OPERATORS AND FUNCTIONS

A PEER SUITE TRAINING MANUAL

CALLAHAN & ASSOCIATES, INC.
www.Callahan.com
analystsupport@callahan.com

PEER BETA

Table of Contents

Introduction 2

Cycle Operators 3

Functions 101: Growth Ratios 5

1. Growth 5
2. Growth# 6
3. GrowthYTD 6
4. GrowthNCUA or GrowthYTDAnn 6
5. GrowthQtr 7
6. GrowthQtrAnn 7

Functions 201: Calculations 8

7. Annual 8
8. FullYear 9
9. Change 9
10. QtrChange 9
11. YTDChange 10
12. Booked 10
13. QtrAve 10
14. QtrAveAnn 10
15. hSum 11
16. hAverage 11
17. YTD 11
18. Count 11

Functions 301: Benchmarking 12

19. Rank 12
20. ReverseRank 12
21. Percentile 12
22. ReversePercentile 12
23. AtPercent 12
24. Peer 12

Introduction

What Are Functions?
Functions simplify more complex formulas by doing the work for you! Functions are commands you can use on custom displays in Peer to modify account codes. They are “shortcuts” for calculations that finance executives commonly use in evaluating financial performance.

> **For example**, when you type “Growth” then an account code bracketed by parentheses, e.g. Growth(a010), Peer will calculate annual asset growth so you don’t have to know the growth equation yourself.

What is an Operator?
In addition to normal mathematical operators (+, -, /, *), Peer also includes Cycle Operators. Cycle operators don’t change the value of an account code, but rather they point the program to a specific cycle for that code. Cycle operators can be relative to the current cycle (:.25 goes backward one quarter; :1 goes back one full year; :dec goes back to the last year-end cycle). Or they can be specific = :2Q22 will select the value for an account for the second quarter of 2022. See page 3 for more details on how and when they are used.

Helpful Hints

- **Have your 5300 handy.**
  You can print out the latest 5300 Call Report under the “Data Sources” menu in the top navigation of Peer.

- **Put an “a” in front of all 5300 account codes.**
  Many account codes are numeric values and peer will read any number as just that, a number. So, when building custom displays, just put an “a” in front of any account code from the NCUA 5300 Call Report.

- **Casing doesn’t matter.**
  Functions are NOT case sensitive. We are just using mixed case letters in this document to make them easier to read and remember.

- **Terminology.**
  When a function is used with an account, we refer to this as a **formula**. So, Growth(members) is the *formula* for member growth; Annual(a270) is the *formula* for a credit union’s annual marketing expenses. “Growth” and “annual” are the functions in these two examples.

- **You can use account codes in equations.**
  When using functions, you can add, subtract, multiply and divide account codes just like any number within the parenthesis. So, to see total auto loan growth, you can type **Growth(a385 + a370),** where a385 is new and a370 is used vehicle loans.

Cycle Operators

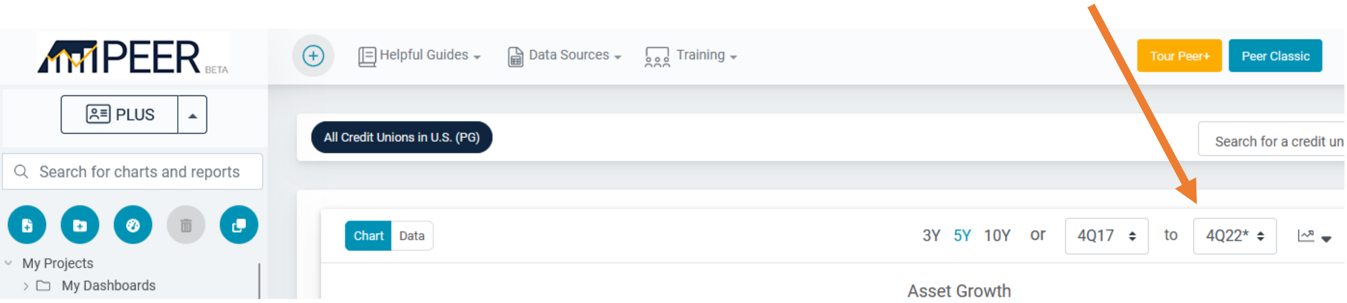
Cycle

We include the **Cycle** function in this section because all **Growth** function use data from prior periods in the calculation. Therefore, in order to understand how we calculate each different growth function, you need to be familiar with the cycle function.

