Programming IVolatility SDK

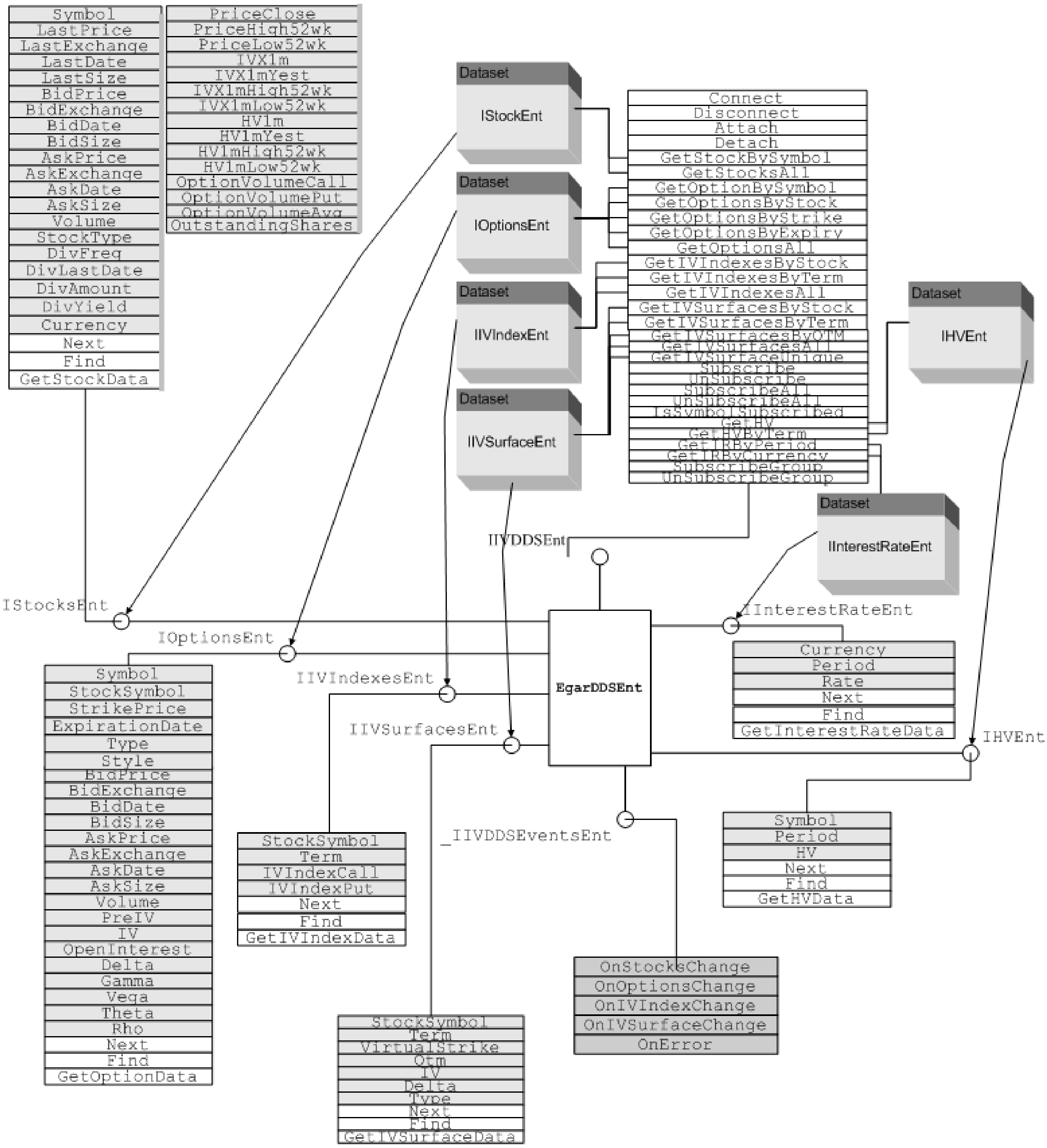
# 1 Introduction

The IVolatility SDK is a set of software libraries with simple data access methods to U.S. Stocks, Futures, options quotes and derivatives data such as implied volatilities, Greeks, volatility surface etc. With industry standard COM interface you get an unprecedented flexibility with feeding 20 minutes delayed quotes and derivatives data into custom application for Microsoft Windows. IVolatility SDK also includes DDE server so you can access 20 minutes delayed volatility data in Microsoft Excel.

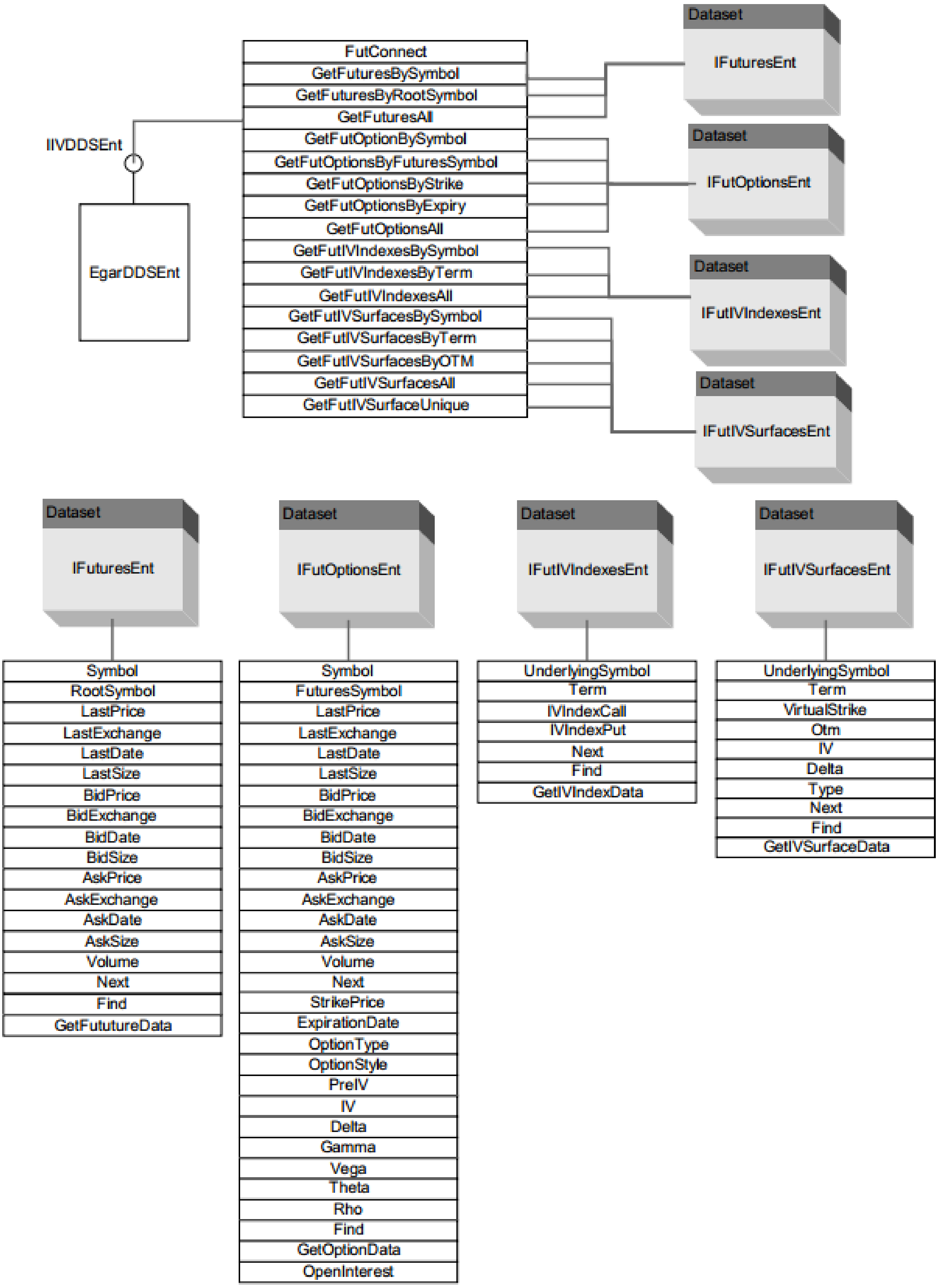
The SDK offers both event-driven and snapshot based interfaces for flexible data access. For example you can get options chain snapshot for specified underlying stock using getOptionsbyStock()method. You can also subscribe for 20 minutes delayed data using one or several subscription methods (SubscribeAll()method, for example, will subscribe you to all instruments you are authorized to access). Upon successful subscription a pre-defined callback method (onOptionsChange method to receive individual option contracts quotes and volatilities changes) will be called every time the value is updated.

