

Introduction to BQL

Using Bloomberg Query Language in Excel

BQL stands for Bloomberg Query Language, which is the language used to perform analytics with and retrieve data from the Bloomberg Database. New functions have been added to the Bloomberg Excel Add-in that can be used to retrieve data using BQL.

Like the Excel functions BDH and BDP, it's possible to retrieve both current and historical data using BQL. Uniquely, BQL is also capable of performing complex tasks like screening, interval calculations, time series calculations, and supports the declaration of custom fields.

A useful feature of BQL is the ability to manipulate data before retrieval. This includes arithmetic operations, rounding, finding the highest number in a set, etc. The advantage of performing calculations within BQL rather than in Excel is that it retrieves fewer pieces of information through the API and is therefore easier to process.

BQL Basics

There are two ways to perform a BQL query in Excel: the functions BQL and BQL.Query. Both functions are equally capable, but use different syntaxes. There are three types of inputs used when querying in BQL: a *Universe*, an *Expression*, and *Parameters*.

The universe is the subject of the query and is what the queried data describes. A universe can be a single security, a list of securities, an index, etc. For example, to query the most recent price of IBM, the universe would be "IBM US Equity". To query the most recent prices of both IBM and Apple, the universe would be "['IBM US Equity', 'AAPL US Equity']".

An expression, also known as a *field*, is the type of data that will be retrieved. For example, when querying the most recent price of IBM, the expression used is "PX_LAST". An expression can contain a single field, a combination of fields, and operations applied to the fields. For a complete list of fields available for use, see FLDS <GO>.

A parameter is an optional addition to a query that can further specify an expression. While querying the closing price of IBM on 12/13/2017, the date would be a parameter. In this case, the proper notation for this parameter would be "DATES=2017-12-13".

BQL.Dates, BQL.Params, BQL.Expr, and BQL.List

Notably, BQL functions follow a different dating convention than Excel or other Bloomberg functions. For example, to be usable in BQL, the date 7/25/2017 must be converted into the string "2017-07-25". This conversion may be completed automatically by inputting a regular date into the BQL.Date function.

The other functions, like BQL.Date, are useful for converting cell references in Excel into usable inputs for the BQL and BQL.Query functions. BQL.Params is used to convert inputs into a parameter, BQL.Expr is used to convert inputs into an expression, and BQL.List is used to convert a cell referenced range of securities into a list. An example of each function and its output are in the following table.

Function	Input	Output
BQL.Params	BQL.Params("DATES","2017-12-13")	DATES=2017-12-13
BQL.Expr	BQL.Expr("DATES=RANGE","2017-01-01","2017-12-31")	DATES=RANGE(2017-01-01,2017-12-31)
BQL.List	BQL.List("IBM US Equity","AAPL US Equity",	['IBM US EQUITY','AAPL US EQUITY']

Using BQL

BQL follows the form (“Universe”, “Expression”, “Parameters”). Below is an example which retrieves the closing price of IBM on 12/13/2017.

```
=BQL("IBM US Equity","PX_LAST","DATES=2017-12-13")
```

BDH rounds its digits to four decimal places, but BQL rounds to the thirteenth. It’s possible to reconcile this difference by using the “round” function in the expression. An example is below.

```
=BQL("IBM US Equity","round(PX_LAST,4)","DATES=2017-12-13")
```

Instead of manually inputting a string, a cell reference could be used instead. Assume the date was entered in cell A1.

```
=BQL("IBM US Equity","round(PX_LAST,4)",BQL.Params("DATES",A1))
```

BQL can also be used to perform interval calculations. In this example, the daily closing prices of IBM are retrieved and then averaged using the “avg” function. Only the final result of the calculation is returned to Excel.

```
=BQL("IBM US Equity","avg(PX_LAST)","DATES=RANGE(2017-01-01,2017-12-31)")
```

Combinations of functions can be used in an expression. In this example, the closing prices of IBM are retrieved, #N/A’s are excluded, then the prices are rounded to four decimals, and finally the prices are averaged.

```
=BQL("IBM US Equity","avg(round(dropna(PX_LAST),4))","DATES=RANGE(2017-01-01,2017-12-31)")
```

As mentioned, BQL supports the declaration of custom fields. Below is an example that defines “#AvgPX” as the average closing prices in January of 2017, then uses it in the expression to retrieve the “#AvgPX” of IBM.

```
=BQL("IBM US Equity","avg(PX_LAST(START=2017-01-01,END=2017-01-31)) as #AvgPX","#AvgPX")
```

Using BQL.Query

BQL.Query may contain the clauses “get”, “for”, “with”, and “let”. The “get” is used for expressions, the “for” clause specifies the universe, the “with” clause contains the parameters, and the “let” clause can be used to define custom fields.

Below are the first five examples used in the BQL section, but accessed through BQL.Query.

1. Price Quote

```
=BQL.Query("get(PX_LAST) for('IBM US Equity') with(DATES=2017-12-13)")
```

2. Rounded Price Quote

```
=BQL.Query("get(round(px_last,4)) for('IBM US EQUITY') with(DATES=2017-12-13)")
```

3. Rounded Price Quote with Cell-Referenced Date

```
=BQL.Query("get(round(px_last,4)) for('IBM US EQUITY') with("&BQL.Params("DATES",A1)&"))
```

4. Average Price

```
=BQL.Query("get(avg(PX_LAST)) for('IBM US EQUITY') with(DATES=RANGE(2017-01-01, 2017-12-31)))")
```

5. Average Rounded Price

```
=BQL.Query("get(avg(round(dropna(PX_LAST),4))) for('IBM US EQUITY')  
with(DATES=RANGE(2017-01-01,2017-12-31)))")
```

Here is an example using the “let” clause. This example defines the average closing price in January of 2017 as “#StartAvg”, the average closing price in December of 2017 as “#EndAvg”, and “#StartAvg” divided by “#EndAvg” minus one as “#Return”. The query then retrieves “#Return” for IBM.

```
=BQL.Query("let(#StartAvg = avg(PX_LAST(START=2017-01-01,END=2017-01-31));  
#EndAvg = avg(PX_LAST(START=2017-12-01,END=2017-12-31)); #Return = #EndAvg / #StartAvg - 1;  
get(#Return) for('IBM US EQUITY'))")
```

It’s worth noting that there are multiple syntactically correct ways of writing the same query. For example, it doesn’t matter what order the clauses appear within the query. Another example: instead of specifying DATES=RANGE, one could specify both a START_DATE and END_DATE to produce the same result.

August 2019

Resources

Further documentation on BQL is available with HELP BQLX <GO>.

See *BQL Bloomberg Fundamentals: Data Parameters & Associative Columns* for help with parameters.

See accompanying BQL_Excel_Examples workbook at www.fintools.com.