The **Cycle** function is indicated with a colon **":"** and a number and is used to pull historical data for account codes from any time cycle. It is different from all the other functions in that it is used AFTER the account code.

Rather than making any calculations, the ":" function simply pulls data from the time period you specify. Typing ":" will denote that you want to find data from previous business cycles; the number after the colon indicates how far back in time you want to go, with each 0.25 equaling one quarter.

It is important to note that the ":" function is dependent on the cycle that you have currently selected at the top of the screen. In the screenshot below, Peer will calculate backward from 4Q 2022.



An example using total loans, or 025b on the 5300 Call Report:

- Your loan balance from the previous quarter = a025b**:0.25**
- Your loan balance six months ago = a025b**:0.5**
- Your loan balance nine months ago = a025b**:0.75**
- Your loan balance for the same quarter last year = a025b**:1**

Remember, in all levels of Peer, you must put an "a" in front of account codes from the 5300.

You may also apply the cycle operator to an entire formula as long as you enclose the formula in parentheses. For example,

(loans+investments+cash):3 is equivalent to: loans:3+investments:3+cash:3

There is one additional special cycle character you may encounter in growth calculations – **:dec**. This function always selects the value from the previous year's end. If you want to look at data from last year's end, use a396**:dec**.

If you don’t want to calculate how far back to go, you could use **:2Q22** instead (for example) to pull data from the second quarter of 2022. It can be used with formulas, as well as individual account codes and defined facts. For example, total assets from the first quarter of 2022 = Assets:**1Q22**

One drawback of this notation is that the displays would need to be modified in future quarters to produce equivalent results relative to those quarters. This is more useful as a quick shortcut for temporary displays.

Functions 101: Growth Calculations

Growth is often viewed as a key indicator of success, so Peer makes it easy to calculate your growth rates in five different ways. There is also a cycle function that is essential to understanding growth calculations. This section covers the following seven functions:

1. **Growth** – used to calculate the year-over-year, or actual 12-month, growth rate.
2. **Growth3** – calculates the compound annual growth rate over a variety of time periods.
3. **GrowthYTD** – calculates the growth rate from the prior year-end.
4. **GrowthNCUA / GrowthYTDAnn** – the projected annual growth date based on year-to-date performance.
5. **GrowthQtr** – calculates your growth over the previous quarter.
6. **GrowthQtrAnn** – forecasts a 12 month growth rate based on the current quarter's growth.

1. Growth

The **Growth** function is used to calculate year-over-year, or actual 12-month growth. Since it is calculating the rate of growth, it is displayed as a percentage and typically used on line charts or in tables. It's the most commonly used function in Peer!

This, and all other functions with the exception Cycle, are placed in front of an account code or codes in parentheses. So, to calculate the actual 12-month growth of loans you type **Growth(a025b) or Growth(loans)[1].**

The growth function calculates the rate of change by taking the difference between the results today and one year ago and dividing by the value one year ago. It saves you from having to type out the full formula using cycle functions as such:

$$\textbf{Growth(a025b)} \quad \text{or} \quad \textbf{Growth(loans)} = \frac{(\text{a025b} - \text{a025b:1})}{\text{a025b:1}} = \frac{(\text{loans} - \text{loans:1})}{\text{loans:1}}$$

You can also use Growth on ratios. We already have a shortcut for return on assets which is just "ROA". By typing **Growth(ROA),** you are asking for the percentage growth rate of your ROA compared to this time last year.