The IVolatility SDK consists of several class interfaces that will be described in greater details below. The core interface IVDDSEntis used to initialize the system (to establish connection and perform authorization on IVolatility.com database server). Upon successful object creation you should first do a Connect()request to perform initial connection and authorization.

# 2 IVolatility SDK Interfaces Diagram




**IVolatility SDK Stock Interface diagram**




**IVolatility SDK Futures Interface diagram**

”DDS Enterprise Library”is a core ActiveX object of IVolatility SDK. IIVDDSEnt is the main interface to get access to the data. It has methods to initialize the system (Connect, Disconnect) and several methods to retrieve the data and subscribe for data updates. All data is organized by collections i.e. IStocksEnt, IOptionsEnt, IIinterestRateEnt, IIVIndexesEnt and IIVSurfacesEnt. To get the data snapshot use appropriate IIVDDSEnt Get... method. Get... method creates ‘in-memory’ table with internal pointer to ‘current’ row. Method ‘Next’ moves internal pointer to the next row. Method ‘Find’ moves internal pointer to the first row matching requested stock, futures or options symbol.

For data update use ‘Subscribe…’ method and implement appropriate On…Change callback method as well. ‘Data update’ events are normaly stored in the same IStocksEnt, IOptionsEnt, IIVIndexesEnt or IIVSurfacesEnt collections containing only trends.

# 3 IVDDSEnt interface.

## 3.1 Initialization

Method Connect() should be used to establish connection with IVolatility master data source for stock data access. You should always call this method first.

*Definition:*

```
HRESULT Connect([in] BSTR UserName, [in] BSTR Password);
```

*Parameters:*

UserName – username
Password – password

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
```

Method FutConnect() should be used to establish connection with IVolatility master data source for futures data access. . You should always call this method first.

*Definition:*

```
HRESULT FutConnect([in] BSTR UserName, [in] BSTR Password);
```

The Disconnect() method is used to disconnect from master data source. You should always call this method before closing your application.

*Definition:*

```
HRESULT Disconnect();
```

*Parameters:*

No parameters

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
...
dds.Disconnect
```

## 3.2 Stock data request

Method GetStockBySymbol and GetStocksAll returns a Stocks collection. See IStock interface description below to see how to work with Stocks collection.

*Definition:*

```
HRESULT GetStockBySymbol([in] BSTR Symbol, [out,retval] IStocks** pStocks);
HRESULT GetStocksAll([out, retval] IStocks **pStocks);
```

*Parameters:*

Symbol – ticker symbol.
pStocks – IStock collection interface

*Example 1:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim stck As StocksEnt
Set stck = dds. GetStockBySymbol (Symbol)
…
dds.Disconnect
```

*Example 2:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim stcksCol As StocksEnt
Set stcksCol = dds.GetStocksAll
….
dds.Disconnect
```

## 3.3 Options data request

Options data can be initialized in a collection object. You can fill in this collection by only one option contract or by a group of option contracts. There are several methods that allow you to fill Options collection with:

### 3.3.1 Single option contract

*Definition:*

```
HRESULT GetOptionBySymbol ([in] BSTR Symbol, [out, retval] IOptions** pOptions);
```

*Parameters:*

Symbol – option symbol
pData – IOptions collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim optn As OptionsEnt
Set optn = dds. GetOptionBySymbol ("IBMDA")

dds.Disconnect
```

### 3.3.2 Options chain for specified underlying

*Definition:*

```
HRESULT GetOptionsByStock([in] BSTR Symbol, [out, retval] IOptions** pOptions);
```

*Parameters:*

Symbol – Stock symbol
pOptions – IOptions collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim optnsCollection As OptionsEnt
Set optnsCollection = dds. GetOptionsByStock (Symbol)
```

```
…
dds.Disconnect
```

### 3.3.3 Options for specified underlying and strike

*Definition:*

HRESULT GetOptionsByStrike([in] BSTR StockSymbol, [in] DOUBLE StrikePrice, [out, retval] IOptions** pOptions);

*Parameters:*

StockSymbol – Stock symbol.
StrikePrice – option Strike price.
pOptions – IOptions collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim optnsCollection As OptionsEnt
Set optnsCollection = dds.GetOptionsByStrike (StockSymbol, StrikePrice)
….
dds.Disconnect
```

### 3.3.4 Options for specified underlying and expiration

*Definition:*

HRESULT GetOptionsByExpiry([in] BSTR StockSymbol, [in] DATE ExpirationDate, [out, retval] IOptions** pOptions);

*Parameters:*

StockSymbol – Stock symbol.
ExpirationDate – option expiration date
pOptions – IOptions collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim optnsCollection As OptionsEnt
Set optnsCollection = dds. GetOptionsByExpiry (StockSymbol, ExpirationDate)
….
dds.Disconnect
```

### 3.3.5 All options for all underlyings

This method returns a collection of all options that specified client is authorized to receive.

*Definition:*

HRESULT GetOptionsAll([out, retval] IOptions** pOptions);

*Parameters:*

pOptions – IOptions collection reference

*Example:*

```
Dim dds As New IVDDSEnt
```

```
dds.Connect username, password
Dim optnsCollection As OptionsEnt
Set optnsCollection = dds.GetOptionsAll
dds. Detach enOptionData
….
dds.Disconnect
```

## 3.4 IV Index data request

Observation of methods returning Implied Volatility Index (IV Index).

### 3.4.1 IV Indexes for specified stock

*Definition:*

HRESULT GetIVIndexesByStock([in] BSTR StockSymbol, [out, retval] IIVIndexes** pIVIndexes);

*Parameters:*

StockSymbol – underlying symbol
pIVIndexes – IVIndexes collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivxCollection As IVIndexesEnt
Set ivxCollection = dds. GetIVIndexesByStock(StockSymbol)
…
dds.Disconnect
```

### 3.4.2 IV Indexes for specified stock and term (IV index horizon)

*Definition:*

HRESULT GetIVIndexesByTerm([in] BSTR StockSymbol, [in] SHORT Term, [out, retval] IIVIndexData** pIVIndexData);

*Parameters:*

StockSymbol – underlying symbol
Term – term (in calendar days)
pIVIndexes – IVIndexes collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivxCollection As IVIndexesEnt
Set ivxCollection = dds. GetIVIndexesByTerm (StockSymbol, Term)
…
dds.Disconnect
```

### 3.4.3 IV Indexes for all stocks

GetIVIndexesAll returns a collection of IV Indexes for all instruments current user is authorized to receive.

