

LeanData Routing

Round Robin Distribution Guide





Overview	3
How it works	3
Assignment scenarios	4
Members Returning from Unavailability	4
All Members are Unavailable Simultaneously	5
How to Reset Pointers	5
Scenario	5
Frequently Asked Questions:	8
Summary	8



Overview

LeanData's Round Robin feature includes distribution of Leads, Contacts, Accounts, Opportunities, or Cases to a group of Users or Queues within a given pool. This guide will provide an overview on how LeanData's Round Robin functionality works to assign records and how the product determines assignment order for different scenarios.

Note: This guide only describes the behavior of LeanData's Round Robin distribution prior to LeanData Version 1.559. Subsequent versions will feature a distribution model that assigns to the next available User in sequence for unweighted pools.

How it works

LeanData's Round Robin assignment does not necessarily distribute in the chronological order of the pool, but is configured to prioritize predictability while subject to any weightings or other restrictions on the distribution of records (schedules, capping, pausing, etc) .

LeanData's Round Robin distribution functionality utilizes pointer values to determine which member is up next for assignment in the Round Robin pool. A pointer value is a numerical value that is associated with each round robin member, to help keep track of who should receive the next record. Whichever available User has the lowest pointer value at the time of assignment will receive the record. While these pointer values are not visible on the Round Robin page itself, LeanData will keep track of them implicitly.

After a record is assigned to a user, their pointer value will increase to reflect that assignment.

When multiple available Users have the lowest pointer value, LeanData will select the User who appears earliest in the pool (from top to bottom).

When a new pool is created, all members in a given pool will have no pointer value. In this case, LeanData will start the Round Robin distribution from the top of the list and work its way down the list of members in the given pool, as all members start with an equal pointer value.

If a new User is added into an existing pool, LeanData will continue normal distribution until their turn arrives, at which point, their pointer value will be set equal to the lowest pointer value available in the pool.

Assignment scenarios

Members Returning from Unavailability

Since we use the pointers method, when a User becomes unavailable (Working Hours, Vacations, etc) and then returns to availability, certain reset behavior needs to occur to ensure that subsequent assignments are not skewed disproportionately.

When a User returns to availability in the pool, LeanData will continue to assign records normally until the returning User's turn arrives in the pool. They will receive the record when their turn arrives, and then their pointer value will reset.

If their turn occurs **before** the member with the lowest pointer value, their pointer value will reset to the lowest pointer value and also increment up for the latest assignment.

If their turn occurs **after** the member with the lowest pointer value, their pointer value will be reset to match the lowest pointer value.

Note: A User's pointer value will only be considered as the lowest pointer value if they are available and their pointer value has already been reset after their latest return to the pool.

Example

There are 3 Round Robin pool members: Alex, Christina, and Stephen.

Alex goes on vacation and is skipped in the Round Robin distribution for a week. When he returns, assignments will continue sequentially in the pool until his next turn arrives in the pool, upon which his pointer value will be reset based on the lowest pointer value in the pool.

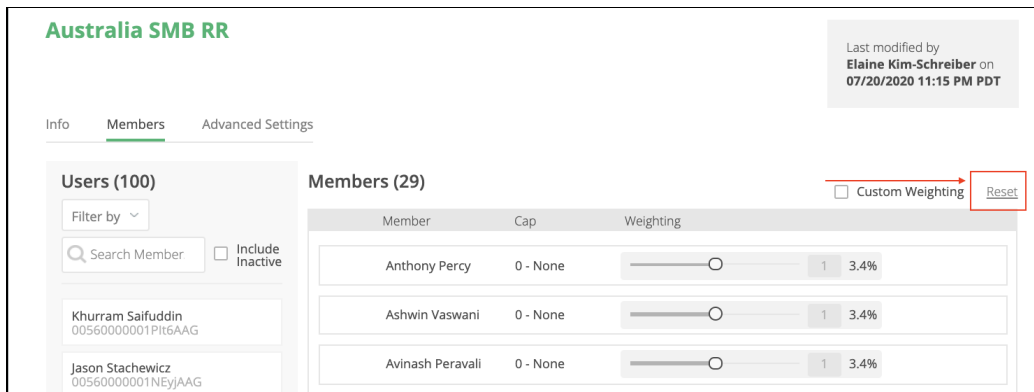
Member	Pointer Values when Alex goes on Vacation	Alex returns but will wait until his turn arrives in the pool	After Alex receives his first Lead after returning
Alex	3	3	11 (Min PV +1)
Christina	3	Next up → 9	10
Stephen	3	9	10

All Members are Unavailable Simultaneously

If all members of the Round Robin pools are unavailable simultaneously, such as a company holiday where everyone is unavailable, once everyone returns, the pool will assign the next record to whomever was due to receive the next record before the unavailability. Subsequent assignments will then reset based on the pointer value of the first user to get an assignment.