FAQ – Why is your growth different from NCUA's FPR report?
NCUA calculates growth differently on their Financial Performance Reports (FPR). They are projecting an annual growth rate using your year-to-date growth rate. There is a function that uses the same calculation, **GrowthNCUA**, described on page 5.

[1] "Loans" is a defined fact, or shortcut, included in Peer for commonly used accounts. See our Defined Facts Manual for more information.

2. Compound Annual Growth Rates (CAGR) – Growth#

The standard Growth function can also be manipulated to calculate compound growth over **2, 3, 4, 5 or 10** year periods just by adding the number after Growth. So, to calculate your membership's 3-year CAGR, you type **Growth3(members) or Growth3(a083b).**

A compound annual growth rate, or CAGR, measures the average of how much a statistic has grown each year for a given period of time, almost always longer than one year. Investopedia describes it as a, "Mathematical formula that provides a 'smoothed' rate of return."

Peer calculates compound growth by taking the ratio of the results today over the prior period results, to the nth root, with n being the number of years into the past you have specified. That number is then subtracted by 1 to give a percentage.

- 2-year compound annual growth: **Growth2(a083)** = (a083/a083:2)^(1/2) – 1
- 3-year compound annual growth: **Growth3(a083)** = (a083/a083:3)^(1/3) – 1
- 4-year compound annual growth: **Growth4(a083)** = (a083/a083:4)^(1/4) – 1
- 5-year compound annual growth: **Growth5(a083)** = (a083/a083:5)^(1/5) – 1
- 10-year compound annual growth: **Growth10(a083)** = (a083/a083:10)^(1/10) – 1

3. GrowthYTD

GrowthYTD, also read as "year-to-date growth", measures your rate of growth since the beginning of the year. It calculates the rate of change by taking the difference between the results of the current cycle and the prior year-end results, then dividing by the prior year-end results.

$$\textbf{GrowthYTD(a025b)} = \frac{(\text{a025b} - \text{a025b:dec})}{\text{a025b:dec}}$$

Note the use of the cycle function **:dec** in this formula rather than numerical designations. This is how Peer knows to find the prior year-end rather than look back a set number of cycles.

4. GrowthNCUA or GrowthYTDAnn

Another growth function is **GrowthNCUA** which matches what NCUA uses on its Financial Performance Reports (FPRs). This function *projects* your full calendar year growth based on the rate of growth so far this year. If it's easier to remember, you can use **GrowthYTDAnn** (which stands for YTD growth annualized) instead of GrowthNCUA.

GrowthNCUA projects an annual growth rate using your year-to-date growth rate. So, NCUA's FPR growth rates are calculated as such:

- At 1Q: GrowthNCUA(a025b) = ((a025b – a025b:dec)/(a025b:dec)) * 4/1
- At 2Q: GrowthNCUA (a025b) = ((a025b – a025b:dec)/(a025b:dec)) * 4/2
- At 3Q: GrowthNCUA(a025b) = ((a025b – a025b:dec)/(a025b:dec)) * 4/3
- At 4Q: GrowthNCUA (a025b) = ((a025b – a025b:dec)/(a025b:dec)) * 4/4

5. GrowthQtr

Our final two growth functions look at a credit union's performance over just the last 3-month period. **GrowthQtr** calculates the percent change over the previous quarter.

$$\textbf{GrowthQtr(a025b)} \quad = \quad \text{a025b-a025b:.25}$$

6. GrowthQtrAnn

Our final growth function is **GrowthQtrAnn** which project a full year's growth based on your most recent quarter's growth rate. GrowthQtrAnn calculates the percent change over the previous quarter and then annualizes that growth at a compounded rate.

$$\textbf{GrowthQtrAnn(a025b)} \quad = \quad \frac{\text{a025b-a025b:.25}}{\text{a025b:.25}} \quad \text{\textasciicircum}4$$

Functions 201: Calculations

This chapter includes function that are designed to be used with either YTD or continually accruing items, but not both. It is important to use the correct one when building your displays. Since some account codes are reported year-to-date, we've created function that automatically know how to calculate changes correctly.

