

FACT-PATTERN RESEARCH

Fact-Pattern Decision Research with Docket Navigator

Find court decisions by describing a fact pattern, legal issue, procedural posture, or outcome instead of building filters by hand.

HOW TO USE THIS DOCUMENT

Upload this PDF to the AI assistant you use with the Docket Navigator connector and ask it to create a skill from it. The pages that follow are the complete skill: the numbered sections are its instructions, and its frontmatter is listed at the end. The assistant needs the Docket Navigator MCP connector connected and authenticated before the skill will run.

A researcher working a live matter thinks in facts and postures. They know their opponent's damages expert used whole-machine revenue, or that a survey they consider junk carried a preliminary injunction. They should not have to reverse-engineer which filter encodes that before research can begin.

Take the description they give you and turn it into structured Docket Navigator research. Docket Navigator supplies the data, the annotations, and the source links. Whether a decision is genuinely responsive is a judgment call, and it has to come from the record rather than from a case name that looks close.

1. Break the assignment into dimensions

Do this before calling anything. It takes a moment and it gives you a standard to check results against later.

DIMENSION	THE QUESTION	EXAMPLE
Facts	What happened?	Royalty ran off whole-machine revenue
Doctrine	What issue decides it?	Apportionment, entire market value rule
Vehicle	What filing produced the ruling?	Rule 702 motion to exclude
Direction	Which way did it come out?	Excluded or limited the opinion
Forum	Court, judge, or agency?	D. Del., or any district court
Time	Recency bounds?	2015 onward

Unspecified dimensions are fine. What matters is noticing which ones the user cares about, since a wrong result will be wrong about one of those. Ask only when a genuine ambiguity would change the result set, and ask once.

Listen for loose direction language. "Thrown out or cut back" covers partial grants, and a search limited to full grants drops the partial-grant orders that often show most precisely where a court drew the line.

2. Pick the library

Match subject matter to patent, trademark, copyright, tradesecret, antitrust, securities, upc, omnibus, or profile. Use omnibus when a matter crosses subject matter. Ask when the mapping is unclear, since library choice determines every target and criterion name downstream.

Target and criterion names differ between libraries. Confirm them through a discovery call for the library you are actually in rather than carrying a name over from another one.

3. Plan before building

Call `docnav_plan_search_workflow` and pass the user's own words into `question`. Paraphrasing into doctrine first throws away the facts the planner uses to route the request. Add `intent_hint` for the workflow family you expect.

Reach for `docnav_discover_search_options` when no planner candidate fits or a planned draft needs modifying.

4. Check the plan against Step 1

Read the returned motion category, criteria template, and definition, then confirm they describe the assignment you decomposed. Does the motion category match the procedural vehicle asked about? Does the analytical object match what the user is counting, given that motion decisions, documents, cases, and accusations carry different denominators?

A returned confidence score reports how well the request matched a recipe. It does not confirm the recipe answers the question. When the plan misses, say so and rebuild through discovery rather than proceeding quietly.

5. Resolve values, then validate one at a time

Use `docnav_lookup_search_values` for anything from a controlled vocabulary. Values are product ids rather than the names a human would type. Feed resolved ids back through the planner via `known_entities` so it composes target-specific shapes for you.

Two distinctions worth holding onto:

- Document Type is the procedural vehicle. Legal Issue is the doctrine. They are separate filters and swapping one for the other silently changes the question.
- Filtering on a motion Document Type returns the order deciding that motion, which is what makes this workflow work.

Build the search up through `docnav_validate_and_preview_search` one criterion at a time. When something breaks you will know which criterion broke it, and the count trajectory tells you whether each filter is biting. Read the returned filter description before trusting the count, since it is the connector restating your search in its own words.

6. Read results for substance

A returned row is a candidate. Use the annotation fields to see what the court actually did, then sort candidates into four buckets and carry the labels through to the report:

- **Directly responsive.** Same posture, same issue, same direction.
- **Contrary authority.** Same posture and issue, opposite direction.
- **Adjacent.** Related issue or different posture. Context rather than support.
- **Not verified.** The record does not confirm the match. Say so rather than dropping it or promoting it.

Run the opposite direction as its own search, holding every other criterion constant so the two sets stay comparable. A researcher who brings a one-sided list into a brief meets the other cases in the opposition's response.

Watch two things when reporting outcomes. How a motion came out is a different field from how an accusation resolved, so an order excluding an expert is a motion result and filtering on case outcome will miss most of them. And in patent matters the patentee is not always the plaintiff, since declaratory judgment actions, ITC investigations, and PTAB proceedings all invert or rename the roles.

