LeanData Routing

Best Practices Vol. 1

Overview

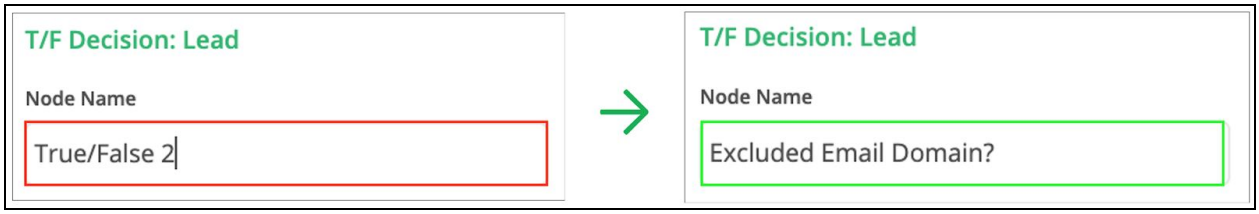
Although LeanData's Routing products are designed to be customizable and flexible to accommodate your specific organizational needs, we have found that many of our customers have success when following certain best practices in designing and building their Routing graphs.

This guide will introduce some of these best practices, but it is important to remember that these are offered as ***recommendations*** only. Your specific organizational practices may make some of these best practices irrelevant or less useful to you. Nevertheless, we are confident that you will find some best practices that will be beneficial to your use of LeanData Routing.

General Graph Design Best Practices

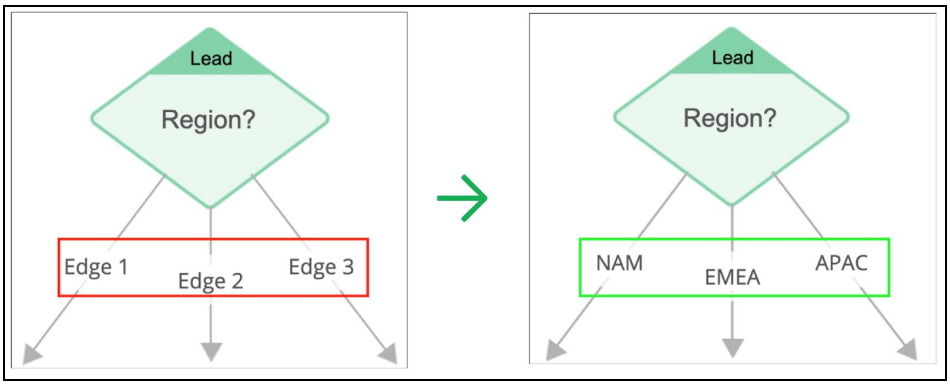
Name Your Nodes Clearly

Name your nodes meaningfully as if some else would eventually need to take over, so that they can easily interpret each node's purpose without having to open it and check the logic. This will also help yourself find nodes with Cmd+F (or Ctrl+F) when you can't find particular nodes.



Name Your Edges Clearly

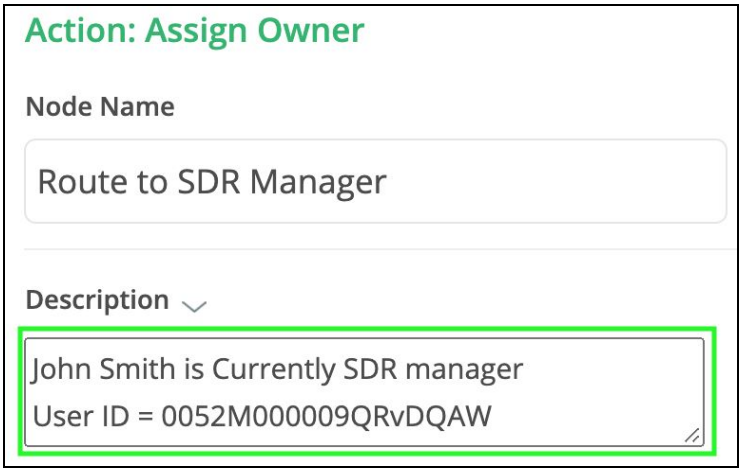
Name your node edges helpfully (not edge 1, edge 2, etc). Also, edges reflect the ***outcome*** of the node they are proceeding from, ***not the destination*** they are directed towards, so you should name them accordingly.



Utilize the Node Description Field

Utilize the node description field to state any reasoning and purpose for a node. This will help future admins (including yourself) understand the intent behind the node and whether it is still relevant as they are reviewing.