7. **Annual –** used only with YTD items, this function estimates a full year's performance based on current year-to-date performance.
8. **FullYear –** also used only with YTD items, this function calculates the actual 12-month value for the prior 12 months for any YTD account.
9. **Change –** calculates the absolute difference between this year and last for any account. Most commonly used with continually accruing accounts.
10. **QtrChange –** used with continually accruing items, it calculates the difference between quarters.
11. **YTDChange** – used with year-to-date accounts, it calculates the change in value over the course of the year.
12. **Booked –** used with year-to-date accounts, it calculates the dollar or number booked in each quarter.
13. **QtrAve –** used with balance sheet items to calculate yield, it averages the current cycle with the prior year-end.
14. **QtrAveAnn –** used with year-to-date items, this function averages this year's annualized value with the prior full year's actual value.
15. **hSum –** used to find the historical sum of a value across all quarters. A number ("**hSum3**") can be added to limit the history to either 3, 5, or 7 years.
16. **hAverage** – calculates the average value across every historical quarter. A number ("**hAverage3**") can be added to limit the history to either 3, 5, or 7 years.
17. **YTD** – Calculates a year-to-date total for values reported on a period-by-period frequency. Used for Callahan datasets outside of the 5300 (non-interest income, etc.).
18. **Count** – Calculates the number of credit unions in your peer group that report a non-zero value for your given metric.

7. Annual

The **Annual** function is used to project full year amounts for items that are reported as year-to-date (YTD) on the call report. So **Annual(a031b)** projects your full year loan originations, assuming they continue at the same rate as they have so far this year.

Annualizing a YTD line item enables you to make meaningful comparisons across multiple time periods, regardless of which period you are using for your comparison. One common use is in expense and productivity ratios, such as Operating Expense to Average Assets or Average Loan Originations per Employee.

This function is only used on account codes that are reported year-to-date on the Call Report. Here are some examples:

Description	Example Account Code	Where on Call Report

Year-to-Date Loan Originations	a031b	Page 6
All Income & Expense Items *(Ex. Interest Income)*	a115	Page 5
Loan Charge-offs & Recoveries *(Ex. Total C/Os)*	a550	Page 8
Schedule A Specialized Lending *(Ex. 1st Mortgages Granted)*	aRL0017	Page 11

Peer annualizes a number by multiplying the value by 4 and then dividing by the current quarter. You do not need to make these calculations – Peer is running all this for you in the background! The account used in this example, a031b, is the account code for year-to-date loan originations.

- If your current cycle is 1Q: annual(a031b) = (a031b * 4)/1
- If your current cycle is 2Q: annual(a031b) = (a031b * 4)/2
- If your current cycle is 3Q: annual(a031b) = (a031b * 4)/3
- If your current cycle is 4Q: annual(a031b) = (a031b * 4)/4

8. FullYear

The **FullYear** function is also used specifically with year-to-date (YTD) line items to give you the actual accumulated value over the previous 12 months. Since YTD line items "reset" at the beginning of each year, it is difficult to see the absolute change in YTD items across different calendar years. The FullYear function allows you to easily see the one year accrual in any YTD item of your choosing.

Peer calculates a YTD item's full year accumulation by adding the YTD amount to the difference between the previous year-end's value and the value from one year ago. This one is easier to see in a formula!

FullYear(a031b) = a031b + (a031b:dec – a031b:1)

where a031b is this year's accrual; a031b:dec is total loan originations as of December 31 the prior year; and a031:1 is the value of a031b at the same cycle last year.

9. Change

Unlike Growth which calculates a percentage rate, the **Change** function calculates the absolute difference between this year and last for any continually accruing account code. It is the numerator of the Growth calculation.

