use selects the symbol and the number of headlines:

```
In[•]:= news = twsNews["Symbol" → "AAPL", "Count" → 5] (* twsNews["AAPL", 5] *)
```

```
Out[•]=
⟨| Date →
   {Fri 3 Feb 2023 14:54:50 GMT−5, Fri 27 Jan 2023 14:13:21 GMT−5, Mon 23 Jan 2023 18:18:32 GMT−5,
    Wed 18 Jan 2023 13:57:59 GMT−5, Tue 3 Jan 2023 12:15:50 GMT−5},
  ArticleId → {BRFG$13a0c706, BRFUPDN$1391468b, BRFUPDN$13890e37,
    BRFUPDN$137f91d7, BRFUPDN$1361b758}, Headline →
  {{A:800015:L:en:K:n/a:C:0.6937490105628967}Apple recoups earlier losses
     as investors keep the spotlight on the many bright spots in DecQ,
    {A:800015:L:en:K:n/a:C:0.9852834939956665}!BofA Securities
     reiterated Apple (AAPL) coverage with Neutral and target $153,
    {A:800015:L:en:K:0.39:C:0.3865243196487427}!Deutsche Bank
     reiterated Apple (AAPL) coverage with Buy and target $160,
    {A:800015:L:en:K:0.68:C:0.6816690564155579}!Canaccord Genuity
     reiterated Apple (AAPL) coverage with Buy and target $170,
    {A:800015:L:en:K:-0.97:C:0.97}!Exane BNP Paribas downgraded
     Apple (AAPL) to Neutral with target $140}|⟩
```

```
In[•]:= LastestNews = news["ArticleId"][[2]]
```

```
Out[•]=
BRFUPDN$1391468b
```

```
In[•]:= twsArticle[LastestNews]
```

```
Out[•]=
LibraryFunctionError[LIBRARY_FUNCTION_ERROR, 6]
```

In[ ]:=
```
twsLog[]
```

Out[ ]=
```
setup unhandled exception translator
Connected to 127.0.0.1:7496 clientId:5
got account list: DF220168,DU220169,DU220170,DU220171,DU220172,DU220173,
TWS data initialized
cancel account All update subscription
reqAccountUpdates DF220168
reqAccountUpdates DU220169
reqAccountUpdates DU220170
reqAccountUpdates DU220171
reqAccountUpdates DU220172
reqAccountUpdates DU220173
IB message [2100] API client has been unsubscribed from account data.
IB message [200] No security definition has been found for the request
news providers: BRFG+BRFUPDN+DJNL id=10014
got news 10014 BRFG$13a0c706
got news 10014 BRFUPDN$1391468b
got news 10014 BRFUPDN$13890e37
got news 10014 BRFUPDN$137f91d7
got news 10014 BRFUPDN$1361b758
historicalNewsEnd 10014 hasMore=1
IB message [10172] Failed to request news article:Not allowed
IB message [10172] Failed to request news article:Not allowed
```

Order management

In[ ]:=
```
myAccount = accounts[[4]]
```

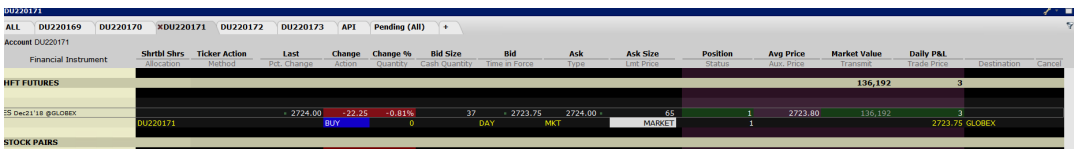
Out[ ]=
```
DU220171
```

Place order.

Orders can be defined in Mathematica and transmitted to TWS using the twsPlaceOrder function. Note that there is no short form for this function.

In[ ]:=
```
order1 = twsPlaceOrder[<|"Symbol" → "ESZ2",
    "Exchange" → "GLOBEX", "SecType" → "FUT", "Action" → "BUY",
    "Quantity" → 1, "OrderType" → "MKT", "Account" → myAccount|>]
```

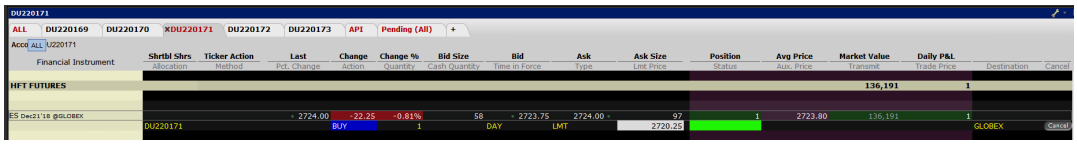
Out[ ]=
```
18 298
```



In[ ]:=
```
order2 = twsPlaceOrder[<|"Symbol" → "ESZ2", "Exchange" → "GLOBEX",
    "SecType" → "FUT", "Action" → "BUY", "Quantity" → 1,
    "OrderType" → "LMT", "LmtPrice" → 3700.25, "Account" → myAccount|>]
```

Out[ ]=
```
18 281
```

```
;;
```



In[ ]:=
```
orderBAD = twsPlaceOrder["Action" → "BUY",
  "Quantity" → 100, "Symbol" → "INTC", "OrderType" → "BADTYPE"]
```

Out[ ]=
```
LibraryFunctionError[LIBRARY_FUNCTION_ERROR, 6]
```

```
twsError[]
```

Out[ ]=
```
Wolfram Library Exception
Place order error. Error validating
  request:-'bE' : cause - Invalid order type was entered
```

In[ ]:=
```
order3 = twsPlaceOrder["Action" → "BUY",
  "Quantity" → 100, "Symbol" → "AAPL", "OrderType" → "LMT",
  "LmtPrice" → 140.25, "TIF" → "GTC", "Account" → myAccount]
```

Out[ ]=
```
18 299
```

Modify order

To modify order we should pass order id to twsPlaceOrder with new parameters

In[ ]:=
```
twsPlaceOrder[<|"Action" → "BUY",
  "Quantity" → 200, "Symbol" → "AAPL", "OrderType" → "LMT",
  "LmtPrice" → 138.60, "TIF" → "DAY", "OrderId" → order3|>]
```

Out[ ]=
```
18 282
```

```
twsPlaceOrder[<|"OrderId" → order2, "Symbol" → "ESz2",
  "Exchange" → "GLOBEX", "SecType" → "FUT", "Action" → "BUY",  "Quantity" → 1,
  "OrderType" → "LMT", "LmtPrice" → 2705.25, "Account" → myAccount|>]
```

Out[ ]=

2

Cancel order

In[ ]:=
```
twsCancelOrder[order2]
```

In[ ]:=
```
twsCancelOrder[order3]
```

Bracket order

In[ ]:=
```
parentOrder = twsPlaceOrder[<|"Action" → "BUY", "Quantity" → 100,
    "Symbol" → "AAPL", "OrderType" → "LMT", "LmtPrice" → 140.5,
    "TIF" → "GTC", "Transmit" → False, "Account" → myAccount|>];
takeOrder = twsPlaceOrder[<|"Action" → "SELL", "Quantity" → 100,
    "Symbol" → "AAPL", "OrderType" → "LMT", "LmtPrice" → 147.90, "TIF" → "GTC",
    "Transmit" → False, "ParentId" → parentOrder, "Account" → myAccount|>];
stopOrder = twsPlaceOrder[<|"Action" → "SELL", "Quantity" → 100,
    "Symbol" → "AAPL", "OrderType" → "STP", "AuxPrice" → 138.50, "TIF" → "GTC",
    "Transmit" → True, "ParentId" → parentOrder, "Account" → myAccount|>];
```

The rule is to add ParentId parameter to all child orders. Transmit flag should be False for all orders except the last one.