- You may also add user IDs and names to the description if it would help you discern who a record should be going to.



Graph Structure

When designing your graph from the start, you may be overwhelmed by the endless possibilities on how to structure your organizational routing rules. The following framework will help to give a general outline on the major sections of a graph, and how they are laid out sequentially. From there, you can add your specific details, exceptions or modifications so that it works for your needs.

Many graphs follow this general format:

1. **Triggers & Filters**
    - What must happen for LeanData to start evaluating a new or existing record?
    - Which records should not be evaluated, or treated separately from the rest of your records?
2. **Decisions**
    - Should there be an initial split into separate branches from the start, based on lead source, geography, record type, or tier? Or should that happen later?
3. **Matching**
    - Would it matter if there are related records in the system?
    - You may use multiple types of matching sequentially.
4. **Decisions (Post-Matching)**
    - Checking criteria on the matched record if there are matches
    - Further segmenting records that don't find a match
5. **Actions**
    - What do you want to do with the record after passing through the prior logic and arriving at its destination? Is it simply assigning the record? To whom? Are there other fields to update? Sending notifications? Converting or creating other records?

Your graph may look different than the framework given above. The goal isn't to make your graph conform exactly to some pre-existing template, but to provide a starting point and generally applicable outline to guide you when building out your graph initially.

We will be using this framework to categorize other best practices pertaining to each section of your graph.

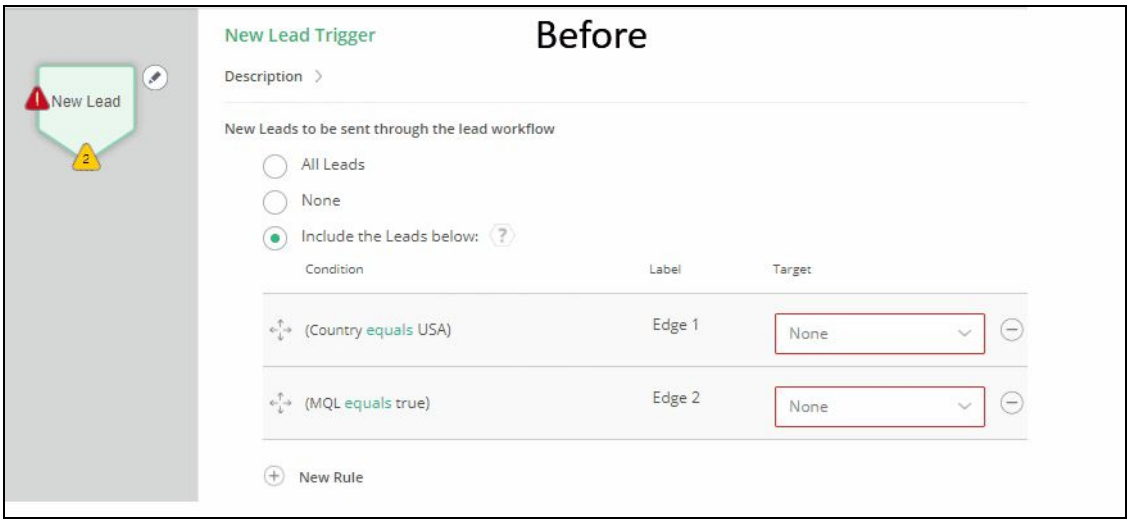
Triggers & Filters

Separate Triggers from Filters

Keep your triggers and filters separate, combining them can lead to unexpected triggering.

It is usually best to focus your update triggers around the actual field that will be changed, any other additional criteria should be handled in a follow up decision node. Including filters to go along with that may result in records getting updated too often.

With the New Lead node, we generally want to follow the same principle, including all Leads within the New Lead trigger, and immediately follow up with a decision node to check criteria on whether or not a record should be routed. This allows you to see detailed routing insights for every single record regardless if it was routed or not.