For example, to calculate how much your loan portfolio has grown in absolute dollars, you would use:

Change(a025b) = (a025b – a025b:1)

Change displays as a dollar or number, whichever format the original account code is in. Since it's a volume of change, column charts may be more appropriate.

10. QtrChange

The **QtrChange** function is used to calculate the absolute difference between this quarter and last quarter for balance sheet items or other accounts that are continually accruing. It will display as a dollar or number, whichever format the original account code is in. Since it's a volume of change, column charts may be more appropriate.

QtrChange(a025b) = (a025b – a025b:.25)

11. YTDChange

The YTDChange function is used to calculate the change in value over the course of the year for balance sheet items and other accounts which are reported as of a point-in-time. This function takes the value for the quarter and subtracts the previous year-end value.

YTDChange(a025b) = (a025b – a025b:dec)

12. Booked

Booked is similar to QtrChange but is used on year-to-date items. It is a quick and easy way to calculate how much you have booked in any year-to-date account in any given quarter.

Since credit unions report some items as a year-to-date number, this function parses which cycle you are in to calculate the change from the prior quarter. Unlike continually accruing items, YTD accounts reset each January 1, so the Booked function knows to treat first quarter differently. In other quarters, it subtracts the prior cycle value from the current one.

IF Cycle = 1Q **Booked(a025b)** = (a025b) ELSE **Booked(a025b)** = (a025b – a025b:.25)

13. QtrAve

The **QtrAve** function is used only continually accruing items and calculates the average balance with the prior year-end. It is primarily used when calculating yields.

QtrAve(a025b) = (a025b + a025b:dec)/2

Let's look at how we calculate the yield on the loan portfolio. Simply dividing YTD loan interest income (a110) by the current quarter's loan balance (loans) will understate your yield (assuming you are growing the portfolio) or overstate your yield (if you are shrinking it). This is because YTD loan interest income is measured as an accumulated amount over the year, while loan balances are a static value at that specific quarter. In order to get a more accurate measurement of yields, we would instead use a110/QtrAve(loans)

14. QtrAveAnn

There is another function similar to QtrAve that works with year-to-date account codes: **QtrAveAnn**. This function calculates this year's annualized data and then averages this value with last year's actual value.
QtrAveAnn(a270) = (annual(a270) + a270:dec)/2

15. hSum

The **hSum** function calculates the sum of a value across all historical quarters. This is useful when calculating the number of mergers undergone by a credit union. The historical range can be limited to 3, 5, or 7 years by adding the relevant number to the end of the function.

hSum(merger) = # of mergers all-time
hSum5(merger) = # of mergers over the past 5 years

This function is also useful for values reported as YTD, that have been "booked" to isolate quarterly values. **Note:** YTD values must be booked before using the **hSum** function.

hSum(booked(dividends)) = Total dividends issued by a credit union, all-time

16. hAverage

The **hAverage** function takes the average of a value across all historical quarters. This function is useful for ratios and values reported as YTD, that have been "booked" to isolate quarterly values. The historical range can be limited to 3, 5, or 7 years by adding the relevant number to the end of the function.

hAverage(ROA) = Average ROA, all-time
hAverage5(ROA) = Average ROA over the past 5 years
hAverage(booked(arl0048)) = Average real estate loans originated per quarter, all time

17. YTD

The **YTD** function is used to calculate a year-to-date total for values reported on a period-by-period basis, usually monthly. The function will aggregate the designated value for all periods from the current cycle back to and including the most recent January. This function is meant to be used with Callahan's proprietary datasets that use monthly reporting structures. There are no account codes on the 5300 that require this function, as all codes are already reported as year-to-date values.

YTD(x131z) = The total amount ($) of fees refunded year-to-date

18. Count

The **Count** function fills a similar role to some uses of pivot tables in Excel. The Count function is used exclusively with a peer group of institutions active in your comparison set. When combined with a 5300 account code, the Count function will tell you how many institutions in your peer group report a non-zero value for this metric. This function can often be used in comparison with the defined fact "**Peersize"**, which gives the value for the <u>total</u> number of credit unions in your peer group.