*Definition:*

HRESULT GetIVIndexesAll([out, retval] IIVIndexes** pIVIndexes);

*Parameters:*

pIVIndexes – IVIndexes collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivxCollection As IVIndexesEnt
Set ivxCollection = dds. GetIVIndexesAll
…
dds.Disconnect
```

## 3.5 IV Surface Data Request

Observation of methods returning Implied Volatility Surface (IV Surface).

### 3.5.1 IV Surface data for specified stock

*Definition:*

HRESULT GetIVSurfacesByStock([in] BSTR StockSymbol, [out, retval] IIVSurfaces** pIVSurfaces);

*Parameters:*

StockSymbol – underlying symbol.
pIVSurfaces – IVSurfaces collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivsCollection As IVSurfaces
Set ivsCollection = dds.GetIVSurfacesByStock(StockSymbol)
...
dds.Detach enIVIndexData
dds.Disconnect
```

### 3.5.2 IV Surface data for specified stock and term (surface horizon)

*Definition:*

HRESULT GetIVSurfacesByTerm([in] BSTR StockSymbol, [in] SHORT Term, [out, retval] IIVSurfaces** pIVSurfaces);

*Parameters:*

StockSymbol – underlying symbol
Term – term (in calendar days)
pIVSurfaces – IVSurfaces collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivsCollection As IVSurfacesEnt
Set ivsCollection = dds. GetIVSurfacesByTerm(StockSymbol, Term)
…
dds.Disconnect
```

### 3.5.3 IV Surface data for specified stock and moneyness

*Definition:*

HRESULT GetIVSurfacesByOTM([in] BSTR StockSymbol, [in] SHORT OTM, [out, retval] IIVSurfaces** pIVSurfaces);

*Parameters:*

StockSymbol – underlying symbol
OTM - moneyness
pIVSurfaces – IVSurfaces collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivsCollection As IVSurfacesEnt
Set ivsCollection = dds. GetIVSurfacesByOTM (StockSymbol, OTM)
```

### 3.5.4 IV Surface data for all stocks

GetIVSurfacesAll returns a collection of IV Surface records for all instruments current user is authorized to receive.

*Definition:*

HRESULT GetIVIndexesAll([out, retval] IIVSurfaces** pIVSurfaces);

*Parameters:*

pIVIndexes – IVSurfaces collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivsCollection As IVSurfacesEnt
Set ivsCollection = dds. GetIVSurfacesAll
…
dds.Disconnect
```

### 3.5.5 Single IV Surface point

Method GetIVSurfaceUnique returns a single volatility point on the Volatility Surface curve for selected instrument and specified term, moneyness (OTM value) and Call/Put curve portion (Call for virtual strikes above spot, Put for strikes below spot)

*Definition:*

HRESULT GetIVSurfaceUnique([in] BSTR StockSymbol, [in] SHORT Term, [in] SHORT OTM, [in] OptionTypeEnum Type, [out, retval] IIVSurfaces** pIVSurfaceData)

*Parameters:*

StockSymbol – underlying symbol
Term – term (days)
OTM – moneyness
Type – enCall or enPut
pIVSurfaces – IVSurfaces collection reference

*Example:*

```
Dim dds As New IVDDSEnt
dds.Connect username, password
Dim ivs As IVSurfacesEnt
Set ivs = dds. GetIVSurfaceUnique(StockSymbol, Term, OTM, Type)
…
dds.Disconnect
```

## 3.6 Historical Volatility (HV) Data Request

Method GetHV returns HV data collection for given symbol, `GetHVByTerm` returns HV value for given symbol and term. See HVEnt interface description below to see how to work with HV data collection.

*Definition:*

```
HRESULT GetHV([in] BSTR Symbol, [out, retval] IHVEnt** pHV);
HRESULT GetHVByTerm([in] BSTR Symbol, [in] SHORT Term, [out, retval] FLOAT*
HV);
```

*Parameters:*

Symbol – security symbol.
Period - term

Example:

```
Dim dds As New IVDDSEnt
Dim HV30 As Float
dds.Connect username, password
dds.GetHVByTerm(“MSFT”, 30, HV30)
….
dds.Disconnect
```

## 3.7 Interest Rate Data Request

Method GetIRByPeriod returns IR value for given currency and period, GetIRByCurrency returns Interest rate data collection. See InterestRateEnt interface description below to see how to work with Interest rate collection.

*Definition:*

```
HRESULT GetIRByPeriod([in] BSTR Currency, [in] LONG Period, [out, retval] FLOAT* Rate);
HRESULT GetIRByCurrency([in] BSTR Currency, [out, retval] IInterestRateEnt** pInterestRate);
```

*Parameters:*

Currency – currency symbol.
Period - period

Example:

```
Dim dds As New IVDDSEnt
Dim rate As Float
dds.Connect username, password
dds.GetIRByPeriod(“USD”, 60, rate)
….
dds.Disconnect
```

## 3.8 Futures Data Request

Futures data is stored in in FuturesEntcollection. There are several methods returning futures collection.

### 3.8.1 Futures data for specified symbol

*Definition:*

HRESULT GetFutureBySymbol([in] BSTR Symbol, [out,retval] IFuturesEnt** pData);

*Parameters:*

Symbol –futuressymbol.

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futuresCollection As FuturesEnt

Set futuresCollection  = dds. GetFutureBySymbol  (symbol)
….
dds.Disconnect
```

### 3.8.2 Futures for specified futures root symbol

*Definition:*

HRESULT GetFutureByRootSymbol([in] BSTR Symbol, [out,retval] IFuturesEnt** pFutures);

*Parameters:*

Symbol –futures root symbol.

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futuresCollection As FuturesEnt

Set futuresCollection  = dds.GetFutureByRootSymbol (symbol)
….
dds.Disconnect
```

### 3.8.3 All futures

*Definition:*

HRESULT GetFuturesAll([out, retval] IFuturesEnt** pFutures);

*Parameters:*

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futuresCollection As FuturesEnt

Set futuresCollection  = dds.GetFuturesAll
….
dds.Disconnect
```

## 3.9 Futures Options Data Request

Futures options data is stored in FutOptionsEnt collection. There are several methods returning futures options collection.

### 3.9.1 Single futures option contract

*Definition:*

HRESULT GetFutOptionBySymbol([in] BSTR Symbol, [out,retval] IFutOptionsEnt** pData);

*Parameters:*

Symbol –option symbol.