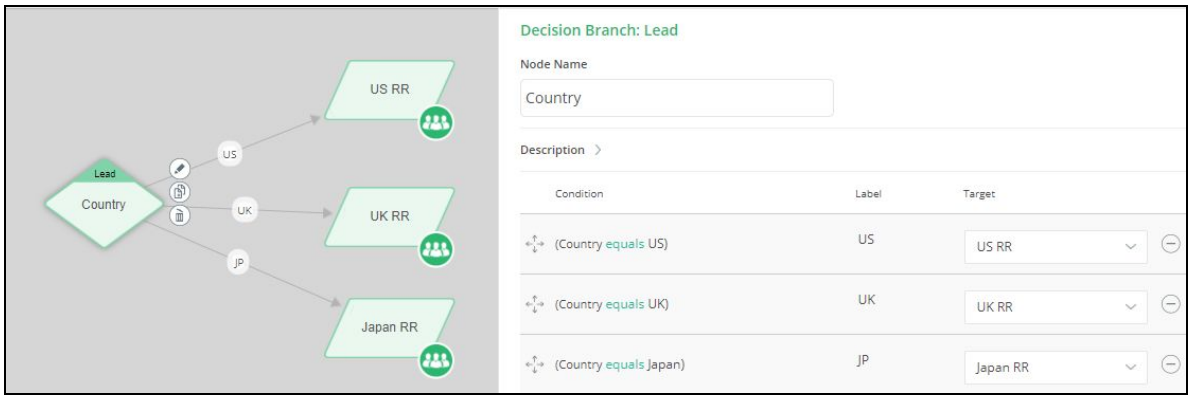
For more information on triggers, please refer to the New Entry Node Guide and the Updated Entry Node Guide.

Decisions

Keep Branch Node Edges Mutually Exclusive

Keep the criteria of branch nodes mutually exclusive if possible.

When working with branch nodes, you want each condition to be unique so that when a record is being evaluated, only one edge will be satisfied. If your criteria happen to be too vague, your records will pass through the first edge it satisfies, never giving it the opportunity to be evaluated by other conditions in the node.

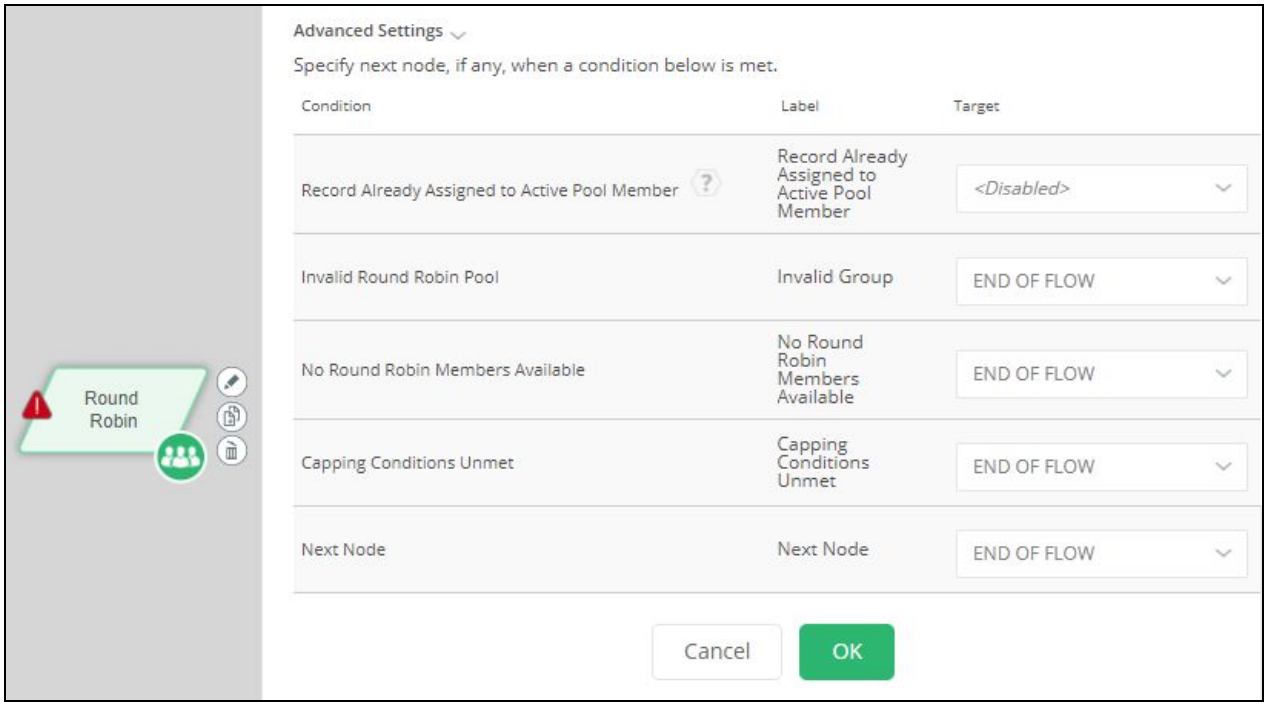


For more information on branch nodes, please refer to the Branch Decision Node Guide.

