

F.ExtractFormulaHistory

The ExtractFormulaHistory function is used for extracting one or more items for one security, an index or a list of securities over time. The function is using the FactSet Query Language (FQL), which is a proprietary data retrieval language used to access a time-series of FactSet data.

1. FactSet Query Language

Some advantages of FQL include:

- + **The ability to specify dates for any database using the same formats.**
With FQL, date formats are flexible. You can use a number of consistent date formats (defined by FQL) for all databases which makes using and combining data from different databases easier than ever.
- + **The ability to iterate items, formulas, and functions at any frequency.**
With FQL, you can iterate items, formulas, and functions at any frequency. For example, you can request a series of weekly price to earnings ratios.

To request a time-series of data, a start date, end date and frequency needs to be specified. If a date is not specified, data is returned from the most recent time period. The dates can be designated as absolute dates or relative dates.

1.1 FQL Date Format - Absolute Dates

Absolute dates indicate a specific day, month-end, fiscal quarter-end, calendar quarter-end, fiscal year-end, or calendar year-end as depicted in the examples below:

- + A day: MM/DD/YYYY (e.g. 7/11/1999)
Note: DD/MM/YYYY is not a valid date format
- + A month-end: MM/YYYY (e.g. 6/1999)
- + A fiscal quarter-end: YY/FQ or YYYY/FQ (e.g. 1999/1F, 2000/3F, 2001/2F)
- + A calendar quarter-end: YY/CQ or YYYY/CQ (e.g. 1999/1C, 00/3C, 2001/1C)
- + A fiscal year-end: YY or YYYY (e.g. 2000, 01, 1999)

1.2 FQL Date Format - Relative Dates

Relative dates represent a date relative to the most recently updated period. For example, 0 (zero) represents the most recently updated period; -1 represents the time period prior to the most recently updated.

The zero date is determined by the default time period or the natural frequency of the data being requested. Zero (0) when used with monthly data indicates the most recent month end. Negative one (-1) when used with annual data indicates one fiscal year prior to the most recently updated fiscal year.

List of Relative Date Arguments:

D	0D is the most recent trading day, -1D is one trading day prior.
WE	0WE is the most recent trading week end, -1AW is the one actual week (7 days) prior to the most recent.
W	0W is the last day of the most recent trading week (usually Friday), -1W is the last trading day of the prior week.
AM	0AM is the most recent trading day, -1AM is the same day, one actual month prior.
M	0M is the last trading day of the most recent month, -1M is the last trading day of the prior month.
AQ	0AQ is the most recent trading day, -1AQ is the same day 3 months prior
Q	0Q is the last trading day of the company's most recent fiscal quarter, -1Q is the last day of the prior fiscal quarter.
CQ	0CQ is the last trading day of the most recent calendar quarter (March, June, September, or December), -1CQ is the last trading day of the prior calendar quarter.
AY	0AY is the most recent trading day, -1AY is one actual year (365 days) prior
Y	0Y is the last trading day of the company's most recent fiscal year, -1Y is the last trading day of the prior fiscal year.
CY	0CY is the last trading day of the most recent calendar year (the last trading day in December), -1CY is the last trading day of the prior calendar year.

2. F.ExtractFormulaHistory syntax

The syntax for the F.ExtractFormulaHistory function is:

data = F.ExtractFormulaHistory(ids, items, date, optional arguments)

where,

data	variable name for the data returned
ids	CellString array with a list of one or multiple security identifiers
items	CellString array with a list of one or more FactSet data items in the Screening language
dates	Date range and frequency entered using actual or relative dates. A valid FactSet frequency (e.g. 'd') dates Alternate method of entering dates entered in start:end:freq format. (e.g. '20101215:20110115:d')

Optional arguments,

curr	The currency in which the data is to be returned, using a string with the three character ISO code (e.g. 'USD' or 'EUR').
-------------	---