Algo Orders

Adaptive

In[ ]:=
```
order4 = twsPlaceOrder[<|"Symbol" → "AAPL", "Action" → "BUY",
    "Quantity" → 100, "OrderType" → "MKT", "Account" → myAccount,
    "Algo" → "Adaptive", "AlgoParams" → "adaptivePriority=Normal"|>]
```

Out[ ]=

19 538

In[ ]:=
```
twsCancelOrder[order4]
```

ArrivalPrice

```
In[ ]:= order5 = <|"Action" → "BUY", "TotalQuantity" → 100,
          "OrderType" → "LMT", "LmtPrice" → 140.5, "Tif" → "DAY", "Transmit" → True,
          "Account" → myAccount, "AlgoStrategy" → "ArrivalPx",
          "AlgoParams" → <|"maxPctVol" → 0.2, "riskAversion" → "Neutral"|>|>
Out[ ]=
⟨|Action → BUY, TotalQuantity → 100, OrderType → LMT, LmtPrice → 140.5,
 Tif → DAY, Transmit → True, Account → DU220173, AlgoStrategy → ArrivalPx,
 AlgoParams → ⟨|maxPctVol → 0.2, riskAversion → Neutral|⟩|⟩

In[ ]:= order5 = twsPlaceOrderEx[<|"Symbol" → "AAPL", "Order" → order5|>]
Out[ ]=
18 296

In[ ]:= order5 = twsPlaceOrder[<|"Symbol" → "AAPL", "Action" → "BUY",
            "Quantity" → 100, "OrderType" → "MKT", "Account" → myAccount,
            "Algo" → "ArrivalPx", "AlgoParams" → <|"MaxPctVol"
               → 0.2, "riskAversion" → "Neutral"|>|>]
   LibraryFunction : Argument ⟨|MaxPctVol → 0.2, riskAversion → Neutral |⟩ at position 8 should be a
     UTF8String.
Out[ ]=
LibraryFunction[ Function name: wll _twsPlaceOrder
                 Argument count: 15 ][
 DU220173, BUY, 100, AAPL, STK, SMART, ArrivalPx,
 ⟨|MaxPctVol → 0.2, riskAversion → Neutral|⟩, MKT, 0, 0, DAY, True, 0, 0]

In[ ]:= order = <|"Action" → "BUY", "TotalQuantity" → 1000,
          "OrderType" → "LMT", "LmtPrice" → 140.75,
          "FaGroup" → "Test Group 1", "FaMethod" → "AvailableEquity"|>
Out[ ]=
⟨|Action → BUY, TotalQuantity → 1000, OrderType → LMT,
 LmtPrice → 140.75, FaGroup → Test Group 1, FaMethod → AvailableEquity|⟩

In[ ]:= twsPlaceOrderEx[<|"Symbol" → "AAPL", "Order" → order|>]
Out[ ]=
18 297
```

VWAP

```
In[ ]:= order6 = <|"Action" → "BUY", "TotalQuantity" → 100,
          "OrderType" → "LMT", "LmtPrice" → 185.00, "Tif" → "DAY", "Transmit" → True,
          "Account" → myAccount, "AlgoStrategy" → "Vwap", "AlgoParams" → <|
            "maxPctVol" → 0.15, "startTime" → "09:00:00", "endTime" → "15:55:00",
            "allowPastEndTime" → False, "noTakeLiq" → True, "speedUp" → True|>|>
Out[ ]=
⟨|Action → BUY, TotalQuantity → 100, OrderType → LMT, LmtPrice → 185.,
 Tif → DAY, Transmit → True, Account → DU220173, AlgoStrategy → Vwap,
 AlgoParams → ⟨|maxPctVol → 0.15, startTime → 09:00:00, endTime → 15:55:00,
    allowPastEndTime → False, noTakeLiq → True, speedUp → True|⟩|⟩

In[ ]:= order6 = twsPlaceOrderEx[<|"Symbol" → "AAPL", "Order" → order6|>]
Out[ ]=
19 846
```

Accumulate/Distribute

```
order7 = <|"Action" → "BUY", "TotalQuantity" → 1000,
  "OrderType" → "LMT", "LmtPrice" → 185.00, "Tif" → "DAY", "Transmit" → True,
  "Account" → myAccount, "AlgoStrategy" → "AD", "AlgoParams" → <|
    "componentSize" → 100, "timeBetweenOrders" → 90, "randomizeTime20" → True,
    "randomizeSize55" → True, "catchUp" → True, "waitForFill" → True,
    "activeTimeStart" → "09:00:00", "activeTimeEnd" → "15:55:00"|>|>
Out[ ]=
⟨|Action → BUY, TotalQuantity → 1000, OrderType → LMT, LmtPrice → 185.,
 Tif → DAY, Transmit → True, Account → DU220173, AlgoStrategy → AD,
 AlgoParams → ⟨|componentSize → 100, timeBetweenOrders → 90,
    randomizeTime20 → True, randomizeSize55 → True, catchUp → True,
    waitForFill → True, activeTimeStart → 09:00:00, activeTimeEnd → 15:55:00|⟩|⟩

In[ ]:= order7 = twsPlaceOrderEx[<|"Symbol" → "AAPL", "Order" → order7|>]
Out[ ]=
19 545

In[ ]:= twsLog[]
Out[ ]=
order params: |<|"Action" -> "BUY", "TotalQuantity" -> 1000, "OrderType" ->
  "LMT", "LmtPrice" -> 185., "Tif" -> "DAY", "Transmit" -> True, "Account" ->
  "DU220173", "AlgoStrategy" -> "AD", "AlgoParams" -> <|"componentSize" ->
  100, "timeBetweenOrders" -> 90, "randomizeTime20" -> True, "randomizeSize55"
  -> True, "catchUp" -> True, "waitForFill" -> True, "activeTimeStart"
  -> "09:00:00 US/Eastern", "activeTimeEnd" -> "15:55:00 US/Eastern"|>|>|
IB message [10315] activeTimeEnd: The time entered is
  invalid. The correct format is hh:mm:ss. E.g.: 15:00:00 in
  UTC. No date should be specified, current date is assumed.
```

Open orders

```
In[•]:= openorders = twsOpenOrders[<|"Account" → myAccount|>]
```

```
Out[•]=
⟨|OrderId → {18 299}, Symbol → {AAPL}, Action → {BUY}, OrderType → {LMT},
 LmtPrice → {140.25}, AuxPrice → {0}, Quantity → {100}, Status → {PreSubmitted}|⟩
```

```
In[•]:= Transpose@Dataset[openorders]
```

```
Out[•]=
```

OrderId	Symbol	Action	OrderType	LmtPrice	AuxPrice	Quantity	Status
18 299	AAPL	BUY	LMT	140.25	0	100	PreSubmitted

Executions

A list of executed orders is obtained using the twsExecutions function:

```
In[•]:= myexecutions = twsExecutions["Account" → myAccount]
        (* Abbreviated form: twsExecutions[] *)
```

```
Out[•]=
⟨|ExecId → {0000e1a7.6357b1a7.01.01}, Symbol → {ESZ2},
 Date → {2022-10-25 11:49:29}, Side → {BOT}, Qty → {1},
 Price → {3859}, Account → {DU220173}, OrderId → {18 298}|⟩
```

```
In[•]:= Transpose@Dataset[myexecutions]
```

```
Out[•]=
```

