Custom formulas in PORT<GO>

Bloomberg

Custom formulas allow users to create custom expressions that are derived from standard fields available in PORT. Examples of custom expressions include applying mathematical operations to other PORT fields, using logical operators, incorporating cross-sectional operations and more. In order to create a custom formula, users can run PORT<GO> on the Bloomberg terminal, go to the *Characteristics* tab, right-click on the column header row, select Add/Remove Fields and then click the pencil icon next to the Create formula option under the Custom Formulas section of the Edit Template dialog. In addition to adding custom formulas as fields on Characteristics tab, you can also use custom formulas as basis for custom classifications (custom partitions) in UNCL<GO>, or reference custom formulas in portfolio optimizer.

Custom formulas are constructed by combining standard and custom PORT fields, constants and operators. As an example of a simple formula let's calculate the PE ratio by dividing price by earnings per share. The formula will look like this:
PX_LAST_EOD/ BEST_EPS

Figure 1. Formula Editor in PORT<GO>

In this example PX_LAST_EOD and BEST_EPS are examples of standard PORT fields and '/' is a division operator. PORT fields can be typed directly into the formula using field mnemonics (Example: PX_LAST_EOD). Alternatively, standard PORT fields can be selected from the field picker in the Formula Editor available in PORT. Note that in order to incorporate custom data variables defined in the Custom Data Editor (CDE<GO>) into a formula, users can select the appropriate CDE field from the field picker. Custom formulas that you have created previously, or that have been shared with you could also be used as fields in custom formulas.

In addition to PORT fields and operators, constants can also be used in formulas. Constants can be of the following data types: strings, numbers, or dates. Strings should be enclosed in double quotes. An example of a string is "My string". Numbers can be typed in directly. For decimal fractions make sure to include a leading zero if applicable (Example: 0.25). Dates can be specified as YYYY-MM-DD (Example: 2015-12-31).
If a value for a given field is not applicable, you can refer to it as NA. For example, the

following formula returns delta adjusted exposure for securities for which a delta measure is available and market values for other security types: IF(DELTA != NA,DELTA_ADJ_EXPOSURE, MKT_VAL).

Complex formulas can include optional formula variables. For example, if a user wants to incorporate the custom PE ratio that we defined above into a formula, they can define it as an optional formula variable. Formula variables consist of two parts: a Name and an Expression. The name has a default value of EXPN, where N is a number, but can be customized. In our example, we can rename EXP1 to PE and define the Expression as PX_LAST_EOD/ BEST_EPS. Then, we can use PE as a component of the formula. Standard PORT fields can be further customized by using field overrides. An example of using an override is specifying whether you want to derive the value of a given security from the portfolio or from the benchmark. For example, to derive the market value of the security from the portfolio, write MKT_VAL(PORTFOLIO_ID=CURRENT_POR). On the other hand, if you want to derive the market value of the security from the benchmark, write MKT_VAL(PORTFOLIO_ID=CURRENT_BMK). Note that in order to use portfolio and benchmark analytics in the same formula, you must select **Single Column** from the *Show formula values for portfolio and benchmark securities in* section of the *Save As* window when saving your formula.

Field values that are used in the formula are retrieved using your analysis As Of date. In order to reference field values on dates prior to the As Of date, you can use LAG() function. Example: LAG(PX_LAST_OED, 1D). The output of this formula is security's closing price one day prior to the analysis date. Note that for the 1D parameter value, we are using calendar days. So if the as of date is Monday, LAG(PX_LAST_OED, 1D) refers to the previous Sunday. If security is not traded on weekends, Friday closing price will be returned in this case. LAG function takes two arguments. The first argument is the field mnemonic, and the second argument is a relative date in the format of nCT, where

- n is the number of time periods you want to go back. Note that n has to be positive.
- C is an *optional* parameter that refers to the end of the time period. For example, 1CM refers to the last day of the previous month.
- T is a time unit with valid values of D for days, W for weeks, M for months, Q for quarters, S for semi-annuals, Y for years.