Always Consider Fallback and Default Edges

Fallback and default edges can ensure that every record ends up getting routed, rather than terminating at some midpoint in your flow. An example would be an invalid/inactive user fallback when routing to a matched Account. Even though inactive users might not be a problem for a specific organization, it's always best to be on the safe side. E.g someone leaving the company, on vacation, etc.

Additionally, existing records may have legacy values that you may no longer use, but you may still want to route those records at some point. Future records may have different values that you cannot anticipate. Using your fallback and default edges will ensure they are accounted for, even while you consider how to specifically adust your flow to route them appropriately.

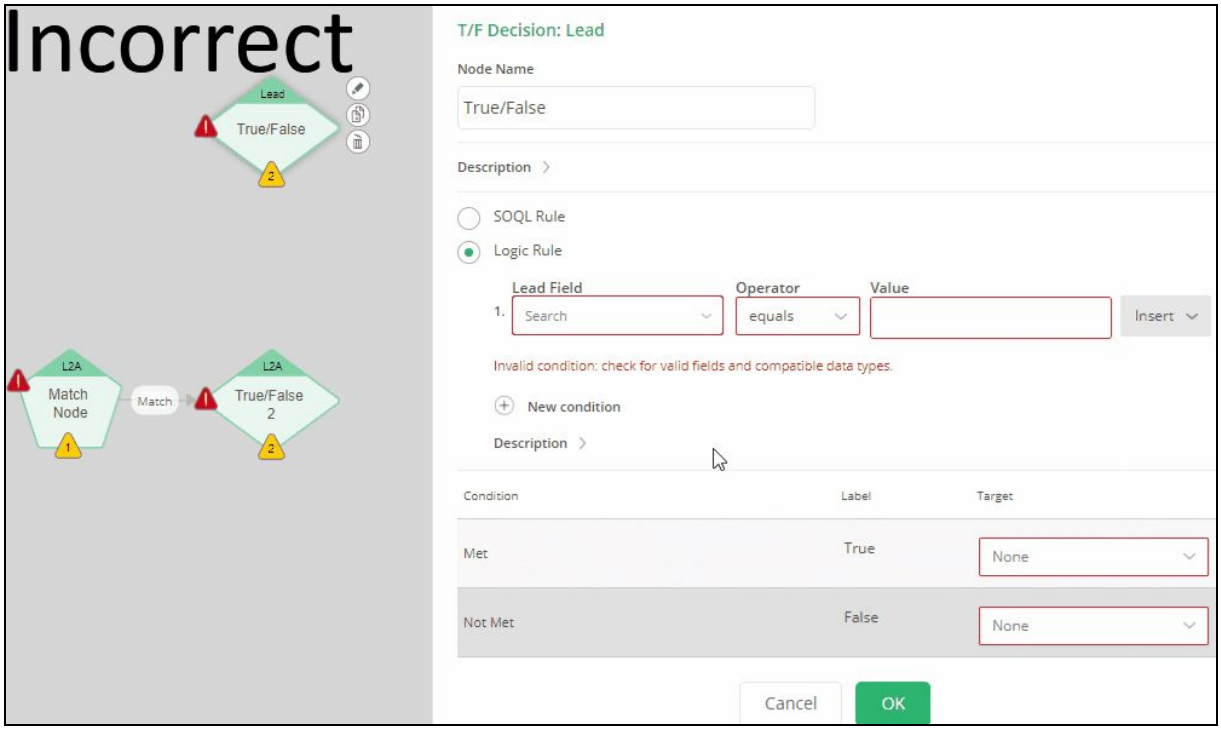


Matching

LeanData Routing Operates Independently from Tagging

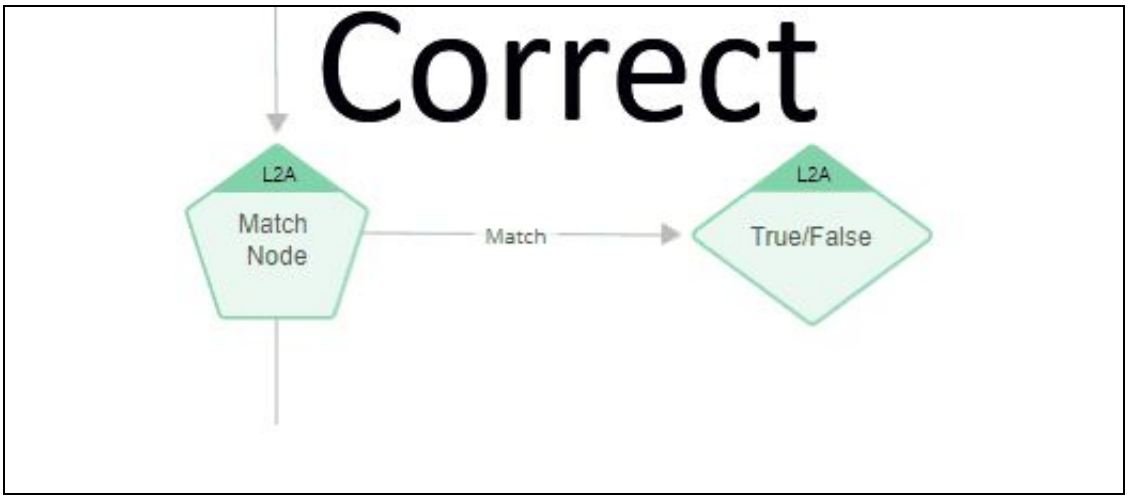
One common misconception is that LeanData Routing relies on matches found via our Tagging Product. These two products operate independently.