ExecId	Symbol	Date
0000e1a7.6357b1a7.01.01	ESZ2	2022–10–25 11:49:29

Exit All Positions MOO or MOC

```
ExitAllPositions[account_, ordertype_ : "MOO"] := Module[{positions},
  positions = twsPositions[account];
  If[ordertype == "MOC",
   Do[twsPlaceOrder["Action" → "SELL",
       "Quantity" → positions[[ ;; , i]]["Position"],
       "Symbol" → positions[[ ;; , i]]["Symbol"], "OrderType" → ordertype,
       "TIF" → "DAY", "Account" → account], {i, Length@positions["Symbol"]}];];
  If[ordertype == "MOO",
   Do[twsPlaceOrder["Action" → "SELL",
       "Quantity" → positions[[ ;; , i]]["Position"],
       "Symbol" → positions[[ ;; , i]]["Symbol"], "OrderType" → "MKT",
       "TIF" → "OPG", "Account" → account], {i, Length@positions["Symbol"]}];];]
```

Options

Stocks

In[ ]:=
```
aaplOpts = twsOptionParams["AAPL"]
```

Out[ ]=

⟨| Expiration → { Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 ,
Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 ,
Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 ,
Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 , Fri 17 Feb 2023 ,
Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 ,
Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 ,
Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 ,
Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 24 Feb 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 ,
Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 ,
Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 ,
Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 , Fri 3 Mar 2023 ,
Fri 3 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 ,
Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 ,
Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 ,
Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 10 Mar 2023 , Fri 17 Mar 2023 ,
Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 ,
Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 ,
Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 17 Mar 2023 ,
Fri 17 Mar 2023 , Fri 17 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 ,
Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 ,
Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 ,
Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 , Fri 24 Mar 2023 ,
Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 ,
Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 ,
Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 ,
Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 31 Mar 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 ,
Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 ,

Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 ,
Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 , Fri 21 Apr 2023 ,
Fri 21 Apr 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 ,
Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 ,
Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 ,
Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 , Fri 19 May 2023 ,
Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 ,
Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 ,
Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 ,
Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 16 Jun 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 ,
Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 ,
Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 ,
Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 , Fri 21 Jul 2023 ,
Fri 21 Jul 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 ,
Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 ,
Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 ,
Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 18 Aug 2023 , Fri 15 Sep 2023 ,
Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 ,
Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 ,
Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 15 Sep 2023 ,
Fri 15 Sep 2023 , Fri 15 Sep 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 ,
Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 ,
Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 ,
Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 , Fri 20 Oct 2023 ,
Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 ,
Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 ,
Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 ,
Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 15 Dec 2023 , Fri 19 Jan 2024 , Fri 19 Jan 2024 ,
Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 ,
Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 ,
Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 , Fri 19 Jan 2024 ,

Fri 19 Jan 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024,
Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024,
Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024,
Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 15 Mar 2024, Fri 21 Jun 2024,
Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024,
Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024,
Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 21 Jun 2024,
Fri 21 Jun 2024, Fri 21 Jun 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024,
Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024,
Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024,
Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024, Fri 20 Sep 2024,
Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025,
Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025,
Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025,
Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 17 Jan 2025, Fri 20 Jun 2025, Fri 20 Jun 2025,
Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025,
Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025,
Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025, Fri 20 Jun 2025,
Fri 20 Jun 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025,
Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025,
Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025,
Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025, Fri 19 Dec 2025},

```
Strike → {35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110,
  113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128,
  129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144,
  145, 146, 147, 148, 149, 150, 152.5, 155, 157.5, 160, 162.5, 165, 167.5, 170,
  172.5, 175, 177.5, 180, 182.5, 185, 190, 195, 200, 205, 210, 215, 220, 225, 230,
  235, 240, 245, 250, 255, 260, 265, 270, 275, 280, 285, 290, 295, 300, 310, 320},
TradingClass → {AAPL, AAPL, AAPL, AAPL, AAPL, AAPL, AAPL, AAPL,
  AAPL, AAPL, AAPL, AAPL, AAPL, AAPL, AAPL, AAPL, AAPL, AAPL},
Multiplier → {100, 100, 100, 100, 100, 100, 100, 100, 100,
  100, 100, 100, 100, 100, 100, 100, 100, 100}|⟩
```

In[•]:=
```
AAPLP140 = twsOptionContract["AAPL", aaplOpts["Expiration"][[3]], 140 , "P"]
```

Out[ ]=

LibraryFunction[Function name: wll _twsOptionContract
Argument count: 8][

AAPL, OPT, SMART, 100, , Fri 3 Mar 2023 , 140, P]

Futures

In[ ]:=
```
ESoptions =
 twsOptionParams[<|"Symbol" → "ESH3", "SecType" → "FUT", "Exchange" → "CME"|>]
```

Out[ ]=
```
⟨| Expiration → { Thu 16 Feb 2023 , Fri 17 Feb 2023 , Tue 21 Feb 2023 ,
    Wed 22 Feb 2023 , Thu 23 Feb 2023 , Fri 24 Feb 2023 , Mon 27 Feb 2023 ,
    Tue 28 Feb 2023 , Wed 1 Mar 2023 , Thu 2 Mar 2023 , Fri 3 Mar 2023 , Mon 6 Mar 2023 ,
    Tue 7 Mar 2023 , Wed 8 Mar 2023 , Thu 9 Mar 2023 , Fri 10 Mar 2023 , Mon 13 Mar 2023 ,
    Tue 14 Mar 2023 , Wed 15 Mar 2023 , Thu 16 Mar 2023 , Fri 17 Mar 2023 } ,
  Strike → {100, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700,
    1800, 1900, 2000, 2100, 2150, 2200, 2250, 2275, 2300, 2325, 2350, 2375, 2400,
    2425, 2450, 2475, 2500, 2550, 2600, 2650, 2700, 2750, 2800, 2850, 2900, 2910,
    2920, 2930, 2940, 2950, 2960, 2970, 2980, 2990, 3000, 3010, 3020, 3025, 3030,
    3040, 3050, 3060, 3070, 3075, 3080, 3090, 3100, 3110, 3120, 3125, 3130, 3140,
    3150, 3160, 3170, 3175, 3180, 3190, 3200, 3210, 3220, 3225, 3230, 3240, 3250,
    3260, 3270, 3275, 3280, 3290, 3300, 3310, 3320, 3325, 3330, 3340, 3350, 3360,
    3370, 3375, 3380, 3390, 3400, 3410, 3420, 3425, 3430, 3440, 3450, 3460, 3470,
    3475, 3480, 3490, 3500, 3510, 3520, 3525, 3530, 3540, 3550, 3560, 3570, 3575,
    3580, 3590, 3600, 3610, 3620, 3625, 3630, 3640, 3650, 3660, 3670, 3675, 3680,
    3685, 3690, 3695, 3700, 3705, 3710, 3715, 3720, 3725, 3730, 3735, 3740, 3745,
    3750, 3755, 3760, 3765, 3770, 3775, 3780, 3785, 3790, 3795, 3800, 3805, 3810,
    3815, 3820, 3825, 3830, 3835, 3840, 3845, 3850, 3855, 3860, 3865, 3870, 3875,
    3880, 3885, 3890, 3895, 3900, 3905, 3910, 3915, 3920, 3925, 3930, 3935, 3940,
    3945, 3950, 3955, 3960, 3965, 3970, 3975, 3980, 3985, 3990, 3995, 4000, 4005,
    4010, 4015, 4020, 4025, 4030, 4035, 4040, 4045, 4050, 4055, 4060, 4065, 4070,
    4075, 4080, 4085, 4090, 4095, 4100, 4105, 4110, 4115, 4120, 4125, 4130,
    4135, 4140, 4145, 4150, 4155, 4160, 4165, 4170, 4175, 4180, 4185, 4190,
    4195, 4200, 4205, 4210, 4215, 4220, 4225, 4230, 4235, 4240, 4245, 4250,
    4255, 4260, 4265, 4270, 4275, 4280, 4285, 4290, 4295, 4300, 4305, 4310,
    4315, 4320, 4325, 4330, 4335, 4340, 4345, 4350, 4355, 4360, 4365, 4370,
    4375, 4380, 4390, 4400, 4410, 4420, 4425, 4430, 4440, 4450, 4460, 4470,
    4475, 4480, 4490, 4500, 4510, 4520, 4525, 4530, 4540, 4550, 4560, 4570,
    4575, 4580, 4590, 4600, 4610, 4620, 4630, 4640, 4650, 4660, 4670, 4680,
    4690, 4700, 4710, 4720, 4730, 4740, 4750, 4760, 4770, 4780, 4800, 4850,
    4900, 4950, 5000, 5050, 5100, 5150, 5200, 5250, 5300, 5350, 5400, 5450,
    5500, 5550, 5600, 5700, 5800, 5900, 6000, 6100, 6200, 7100, 8100, 9100},
  TradingClass → {E1D, E1C, E2D, E3D, E1B, E2C, ES, E4D, E1A, E2B,
    E3C, E3B, E4C, E2A, EW, E4A, EW3, EW2, EW1, EW4}, Multiplier →
  {50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50} |⟩
```