Note that LAG function cannot be nested, i.e. LAG(LAG(PX_LAST_OED, 1D), 1D) is not supported, so LAG(PX_LAST_OED, 2D) should be used instead.

Scenario output using the data available on the Scenarios tab in PORT can be incorporated into the custom formulas. In order to do this, make sure that the appropriate scenarios are defined using Scenario Editor on the Scenarios tab in PORT. Then create a custom formula that references one of the scenario fields such as Profit and Loss percent or Stress Market Value. Once you have selected the scenario field under Show All Fields, you can select the name of the scenario that is going to be used in the custom formula. For full valuation scenarios you can also select scenario horizon. If you need to modify already created formula expression that references custom scenario, delete an existing expression and then recreate it by selecting the appropriate scenario name from the field picker. Note that you need to be explicitly enabled to reference Full Valuation scenarios.

Once the formula is entered, you need to save it before it can be used in PORT. When saving the formula, you need to provide a formula name, an optional description, as well as the expected output data type for the formula. Valid output data types are:

- Integer
- Number
- String
- Date

Formulas can also be shared with other Bloomberg users. You can grant users either read-only or read-write access to shared formulas.

Once the formula is saved, you can add it as a column on the *Characteristics* tab in PORT by clicking on the blue plus icon next to the formula name. After the formula has been added to the list of available fields, you can customize how you want to aggregate this formula by clicking on the pencil icon and selecting the appropriate Aggregation Methodology. Weighted Average aggregation methodology allows you to further customize the aggregation by picking the appropriate Numerator and Denominator values. Options for Numerator and Denominator include Notional Exposure, Market Value, Contribution to Duration, etc. You can also specify if you want to see the formula displayed for portfolio, benchmark or difference columns.

Custom formulas can also be referenced when creating custom classifications in UNCL<GO>, as well as when customizing goals, constraints and security properties in portfolio optimizer.

Below you can see the list of operators that are supported in custom formulas.

Mathematical Operators:

Operator	Description	Example
+	Sum	LIBOR_OAS+50
-	Difference	LIBOR_OAS-50
/	Division	LIBOR_OAS/2
*	Multiplication	LIBOR_OAS*0.5
^	Power	LIBOR_OAS^2
EXP	Exponent	EXP(LIBOR_OAS)
ABS	Absolute Value	ABS(LIBOR_OAS)
SIGN	Sign	SIGN(LIBOR_OAS)
LOG	Logarithm	LOG(LIBOR_OAS)
LN	Natural Logarithm	LN(LIBOR_OAS)
SQRT	Square Root	SQRT(LIBOR_OAS)
ROUND	Round function with number of precision	ROUND(EARN_YLD_HIST, 2)
MAX	Returns the largest value among the arguments	MAX(0.1* CUR_MKT_CAP, POSITION*PX_LAST_EOD)
MIN	Returns the smallest value among the arguments	MIN(0.1* CUR_MKT_CAP, POSITION*PX_LAST_EOD)
MEDIAN	Returns the median value among the arguments	MEDIAN(RATING_NUMERICAL_VALUE(RATING=RTG_FITCH_NUM), RATING_NUMERICAL_VALUE(RATING=RTG_SP_NUM), RATING_NUMERICAL_VALUE(RATING=RTG_MOODY_NUM))

Logical Operators:

Operator	Description	Example
==	Equal	LIBOR_OAS==100
!=	Not equal	LIBOR_OAS!=100
<	Less than	LIBOR_OAS<100
<=	Less than or equal	LIBOR_OAS<=100
>	Greater than	LIBOR_OAS>100
>=	Greater than or equal	LIBOR_OAS>=100
OR	Returns true if either the first, or the second argument is true	LIBOR_OAS>=100 OR YLD_CUR_BID>2
AND	Returns true if both the first and the second arguments are true	LIBOR_OAS>=100 AND YLD_CUR_BID>2
NOT	Returns false if the argument is true and vice versa	NOT(LN_COVENANT_LITE)
IN	Returns true if the first argument is in the list specified in the second argument	IN(SECURITY_TYPE, ["ABS", "Agency CMOs", "CMBS"])