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futOptionsCollection As FutOptionsEnt

Set futOptionsCollection  = dds.GetFutOptionBySymbol (symbol)
….
dds.Disconnect
```

### 3.9.2 Options chain for specified futures symbol

*Definition:*

GetFutOptionsByFutureSymbol([in] BSTR Symbol, [out, retval] IFutOptionsEnt** pOptions);

*Parameters:*

Symbol –futures symbol.

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futOptionsCollection As FutOptionsEnt

Set futOptionsCollection  = dds.GetFutOptionsByFutureSymbol (symbol)
….
dds.Disconnect
```

### 3.9.3 Options for specified futures symbol and strike price

*Definition:*

HRESULT GetFutOptionsByStrike([in] BSTR Symbol, [in] DOUBLE StrikePrice, [out, retval] IFutOptionsEnt** pOptions);

*Parameters:*

Symbol –futuressymbol.
StrikePrice – option strike

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futOptionsCollection As FutOptionsEnt

Set futOptionsCollection  = dds.GetFutOptionsByStrike (symbol, strike)
….
dds.Disconnect
```

### 3.9.4 Options for specified futures symbol and expiration date

*Definition:*

GetFutOptionsByExpiry([in] BSTR Symbol, [in] DATE ExpirationDate, [out, retval] IFutOptionsEnt** pOptions);

*Parameters:*

Symbol –futuressymbol.
ExpirationDate – option expiration date

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
```

```
Dim futOptionsCollection As FutOptionsEnt

Set futOptionsCollection  = dds.GetFutOptionsByExpiry(symbol,expirationDate)
….
dds.Disconnect
```

### 3.9.5 All options for all futures

*Definition:*

GetFutOptionsAll([out, retval] IFutOptionsEnt** pOptions);

*Parameters:*

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futOptionsCollection As FutOptionsEnt

Set futOptionsCollection  = dds.GetFutOptionsAll
….
dds.Disconnect
```

## 3.10 Futures IV Index Data Request

Futures IV Index data is stored in FutIVIndexesEnt collection. There are several methods returning Futures IV Index collection.

### 3.10.1 IV Indexes for specified futures symbol

*Definition:*

GetFutIVIndexesBySymbol([in] BSTR Symbol, [out, retval] IFutIVIndexesEnt** pIVIndexes);

*Parameters:*

Symbol – futures underlying.

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVIndexesCollection As FutIVIndexesEnt

Set futOptionsCollection  = dds.GetFutIVIndexesBySymbol(symbol)
….
dds.Disconnect
```

### 3.10.2 IV Indexes for specified futures symbol and term

*Definition:*

GetFutIVIndexesByTerm([in] BSTR Symbol, [in] SHORT Term, [out, retval] IFutIVIndexesEnt** pIVIndexData);

*Parameters:*

Symbol – futures underlying.

Term – term (in calendar days)

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVIndexesCollection As FutIVIndexesEnt

Set futOptionsCollection  = dds. GetFutIVIndexesByTerm(symbol, term)
….
dds.Disconnect
```

### 3.10.3 IV Index data for all futures

*Definition:*

GetFutIVIndexesAll([out, retval] IFutIVIndexesEnt** pIVIndexes);;

*Parameters:*

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVIndexesCollection As FutIVIndexesEnt

Set futOptionsCollection  = dds. GetFutIVIndexesAll
….
dds.Disconnect
```

## 3.11 Futures IV Surface Data Request

Futures IV Surface data is stored in FutIVSurfacesEnt collection object. There are several methods returning  Futures IV Surface collection.

### 3.11.1 IV Surface data for specified futures symbol

*Definition:*

GetFutIVSurfacesBySymbol([in] BSTR Symbol, [out, retval] IFutIVSurfacesEnt** pIVSurfaces);

*Parameters:*

Symbol – futures underlying.

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVSurfacesCollection As FutIVSurfacesEnt

Set futIVSurfacesCollection = dds.GetFutIVSurfacesBySymbol (symbol)
….
dds.Disconnect
```

### 3.11.2 IV Surface data for specified futures symbol and term

*Definition:*

GetFutIVSurfacesByTerm([in] BSTR Symbol, [in] SHORT Term, [out, retval] IFutIVSurfacesEnt** pIVSurfaces);

*Parameters:*

Symbol – futures underlying.

Term – term (in calendar days)

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVSurfacesCollection As FutIVSurfacesEnt

Set futIVSurfacesCollection = dds. GetFutIVSurfacesByTerm (symbol, term)
….
dds.Disconnect
```

### 3.11.3 IV Surface for specified futures and moneyness

*Definition:*

GetFutIVSurfacesByOTM([in] BSTR Symbol, [in] DOUBLE OTM, [out, retval] IFutIVSurfacesEnt** pIVSurfaces);

*Parameters:*

Symbol – futures underlying.

OTM – moneyness

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVSurfacesCollection As FutIVSurfacesEnt
```

```
Set futIVSurfacesCollection = dds. GetFutIVSurfacesByOTM (symbol,otm)
….
dds.Disconnect
```

### 3.11.4 IV Surface data for all futures

*Definition:*

GetFutIVSurfacesAll([out, retval] IFutIVSurfacesEnt** pIVSurface);

*Parameters:*

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVSurfacesCollection As FutIVSurfacesEnt

Set futIVSurfacesCollection = dds.GetFutIVSurfacesAll
….
dds.Disconnect
```

### 3.11.5 Single IV Surface point

*Definition:*

GetFutIVSurfaceUnique([in] BSTR StockSymbol, [in] SHORT Term, [in] DOUBLE OTM, [in] OptionTypeEnum Type, [out, retval] IFutIVSurfacesEnt** pIVSurfaceData);

*Parameters:*

Symbol – futures underlying.
Term – term (days)
OTM – moneyness
Type – enCall or enPut

Example:

```
Dim dds As New IVDDSEnt
dds. FutConnect username, password
Dim futIVSurfacesCollection As FutIVSurfacesEnt

Set futIVSurfacesCollection = dds.GetFutIVSurfaceUnique (symbol,term,otm,type)
….
dds.Disconnect
```

# 4 IStocksEnt Interface

IStocks interface provides a set of methods to iterate over Stock data collection and access each item properties.

| | |
|---|---|
| Symbol | HRESULT Symbol([out, retval] BSTR *pVal)<br>Stock Symbol |
| LastPrice | HRESULT LastPrice([out, retval] FLOAT *pVal)<br>LastPrice |
| LastExchange | HRESULT LastExchange([out, retval] BSTR *pVal)<br>Exchange of last price |
| LastDate | HRESULT LastDate([out, retval] DATE *pVal)<br>Time or date of last price |
| LastSize | HRESULT LastSize([out, retval] LONG *pVal)<br>Last size |
| BidPrice | HRESULT BidPrice([out, retval] FLOAT *pVal)<br>Bid Price |
| BidExchange | HRESULT BidExchange([out, retval] BSTR *pVal)<br>Exchange of bid price |
| BidDate | HRESULT BidDate([out, retval] DATE *pVal)<br>Time or date of bid price |
| BidSize | HRESULT BidSize([out, retval] LONG *pVal)<br>Bid size |
| AskPrice | HRESULT AskPrice([out, retval] FLOAT *pVal)<br>Ask Price |
| AskExchange | HRESULT AskExchange([out, retval] BSTR *pVal)<br>Exchange of ask price |
| AskDate | HRESULT AskDate([out, retval] DATE *pVal)<br>Time or date of ask price |
| AskSize | HRESULT AskSize([out, retval] LONG *pVal)<br>Ask size |
| Volume | HRESULT Volume([out, retval] long *pVal)<br>Volume |
| StockType | HRESULT StockType([out, retval] StockTypeEnum *pVal)<br>enum StockTypeEnum<br>{ enIndex = 0,<br>enStock = 1<br>} StockTypeEnum;<br>Returns security type. |
| PriceClose | HRESULT PriceClose([out, retval] FLOAT *pVal);<br>Close price as of yesterday |
| PriceHigh52wk | HRESULT PriceHigh52wk([out, retval] FLOAT *pVal);<br>Highest price for the last year |
| PriceLow52wk | HRESULT PriceLow52wk([out, retval] FLOAT *pVal);<br>Lowest price for the last year |