In[ ]:=
```
ESoptions["Expiration"] = Sort@ESoptions["Expiration"];
ESoptions
```

Out[ ]=

⟨| Expiration → { Thu 16 Feb 2023 , Fri 17 Feb 2023 , Tue 21 Feb 2023 ,
Wed 22 Feb 2023 , Thu 23 Feb 2023 , Fri 24 Feb 2023 , Mon 27 Feb 2023 ,
Tue 28 Feb 2023 , Wed 1 Mar 2023 , Thu 2 Mar 2023 , Fri 3 Mar 2023 , Mon 6 Mar 2023 ,
Tue 7 Mar 2023 , Wed 8 Mar 2023 , Thu 9 Mar 2023 , Fri 10 Mar 2023 , Mon 13 Mar 2023 ,
Tue 14 Mar 2023 , Wed 15 Mar 2023 , Thu 16 Mar 2023 , Fri 17 Mar 2023 } ,

Strike → {100, 800, 900, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700,
1800, 1900, 2000, 2100, 2150, 2200, 2250, 2275, 2300, 2325, 2350, 2375, 2400,
2425, 2450, 2475, 2500, 2550, 2600, 2650, 2700, 2750, 2800, 2850, 2900, 2910,
2920, 2930, 2940, 2950, 2960, 2970, 2980, 2990, 3000, 3010, 3020, 3025, 3030,
3040, 3050, 3060, 3070, 3075, 3080, 3090, 3100, 3110, 3120, 3125, 3130, 3140,
3150, 3160, 3170, 3175, 3180, 3190, 3200, 3210, 3220, 3225, 3230, 3240, 3250,
3260, 3270, 3275, 3280, 3290, 3300, 3310, 3320, 3325, 3330, 3340, 3350, 3360,
3370, 3375, 3380, 3390, 3400, 3410, 3420, 3425, 3430, 3440, 3450, 3460, 3470,
3475, 3480, 3490, 3500, 3510, 3520, 3525, 3530, 3540, 3550, 3560, 3570, 3575,
3580, 3590, 3600, 3610, 3620, 3625, 3630, 3640, 3650, 3660, 3670, 3675, 3680,
3685, 3690, 3695, 3700, 3705, 3710, 3715, 3720, 3725, 3730, 3735, 3740, 3745,
3750, 3755, 3760, 3765, 3770, 3775, 3780, 3785, 3790, 3795, 3800, 3805, 3810,
3815, 3820, 3825, 3830, 3835, 3840, 3845, 3850, 3855, 3860, 3865, 3870, 3875,
3880, 3885, 3890, 3895, 3900, 3905, 3910, 3915, 3920, 3925, 3930, 3935, 3940,
3945, 3950, 3955, 3960, 3965, 3970, 3975, 3980, 3985, 3990, 3995, 4000, 4005,
4010, 4015, 4020, 4025, 4030, 4035, 4040, 4045, 4050, 4055, 4060, 4065, 4070,
4075, 4080, 4085, 4090, 4095, 4100, 4105, 4110, 4115, 4120, 4125, 4130,
4135, 4140, 4145, 4150, 4155, 4160, 4165, 4170, 4175, 4180, 4185, 4190,
4195, 4200, 4205, 4210, 4215, 4220, 4225, 4230, 4235, 4240, 4245, 4250,
4255, 4260, 4265, 4270, 4275, 4280, 4285, 4290, 4295, 4300, 4305, 4310,
4315, 4320, 4325, 4330, 4335, 4340, 4345, 4350, 4355, 4360, 4365, 4370,
4375, 4380, 4390, 4400, 4410, 4420, 4425, 4430, 4440, 4450, 4460, 4470,
4475, 4480, 4490, 4500, 4510, 4520, 4525, 4530, 4540, 4550, 4560, 4570,
4575, 4580, 4590, 4600, 4610, 4620, 4630, 4640, 4650, 4660, 4670, 4680,
4690, 4700, 4710, 4720, 4730, 4740, 4750, 4760, 4770, 4780, 4800, 4850,
4900, 4950, 5000, 5050, 5100, 5150, 5200, 5250, 5300, 5350, 5400, 5450,
5500, 5550, 5600, 5700, 5800, 5900, 6000, 6100, 6200, 7100, 8100, 9100},

TradingClass → {E1D, E1C, E2D, E3D, E1B, E2C, ES, E4D, E1A, E2B,
E3C, E3B, E4C, E2A, EW, E4A, EW3, EW2, EW1, EW4}, Multiplier →
{50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50, 50} |⟩

```
In[ ]:= ESCall3885 = twsOptionContract[<|"Symbol" → "ES",
          "SecType" → "FOP", "Exchange" → "CME", "Strike" → 4135,
          "Multiplier" → "50", "Right" → "C", "Expiration" → Thu 16 Feb 2023 |>]
Out[ ]=
E3DG3 C4135

In[ ]:= DateString[Thu 16 Feb 2023, "ISODate"]
Out[ ]=
2023-02-16

In[ ]:= twsLog[]
Out[ ]=
OptionContract request symbol=ES sectype:FOP
  exchange:CME mult:50 class: exp:2023-3-9 strike:4125 right:C
IB message [200] No security definition has been found for the request
OptionContract request symbol=ES sectype:FOP
  exchange:CME mult:50 class: exp:2023-2-14 strike:4125 right:C
OptionContract request symbol=ES sectype:FOP
  exchange:CME mult:50 class: exp:2023-2-15 strike:4125 right:C
OptionContract request symbol=ES sectype:FOP
  exchange:CME mult:50 class: exp:2023-2-17 strike:4135 right:C
```