Conditional Operators:

Operator	**Description**	**Example**
IF	Returns either the second or the third parameter depending on whether the first parameter is TRUE or FALSE	IF(LIBOR_OAS<300, EFFECTIVE_DURATION, 0.25*EFFECTIVE_DURATION)
AVAIL	If the value of the first argument is NA, the second argument is returned. BOTH ARGUMENTS HAVE TO BE VALID FIELDS (CONSTANTS ARE NOT SUPPORTED)	AVAIL(DIVIDEND_YIELD, YLD_CNV_BID)
ZNAV	Returns parameter if it is available, or 0 if it is NA	ZNAV(DIVIDEND_YIELD)

Cross-sectional Operators:

Cross-sectional operators allow you to apply an operator on all securities in a portfolio/benchmark, or a specific group of securities that are a subset of portfolio or benchmark. Cross-sectional operators take two arguments: the first required argument is the name of the field that you want to apply the operator on, and the second optional argument specifies the name of the group. For example, GROUPSUM(MKT_VAL, INDUSTRY_SECTOR) sums market values of portfolio or benchmark securities that belong to the same BICS Sector. GROUPSUM(MKT_VAL) sums market values for all securities in a portfolio or benchmark. In order to use the grouping that is currently selected in PORT in the cross-sectional operator, you can use the keyword RELATIONS. In this case GROUPSUM(MKT_VAL, RELATIONS) sums up the market values for securities in the same group using the grouping model that is selected in PORT when the formula is evaluated.

Operator	**Description**	**Example**
GROUPSUM	Returns the sum of values for all portfolio securities	GROUPSUM(MKT_VAL)
GROUPAVG	Returns the average value for all portfolio securities	GROUPAVG(MKT_VAL)
GROUPMIN	Returns the smallest value for all portfolio securities	GROUPMIN(MKT_VAL)
GROUPMAX	Returns the largest value for all portfolio securities	GROUPMAX(MKT_VAL)
GROUPMEDIAN	Returns the median value for all portfolio securities	GROUPMEDIAN(MKT_VAL)
GROUPCOUNT	Returns the number of values that are not NAs for all portfolio/benchmark securities. The return value is the same for all securities in the portfolio or in the benchmark.	GROUPCOUNT(MKT_VAL)

String Operators:

Operator	**Description**	**Example**
LEN	Returns the length of a string	LEN(ID_CUSIP)

LEFT	Returns a string that contains a specified number of characters from the left side of the string	LEFT(ID_CUSIP, 2)
RIGHT	Returns a string that contains a specified number of characters from the right side of the string	RIGHT(ID_CUSIP, 2)
STARTSWITH	Returns 1 if string in ARG1 starts with string in ARG2, and 0 otherwise	STARTSWITH(ID_CUSIP, ‘36’)
CONCAT	Concatenates two arguments	CONCAT(ID_CUSIP, SECURITY_DES)
TOLOWER	Converts argument to lower case	TOLOWER(SECURITY_DES)
TOUPPER	Converts argument to UPPER case	TOUPPER(SECURITY_DES)
REPLACE	Replaces part of a text string, based on the number of characters specified, with a different text string	REPLACE(ID_CUSIP, 1, 2, ‘AA’) – starting from the first position, we are replacing 2 characters of ID_CUSIP with AA

Date Operator:

Operator	**Description**	**Example**
TODAY	Returns today’s date	TODAY()

Time-series Operator:

Operator	**Description**	**Example**
LAG	Returns the value of the field as of a historical date	LAG(PX_LAST_OED, 1D)

Contact Us

To learn more about Bloomberg’s Portfolio & Risk Analytics, contact your Bloomberg account representative or press the <HELP> key twice on the Bloomberg Professional® service.