

Routes

The next thing we'll need to add to our process are routes. **Routes** are typically nodes where decisions are made and, based on the rules of your process, where the flow of the application may be altered. Strictly speaking, the position of a route in a swimlane doesn't really matter and will not change the logic of your process. For practicality's sake, however, it will be best to make it as legible as possible.

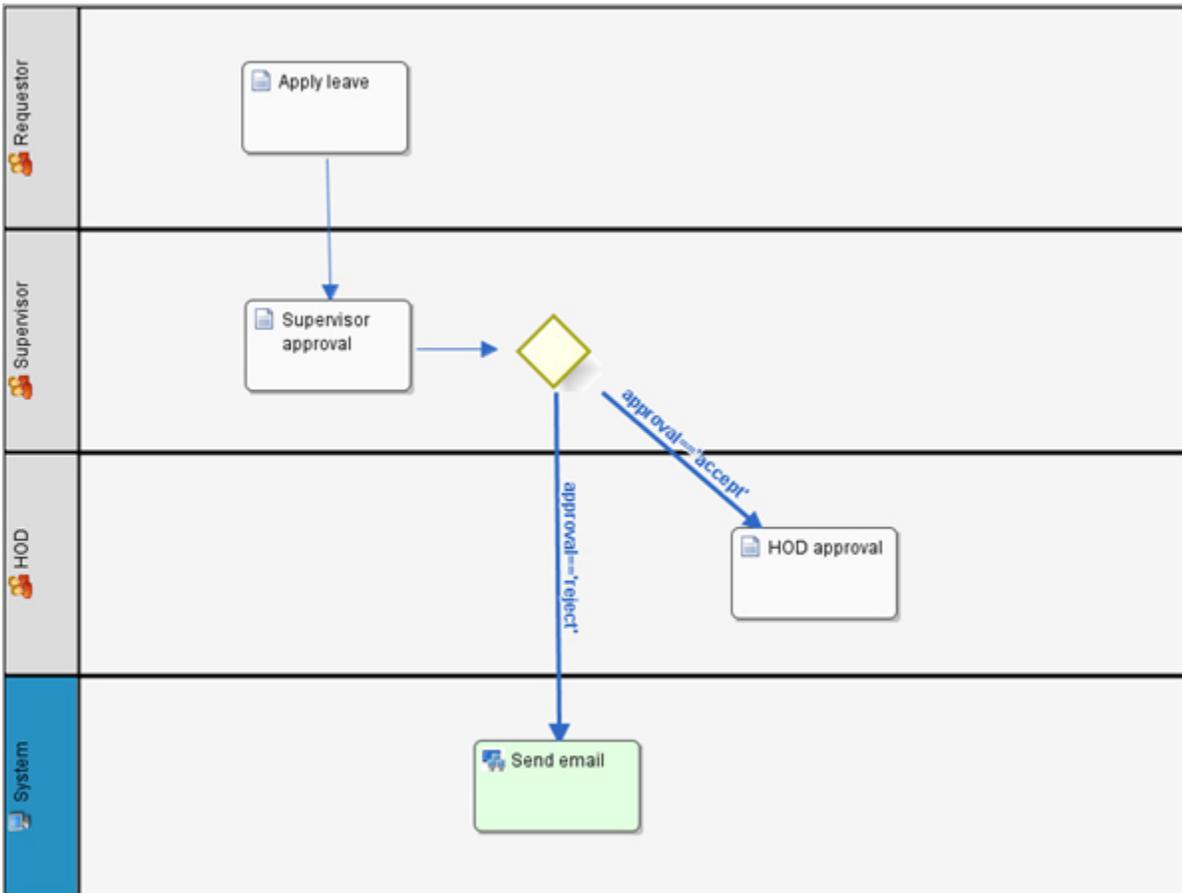
Before modifying our process, there are 2 general concepts that you should be familiar with when dealing with routes. Given that routes are generally used to affect flow and insert logic and rules into your process, there are 3 types of common constructs for routes. They are:

Decisions

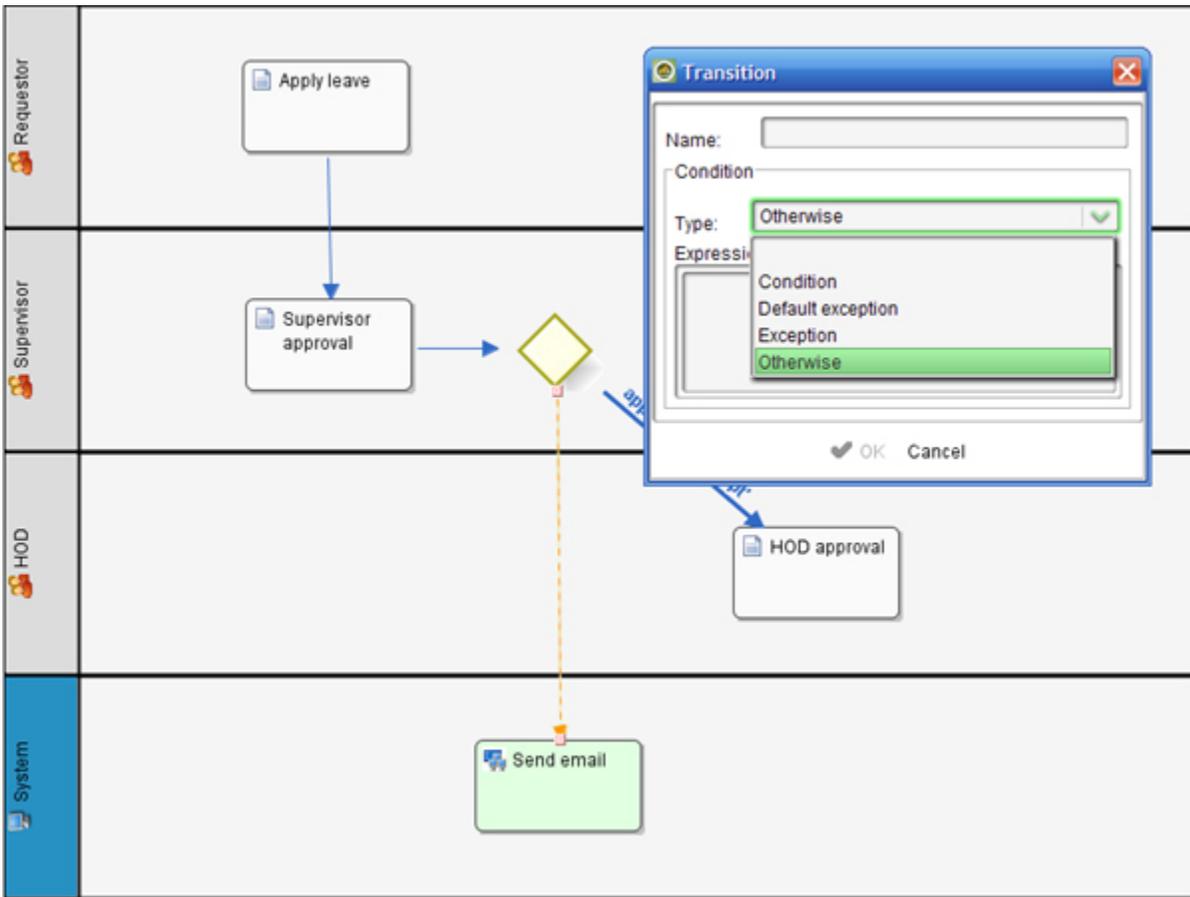
Decisions are used when you want to insert a point in your process where the flow will be determined by a set of rules. For instance, in our Leave Application process, once a requestor submits a leave request, a review by the supervisor will be required. You will see in the figure below that the transition from the requestor to a supervisor happens in a straight line, which tells us that whenever a requestor submits a request, that request will always go to the supervisor and there are no other circumstances or conditions in which that will not be true.

Now, what the supervisor decides to do will result in a decision: either choose to accept or reject the application. If the supervisor rejects it, a mail will be sent to the requestor. If the supervisor accepts it, the application will be forwarded to the HOD.