Index Options - Regular Monthly

```
In[ ]:= SPXoptions =
         twsOptionParams[<|"Symbol" → "SPX", "SecType" → "IND", "Exchange" → "CBOE"|>]
Out[ ]=
⟨| Expiration →
  {Thu 16 Feb 2023, Thu 16 Feb 2023, Thu 16 Feb 2023, Thu 16 Feb 2023, Thu 16 Feb 2023,
   Thu 16 Feb 2023, Fri 17 Feb 2023, Fri 17 Feb 2023, Fri 17 Feb 2023, Tue 21 Feb 2023,
   Tue 21 Feb 2023, Tue 21 Feb 2023, Wed 22 Feb 2023, Wed 22 Feb 2023, Wed 22 Feb 2023,
   Thu 23 Feb 2023, Thu 23 Feb 2023, Thu 23 Feb 2023, Fri 24 Feb 2023, Fri 24 Feb 2023,
   Fri 24 Feb 2023, Mon 27 Feb 2023, Mon 27 Feb 2023, Mon 27 Feb 2023, Tue 28 Feb 2023,
   Tue 28 Feb 2023, Tue 28 Feb 2023, Wed 1 Mar 2023, Wed 1 Mar 2023, Wed 1 Mar 2023,
   Thu 2 Mar 2023, Thu 2 Mar 2023, Thu 2 Mar 2023, Fri 3 Mar 2023, Fri 3 Mar 2023,
   Fri 3 Mar 2023, Mon 6 Mar 2023, Mon 6 Mar 2023, Mon 6 Mar 2023, Tue 7 Mar 2023,
   Tue 7 Mar 2023, Tue 7 Mar 2023, Wed 8 Mar 2023, Wed 8 Mar 2023, Wed 8 Mar 2023,
   Thu 9 Mar 2023, Thu 9 Mar 2023, Thu 9 Mar 2023, Fri 10 Mar 2023, Fri 10 Mar 2023,
   Fri 10 Mar 2023, Mon 13 Mar 2023, Mon 13 Mar 2023, Mon 13 Mar 2023, Tue 14 Mar 2023,
```

Tue 14 Mar 2023, Tue 14 Mar 2023, Wed 15 Mar 2023, Wed 15 Mar 2023, Wed 15 Mar 2023,
Thu 16 Mar 2023, Thu 16 Mar 2023, Thu 16 Mar 2023, Thu 16 Mar 2023, Thu 16 Mar 2023,
Thu 16 Mar 2023, Fri 17 Mar 2023, Fri 17 Mar 2023, Fri 17 Mar 2023, Mon 20 Mar 2023,
Mon 20 Mar 2023, Mon 20 Mar 2023, Tue 21 Mar 2023, Tue 21 Mar 2023, Tue 21 Mar 2023,
Fri 24 Mar 2023, Fri 24 Mar 2023, Fri 24 Mar 2023, Fri 31 Mar 2023, Fri 31 Mar 2023,
Fri 31 Mar 2023, Fri 14 Apr 2023, Fri 14 Apr 2023, Fri 14 Apr 2023, Thu 20 Apr 2023,
Thu 20 Apr 2023, Thu 20 Apr 2023, Fri 21 Apr 2023, Fri 21 Apr 2023, Fri 21 Apr 2023,
Fri 28 Apr 2023, Fri 28 Apr 2023, Fri 28 Apr 2023, Thu 18 May 2023, Thu 18 May 2023,
Thu 18 May 2023, Fri 19 May 2023, Fri 19 May 2023, Fri 19 May 2023, Wed 31 May 2023,
Wed 31 May 2023, Wed 31 May 2023, Thu 15 Jun 2023, Thu 15 Jun 2023, Thu 15 Jun 2023,
Fri 16 Jun 2023, Fri 16 Jun 2023, Fri 16 Jun 2023, Fri 30 Jun 2023, Fri 30 Jun 2023,
Fri 30 Jun 2023, Thu 20 Jul 2023, Thu 20 Jul 2023, Thu 20 Jul 2023, Fri 21 Jul 2023,
Fri 21 Jul 2023, Fri 21 Jul 2023, Mon 31 Jul 2023, Mon 31 Jul 2023, Mon 31 Jul 2023,
Thu 17 Aug 2023, Thu 17 Aug 2023, Thu 17 Aug 2023, Thu 14 Sep 2023, Thu 14 Sep 2023,
Thu 14 Sep 2023, Fri 29 Sep 2023, Fri 29 Sep 2023, Fri 29 Sep 2023, Thu 19 Oct 2023,
Thu 19 Oct 2023, Thu 19 Oct 2023, Thu 16 Nov 2023, Thu 16 Nov 2023, Thu 16 Nov 2023,
Thu 14 Dec 2023, Thu 14 Dec 2023, Thu 14 Dec 2023, Fri 29 Dec 2023, Fri 29 Dec 2023,
Fri 29 Dec 2023, Thu 18 Jan 2024, Thu 18 Jan 2024, Thu 18 Jan 2024, Thu 15 Feb 2024,
Thu 15 Feb 2024, Thu 15 Feb 2024, Thu 14 Mar 2024, Thu 14 Mar 2024, Thu 14 Mar 2024,
Thu 20 Jun 2024, Thu 20 Jun 2024, Thu 20 Jun 2024, Thu 19 Dec 2024, Thu 19 Dec 2024,
Thu 19 Dec 2024, Thu 18 Dec 2025, Thu 18 Dec 2025, Thu 18 Dec 2025, Thu 17 Dec 2026,
Thu 17 Dec 2026, Thu 17 Dec 2026, Thu 16 Dec 2027, Thu 16 Dec 2027, Thu 16 Dec 2027},

```
Strike → {100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200, 1300,
  1400, 1500, 1600, 1700, 1750, 1800, 1850, 1900, 1950, 1975, 2000, 2050, 2100,
  2150, 2200, 2250, 2300, 2350, 2400, 2450, 2500, 2525, 2550, 2575, 2600, 2625,
  2650, 2675, 2700, 2725, 2750, 2775, 2800, 2825, 2850, 2870, 2875, 2880, 2890,
  2900, 2910, 2920, 2925, 2930, 2940, 2950, 2960, 2970, 2975, 2980, 2990, 3000,
  3010, 3020, 3025, 3030, 3040, 3050, 3060, 3070, 3075, 3080, 3090, 3100, 3110,
  3120, 3125, 3130, 3140, 3150, 3155, 3160, 3165, 3170, 3175, 3180, 3185, 3190,
  3195, 3200, 3205, 3210, 3215, 3220, 3225, 3230, 3235, 3240, 3245, 3250, 3255,
  3260, 3265, 3270, 3275, 3280, 3285, 3290, 3295, 3300, 3305, 3310, 3315, 3320,
  3325, 3330, 3335, 3340, 3345, 3350, 3355, 3360, 3365, 3370, 3375, 3380, 3385,
  3390, 3395, 3400, 3405, 3410, 3415, 3420, 3425, 3430, 3435, 3440, 3445, 3450,
  3455, 3460, 3465, 3470, 3475, 3480, 3485, 3490, 3495, 3500, 3505, 3510, 3515,
  3520, 3525, 3530, 3535, 3540, 3545, 3550, 3555, 3560, 3565, 3570, 3575, 3580,
```

```
    3585, 3590, 3595, 3600, 3605, 3610, 3615, 3620, 3625, 3630, 3635, 3640, 3645,
    3650, 3655, 3660, 3665, 3670, 3675, 3680, 3685, 3690, 3695, 3700, 3705, 3710,
    3715, 3720, 3725, 3730, 3735, 3740, 3745, 3750, 3755, 3760, 3765, 3770, 3775,
    3780, 3785, 3790, 3795, 3800, 3805, 3810, 3815, 3820, 3825, 3830, 3835, 3840,
    3845, 3850, 3855, 3860, 3865, 3870, 3875, 3880, 3885, 3890, 3895, 3900, 3905,
    3910, 3915, 3920, 3925, 3930, 3935, 3940, 3945, 3950, 3955, 3960, 3965, 3970,
    3975, 3980, 3985, 3990, 3995, 4000, 4005, 4010, 4015, 4020, 4025, 4030, 4035,
    4040, 4045, 4050, 4055, 4060, 4065, 4070, 4075, 4080, 4085, 4090, 4095, 4100,
    4105, 4110, 4115, 4120, 4125, 4130, 4135, 4140, 4145, 4150, 4155, 4160, 4165,
    4170, 4175, 4180, 4185, 4190, 4195, 4200, 4205, 4210, 4215, 4220, 4225,
    4230, 4235, 4240, 4245, 4250, 4255, 4260, 4265, 4270, 4275, 4280, 4285,
    4290, 4295, 4300, 4305, 4310, 4315, 4320, 4325, 4330, 4335, 4340, 4345,
    4350, 4355, 4360, 4365, 4370, 4375, 4380, 4385, 4390, 4395, 4400, 4405,
    4410, 4415, 4420, 4425, 4430, 4435, 4440, 4445, 4450, 4460, 4470, 4475,
    4480, 4490, 4500, 4510, 4520, 4525, 4530, 4540, 4550, 4560, 4570, 4575,
    4580, 4590, 4600, 4610, 4620, 4625, 4650, 4675, 4700, 4725, 4750, 4775,
    4800, 4825, 4850, 4875, 4900, 4925, 4950, 4975, 5000, 5025, 5050, 5075,
    5100, 5125, 5150, 5175, 5200, 5225, 5250, 5275, 5300, 5325, 5350, 5375,
    5400, 5450, 5500, 5550, 5600, 5700, 5800, 5900, 6000, 6100, 6200, 6300,
    6400, 6500, 6600, 6700, 6800, 6900, 7000, 7100, 7200, 7300, 7400, 7600,
    7800, 8000, 8200, 8300, 8400, 8600, 8800, 9000, 9200, 9600, 10000, 12000},
  TradingClass → {SPX, SPX, SPXW, SPXW, SPXW, SPX},
  Multiplier → {100, 100, 100, 100, 100, 100} |⟩
```