Count(a396) = The number of credit unions in your peer group that offer credit card loan products.

Functions 301: Benchmarking

The six functions in this section allow you to **benchmark**, or **rank,** your performance against a set of peer credit unions. They are designed to not only tell you if you are above or below a peer group average, but specify where in the peer group your performance falls in order to set improvement goals. They can be used to build scorecards or on charts to show how your performance ranking has changed over time.

19. **Rank –** calculates where your performance stands in comparison to other credit unions as a numerical rank, where the highest values are assigned the lowest ranks.
20. **ReverseRank –** same as above, but used when higher value is given a higher rank.
21. **Percent –** similar to Rank, this calculates a percentile ranking, where a higher value is assigned a higher rank.
22. **ReversePercent –** Similar to Percent, but higher values receive lower percentiles.
23. **AtPercent –** used to display a metric's explicit value for a specified percentile.
24. **Peer –** returns the peer group average for any metric.

Note: Before using any of these functions, make sure the credit union you want to analyze is set as the Primary. The comparison set should include either individual credit unions or a peer group containing a group of credit unions that you want to benchmark again. If you select multiple peer groups, these functions look at each credit union within those groups individually to determine where your performance stands within the whole.

19. Rank

The **Rank** function can be used with any account or ratio and returns a numerical rank for your performance within the peer group comparison set. **Rank** is used on values where higher is generally considered "better".

For example, **Rank(ROA)** will return a numerical value for where the primary credit union's ROA falls within the comparison set. It returns a number from 1 (the best) to the maximum number of credit unions in your comparison set. Our software ranks discreet values, not necessarily equal to the number of credit unions in your peer group.

You can use this, and the other function in this section, in conjunction with other functions. For example, **Rank(Growth(members))** will return your primary institution's ranking in Member Growth compared to the rest of the institutions in the comparison set.

20. ReverseRank

ReverseRank works the same way as Rank but should be used with metrics where lower is better. Examples include ratios like Operating Expense/Average Assets, Efficiency or Delinquency.

As an example, **ReverseRank(a041b/loans)** will return a numerical rank for how the primary institution's delinquency ratio measures against its peer comparisons. The lower the value, the better the ranking, with 1 being the best.

21. Percentile

Similar to Rank, the **Percentile** function shows how a credit union compares to all the institutions in the comparison set. However, rather than return a numerical rank, this function returns the percentile your credit union falls into for that metric. **Percentile** assumes that higher is better.

Our software ranks all the values for a performance metric, e.g. ROA, from the credit unions in the peer comparisons you have selected. The percentile ranking is then determined via a simple equation to ensure that the incremental differences between each rank is uniform throughout.

22. ReversePercentile

The **ReversePercentile** function works the same way as Percentile but should be used when you are looking at metrics were lower is better. Examples include ratios like Operating Expense/Average Assets, Efficiency or Delinquency.

23. AtPercent

The **AtPercent** function requires the user to specify both a metric (an account code, a ratio, or a formula), as well as a performance level. It returns a metric's explicit value for the specified percentile.

For example, to show how fast credit unions in your comparison set are growing their membership in the top quartile of performance, you would specify

AtPercent(75, growth(members))

Where 75 indicates the top quartile and growth(members) is the formula for member growth.

24. Peer

The **Peer** function will return the average for your entire comparison set for whatever metric you specify. For example, typing **Peer(ROA)** on a table or a line chart, will return the average ROA for all peers selected, compared to **ROA** which would return the value for the primary institution or group.

Peer group averages are computed by totaling the values for each individual account and dividing each sum by the number of credit unions in the peer group. All ratios and other formulas are then computed using these average account values. This confers greater weights to credit unions with higher values in each account.