| | |
|---|---|
| IVX1m | HRESULT IVX1m([out, retval] FLOAT *pVal);<br>1-m IV Index as of yesterday close |
| IVX1mYest | HRESULT IVX1mYest([out, retval] FLOAT *pVal);<br>Day before yesterdays value of IV Index |
| IVX1mHigh52wk | HRESULT IVX1mHigh52wk([out, retval] FLOAT *pVal);<br>Highest value of 1m IV Index for the last year |
| IVX1mLow52wk | HRESULT IVX1mLow52wk([out, retval] FLOAT *pVal);<br>Lowest value of 1m IV Index for the last year |
| HV1m | HRESULT HV1m([out, retval] FLOAT *pVal);<br>Historical Volatility (HV) 1m as of yesterday |
| HV1mYest | HRESULT HV1mYest([out, retval] FLOAT *pVal);<br>Day before yesterdays value of HV 1m |
| HV1mHigh52wk | HRESULT HV1mHigh52wk([out, retval] FLOAT *pVal);<br>Highest value of HV 1m for the last year |
| HV1mLow52wk | HRESULT HV1mLow52wk([out, retval] FLOAT *pVal);<br>Lowest value of HV 1m for the last year |
| OptionVolumeCall | HRESULT OptionVolumeCall([out, retval] long *pVal);<br>Total Calls Volume |
| OptionVolumePut | HRESULT OptionVolumePut([out, retval] long *pVal);<br>Total Puts Volume |
| OptionVolumeAvg | HRESULT OptionVolumeAvg([out, retval] FLOAT *pVal);<br>One month average of total Calls and Puts Volume |
| OutstandingShares | HRESULT OutstandingShares([out, retval] FLOAT *pVal);<br>Outstanding Shares |
| DivFreq | HRESULT DivFreq([out, retval] long *pVal)<br>Dividend frequency code:<br>4 - Quarterly<br>2 - Semi-annualy<br>1 - Annualy<br>0 – No dividends |
| DivLastDate | HRESULT DivLastDate([out, retval] DATE *pVal)<br>Last dividend date |
| DivAmount | HRESULT DivAmount([out, retval] FLOAT *pVal)<br>Dividend amount or yield |
| Currency | HRESULT Currency([out, retval] BSTR *pVal)<br>Currency |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next elemnt in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal)<br>Find symbol in collection |

*Example:*

```
Dim stck As StocksEnt
Set stck = dds.GetStocksAll
Do
Debug.Print  stck.Symbol & " " & stck.LastPrice
```

```
Loop While stck.Next
```

# 5 IOptionsEnt interface

IOptionsEnt interface provides a set of methods to iterate over Options dataset collection and access each item properties.

| | |
|---|---|
| Symbol | HRESULT StockSymbol([out, retval] BSTR*pVal);<br>Stock symbol |
| StockSymbol | HRESULT StockSymbol([out, retval] BSTR *pVal)<br>Underlying symbol |
| OptionStyle | HRESULT OptionStyle([out, retval] OptionStyleEnum *pVal)<br>enum OptionStyleEnum<br>{<br>enAmerican = 0,<br>enEuropean = 1<br>} OptionStyleEnum;<br>Option style American or European |
| BidPrice | HRESULT BidPrice([out, retval] FLOAT *pVal)<br>Bid Price |
| BidExchange | HRESULT BidExchange([out, retval] BSTR *pVal)<br>Exchange of bid price |
| BidDate | HRESULT BidDate([out, retval] DATE *pVal)<br>Time or date of bid price |
| BidSize | HRESULT BidSize([out, retval] LONG *pVal)<br>Bid size |
| AskPrice | HRESULT AskPrice([out, retval] FLOAT *pVal)<br>Ask Price |
| AskExchange | HRESULT AskExchange([out, retval] BSTR *pVal)<br>Exchange of ask price |
| AskDate | HRESULT AskDate([out, retval] DATE *pVal)<br>Time or date of ask price |
| AskSize | HRESULT AskSize([out, retval] LONG *pVal)<br>Ask size |
| StrikePrice | HRESULT StrikePrice([out, retval] float *pVal);<br>Strike price |
| ExpiratioDate | HRESULT ExpirationDate([out, retval] DATE *pVal);<br>Expiration date |
| IV | HRESULT IV([out, retval] float *pVal);<br>Implied Volatility (interpolated value) |
| PreIV | HRESULT IV([out, retval] float *pVal);<br>Implied Volatility (not interpolated value) |
| Delta | HRESULT Delta([out, retval] float *pVal);<br>Delta |
| Gamma | HRESULT Gamma([out, retval] float *pVal);<br>Gamma |
| Vega | HRESULT Vega([out, retval] float *pVal);<br>Vega |
| Theta | HRESULT Theta([out, retval] float *pVal); |

| | |
|---|---|
| | Theta |
| Rho | HRESULT Rho([out, retval] float *pVal);<br>Rho |
| Volume | HRESULT Volume([out, retval] long *pVal);<br>Volume |
| OpenInterest | HRESULT OpenInterest([out, retval] long *pVal);<br>Open Interest |
| GetOptionData | HRESULT GetOptionData([out] BSTR* Symbol, [out] BSTR* StockSymbol, [out] OptionTypeEnum* OptionType, [out] float* StrikePrice, [out] DATE* ExpirationDate, [out] float* IV, [out] float* Bid, [out] float* Ask, [out] float* Delta, [out] float* Vega, [out] float* Gamma,<br>[out] float* Theta, [out] float* Rho, [out] long* Volume);<br>Get all Options properties in a single request. |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next elemnt in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal) |

*Example:*

```
Dim optn As IOptionsEnt
Symbol = "MSFT"
Set optn = dds.GetOptionsByStock(Symbol)
Do
        Debug.Print  optn.Symbol & " " & optn.IV
Loop While optn.Next
```

# 6 IVIndexesEnt interface

IVIndexes interface provides a set of methods to iterate over data collection and access each item properties.

| | |
|---|---|
| StockSymbol | HRESULT StockSymbol([out, retval] BSTR *pVal)<br>Stock Symbol |
| Term | HRESULT Term([out, retval] short *pVal)<br>IVX term (days) |
| IVIndexCall | HRESULT IVIndexCall([out, retval] float *pVal)<br>IVIndexCall – IV Index for Call options |
| IVIndexPut | HRESULT IVIndexPut([out, retval] float *pVal)<br>IVIndexPut – IV Index for Put options |
| GetIVIndexData | HRESULT GetIVIndexData([out] BSTR* StockSymbol, [out] short* Term, [out] float* IVIndexCall, [out] float* IVIndexPut)<br>Get all IVIndex properties in a single request. |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next element in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal) |

*Example:*

```
Dim ivxs As IVIndexes
Symbol = “MSFT”
Set ivxs = dds.GetIVIndexesByStock(Symbol)
Do
   Debug.Print  ivxs.Symbol & “ ” & ivxs. Term & “ ” & ivxs. IVIndexCall
Loop While ivxs.Next
```

# 7 IVSurfacesEnt interface

IVSurfaces interface provides a set of methods to iterate over data collection and access each item properties.