In[ ]:=
```
twsOptionContract[<|"Symbol" → "SPX", "SecType" → "OPT",
  "Exchange" → "CBOE", "Expiration" → SPXoptions["Expiration"][[1]],
  "Strike" → 3720, "Multiplier" → "100", "Right" → "C" |>]
```

Out[ ]=
```
SPX   230217C03720000
```

In[ ]:=
```
twsLogClear[]
```

In[ ]:=
```
twsLog[]
```

Out[ ]=

Index Options - Weekly

In[ ]:=
```
SPXoptions = twsOptionParams[<|
   "Symbol" → "SPX:SPXW", "SecType" → "IND", "Exchange" → "CBOE"|>]
```

Out[ ]=

⟨| Expiration →

{Thu 16 Feb 2023, Thu 16 Feb 2023, Thu 16 Feb 2023, Fri 17 Feb 2023, Fri 17 Feb 2023,
Fri 17 Feb 2023, Tue 21 Feb 2023, Tue 21 Feb 2023, Tue 21 Feb 2023, Wed 22 Feb 2023,
Wed 22 Feb 2023, Wed 22 Feb 2023, Thu 23 Feb 2023, Thu 23 Feb 2023, Thu 23 Feb 2023,
Fri 24 Feb 2023, Fri 24 Feb 2023, Fri 24 Feb 2023, Mon 27 Feb 2023, Mon 27 Feb 2023,

Mon 27 Feb 2023, Tue 28 Feb 2023, Tue 28 Feb 2023, Tue 28 Feb 2023, Wed 1 Mar 2023,
Wed 1 Mar 2023, Wed 1 Mar 2023, Thu 2 Mar 2023, Thu 2 Mar 2023, Thu 2 Mar 2023,
Fri 3 Mar 2023, Fri 3 Mar 2023, Fri 3 Mar 2023, Mon 6 Mar 2023, Mon 6 Mar 2023,
Mon 6 Mar 2023, Tue 7 Mar 2023, Tue 7 Mar 2023, Tue 7 Mar 2023, Wed 8 Mar 2023,
Wed 8 Mar 2023, Wed 8 Mar 2023, Thu 9 Mar 2023, Thu 9 Mar 2023, Thu 9 Mar 2023,
Fri 10 Mar 2023, Fri 10 Mar 2023, Fri 10 Mar 2023, Mon 13 Mar 2023, Mon 13 Mar 2023,
Mon 13 Mar 2023, Tue 14 Mar 2023, Tue 14 Mar 2023, Tue 14 Mar 2023, Wed 15 Mar 2023,
Wed 15 Mar 2023, Wed 15 Mar 2023, Thu 16 Mar 2023, Thu 16 Mar 2023,
Thu 16 Mar 2023, Fri 17 Mar 2023, Fri 17 Mar 2023, Fri 17 Mar 2023, Mon 20 Mar 2023,
Mon 20 Mar 2023, Mon 20 Mar 2023, Tue 21 Mar 2023, Tue 21 Mar 2023,
Tue 21 Mar 2023, Fri 24 Mar 2023, Fri 24 Mar 2023, Fri 24 Mar 2023, Fri 31 Mar 2023,
Fri 31 Mar 2023, Fri 31 Mar 2023, Fri 14 Apr 2023, Fri 14 Apr 2023, Fri 14 Apr 2023,
Fri 21 Apr 2023, Fri 21 Apr 2023, Fri 21 Apr 2023, Fri 28 Apr 2023, Fri 28 Apr 2023,
Fri 28 Apr 2023, Fri 19 May 2023, Fri 19 May 2023, Fri 19 May 2023, Wed 31 May 2023,
Wed 31 May 2023, Wed 31 May 2023, Fri 16 Jun 2023, Fri 16 Jun 2023, Fri 16 Jun 2023,
Fri 30 Jun 2023, Fri 30 Jun 2023, Fri 30 Jun 2023, Fri 21 Jul 2023, Fri 21 Jul 2023,
Fri 21 Jul 2023, Mon 31 Jul 2023, Mon 31 Jul 2023, Mon 31 Jul 2023, Fri 29 Sep 2023,
Fri 29 Sep 2023, Fri 29 Sep 2023, Fri 29 Dec 2023, Fri 29 Dec 2023, Fri 29 Dec 2023},

```
Strike → {200, 400, 600, 800, 1000, 1100, 1200, 1300, 1400, 1500, 1600, 1700,
  1800, 1900, 1975, 2000, 2100, 2150, 2200, 2250, 2300, 2350, 2400, 2450, 2500,
  2550, 2575, 2600, 2625, 2650, 2675, 2700, 2725, 2750, 2775, 2800, 2825, 2850,
  2870, 2875, 2880, 2890, 2900, 2910, 2920, 2925, 2930, 2940, 2950, 2960, 2970,
  2975, 2980, 2990, 3000, 3010, 3020, 3025, 3030, 3040, 3050, 3060, 3070, 3075,
  3080, 3090, 3100, 3110, 3120, 3125, 3130, 3140, 3150, 3155, 3160, 3165, 3170,
  3175, 3180, 3185, 3190, 3195, 3200, 3205, 3210, 3215, 3220, 3225, 3230, 3235,
  3240, 3245, 3250, 3255, 3260, 3265, 3270, 3275, 3280, 3285, 3290, 3295, 3300,
  3305, 3310, 3315, 3320, 3325, 3330, 3335, 3340, 3345, 3350, 3355, 3360, 3365,
  3370, 3375, 3380, 3385, 3390, 3395, 3400, 3405, 3410, 3415, 3420, 3425, 3430,
  3435, 3440, 3445, 3450, 3455, 3460, 3465, 3470, 3475, 3480, 3485, 3490, 3495,
  3500, 3505, 3510, 3515, 3520, 3525, 3530, 3535, 3540, 3545, 3550, 3555, 3560,
  3565, 3570, 3575, 3580, 3585, 3590, 3595, 3600, 3605, 3610, 3615, 3620, 3625,
  3630, 3635, 3640, 3645, 3650, 3655, 3660, 3665, 3670, 3675, 3680, 3685, 3690,
  3695, 3700, 3705, 3710, 3715, 3720, 3725, 3730, 3735, 3740, 3745, 3750, 3755,
  3760, 3765, 3770, 3775, 3780, 3785, 3790, 3795, 3800, 3805, 3810, 3815, 3820,
  3825, 3830, 3835, 3840, 3845, 3850, 3855, 3860, 3865, 3870, 3875, 3880, 3885,
  3890, 3895, 3900, 3905, 3910, 3915, 3920, 3925, 3930, 3935, 3940, 3945, 3950,
  3955, 3960, 3965, 3970, 3975, 3980, 3985, 3990, 3995, 4000, 4005, 4010, 4015,
  4020, 4025, 4030, 4035, 4040, 4045, 4050, 4055, 4060, 4065, 4070, 4075, 4080,
```

```
    4085, 4090, 4095, 4100, 4105, 4110, 4115, 4120, 4125, 4130, 4135, 4140, 4145,
    4150, 4155, 4160, 4165, 4170, 4175, 4180, 4185, 4190, 4195, 4200, 4205, 4210,
    4215, 4220, 4225, 4230, 4235, 4240, 4245, 4250, 4255, 4260, 4265, 4270, 4275,
    4280, 4285, 4290, 4295, 4300, 4305, 4310, 4315, 4320, 4325, 4330, 4335, 4340,
    4345, 4350, 4355, 4360, 4365, 4370, 4375, 4380, 4385, 4390, 4395, 4400,
    4405, 4410, 4415, 4420, 4425, 4430, 4435, 4440, 4445, 4450, 4460, 4470,
    4475, 4480, 4490, 4500, 4510, 4520, 4525, 4530, 4540, 4550, 4560, 4570,
    4575, 4580, 4590, 4600, 4610, 4620, 4625, 4650, 4675, 4700, 4725, 4750,
    4775, 4800, 4825, 4850, 4875, 4900, 4925, 4950, 4975, 5000, 5025, 5050,
    5075, 5100, 5125, 5150, 5175, 5200, 5225, 5250, 5275, 5300, 5325, 5350,
    5375, 5400, 5450, 5500, 5550, 5600, 5700, 5800, 5900, 6000, 6100, 6200,
    6300, 6400, 6500, 6600, 6800, 7000, 7200, 7400, 7600, 7800, 8000, 8200},
  TradingClass → {SPXW, SPXW, SPXW}, Multiplier → {100, 100, 100}|⟩
```

