Service Network Coverage: Finding Contractor Coverage Gaps

MCP Server · https://www.easyterritory.com/documentation/ezt-mcp/service-network-coverage/

If your business relies on a network of contractors (technicians, repair shops, installers) to serve customers, the first question is always the same: **where are we short?** Service Network Coverage answers that question for one trade at a time. For every customer, claim, store, or ZIP you care about, it counts how many qualified contractors can reach it. Then it shows which places have none, which depend on a single contractor, and which are over-served.

The tool is `analyze_network_coverage`. It reads your data and reports. It does not assign work, move territories, or change your source layers.

Who uses it

- Home warranty and home services networks checking coverage for each trade
- Insurance repair and managed-repair programs checking shop coverage
- Facilities and retail service teams checking contractor coverage across a store portfolio
- Any operations team that needs a recruiting list or a single-contractor risk list

What you need

1. **A contractor layer.** One row per contractor location, with coordinates or an address. Load it with `ingest_accounts` and tell the server it is a contractor list, declaring the columns you'll use:
2. `layer_role: "contractors"`
3. `trade_field`: the trade or trades a contractor serves. A contractor listing several trades (for example, `HVAC, Plumbing`) counts for each one.
4. `tier_field` (optional): for example `Preferred` / `Standard`
5. `capacity_field` (optional): for example jobs per week
6. `service_zip_field` (optional): the ZIPs a contractor has agreed to serve

Declare these columns when you load the layer. Columns you don't declare are not kept.

1. **A demand layer.** The places that need service: customers, open claims, stores, or ZIP center points. Add a numeric demand column, such as requests per year or revenue, if you want gaps weighted by demand.
2. **A coverage rule.** You choose it; the server never guesses (see below).
3. **A trade.** One trade per run. Use `*` to ignore trade.

If anything is missing or ambiguous, for example two point layers and no indication of which holds contractors, the server replies with a question for you rather than guessing.

Choosing a coverage rule

Rule	Ask for it like this	How "reach" is measured	Good for
Drive-time limit	"…within 45 minutes"	A calibrated travel-model estimate (no provider calls) from the demand point to each contractor, converted to minutes. There is no per-call provider cost or origin limit.	First pass at any scale
Distance limit	"…within 40 km"	Straight-line kilometers	Simple radius policies
Drive-time areas	"…use real drive-time areas for these contractors"	Polygons on the Azure Maps road network, built first with `isochrone_build` (up to 25 contractors and 5 bands per call), then used by the coverage check	Confirming the gaps that matter on real roads
Service ZIPs	"…based on each contractor's service ZIPs"	A demand point is covered if its ZIP is in a contractor's declared list	Contractual coverage

A common pattern: run the drive-time limit across the whole network, then build real drive-time areas only for the contractors near the gaps and re-run with those areas. The result often suggests this next step for you.

The travel estimate is an estimate. It doesn't see rivers, lakes, or sparse rural roads. When you report results, say which rule you used.

What you get back

- **A coverage layer** (`<demand>_coverage_<trade>`). Every demand point, with:
- contractor count, covered (yes/no), and single-source (yes/no)
- a coverage class: **none** (red), **one** (amber), **two or more** (green), **saturated** (blue, at or above a threshold you can set)
- nearest contractor and estimated minutes
- the list of covering contractors
- a **gap priority** score from 0 to 100 (weights adjustable)
- a capacity ratio, when you declared a capacity column
- **A gaps layer** (`<demand>_gaps_<trade>`). Only the uncovered points, ready to share or export.
- **Optional gap territory.** Ask for it and the ZIPs whose demand is entirely uncovered are grouped into one territory you can view, export, or push to Territory Designer.
- **A summary** your agent reports in chat: coverage percentage, gap demand, single-source and three-or-more overlap counts, a coverage-depth histogram, reach per contractor, and the top gaps.

Re-running replaces the layers the tool wrote; your contractor and demand layers are never overwritten. To compare two rules side by side, export the first result before re-running.

Example prompts

- "Audit our accounts against our HVAC contractors with a 45-minute limit. Show me the gaps and the single-source areas."
- "Now only Preferred HVAC contractors, and widen it to 60 minutes."
- "Use each contractor's service ZIPs instead of drive time, for plumbing."
- "Group the ZIPs with no coverage into a gap territory and push it to Designer for the recruiting team."
- "Which contractors cover the most single-source accounts?"

Worked example (sample data)

In a sample run, 1,934 Texas accounts were checked against 12 synthetic contractors, 10 of whom do HVAC, using a 45-minute drive-time limit:

- About 60% of accounts were covered.
- 774 accounts had no HVAC contractor within the limit. They fell mostly in West Texas and the Panhandle.
- 807 relied on a single contractor.
- 353 had two or more.

The gaps layer held exactly the 774 red points. The original accounts and contractor layers were unchanged. Filtering the same run to Preferred contractors only, at 60 minutes, gives the preferred-network view.

What it doesn't do (yet)

- It doesn't assign jobs or contractors to areas. To divide work among people, use `auto_build`.
- It doesn't recommend which contractors to recruit, or rank contractors for a specific job.
- It doesn't compute a road drive-time matrix. Use drive-time areas for road-accurate checks.
- It doesn't route. Estimated minutes are not routed drive time or workload.

Next steps

- How EZT Measures Travel for travel estimate vs drive-time areas vs routes
- Points in a Territory to list the accounts inside a gap territory
- Pulling and Pushing Territory Designer Projects to load a contractor layer from Designer or publish gap territories
- Tools Reference → `analyze_network_coverage`, `ingest_accounts`