.

| | |
|---|---|
| StockSymbol | HRESULT StockSymbol([out, retval] BSTR *pVal)<br>Stock symbol |
| Term | HRESULT Term([out, retval] short *pVal)<br>IV Surface term (days) |
| VirtualStrike | VirtualStrike([out, retval] float *pVal) |
| Otm | HRESULT Otm([out, retval] short *pVal);<br>OTM value |
| Type | HRESULT Type([out, retval] OptionTypeEnum *pVal)<br>Type (enCall or enPut) |
| IV | HRESULT IV([out, retval] float *pVal)<br>Impled Volatility |
| Delta | HRESULT Delta([out, retval] float *pVal)<br>Delta |
| GetIVSurfaceData | HRESULT GetIVSurfaceData([out] BSTR* StockSymbol, [out] short* Term, [out] float* VirtualStrike, [out] short* Otm, [out] float* IV, [out] float* Delta, [out] OptionTypeEnum* Type)<br>Get all IVSurfaces properties in a single request. |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next elemnt in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal) |

*Example:*

```
Dim ivs As IVSurfaces
Symbol = “MSFT”
Set ivs = dds.GetIVSurfacesByStock(Symbol)
Do
        Debug.Print  ivs. StockSymbol & “ ” & ivs. Term & “ ” & ivs.Otm & “ ” & ivs. Type & “ ” & ivs.IV
Loop While ivs.Next
```

# 8 HVEnt interface

HVEnt interface provides a set of methods to iterate over data collection and access each item properties.

| | |
|---|---|
| Symbol | HRESULT Symbol([out, retval] BSTR *pVal);<br>Security symbol |
| Period | HRESULT Period([out, retval] LONG *pVal)<br>Period |
| HV | HRESULT HV([out, retval] FLOAT *pVal);<br>Historical Volatility value |
| GetHVData | HRESULT GetHVData([out] BSTR* Symbol,<br>[out] SHORT* Period, [out] FLOAT* HV);<br>Return all data available |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to next element |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL *bVal)<br>Find record set by currency symbol |

*Example:*

```
Dim hvr As HVEnt
Symbol = “MSFT”
Set hvr = dds.GetHV(Symbol)
Do
        Debug.Print  hvr.Period & “ ” & hvr.HV & vbCRLF
Loop While ivs.Next
```

# 9 InterestRateEnt interface

InterestRateEnt interface provides a set of methods to iterate over data collection and access each item properties.

| | |
|---|---|
| Currency | HRESULT Currency([out, retval] BSTR *pVal)<br>Currency symbol |
| Period | HRESULT Period([out, retval] LONG *pVal)<br>Period |
| Rate | HRESULT Rate([out, retval] FLOAT *pVal)<br>Interest Rate value |
| GetInterestRateData | HRESULT GetInterestRateData([out] BSTR* Currency,[out] LONG* Period, [out] FLOAT* Rate);<br>Return all data available |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to next element |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL *bVal)<br>Find record set by currency symbol |

*Example:*

```
Dim ivs As InterestRateEnt
Symbol = “USD”
Set ir = dds.GetIRByCurrency(Symbol)
Do
        Debug.Print  ir.Period & “ ” & ir.Rate & vbCRLF
Loop While ivs.Next
```

# 10 IFuturesEnt interface

IFuturesEnt interface provides a set of methods to iterate over Futures collection and access each item properties

| | |
|---|---|
| Symbol | HRESULT Symbol([out, retval] BSTR *pVal)<br>Futures symbol |
| RootSymbol | HRESULT RootSymbol([out, retval] BSTR *pVal);<br>Futures root symbol |
| LastPrice | Last price<br>HRESULT LastPrice([out, retval] FLOAT *pVal); |
| LastExchange | Exchange of last price<br>HRESULT LastExchange([out, retval] BSTR *pVal); |
| LastDate | Date and time of last price<br>HRESULT LastDate([out, retval] DATE *pVal); |
| LastSize | Last size<br>HRESULT LastSize([out, retval] LONG *pVal); |
| BidPrice | Bid price<br>HRESULT BidPrice([out, retval] FLOAT *pVal); |
| BidExchange | Exchange of bid price<br>HRESULT BidExchange([out, retval] BSTR *pVal); |
| BidDate | Date and time of bid price<br>HRESULT BidDate([out, retval] DATE *pVal); |
| BidSize | Bid size<br>HRESULT BidSize([out, retval] LONG *pVal); |
| AskPrice | Ask price<br>HRESULT AskPrice([out, retval] FLOAT *pVal); |
| AskExchange | Exchange of ask price<br>HRESULT AskExchange([out, retval] BSTR *pVal); |
| AskDate | Date and time of ask date<br>HRESULT AskDate([out, retval] DATE *pVal); |
| AskSize | Ask size<br>HRESULT AskSize([out, retval] LONG *pVal); |
| Volume | Volume<br>HRESULT Volume([out, retval] long *pVal); |
| Next | Move to the next elemnt in a collection. Last element will return VARIANT_FALSE as bVal value<br>HRESULT Next([out,retval] VARIANT_BOOL* bVal); |
| Find | Find symbol in collection<br>HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal); |
| GetFutureData | Futures properties in single request<br>HRESULT GetFutureData([out] BSTR* Symbol, [out] BSTR* RootSymbol, [out] FLOAT* LastPrice, [out] BSTR* LastExchange, [out] DATE* LastDate, [out] LONG* LastSize, [out] FLOAT* BidPrice, [out] BSTR* BidExchange, [out] DATE* BidDate, [out] LONG* BidSize, [out] FLOAT* AskPrice, [out] BSTR* AskExchange, [out] DATE* AskDate, [out] LONG* AskSize, [out] LONG* Volume); |

*Example:*

```
Dim futures As FuturesEnt
Set futures = dds.GetFuturesBySymbol(symbol)
Do
        Debug.Print  futures.Symbol & " " & futures.LastPrice
Loop While futures.Next
```

# 11 IFutOptionsEnt interface

IFutOptionsEnt interface provides a set of methods to iterate over futures options collection and access each item properties

| | |
|---|---|
| Symbol | HRESULT StockSymbol([out, retval] BSTR*pVal);<br>Stock symbol |
| FuturesSymbol | HRESULT FuturesSymbol ([out, retval] BSTR*pVal)<br>Underlying symbol |
| OptionStyle | HRESULT OptionStyle([out, retval] OptionStyleEnum *pVal)<br>enum OptionStyleEnum<br>{<br>enAmerican = 0,<br>enEuropean = 1<br>} OptionStyleEnum;<br>Option style American or European |
| BidPrice | HRESULT BidPrice([out, retval] FLOAT *pVal)<br>Bid Price |
| BidExchange | HRESULT BidExchange([out, retval] BSTR *pVal)<br>Exchange of bid price |
| BidDate | HRESULT BidDate([out, retval] DATE *pVal)<br>Time or date of bid price |
| BidSize | HRESULT BidSize([out, retval] LONG *pVal)<br>Bid size |
| AskPrice | HRESULT AskPrice([out, retval] FLOAT *pVal)<br>Ask Price |
| AskExchange | HRESULT AskExchange([out, retval] BSTR *pVal)<br>Exchange of ask price |
| AskDate | HRESULT AskDate([out, retval] DATE *pVal)<br>Time or date of ask price |
| AskSize | HRESULT AskSize([out, retval] LONG *pVal)<br>Ask size |
| StrikePrice | HRESULT StrikePrice([out, retval] float *pVal);<br>Strike price |
| ExpiratioDate | HRESULT ExpirationDate([out, retval] DATE *pVal);<br>Expiration date |
| IV | HRESULT IV([out, retval] float *pVal);<br>Implied Volatility (interpolated value) |
| PreIV | HRESULT IV([out, retval] float *pVal);<br>Implied Volatility (not interpolated value) |
| Delta | HRESULT Delta([out, retval] float *pVal);<br>Delta |
| Gamma | HRESULT Gamma([out, retval] float *pVal);<br>Gamma |
| Vega | HRESULT Vega([out, retval] float *pVal);<br>Vega |
| Theta | HRESULT Theta([out, retval] float *pVal); |