In[ ]:=
```
SPXWC3900 = twsOptionContract[<|"Symbol" → "SPX", "SecType" → "OPT",
    "Exchange" → "CBOE", "Expiration" → SPXoptions["Expiration"][[2]],
    "Strike" → 3720, "Multiplier" → "100", "Right" → "C" |>]
```

Out[ ]=
```
SPX   230217C03720000
```

In[ ]:=
```
twsMktData[<|"SecType" → "OPT",
  "Exchange" → "CBOE", Symbol → "SPXW  221101C03900000::100"|>]
```

```
Wolfram Library Exception
Unknown instrument OPT CBOE:SPXW  221101C03900000::100
  USD. No security definition has been found for the request
```

Out[ ]=
```
$Failed
```

Market Data for Options

Stock Options

to get option Greeks we should subscribe to underlying market data. The first function call initiates the subscription. Subsequent function calls retrieve the market data.

In[ ]:=
```
twsMktData["AAPL"]
```

Out[ ]=
```
⟨|LastPrice → 151, LastSize → 9, Bid → 151, Ask → 151.08,
 BidSize → 6, AskSize → 5, High → 152.49, Low → 149.36|⟩
```

In[ ]:=
```
twsMktData["Symbol" → AAPLP140, "SecType" → "OPT", "Exchange" → "SMART"]
```

Out[ ]=
```
⟨|LastPrice → 0, LastSize → 0, Bid → 0,
 Ask → 0, BidSize → 0, AskSize → 0, High → 0, Low → 0|⟩
```

In[ ]:=
```
dsAAPLP140 =
 Transpose@Dataset@twsHistoricalData[<|"Symbol" → AAPLP140, "SecType" → "OPT",
     "BarSize" → "5 mins", "Duration" → "1 W"|>]
```

Out[ ]=

Date	Open	High	Low	C
2022–10–24 09:30:00	3.09	3.09	3.09	3
2022–10–24 09:35:00	3.25	3.25	2.9	2
2022–10–24 09:40:00	3.15	3.15	3.15	3
2022–10–24 09:45:00	3.01	3.01	2.96	2
2022–10–24 09:50:00	3.01	3.01	2.97	2
2022–10–24 09:55:00	2.78	2.82	2.78	2
2022–10–24 10:00:00	2.91	3.15	2.9	3
2022–10–24 10:05:00	3.2	3.2	3.2	3
2022–10–24 10:10:00	3.2	3.2	3.2	3
2022–10–24 10:15:00	3.2	3.2	3.2	3
2022–10–24 10:20:00	3.4	3.4	3.35	3
2022–10–24 10:25:00	3.35	3.35	3.35	3
2022–10–24 10:30:00	3.3	3.3	3.3	3
2022–10–24 10:35:00	3.1	3.1	3.1	3
2022–10–24 10:40:00	3.1	3.1	3.1	3
2022–10–24 10:45:00	3.27	3.27	3.27	3
2022–10–24 10:50:00	3.27	3.27	3.27	3
2022–10–24 10:55:00	3.27	3.27	3.27	3
2022–10–24 11:00:00	3.27	3.27	3.27	3
2022–10–24 11:05:00	3.27	3.27	3.27	3

rows 1–20 of **390**

In[ ]:=
```
twsLogClear[]
```

In[ ]:=
```
twsLog[]
```

Out[ ]=
```
IB message [200] No security definition has been found for the request
```

Index Options

```
In[ ]:= twsMktData[<|"Symbol" → "SPX", "SecType" → "IND", "Exchange" → "CBOE"|>]
Out[ ]=
⟨|LastPrice → 0, LastSize → 0, Bid → 0, Ask → 0,
 BidSize → -2 147 483 648, AskSize → -2 147 483 648, High → 0, Low → 0|⟩

In[ ]:= twsMktData["Symbol" → SPXWC3900, "SecType" → "OPT", "Exchange" → "CBOE"]
Out[ ]=
⟨|LastPrice → 0, LastSize → 0, Bid → 0,
 Ask → 0, BidSize → 0, AskSize → 0, High → 0, Low → 0|⟩

In[ ]:= Transpose@Dataset@twsHistoricalData[<|"Symbol" → SPXWC3900, "SecType" → "OPT",
     "Exchange" → "CBOE", "BarSize" → "5 mins", "Duration" → "1 D"|>]
Out[ ]=
```