How to Reset Pointers

Resetting the weighting within the pool will result in resetting the pointer values as well.



Scenario

In the following example, we have a RR Pool with four members. We will move progressively through each scene to explain how events like assignment or becoming unavailable through some mechanism (like Working Hours, Vacation, or Pausing)

	Member	Status	Pointer Value	Notes
Scene 1	A	Unavailable	2	← User A becomes unavailable
	B	Available	3	
	C	Available	8	
	D	Available	2	

	Member	Status	Pointer Value	Notes
Scene 2	A	Available (PV Pending Reset)	2	← User A becomes available (PV not reset yet)
	B	Available	3	
	C	Unavailable	8	← User C becomes unavailable
	D	Unavailable	2	← User D becomes unavailable

	Member	Status	Pointer Value	Notes
Scene 3*	A	Available	$3+1 = 4$	← Lead gets assigned to User A as the next up. PV is reset to Min PV + 1
	B	Available	3	← 3 is the Min PV (User D is not available)
	C	Unavailable	8	
	D	Unavailable	2	

	Member	Status	Pointer Value	Notes
Scene 4	A	Available	4	
	B	Available	3	← since A was assigned the previous Lead, B is up next
	C	Unavailable	8	
	D	Unavailable	2	

	Member	Status	Pointer Value	Notes
Scene 5**	A	Available	4	
	B	Available	3 + 1 = 4	← B is assigned next Lead, and PV increases by 1
	C	Available (PV Pending Reset)	8	← User A becomes available (PV not reset yet)
	D	Available (PV Pending Reset)	2	← User D becomes available (PV not reset yet)

	Member	Status	Pointer Value	Notes
Scene 6***	A	Available	4	
	B	Available	4	
	C	Available	+1 Lead → PV reset to 4	← Lead gets assigned to User C as the next up. PV is reset to Min PV (min PV is 4 as User D's PV has not yet been reset)
	D	Available (PV Pending Reset)	2	

	Member	Status	Pointer Value	Notes
Scene 7***	A	Available	4	
	B	Available	4	
	C	Available	4	
	D	Available	+1 Lead → PV reset to 4	← Lead gets assigned to User D as the next up. PV is reset to Min PV

* We take the minimum pointer value of available Pool Members and add 1 to it

** We did not skip to D even though he has the lowest pointer. We respect the order of the Pool.

*** We reset to the Min Pointer Value (PV) if the User is after the minimum PV member of the Pool. If before, we reset to the Min PV + 1. Users that just became available are not eligible for consideration as min PV.



Frequently Asked Questions:

Why is LeanData ‘skipping’ around the pool?

- The ‘skipping’ effect may occur when the ‘Next Up’ feature is used. After assigning the “next” record to that specific member, LeanData will then proceed to evaluate which member has the lowest pointers and will distribute the next record to that member thereafter.

Why is one member getting all of the Leads today?

- This is unlikely to happen, but since distribution may be affected by many factors, like weighting, availability, and current Lead ownership, it may be that only one member is available when Leads come in. Due to a combination of these factors, it is also possible that one member’s pointer value can fall behind without getting reset, in which case they will receive Leads until their pointer value increases.

How come some members are not receiving any Leads?

- This may be due to their pointer values being higher than the rest of the pool members.

How does weighting affect this distribution?

- LeanData will still distribute records based on the pointer system and will still reset members’ pointers according to the rules above. Weighting will affect how much each assignment will increment the pointer value.

Summary

This guide has provided an overview on how LeanData’s Round Robin functionality works to assign records and how the product determines assignment order for different scenarios. LeanData’s Round Robin assignment does not necessarily distribute in chronological order of pool, but will attempt to maintain the general order of the pool while also considering other factors such as any weightings or other restrictions on the distribution of records (schedules, capping, pausing, etc) .