| | |
|---|---|
| | Theta |
| Rho | HRESULT Rho([out, retval] float *pVal);<br>Rho |
| Volume | HRESULT Volume([out, retval] long *pVal);<br>Volume |
| OpenInterest | HRESULT OpenInterest([out, retval] long *pVal);<br>Open Interest |
| GetOptionData | HRESULT GetOptionData([out] BSTR* Symbol, [out] BSTR* StockSymbol, [out] OptionTypeEnum* OptionType, [out] float* StrikePrice, [out] DATE* ExpirationDate, [out] float* IV, [out] float* Bid, [out] float* Ask, [out] float* Delta, [out] float* Vega, [out] float* Gamma,<br>[out] float* Theta, [out] float* Rho, [out] long* Volume);<br>Get all Options properties in a single request. |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next elemnt in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal) |

*Example:*

```
Dim optn As FutOptionsEnt
Set optn = dds.GetFutOptionsByFutureSymbol(symbol)
Do
        Debug.Print  optn.Symbol & " " & optn.IV
Loop While optn.Next
```

# 12 IFutIVIndexesEnt interface

IFutIVIndexesEnt interface provides a set of methods to iterate over data collection and access each item properties.

| | |
|---|---|
| Underlying Symbol | HRESULT UnderlyingSymbol([out, retval] BSTR *pVal)<br>Stock Symbol |
| Term | HRESULT Term([out, retval] short *pVal)<br>IVX term (days) |
| IVIndexCall | HRESULT IVIndexCall([out, retval] float *pVal)<br>IVIndexCall – IV Index for Call options |
| IVIndexPut | HRESULT IVIndexPut([out, retval] float *pVal)<br>IVIndexPut – IV Index for Put options |
| GetIVIndexData | HRESULT GetIVIndexData([out] BSTR* Symbol, [out] short* Term, [out] float* IVIndexCall, [out] float* IVIndexPut)<br>Get all IVIndex properties in a single request. |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next element in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal) |

# 13 IFutIVSurfacesEnt interface

IFutIVSurfacesEnt interface provides a set of methods to iterate over data collection and access each item properties.

| | |
|---|---|
| UnderlyingSymbol | HRESULT UnderlyingSymbol([out, retval] BSTR *pVal)<br>Stock symbol |
| Term | HRESULT Term([out, retval] short *pVal)<br>IV Surface term (days) |
| VirtualStrike | VirtualStrike([out, retval] float *pVal) |
| Otm | HRESULT Otm([out, retval] short *pVal); |

| | |
|---|---|
| | OTM value |
| Type | HRESULT Type([out, retval]      OptionTypeEnum *pVal)<br>Type (enCall or enPut) |
| IV | HRESULT IV([out, retval] float *pVal)<br>Impled Volatility |
| Delta | HRESULT Delta([out, retval] float *pVal)<br>Delta |
| GetIVSurfaceData | HRESULT GetIVSurfaceData([out] BSTR* Symbol, [out] short* Term, [out] float* VirtualStrike, [out] short* Otm, [out] float* IV, [out] float* Delta, [out] OptionTypeEnum* Type)<br>Get all IVSurfaces properties in a single request. |
| Next | HRESULT Next([out,retval] VARIANT_BOOL* bVal)<br>Move to the next elemnt in a collection. Last element will return VARIANT_FALSE as bVal value |
| Find | HRESULT Find([in] BSTR Symbol, [out,retval] VARIANT_BOOL* bVal) |

# 14 Subscription mechanism

Data update subscription mechanism provides data update using event-driven model. You should enter security symbol, symbol type and dataset type (see Appendix 1).

| | |
|---|---|
| Subscribe | HRESULT Subscribe([in] BSTR Symbol, [in] SymbolTypeEnum SymbType, [in] DataTypeEnum Type)<br><br>Subscribe to received updates for the specified symbol |
| UnSubscribe | HRESULT UnSubscribe([in] BSTR Symbol, [in] SymbolTypeEnum SymbType, [in] DataTypeEnum Type);<br><br>Cancel receiving updates for the specified symbol |
| SubscribeAll | HRESULT SubscribeAll([in] DataTypeEnum Type);<br><br>Subscribe to receive updates for all symbols available for current user |
| UnSubscribeAll | HRESULT UnSubscribeAll([in] DataTypeEnum Type);<br><br>Cancel receiving updates for all symbols available to the current user |
| SubscribeGroup | HRESULT  SubscribeGroup(VARIANT symbolArray, SymbolTypeEnum SymbType, DataTypeEnum Type)<br><br>Subscribe to receive updates for a group of symbols |
| UnSubscribeGroup | HRESULT  UnSubscribeGroup(VARIANT symbolArray, SymbolTypeEnum SymbType, DataTypeEnum Type)<br><br>Cancel receiving updates for a group of symbols |
| IsSymbolSubscribed | HRESULT IsSymbolSubscribed([in] BSTR Symbol, [in] SymbolTypeEnum SymbType, [in] DataTypeEnum Type, [out,retval] VARIANT_BOOL* bVal);<br><br>Check if the symbol subscription is activated – “True” returned if succeed |

*Example:*

```
Dim WithEvents dds as IVDDSEnt
…
dds.Subscribe "MSFT", enEquity , enStockData
…
```

```
if IsSymbolSubscribed("MSFT", enEquity, enStockData) = true then
        dds.UnSubscribe "MSFT", enEquity, enStockData
```

To handle data events you should declare call back methods for each data type.

| | |
|---|---|
| OnStocksChange | HRESULT OnStocksChange([in] IStocksEnt* pData);<br>Stock or Index quote changed |
| OnOptionsChange | HRESULT OnOptionsChange([in] IOptionsEnt* pData);<br>Options data changed |
| OnIVIndexChange | HRESULT OnIVIndexChange([in] IIVIndexesEnt* pData);<br>IVIndex data changed |
| OnIVSurfaceChange | HRESULT OnIVSurfaceChange([in] IIVSurfacesEnt* pData);<br>IVSurface data changed |
| OnFuturesChange | HRESULT OnFuturesChange([in] IFuturesEnt* pData);<br>Futures data changed |
| OnFutOptionsChange | HRESULT OnFutOptionsChange([in] IFutOptionsEnt* pData);<br>Futures option data changed |
| OnFutIVIndexChange | HRESULT OnFutIVIndexChange([in] IFutIVIndexesEnt* pData);<br>Futures IV Index data changed |
| OnFutIVSurfaceChange | HRESULT OnFutIVSurfaceChange([in] IFutIVSurfacesEnt* pData);<br>Futures IV Surface data changed |
| OnError | HRESULT OnError([in] ErrorNumberEnum ErrorNumber, [in] BSTR Description);<br>Internal error occurred while sending or receiving data |

*Example:*

```
Private Sub dds_OnStocksChange(ByVal stckData As StocksEnt)
     Do
         Debug.Print  stck.Symbol & " " & stck.LastPrice
     Loop While stckData.Next
End Sub
```

# 15 DDE Server

General format for DDE requests is “=Service|Topic!Item” where:

- Service – service name (IVDDE)
- Topic – Stock or Option symbol
- Item – data type

See complete description below.

