Make the Right Checklist Apply Itself

Last Modified on 09/04/2026 11:04 am EDT

One rule on a checklist template decides which deals get which paperwork. Design around the three things a rule can't watch and it lands right every time.

Introduction

The moment your office handles more than one kind of deal, a single checklist stops fitting. Commercial needs documents residential doesn't. A lease isn't a sale. Your Texas office and your Arkansas office answer to different rules. So you end up with several templates and somebody adding the right one by hand — until they don't, and a file goes to closing with nothing tracking it.

An auto-apply rule ends that. It's a standing instruction on the template: *whenever a transaction looks like this, put this checklist on it.* A rule reads three things about a transaction: its status, the side your office represents, and its label. Use one or combine all three to be precise.

Most of the real work is upstream of the rule. Because those three fields are the whole vocabulary, how you've named your statuses and labels decides whether a clean rule is even available to write. Get that right and one rule covers a whole stage of your business.

When to Use This

- Your admin adds the same checklist by hand every time a deal goes under contract.
- Commercial and residential deals need different documents and one template is trying to serve both.
- You want a checklist on every Pending status you have, without writing a rule for each one.
- Two offices in two states need different paperwork on transactions in the same account.
- A checklist doubled up on a transaction after somebody changed its label.

Why This Watches a State, Not a Change

People come to auto-apply thinking of it as an alarm: *when a deal moves from Active to Pending, fire this checklist.* That's not what a rule is, and the distinction explains most of the surprises. A rule doesn't watch for an event — it watches for a condition, and it's satisfied whenever a transaction meets it, whether the deal was created that way or moved into it this morning.

That has two consequences worth planning for. The good one is that a transaction created directly in Pending gets its checklist, which an event-based trigger would have missed entirely. The one that bites is that the condition can be met more than once — a deal that leaves Pending and comes back gets the checklist a second time. So the design question isn't "what happens when," it's "what does a transaction look like when this paperwork is due" — and every workaround further down flows from answering that instead.

1. Split Your Checklists Along the Lines a Rule Can Read

Before you write a rule, look at your templates. A rule can only see status, side, and label, so a checklist that serves two different situations has no clean rule to sit on — and the fallback is agents deleting the half that doesn't apply, which nobody does reliably.

Several specific templates beat one large one. Build a template per real combination your office runs — pending listings, pending buyer-side, commercial, lease — so each transaction gets a list where every line applies to it. That's not extra work; it's the same tasks, sorted, and it's what makes the rest of this possible.

2. Set the Rule on the Template

The rule lives on the template itself, on the same screen where its tasks are built — turn it on, then set the conditions you want the transaction to meet. Every condition you add has to be true at the same moment, so a rule aimed at Pending status and Buying side waits for both.

Set a checklist template to add itself to a transaction the moment that transaction's status, side, or label matches the rule you choose.

See Checklist Auto-Apply Rules

> A rule fires every time its conditions are met, not just the first time. A deal that falls through and comes back, or a label edited on a live transaction, can bring a second copy of the same checklist with it. If you're seeing duplicates, start with Duplicate or Unwanted Checklists Are Auto-Applying.

3. Use a Status Category to Cover a Whole Stage

Most offices have more than one status meaning roughly the same thing — *Listing Pending*, *Contract Pending*, *Pending Short Sale*. Writing a rule per status name means maintaining three rules that must never drift apart. Aim at the status *category* instead and one rule catches every status filed inside it, including the one somebody adds next year.

That works best when the status names carry only the stage. Side is detected on its own and property type belongs on the label, so descriptors folded into a status name — *Pending Buyer Commercial* — are what force you back into one rule per name.

Statuses mark where each transaction stands in its lifecycle — so your team reads any file at a glance.

See Transaction Statuses

4. Keep the Label About the Property

Labels are the second half of a precise rule, and they stay useful exactly as long as they describe *what the property is* — Single Family, Condo, Commercial, Land. The temptation is to encode everything else in them too: financing, side, whether there's an HOA. Every extra dimension multiplies your labels and gives a transaction another way to trip a rule you didn't intend.

Labels classify your transactions by property type, and they drive the checklists Pipeline auto-assigns.

See Transaction Labels

5. Scope It to the Offices That Use It

A template belongs to the locations set on it, and that's what keeps one office's paperwork off another office's files. Offices in different states, or franchises with their own compliance packet, get their own location and their own templates — the rule then only ever reaches transactions created there. When a checklist really is company-wide, a master admin can widen it to every location with the all-locations switch instead of copying it around.

Build a new checklist template from scratch — name it, pick the locations that use it, and load it with the tasks a deal depends on.

See Adding a Checklist Template

6. Design Around the Three Things a Rule Can't Watch

Three asks come up constantly, and each has a standard answer that works better than the trigger people were hoping for.

You want it to fire on a status *change*. Rules read a state, not a transition. If the moment genuinely matters, give that moment a status or a label of its own — a *Compliance Review* status, a *Post-Closing* label — and aim the rule at that. The state now *is* the event.

You want it to fire on a date. Nothing dates a checklist onto a transaction. Move the timing down a level: apply the checklist early, on status, and let the tasks inside it carry relative due dates so they come due when they should. The checklist arrives quietly; the deadlines arrive on schedule. See Task Due Dates.

You want a different checklist per agent. Rules don't read who's on the file. Apply the same checklist and aim the *tasks* instead — visibility and @mentions decide who sees each line, which gets you a per-person list out of one shared template. See Task Visibility.

> Rules only look forward. Turning one on today does nothing for transactions that already passed through the qualifying status — those get the checklist added by hand. See Applying a Checklist.