This means you should not rely on fields that LeanData Tagging has populated for your Routing criteria. Rather, you should directly reference fields on matched records rather than relying on tagged versions of those fields on the Lead record. Leads might not have been tagged yet while they are progressing through your router graph. For more information please see [this article in the Help Center](#).



Match Nodes vs Decision Nodes

Match Nodes are focused on identifying one best match. After this match is found, you can apply decision logic to the matched record in a separate node afterwards. You do not have to use another Match node unless you are looking for a different match.



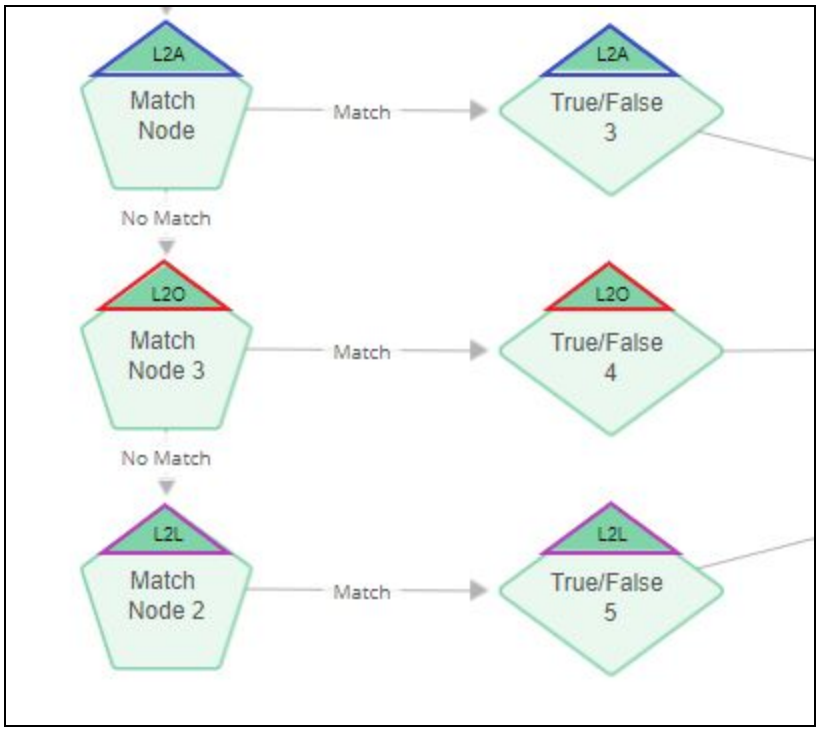
There are times where it may be appropriate to use different Match nodes sequentially, such as matching to different objects.

Decisions (Post-Matching)

Ensure your Decision Nodes Correspond to the Right Object

To avoid any unforeseen errors it's best to check the label of the decision node before placing it on the graph. Decision nodes that reference fields and values on a matched object require a corresponding match node. For example, a L2A decision node requires an Account match before you are able to use it.