## 15.1 Requesting Stock data

| | |
|---|---|
| Service | IVDDE |
| Topic | Stock Symbol |
| Item | • Price<br>• StckVolume<br>• StckLastTime<br>• StckLastExch<br>• StckLastSize<br>• StckBid<br>• StckAsk<br>• StckBidTime<br>• StckAskTime<br>• StckBidExch<br>• StckAskExch<br>• StckBidSize<br>• StckAskSize<br>• DivDate<br>• DivAmount<br>• DivFreq<br>• StckPriceClose<br>• StckPriceHigh52wk<br>• StckPriceLow52wk<br>• Ivx1m<br>• Ivx1mYest<br>• Ivx1mHigh52wk<br>• Ivx1mLow52wk<br>• Hv1m<br>• Hv1mYest<br>• Hv1mHigh52wk<br>• Hv1mLow52wk<br>• StckOptnVolumeCall<br>• StckOptnVolumePut<br>• StckOptnVolumeAvg<br>• OutstandingShares |

*Examples:*

=IVDDE|MSFT!Price – last price for MSFT

=IVDDE|DELL!StckVolume – today volume for DELL

### 15.2 Requesting Options data

| | |
|---|---|
| Service | IVDDE |
| Topic | Option symbol |
| Item | • StockSymbol<br>• Strike<br>• Type<br>• ExpDate<br>• IV<br>• OptnBidTime<br>• OptnAskTime<br>• OptnBidExch<br>• OptnAskExch<br>• OptnBidSize<br>• OptnAskSize<br>• PreIv<br>• Bid<br>• Ask<br>• GrkDelta<br>• GrkVega<br>• GrkGamma<br>• GrkTheta<br>• GrkRho<br>• OptnVolume<br>• OpenInterest |

*Examples:*
=IVDDE|MQFAO!StockSymbol – underlying for option contract MQFAO
=IVDDE|MQFAO!Strike –MQFAO strike
=IVDDE|MQFAO!Type – MQFAO type (Call or Put)
=IVDDE|MQFAO!ExpDate – MQFAO expiration date
=IVDDE|MQFAO!IV –MQFAO implied volatility
=IVDDE|MQFAO!GrkVega – MQFAO Vega
=IVDDE|MQFAO!GrkRho –MQFAO Rho
=IVDDE|MQFAO!OptnVolume – MQFAO daily volume

### 15.3 Requesting IVIndex data

| | |
|---|---|
| Service | IVDDE |
| Topic | Stock symbol |
| Item | IVX[Type]_T[Term], where<br>1. [Type] – option type ( “Call” or “Put”)<br>2. [Term] – term (days) |

*Examples:*
=IVDDE|MSFT!IVXCall_T30 – 30-day Calls IVIndex for MSFT
=IVDDE|MSFT!IVXPut_T90 – 90-day Put IVIndex for MSFT

### 15.4 Requesting IVSurface data

| | |
|---|---|
| Service | IVDDE |
| Topic | Stock symbol |
| Item | IVS[Type]_T[Term]O[OTM]_[Value], where<br>1. [Type] – surface type ("Call", "Put")<br>2. [Term] – term<br>3. [OTM] – OTM<br>4. [Value] – one of these: "VStrike" (virtual strike), "IV", "Delta" |

*Examples:*

=IVDDE|MSFT!IVSCALL_T30O50_IV – 30-day IVSurface implied volatility for 50% Calls OTM
=IVDDE|MSFT!IVSPUT_T90O0_VSTRIKE – virtual strike for 90-day 0 OTM Puts for stock MSFT

### 15.5 Requesting Futures data

| | |
|---|---|
| Service | IVDDE |
| Topic | Futures Symbol |
| Item | • FutPrice<br>• FutVolume<br>• FutLastTime<br>• FutLastExch<br>• FutLastSize<br>• FutBid<br>• FutAsk<br>• FutBidTime<br>• FutAskTime<br>• FutBidExch<br>• FutAskExch<br>• FutBidSize<br>• FutAskSize |

*Examples:*

=IVDDE|BO\10N!FutPrice – last price for BO\10N
=IVDDE|BO\10N!FutVolume – today volume for BO\10N

### 15.6 Requesting Futures Options data

| | |
|---|---|
| Service | IVDDE |
| Topic | Futures option symbol |
| Item | • FutureSymbol<br>• FutOptStrike<br>• FutOptExpDate<br>• FutOptIV<br>• FutOptnBidTime<br>• FutOptnAskTime<br>• FutOptnBidExch<br>• FutOptnAskExch |

| | |
|---|---|
| | • FutOptnBidSize<br>• FutOptnAskSize<br>• FutOptPreIv<br>• FutOptBid<br>• FutOptAsk<br>• FutOptGrkDelta<br>• FutOptGrkVega<br>• FutOptGrkGamma<br>• FutOptGrkTheta<br>• FutOptGrkRho<br>• FutOptnVolume |

*Examples:*
=IVDDE|BO\0N\17500C!FutureSymbol – underlying for option contract BO\0N\17500C
=IVDDE|BO\0N\17500C!FutOptStrike – BO\0N\17500C strike

### 15.7 Requesting Futures IVIndex data

| | |
|---|---|
| Service | IVDDE |
| Topic | Futures underlying symbol |
| Item | IVFX[Type]_T[Term], where<br>3. [Type] – option type ( “Call” or “Put”)<br>4. [Term] – term (days) |

*Examples:*
=IVDDE|BO!IVFX Call_T30 – 30-day Calls IVIndex for BO
=IVDDE|BO!IVFX Put_T90 – 90-day Put IVIndex for BO

### 15.8 Requesting Futures IVSurface data

| | |
|---|---|
| Service | IVDDE |
| Topic | Futures underlying symbol |
| Item | IVFS[Type]_T[Term]O[OTM]_[Value], where<br>5. [Type] – surface type (“Call”, “Put”)<br>6. [Term] – term<br>7. [OTM] – OTM<br>8. [Value] – one of these: “VStrike” (virtual strike), “IV”, “Delta” |

*Examples:*

=IVDDE|BO!IVFSCALL_T30O50_IV – 30-day IVSurface implied volatility for 50% Calls OTM
=IVDDE|BO!IVFSPUT_T90O0_VSTRIKE – virtual strike for 90-day 0 OTM Puts.

# Appendix 1

```
enum DataTypeEnum
{
        enStockData              = 0,
        enOptionData             = 1,
        enIVIndexData            = 2,
        enIVSurfaceData          = 3,
        enMarketStructure        = 4
} DataTypeEnum;

enum OptionTypeEnum
{
        enCall                   = 0,
        enPut                    = 1
} OptionTypeEnum;

enum SymbolTypeEnum
{
                enEquity         = 1,
                enOption         = 2
} SymbolTypeEnum;
```