Date	Open	High	Low	C
2022–10–28 09:30:00	4	8.1	4	6
2022–10–28 09:35:00	7	8.5	6.7	8
2022–10–28 09:40:00	8.5	10	8.5	8
2022–10–28 09:45:00	8.5	9.8	8.2	9
2022–10–28 09:50:00	9	9.5	8.5	9
2022–10–28 09:55:00	9.5	10.8	8.3	8
2022–10–28 10:00:00	9.8	10.4	6.4	7
2022–10–28 10:05:00	8	8.1	6.9	6
2022–10–28 10:10:00	7.3	7.3	4.8	6
2022–10–28 10:15:00	6.7	8.5	6.7	8
2022–10–28 10:20:00	8.1	8.5	7.6	8
2022–10–28 10:25:00	8.6	11.7	8.6	1
2022–10–28 10:30:00	11	12.5	11	1
2022–10–28 10:35:00	10.9	11.8	10.1	1
2022–10–28 10:40:00	11.6	12.8	11.1	1
2022–10–28 10:45:00	13	15.4	12.6	1
2022–10–28 10:50:00	15.3	15.5	14.5	1
2022–10–28 10:55:00	16	16	15	1
2022–10–28 11:00:00	15.9	16.6	14.2	1
2022–10–28 11:05:00	14.5	15	13.5	1

rows 1–20 of **81**

Exercise options

```
twsOptionExercise["Symbol" → AAPLP140, "SecType" → "OPT", Quantity → 1 ]
```

Time & Sales

```
In[ ]:= timesalesGE =
          twsHistoricalTrades["Symbol" → "GE", "SecType" → "STK", "Exchange" → "SMART",
           "FromTime" → "2022-10-18 09:30:00", "ToTime" → "2022-10-18 09:40:00"];

In[ ]:= Dimensions@timesalesGE[[1]]

Out[ ]=
        {3019}

In[ ]:= timesalesGE[[All, 30 ;; 40]] // Dataset // Transpose

Out[ ]=
```

Time	Price	Size	Exchange	Conditions
2022–10–18 09:30:01	70.19	1	FINRA	4 I
2022–10–18 09:30:01	70.19	13	FINRA	4 I
2022–10–18 09:30:01	70.19	4	FINRA	4 I
2022–10–18 09:30:01	70.01	1	FINRA	I
2022–10–18 09:30:01	70.2236	3	FINRA	I
2022–10–18 09:30:01	70.01	1	FINRA	I
2022–10–18 09:30:01	70.23	5	ISLAND	I
2022–10–18 09:30:01	70.01	1	FINRA	I
2022–10–18 09:30:01	70.01	1	FINRA	I
2022–10–18 09:30:01	70.01	1	FINRA	I
2022–10–18 09:30:01	69.77	1	FINRA	I

TWS Log

Get a detailed log of actions taken by MATH-TWS and interactions with the IB TWS application using the twsLog function:

In[ ]:=
```
twsLog[]
```

Out[ ]=
```
IB message [2100] API client has been unsubscribed from account data.
IB message [200] No security definition has been found for the request
news providers: BRFG+BRFUPDN+DJNL id=10010
got news 10010 BRFUPDN$09a61662
got news 10010 BRFUPDN$09a6161f
got news 10010 BRFUPDN$09a6156c
got news 10010 BRFG$09a63dd9
got news 10010 BRFUPDN$09a646ae
historicalNewsEnd 10010 hasMore=1
Got order 79446 status PreSubmitted
execDetails 0000e1a7.5c3be061.01.01
Got order 79446 status Filled
Got order 79446 status Filled
Got order 79447 status PreSubmitted
execDetails 0000e1a7.5c3be065.01.01
Got order 79447 status Filled
Got order 79447 status Filled
Got order 79448 status PreSubmitted
Got order 79448 status PreSubmitted
OptionContract request symbol=AAPL sectype:OPT
  excahnge:SMART mult:100 class: exp:2019-02-01 strike:120 right:P
cname=AAPL  190201P00120000::100
OptionContract request symbol=AAPL sectype:OPT
  excahnge:SMART mult:100 class: exp:2019-02-01 strike:200 right:P
cname=AAPL  190201P00200000::100
OptionContract request symbol=AAPL sectype:OPT
  excahnge:SMART mult:100 class: exp:2019-02-01 strike:120 right:P
cname=AAPL  190201P00120000::100
OptionContract request symbol=AAPL sectype:OPT
  excahnge:SMART mult:100 class: exp:2019-02-01 strike:120 right:P
cname=AAPL  190201P00120000::100
```

The twsLogClear function clears the log:

In[ ]:=
```
twsLogClear[]
```

In[ ]:=
```
twsLog[]
```

Out[ ]=

DISCLAIMER

THIS MATH-TWS SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,

AND/OR NONINFRINGEMENT.THIS SOFTWARE IS NOT OFFICIALLY APPROVED OR ENDORSED BY ANY REGULATORY, GOVERNING OR COMMERCIAL BODY, INCLUDING SEC, FINRA, WOLFRAM AND/OR INTERACTIVE BROKERS.
MUCH EFFORT WAS INVESTED TO ENSURE THE CORRECTNESS, ACCURACY AND USEFULNESS OF THE INFORMATION PRESENTED IN THIS DOCUMENT AND THE SOFTWARE. HOWEVER, THERE IS NEITHER A GUARANTEE THAT THE
INFORMATION IS COMPLETE OR ERROR-FREE, NOR THAT IT MEETS THE USER'S NEEDS. THE AUTHOR AND COPYRIGHT HOLDERS TAKE ABSOLUTELY NO RESPONSIBILITY FOR POSSIBLE CONSEQUENCES DUE TO THIS DOCUMENT
OR USE OF THE SOFTWARE.THE FUNCTIONALITY OF THE SOFTWARE DEPENDS, IN PART, ON THE FUNCTIONALITY OF OTHER SOFTWARE, HARDWARE, SYSTEMS AND SERVICESBEYOND OUR CONTROL. SUCH EXTERNAL COMPONENTS MAY CHANGE OR
STOP TO FUNCTION AT ANY TIME, WITHOUT PRIOR NOTICE AND WITHOUT OUR CONTROL. THEREFORE, THERE CAN BE NO ASSURANCE THAT THE SOFTWARE WOULD WORK, AS EXPECTED OR AT ALL, AT ANY GIVEN TIME.
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES, LOSS, OR OTHER LIABILITY, WHETHER IN ACTION OF CONTRACT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNECTION WITH
THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE, REGARDLESS OF FORM OF CLAIM OR WHETHER THE AUTHORS WERE ADVISED OF SUCH LIABILITIES.
WHEN USING THIS DOCUMENT AND SOFTWARE, USERS MUST VERIFY THE BEHAVIOR CAREFULLY ON THEIR SYSTEM BEFORE USING THE SAME FUNCTIONALITY FOR LIVE TRADES. USERS SHOULD EITHER USE THIS
DOCUMENT AND SOFTWARE AT THEIR OWN RISK, OR NOT AT ALL. ALL TRADING SYMBOLS AND TRADING ORDERS DISPLAYED IN THE DOCUMENTATION ARE FOR ILLUSTRATIVE PURPOSES ONLY

THIS RISK OF LOSS IN ONLINE TRADING OF STOCKS, OPTIONS, FUTURES, FOREX, FOREIGN EQUITIES AND FIXED INCOEM CAN BE SUBSTANTIAL.