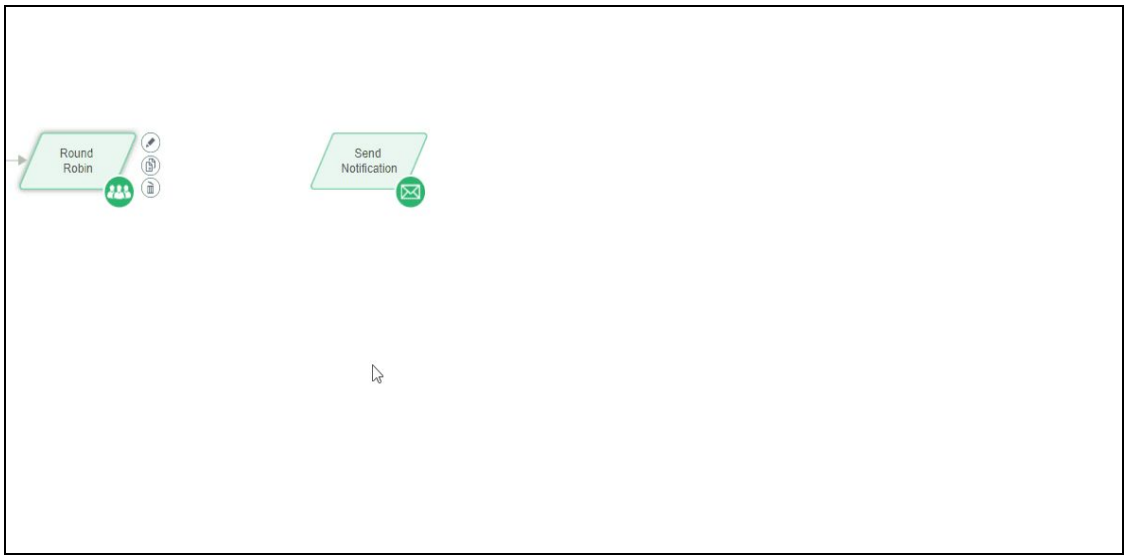
In order to use a decision for a particular matched object, all records that arrive at that decision must have found a match to that object earlier in the graph.



Actions

Your Graph Does Not Have to End at the First Action

Though actions are meant to be the last step in a graph, most nodes allow you to carry out additional actions with the exception of converting or merging leads. An example would be assigning a Lead to a user, then updating a field or sending an email. There are near endless possibilities!



Utilize LeanData's Routing Action Field

LeanData will stamp whatever action it took on a record into a custom field on the routed record: Routing Action.

Routing Action	assigned - owner assignment

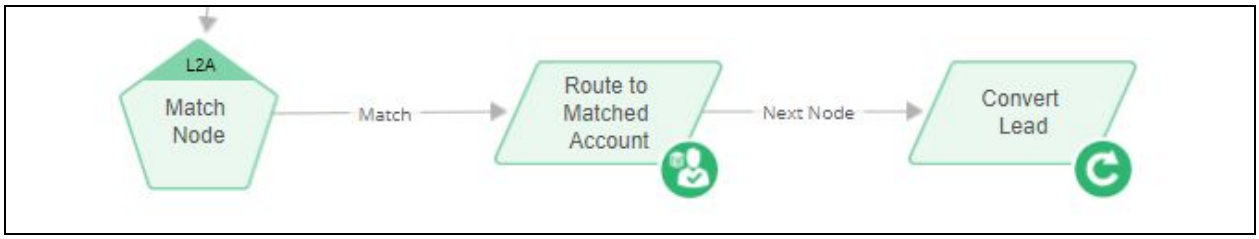
This field contains basic information such as "assigned - owner", "merged", "converted" etc. You can use this field in routing to help target specific use cases like triggering a merged record to re-enter the graph, preventing converted leads from LeanData getting retriggered, and many other options.

This field may also be useful for reporting purposes, if you need a general summary of what actions LeanData took on your records. For a list of these values, please see the Routing Action Guide.

Assign Leads to a User Before Conversion / Merging

Assignments should be handled before converting a Lead. It's best to make all the necessary updates to a record within Lead Router, because once the Lead is converted, it is no longer actionable by Lead Router.

In this illustration we are assigning the Lead to the Matched Account Owner before conversion.



As mentioned at the beginning of this guide, there is no one-size-fits-all approach to building a FlowBuilder graph. These best practices are recommendations to keep in mind as you build your own graph that fits your needs. For more assistance, you may reach out to your LeanData Rep or submit a ticket to the LeanData Support Team.