7. Report so it can be checked

For each decision give the case name and citation, court, judge, date, procedural posture, what the court said was wrong or sufficient, and the Docket Navigator link. Keep sourced findings visibly separate from your own reading of them, since one of the two may end up in front of a court.

Quote the court's characterization where exact wording carries the holding, and keep quotations short. Paraphrase everything else.

Close with what the research did not establish: dimensions you could not filter on, searches that came back empty, coverage you are unsure of. A named gap is one the researcher can work around.

8. Refine rather than rebuild

Follow-ups like "just Delaware" or "only where the court actually excluded" are modifications to the search you already validated. Change one criterion and revalidate, preserving the target and field names. Rebuilding each turn risks returning a different set than the one being refined.

9. Save only when asked

Binders are saved content that colleagues may rely on, so create or edit one only on an explicit request to save. Offering is welcome.

Structure tabs around the questions asked: the broad responsive set, contrary authority, a recency cut, a forum cut. Someone picking the binder up later can then see the shape of the research. Binder results refresh as new data matches the saved criteria, so mention that the binder stays live.

Examples

Example 1: a fact pattern with no doctrine named

USER SAYS

"I'm defending a patent case over a big industrial machine. Their damages guy ran a royalty off the whole machine's revenue and never separated out what the one patented subsystem contributes. I want to move to exclude him."

Decompose first. Facts: whole-product revenue as the base. Doctrine: apportionment and the entire market value rule, neither of which the user named. Vehicle: a Rule 702 motion to exclude. Direction: excluded or limited, which covers partial grants. Forum: district courts. Time: unstated, so leave it open.

Then plan with the user's own wording, confirm the returned plan describes an expert exclusion motion rather than some other vehicle, resolve the doctrine and document type through lookups, and build the search one criterion at a time.

Read the annotations to see what each court actually held, sort into the four buckets, and run the denial side as its own search before reporting.

Result: responsive orders with citations and links, the contrary set alongside them, and a note on anything the record did not confirm.

Example 2: refining without starting over

USER SAYS

"Now just Delaware, and only the ones where the court actually excluded him."

Take the validated search from Example 1, add the court criterion, narrow the motion result, and revalidate. Preserve the target and field names.

Result: a narrowed set the user can trace back to the original research, rather than a fresh search that may not match what they were refining.

Troubleshooting

Tool calls fail, or no Docket Navigator tools are available.

The connector is not connected for this conversation. Have the user check Settings, then Connectors, and confirm Docket Navigator shows as connected. Do not substitute an answer from background knowledge; unsourced case law presented as retrieved research is worse than no answer.

Calls fail with an authentication or credential error.

The session has expired. The user needs to reconnect the connector.

A library or feature returns nothing at all.

Account entitlements vary. Say which library was unavailable rather than silently searching a different one.

A search returns zero results.

Filters combine restrictively, and stacked Legal Issue filters in particular tend to AND together, so an empty set is often expected rather than an error. Remove the most recently added criterion and add back one at a time to find which one is doing the cutting.

A decision the user knows exists is missing.

Docket ingest, document search, and editorial coding are separate layers and the later ones can lag. Open the case or proceeding by number rather than concluding the decision does not exist.

Any other tool failure.

Report what failed and what it blocks. If you fall back to sources outside Docket Navigator, label which findings carry a pinpoint citation and which do not, since checkable citations are the point of this workflow.

Skill frontmatter

These four fields are the skill's YAML frontmatter. Everything before this section is the skill body.

name	docket-navigator-decision-research
description	Find court decisions through the Docket Navigator MCP by describing a fact pattern, legal issue, procedural posture, or outcome instead of building filters by hand. Works across every Docket Navigator library including patent, trademark, copyright, trade secret, antitrust, securities, and UPC. Use whenever someone wants to locate decisions, orders, or rulings in litigation data, including phrasings like "find decisions where...", "has any court held...", "show me rulings on...", or any description of a litigation scenario they want authority for, even when they never name Docket Navigator, a filter, or a doctrine. Also use when refining a result set, explaining why a decision is responsive, finding contrary authority, or saving research to a binder. Do not use for legal questions answerable from background knowledge, for drafting briefs or client work product, or for research in other databases.
compatibility	Requires the Docket Navigator MCP connector, connected and authenticated. Account entitlements determine which libraries and features are searchable. No local scripts, packages, or network access are needed beyond the connector itself.
metadata	author: Docket Navigator · mcp-server: docket-navigator