cal	Calendar setting, arguments include: LOCAL: Uses the local trading calendar for each security. Local exchange holidays will be skipped FIVEDAY: Displays Monday through Friday, regardless of whether there were trading holidays. FIVEDAYEOM: Displays Monday through Friday including a weekend date if it falls on the last day of the month. Where the month-end does not fall on a weekend, the calendar will act just as the standard five-day calendar. SEVENDAY: Displays Monday through Sunday. AAM: For Exchange code uses the calendar of a specific exchange, represented by the exchange code. If there is no calendar available for a specific exchange, the calendar will default to FIVEDAY.
universe	Screening expression to limit the universe
ison	Ison-codes can be used to limit the universe ISON_MSCI_WORLD(0,1) is written as 'ison','msci_world','isonParams','0,1'
isonParams	The arguments within brackets in the ison-code
OFDB	Universe is the constituents of an OFDB file, default directory is Client, if the OFDB is stored in another location the path must be included
OFDBDate	Specific date for the constituents of the OFDB
decimals	Positionally set according to the items in the selection, ie 'decimals',',,3,4,3'
dataType	The optional argument allows users to define a data type for a data item column that is NA for the entire column. This option must be defined for every column/data item requested in the command if it is used at all.
feedback	Setting to control data is not falling forward and display NAs instead of carrying forward values, for those databases that do so (using 'feedback','n').
refresh	This will refresh the connection to FactSet servers to capture the latest database updates. This only needs to be used when a refresh is necessary. It is not recommended to leave this argument in every request made. To use this, the refresh argument should be paired with the value "Y".

Example 1:

In this example extract the last 6 quarters EPS for Exxon Mobile (ticker XOM) using the FQL code FG_EPS.

```
data = F.ExtractFormulaHistory('XOM','FG_EPS(0Q,-5Q,Q)','0Q:-5Q:Q');
```

With the Output:

R:

```
Id Date fg.eps
1 xom 2012-12-31 9.47
2 xom 2013-03-31 9.70
3 xom 2013-06-30 9.82
4 xom 2013-09-30 7.96
5 xom 2013-12-31 7.66
6 xom 2014-03-31 7.37
```

MATLAB:

```
Id: 'xom'
Date: [735234 735324 735415 735507 735599 735689]
fg_eps: [9.4700 9.7000 9.8200 7.9600 7.6600 7.3700]
```

In this example, the date argument is using relative rather than absolute dates. To specify relative dates, enter the number of periods and a period code, such as D for days, W for weeks, or Q for quarters and Y for years. When using relative dates, "0" refers to the most recent time period. Therefore, 0Q refers to the most recent quarter end, while -1Q refers to two quarters ago.

Example 2:

In this example, extract the last 6 periods of pricing and sales data for Microsoft and IBM using the pricing database with the FQL code P_PRICE and the FactSet Fundamentals database for sales data with the FQL code FF_SALES. Both P_PRICE and FF_SALES in this example are used in the items parameter.

```
data = F.ExtractFormulaHistory('MSFT,IBM','P_PRICE(-5,0,Q,USD),FF_SALES(QTR,-5,0,Q,USD)','-5:0:Q');
```

Note: To most efficiently ensure that the dates for the different items (here price and sales) align correctly with the dates field, the dates should be included both in the FQL code and in the dates parameter as specified above.

Example 3:

In this example, extract the price for Google for the date range 12/31/1989 until 12/31/2001 on a monthly frequency. Since there is no available price data for Google starting in 1989, the data would be NA. When using the ExtractFormulaHistory function the data type can be specified for treatment of NA's, for example as a double or integer.

```
data = F.ExtractFormulaHistory('GOOGL','P_PRICE(12/31/1989,12/31/2001,M)',  
'12/31/1989:12/31/2001:M','datatype','DOUBLE');
```

Note: By default, the data type returned is determined by the first value of the items being returned. In the following case the p_price code returned as a character by default because the values for GOOGL are NA (if the request is made for just IBM with the same date range the p_price data is returned as a double since the data is available for IBM). But with the addition of the 'datatype' optional argument, it is possible to specify how the data is returned.

Example 4:

In this example, retrieve the 60 month beta coefficient for Exxon Mobile relative to the S&P 500. The FactSet BETA function measures a security or portfolio's volatility relative to an index. If a security has a beta coefficient greater than one, it is considered more volatile. If a security has a beta coefficient of less than one, its price can be expected to rise and fall more slowly. In the FQL syntax, the BETA function returns the coefficient relative to an index and over any period of time that you specify.

```
R: data = F.ExtractFormulaHistory('XOM','BETA(\SP50\','-0,-59M,M)')  
MATLAB: data = F.ExtractFormulaHistory('XOM','BETA("SP50",-0,-59M,M)')
```

Note: Single quotes in an FQL formula needs to be escaped by a backslash (R) or single quote (MATLAB).

Example 5:

This example is using some of the FQL functions that are available, such as AVG, MAX, MIN, STD, etc. FactSet offers hundreds of functions and a comprehensive alphabetical list of functions is available on Online Assistant page 1410.

Here the average, maximum, minimum, and standard deviation of earnings per share (EPS) values for General Electric are extracted, for the last 6 quarters of history. The EPS values are coming from the FactSet Fundamentals database using the FQL code FF_EPS(QTR,0,-5,Q).

```
data = F.ExtractFormulaHistory('GE','AVG(FF_EPS(QTR,0,-5,Q)) ,MAX(FF_EPS(QTR,0,-5,Q)),MIN(FF_EPS(QTR,0,-5,Q)),STD(FF_EPS(QTR,0,-5,Q))','')
```

Note: The FactSet AVG function returns the average, or mean, value of the data values requested and ignores non-available data over the specified time period. For example, when calculating the average annual sales over the last 10 years, if one out of the 10 years is unavailable, you will still get an average using the 9 years of available data.

The FactSet STD function returns the standard deviation by first finding the average of the data set. Each data item is then subtracted from the average, and the difference is squared. The sum of the squared values is divided by the number of data items minus one. The function returns the square root of that value.

Example 6:

In this example, retrieve the last 2 quarters of EPS and price for the current constituents of the S&P 500 using the FQL codes FF_SALES and P_PRICE and the names of the companies using the FQL code PROPER_NAME. The universe, which is specified as the S&P 500, is based on the FQL code ISON_SP500, returns the current constituents of the S&P 500.

```
data = F.ExtractFormulaHistory('','PROPER_NAME,FF_EPS(QTR,0,-1,Q),P_PRICE(0,-1,Q)','0:-1:Q','ison','SP500');
```

Example 7:

In this example, retrieve proprietary data stored uploaded into FactSet and stored in an Open FactSet Database (OFDB). OFDB is a high-performance multi-dimensional database system used to securely store proprietary numeric and textual data on FactSet.

OFDB is ideal for users who manage large portfolios or maintain extensive historical proprietary databases. OFDB optimizes large, multi-dimensional databases, giving FactSet users highly flexible, factset access to large volumes of complex data that can be used in many different applications. OFDB is based upon Online Analytical Processing technology, which is the basis for multi-dimensional databases.

Here the latest closing price using P_PRICE for the constituents of an OFDB is extracted. The OFDB in this example is saved in the default client directory and is called MyOFDB. To test this example, reference an actual OFDB file that can be accessed using the credentials used.

```
data = F.ExtractFormulaHistory('','P_PRICE(0D)','0','ofdb','MyOFDB');
```

Note: When data is extracted from an OFDB and no directory, such as Client or Personal is specified, the default path is referencing the Client directory within the FactSet workstation.

Note: When using the 'ofdb' parameter, the retrieved constituents are latest available. In order to extract a more dynamic universe of constituents of an OFDB as of a specific date, an 'ofdbDate' argument must be used. The request would thus be set up as:

```
data = F.ExtractFormulaHistory('P_PRICE(0D)', '0', 'ofdb', 'MyOFDB', 'ofdbDate', '20120105')
```

Example 8:

In this example the last year's quarterly Market Value is extracted for a universe defined as the 50 companies with the highest EPS from the selection of US Companies with a Market Value greater than 10 billion US Dollar.

```
R:          data          =          F.ExtractFormulaHistory('P_MARKET_VAL(0,-1Y,Q)', '0:-1Y:Q',
'universe', 'URANKX((P_MARKET_VAL(0,USD)>10000          AND          CONTAINS(P_EXCOUNTRY, \UNITED
STATES\'))=1, FF_EPS(QTR,0))<=50S');
```

```
MATLAB:      data      =      F.ExtractFormulaHistory('P_MARKET_VAL(0,-1Y,Q)', '0:-1Y:Q',
'universe', 'URANKX((P_MARKET_VAL(0,USD)>10000          AND          CONTAINS(P_EXCOUNTRY, "UNITED
STATES"))=1, FF_EPS(QTR,0))<=50S');
```

Note: The universe parameter should only be used for smaller universes (less than 500 securities) when used with the ExtractFormulaHistory factlet.

Example 9:

In this example, for the securities IBM, Microsoft and Nokia, extract the total return from 1 trading day before latest quarterly EPS report date until 1 day after the latest quarterly EPS report date. In this example demonstrating the use of addition, however, subtraction, division and multiplication can be performed as well.

```
data = F.ExtractFormulaHistory('IBM,MSFT,NOK', 'P_TOTAL_RETURN( FF_EPS_RPT_DATE(QTR,0)-
1D, FF_EPS_RPT_DATE(QTR,0)+1D)', '');
```