Given the above, our process will look like the figure below:



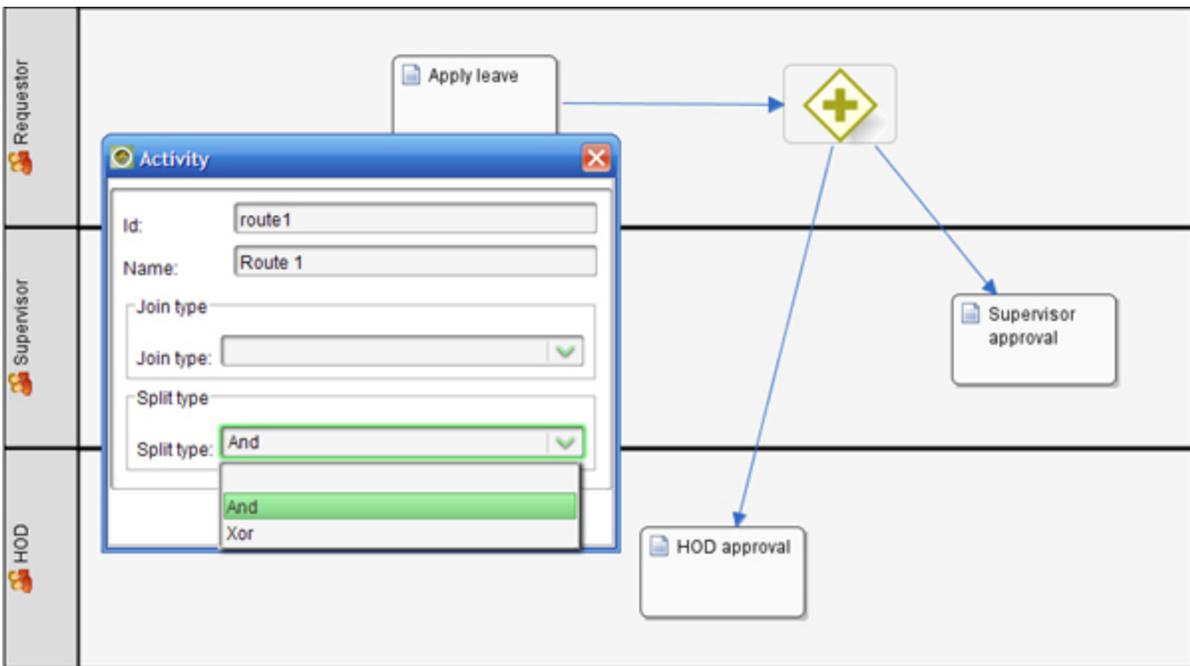
There is, however, a very important caveat to insert here. Based on the rules above, if the approval value is "accept", the application will route to the HOD. If the approval value is "reject", then an email will be sent. However, if a mistake is made in the form or during user input and the value of approval is neither accept or reject, the process will get stuck because there is no valid rule to follow. It is due to this that the best practices of decision routing recommends using the **Otherwise** operator in this given scenario. If we make a slight change to the setting above and put the transition to system as Otherwise, we'll end up with:



Note that effectively, the process remains unaltered. If the value of approval is "accept" it will route to the HOD. Any other value and the email will be sent.

Forks

A fork is used when your process needs to split ways concurrently, meaning that your process will be travelling along 2 separate paths (hence, "fork"). For instance, in our given example, once the requestor submits his leave application, a notification is sent to both his supervisor and HOD at the same time. If that is the case, we'll need to add a route, set the split type to "and" and our process will look like:



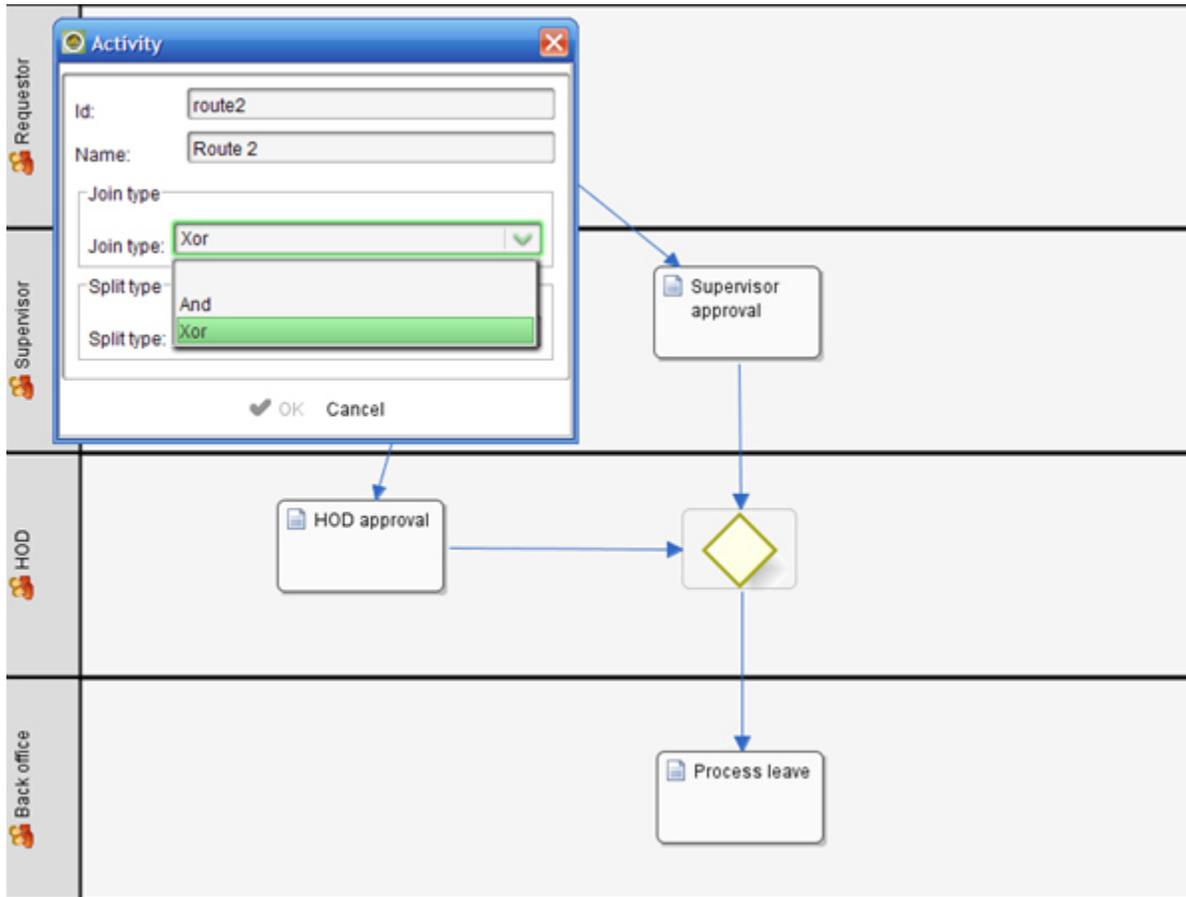
NOTE: If done correctly, a plus sign (+) will appear inside the route.

Joins

Joins usually occur when you have separate paths (usually caused by previous forks) in your process that you intend to merge. In our Leave Application process, once the requestor submits his leave application, both the supervisor and HOD are notified. Once they respond, the application ends up at the back office desk for further processing.

There are 2 types of joins:

1. XOR (or exclusive ORs in geek talk) means that the route is active and considered fulfilled once **any** of the transitions reaches it. So if you use the XOR option in your join route, the back office will be notified the moment either the supervisor or HOD responds.
2. AND means that the route will wait for **all** the transitions to be completed. So if you use the AND option, the back office will only be notified if and when both the supervisor and the HOD have responded.



To sum it up, once the requestor submits an application, the supervisor will decide on whether to accept or reject. If the supervisor accepts, the application will go to the HOD for approval. Two routes are needed here: one for when the supervisor approves and another for when the HOD approves. Again, while the position of the routes in specific swimlanes doesn't really matter, it is ideal to keep the process legible.

After adding the routes, the process should look like:

