

Delivery & Dispatcher

How automatic order assignment decides which driver or carrier delivers which order, and how to configure it.

- [What the dispatcher does](#)
- [Dispatcher glossary](#)
- [The assignment cycle, step by step](#)
- [Re-checking orders already with a carrier](#)
- [When does an alternative exist?](#)
- [The three assignment mechanisms](#)
- [Last-Option: the final safety net](#)
- [Driver classification and priority](#)
- [Priority Customers \(Platinum\)](#)
- [When an order changes driver](#)
- [Scheduling plans: how the automatic cycle works](#)
- [Where each setting lives](#)
- [Quick reference rules](#)
- [Known system boundaries](#)
- [End-to-end overview](#)
- [Reporting a dispatcher issue](#)
- [Dispatcher FAQ](#)
- [Pausing and resuming delivery](#)
- [Delivery zones](#)
- [External carriers](#)
- [Store delivery settings](#)
- [Backups and restoring configuration](#)

What the dispatcher does

“ **Audience:** Operations and dispatch managers who configure, supervise, or explain automatic order assignment to their team. **Purpose:** Understand *what* the dispatcher does and *why* it makes a given decision.

The Dispatcher is the system that automatically decides **which driver (or external carrier) delivers which order**, and **when**. It runs on its own, on a schedule you control, and re-evaluates every store's pending orders each time it runs.

These pages explain the logic behind those decisions in business terms: what the algorithm optimises for, what determines priority, which settings are available to you, and what to expect — or not expect — from it.

How to read this chapter

You don't need to read it front to back. Most questions land on one page:

If you want to know...	Go to
What a term means	Dispatcher glossary
What happens on each run	The assignment cycle, step by step
Why an order stayed with an external carrier	Re-checking orders already with a carrier
Why one driver was chosen over another	Driver classification and priority
Why an order changed driver	When an order changes driver
Why an order was left unassigned	Last-Option: the final safety net and the Dispatcher FAQ
How often it runs, and how to start or stop it	Scheduling plans: how the automatic cycle works
Where to change a particular setting	Where each setting lives
The whole flow on one page	End-to-end overview
A rule, without the explanation	Quick reference rules
How to report something that looks wrong	Reporting a dispatcher issue

Once the dispatcher has made its decisions, the live board where a person can intervene is the Hub — see [The Hub: what it is and who can do what](#).

What it does not do

The dispatcher is deliberately bounded. It does not try to find a mathematically perfect plan across every driver and every order at once, it never touches an order that has already left the store, and it does not reorder work just because one order is more urgent than another. Those limits are listed in full under [Known system boundaries](#) — they are design decisions, not faults, and knowing them saves a lot of guesswork.

Dispatcher glossary

The terms below are used consistently throughout the Delivery & Dispatcher chapter.

Term	Meaning
Own Fleet / Own Drivers	Your restaurant's own delivery staff — delivery riders and, in some cases, kitchen staff who can also deliver
Third-Party Carrier	An external delivery provider (e.g. a courier platform) used when your own fleet isn't the best option
Priority Customers (Platinum)	An optional feature that gives certain orders — typically loyalty or VIP customers — a better chance of being delivered on time, without unfairly harming regular orders
PG Mode	A first-in-first-out (FIFO) assignment method: orders are assigned to the best available driver in the order they came in
Google OR-Tools	An advanced route-optimisation engine, normally used when there are more pending orders than available drivers, so routes can be planned more efficiently
Last-Option	A final, same-cycle safety net for orders that couldn't be placed by the normal process — see Last-Option: the final safety net
Idle	A driver with zero orders currently assigned
Waiting	A driver who has orders assigned but hasn't left the store yet
On The Way	A driver who has already left the store to deliver at least one order
At-Store Zone	Being within roughly 350 metres of the store — close enough to be treated as "at the store" for assignment purposes
Time To Branch	Estimated time for a driver to arrive back at the store
Time To Delivery	Estimated time for a driver to complete a delivery
Assignment Cycle / Run	One full execution of the algorithm across all your stores, triggered automatically on the schedule you define
Clubbing	Grouping more than one order onto the same driver or carrier for a single trip
Critical	An order that is still unassigned and already past its promised delivery time
Pending	An order that is still unassigned but not yet late

The assignment cycle, step by step

Every time the dispatcher runs, it processes each store independently, following five steps in order.

The five steps

#	Step	What happens
1	Fetch data	Pulls current orders, available drivers, and store and zone configuration. This is also when any order still sitting with a Third-Party carrier from a <i>previous</i> cycle gets re-examined — see Re-checking orders already with a carrier .
2	Cache check	If nothing has changed since the last run, the store is skipped rather than re-processed, to save time and resources.
3	Assign orders	The core decision-making step, covered in The three assignment mechanisms . This is where Third-Party, your own fleet and — optionally — Priority Customers logic all come together.
4	Post-process	Results are categorised (new assignments, still-pending orders, orders sent to Third-Party, and so on), orders that can be grouped onto the same driver are identified, and anything still unassigned gets one final relaxed attempt — see Last-Option: the final safety net — before being marked Pending or Critical.
5	Save and notify	Final assignments are saved, drivers are notified by push notification, and any relevant alarms are processed.

Step 2 is the reason a run can complete almost instantly for a quiet store. A skipped store is not an error, and nothing is lost — the next cycle picks it up the moment anything changes.

What "independently, per store" means in practice

Each store is solved on its own. A driver shortage at one store never pulls drivers from another, and a store whose data hasn't changed is skipped without affecting the rest. This is also why a report of a dispatcher problem should always name the store — see [Reporting a dispatcher issue](#).

Where the deeper detail lives

Steps 1 and 3 carry almost all of the decision-making, and each has its own page:

- Step 1's re-examination of carrier-held orders → [Re-checking orders already with a carrier](#)
- The test for whether a replacement is genuinely available → [When does an alternative exist?](#)
- Step 3's assignment logic → [The three assignment mechanisms](#)
- Step 4's final attempt → [Last-Option: the final safety net](#)

Re-checking orders already with a carrier

Before the dispatcher evaluates any *new* assignments, it first re-examines every order that is **already sitting with a Third-Party carrier from a previous cycle** — meaning a carrier has been given the order but hasn't yet left the store with it.

This happens during step 1 of [The assignment cycle, step by step](#), for every store, every cycle. It never touches brand-new orders or orders already assigned to your own fleet.

The three questions, in order

For each carrier-held order, up to three questions are asked. The first one that resolves the order stops the sequence.

1. Has the carrier been sitting on this order too long?

Governed by **Max time to accept** (set per store, per carrier — see [Where each setting lives](#)).

- Limit exceeded **and** a real alternative exists right now → the order is cancelled with the carrier immediately and freed for reassignment in this same cycle.
- Limit exceeded but **no alternative exists** → the order is left exactly as it is. Cancelling would only leave it worse off, with nowhere better to send it.
- Limit not reached → move to question 2.

2. Is the carrier's own estimate for reaching the store slipping too far?

Governed by **Max time to reach store** (per store, per carrier).

If the carrier's estimated arrival at the store has drifted past this limit, the order is cancelled immediately — regardless of whether an alternative is available. An unreliable estimate from the carrier itself is reason enough to give up on it.

3. Are we still inside the free-cancellation window, and does our own fleet have room?

Governed by **Free-cancellation window** (per carrier, company-wide).

If yes to both, the order is reset to look **unassigned** — only in the dispatcher's working memory. Nothing is saved, and nothing is cancelled with the carrier yet. The order simply **rejoins the normal assignment process for that store**, exactly like any other unassigned order in that cycle.

“ This is not a special "try own fleet" routine. From this point on, the dispatcher has no notion that the order was ever with a carrier — it goes through PG Mode, Google OR-Tools and Third-Party like everything else, under all the same rules.

If the normal process assigns an own driver to it, two things then happen in strict order:

1. A real cancellation is sent to the original Third-Party carrier.
2. **Only if that cancellation succeeds**, the new internal-driver assignment is saved.

If the cancellation fails, step 2 never happens: nothing changes and the order stays with the carrier. This avoids a confusing state where one order looks assigned in two places at once.

If the normal process doesn't place it with an own driver, the in-memory reset is discarded. The order goes back to exactly how it was — still validly assigned to the carrier — as if the reset had never happened. No cancellation is ever sent in this case.

Questions 1 and 2 look similar but compare different things

This is the single most common source of confusion on this page.

Question	What it actually compares	Consequence
Max time to accept	Real elapsed time — now, versus when the order was given to the carrier. Only fails once enough wall-clock time has genuinely passed.	Cancels, but only if an alternative exists
Max time to reach store	The carrier's own promise — their reported arrival estimate, versus when the order was given to them. It never looks at the current time at all. Can fail seconds after assignment, if the carrier's own estimate is already too far out.	Cancels regardless of alternatives
Free-cancellation window	Same style of comparison as <i>Max time to accept</i> (now versus assignment time), but with a different consequence: passing it cancels nothing by itself. It only opens the door to the no-commitment own-fleet attempt described above.	Enables the risk-free retry

What happens if the own-fleet attempt doesn't work out

The order is deliberately **skipped from that same cycle's Third-Party fallback pass** — it is not re-evaluated against carriers again in this run.

This is intentional. The order is already validly assigned to a carrier, so re-running the fallback would be redundant at best, and risks producing a second, conflicting carrier assignment for an order that already has one. It carries on with its existing assignment until the next relevant check — a future cycle, or an actual delivery-time failure.

Why this matters

Without this check, an order could sit with a slow or unresponsive carrier indefinitely, or stay there even after your own fleet became free to take it sooner.

The own-fleet attempt is deliberately risk-free: if it doesn't pan out the order is left exactly as it was, the carrier is none the wiser, and nothing is ever cancelled without a confirmed replacement already lined up.

When does an alternative exist?

The first question on [Re-checking orders already with a carrier](#) only cancels a slow carrier if there is a genuine alternative ready to take over.

That check has two independent halves — whether *your own fleet* could take it, and whether *another Third-Party carrier* could take it. They differ a lot in depth. **Either one being true is enough.**

Own fleet — one simple check

Check	What it verifies	Where it's configured
Free capacity slot	At least one own-fleet driver (in-house or own-fleet contractor) currently has room for one more order	Maximum orders per driver — see Where each setting lives

“ This is intentionally a quick, optimistic check. It only asks "does *any* driver have a free slot" — not whether that driver is close enough, idle, or actually able to take *this* order. It exists purely to answer "is it even worth cancelling the carrier's order?" The real evaluation happens afterwards, once the order rejoins the normal assignment process.

Third-Party — several checks, and only carriers further down the list

An important scoping rule first: this only looks at carriers that come **after** the currently-assigned one in that zone's configured carrier order. Carriers earlier in the list are never considered here, even if they would otherwise qualify. The search stops at the **first** carrier that passes every check below — it does not look for the "best" one.

For each candidate carrier, in order, all of the following must pass:

Check	What it verifies
Enabled & order-type match	The carrier's master switch is on, and it applies to this order's type and sales channel
Payment compatibility	Cash orders don't exceed the carrier's cash-on-delivery limit; card orders are only offered if the carrier accepts cards
Product restrictions	None of the order's products fall into a category this carrier is excluded from
Capacity at this store	The carrier still has a free slot in its shared per-store capacity pool
Distance	The store-to-customer distance is within the carrier's allowed range
Operating schedule	The current time falls within the carrier's configured working hours for this store

The first candidate to pass all six counts as "an alternative exists", and the search stops there.

“ **Deliberately not checked here:** the normal delivery-time-window check is skipped. At this point the system is only confirming that a carrier could *structurally* take the order — enabled, payment, products, capacity, distance, hours. Whether it could deliver it *on time* is evaluated later, if and when the order is actually offered to that carrier.

All of these settings live on the carrier's general or per-store configuration — see [Where each setting lives](#).

The three assignment mechanisms

Every order is ultimately resolved using one — or a combination — of three mechanisms.

Mechanism	When it's used
PG Mode (FIFO)	You have enough own drivers to cover pending orders
Google OR-Tools	There are more pending orders than own drivers, so routes need planning more efficiently
Third-Party Carriers	Your external delivery network

“ Which of PG Mode or Google OR-Tools is used for a given store is decided automatically, from the ratio of pending orders to available drivers — unless you force one specific mode in the scheduling plan. See [Scheduling plans: how the automatic cycle works](#).

How they combine

Zone configuration decides **whether Third-Party gets the first attempt** or only acts as a safety net. Everything else works the same either way.

Standard zone — own fleet first:

1. **Own solver** (PG Mode or Google OR-Tools).
2. **Idle driver retry** — free store drivers the solver didn't use.
3. **Third-Party fallback** — last resort.

Third-Party-first zone — carrier first:

1. **Third-Party carrier**, one attempt. If assigned, done.
2. **Own solver** (PG Mode or Google OR-Tools).
3. **Idle driver retry**.
4. **No further fallback** — the order already had its Third-Party turn.

The rule that keeps it fair

Each order gets **exactly one attempt with each resource type** — own fleet and Third-Party — per assignment cycle.

The zone configuration only decides *which one goes first*, never *how many* attempts an order gets:

Zone type	Attempt order	Total attempts
Standard	Own fleet → Third-Party	2
Third-Party-first	Third-Party → Own fleet	2

Always two. Never three, never one.

Before the order is left for the next cycle

There is one more relaxed attempt within the same cycle — see [Last-Option: the final safety net](#).

If the order is still unassigned after that, the next scheduled run starts completely fresh, including a brand-new Third-Party attempt. How soon that happens is set by the store's scheduling plan.

Last-Option: the final safety net

After the own solver and the Third-Party attempt have each had their one shot at an order, some orders may still be unassigned. Rather than leaving them for the next cycle straight away, the dispatcher makes one more attempt — per store, at the end of the same cycle — called **Last-Option**.

What problem it solves

Last-Option only looks at orders still unassigned after the normal process. Its goal is simple: avoid leaving an order completely unassigned if there is *any* reasonable way to place it, even if that means being more flexible than the normal rules.

How candidates are found

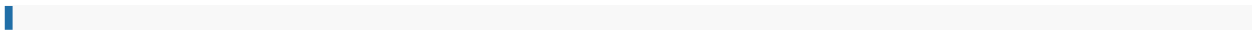
Candidates are sourced differently depending on where they come from, and Last-Option **relaxes some checks** the normal process enforces.

Source	How the candidate is chosen	What's relaxed
Own fleet (in-house and own-fleet contractors)	Best available driver, by capacity and how soon they could leave	The usual delivery-time window is not re-checked
Third-Party carriers	Any carrier already tried and rejected earlier in this same cycle	The original rejection reason is not re-checked — only whether capacity has since freed up

Whichever candidate can start the delivery soonest is the one chosen.

Order of processing

Orders are processed one at a time, **most urgent first**, by promised delivery time.



Priority Customers (Platinum) status does **not** affect which order gets a shot at Last-Option first. Platinum only affects the sequence of multiple orders already on the *same* driver's route.

Step by step

1. Collect the orders still unassigned after the own solver and the Third-Party attempt.
2. Sort them by promised time, most urgent first.
3. For each order in turn, gather candidates:
 - the best available own driver — capacity checked, delivery-time window not re-checked;
 - carriers rejected earlier this cycle, reconsidered only if capacity has since freed up.
4. Pick whichever candidate can start the delivery soonest.
5. If none is available, the order stays unassigned and is marked **Critical** or **Pending**.

If even Last-Option can't place it

The order is categorised as **Critical** (already past its promised time) or **Pending** (not yet late). The very next scheduled run tries again from scratch, including a brand-new Third-Party attempt.

“ **Trade-off to be aware of:** because Last-Option relaxes the normal delivery-time check for own drivers, an order placed through it may end up with a slightly later estimated arrival than the normal tolerance allows. This is intentional — better to get it moving, slightly delayed, than leave it unassigned entirely.

Driver classification and priority

Drivers are ranked by three things: their current status, their type, and how close they are to the store.

By status

Status	Description
Idle	Has zero orders currently assigned — fully available
Waiting	Has orders assigned, but none have left the store yet
On The Way	Has at least one order already out for delivery

By type

Type	Description	Priority
Internal delivery driver	Your dedicated delivery staff	Highest
Internal kitchen staff (can deliver)	Kitchen staff who can also deliver when needed	Medium
External / own-fleet contractor	Externally contracted own-fleet driver	Lower
Third-Party carrier	External delivery platform	Used per zone and configuration rules

Status and location combined

The "At-Store Zone" is roughly 350 metres from the store.

Category	Status	Location	Priority
1	Idle	At store	Highest
2	Idle	Away from store	High

Category	Status	Location	Priority
3	Waiting	At store	Medium
4	Waiting	Away from store	Lower
5	On The Way	Out delivering	Lowest

The reasoning is straightforward: an idle driver already at the store is the fastest possible option, so they are always preferred. A driver already out delivering is the last resort, since assigning them a new order means adding it to an existing route.

Where "at store versus away" actually matters

This distinction is not only used for the ranking above. It changes real decisions in several places:

- **Assignment speed.** An idle driver at the store is assigned instantly, with no further checks. An idle driver away from the store must first pass a delivery-time check.
- **Reassignment priority.** It is one of the main factors when deciding whether to move an order to a different driver — see [When an order changes driver](#).
- **Arrival tracking.** When a driver is confirmed at the store as an order is assigned to them, an arrival timestamp is recorded on that order.
- **Pickup Delay alarms.** The same determination triggers the alarm warning that a driver has been at the store too long without leaving with a ready order.
- **Frozen-driver detection.** A driver considered "at the store" is never flagged as stuck — being at the store is a valid, expected state.
- **Driver status in operational screens.** Labels such as "At Branch", "On The Way" or "Away" come from this same check.
- **Whether the dispatcher re-runs at all.** If no driver's at-store status has changed since the last run — along with everything else being unchanged — the store may be skipped.

See [The assignment cycle, step by step](#).

“ **Upcoming change, currently in internal testing.** Today, whether a driver counts as "at the store" uses an estimated **driving** distance compared against the ~350 m threshold, not a straight-line distance. This is being replaced by a straight-line ("as the crow flies") calculation, because a driving-route estimate can be inaccurate right around a store. The change is not yet active for all accounts; this page will be updated once it ships.

Priority Customers (Platinum)

“ This page only applies if your account has **Priority Customers (Platinum) evaluation enabled**. It is not part of the core flow — it runs as an extra layer on top of it, only for stores where it's switched on, and only when there are Platinum orders to consider.

What it does

When enabled, the dispatcher runs a **double simulation** for each store before committing to a decision:

1. **Normal run** — orders are processed in their regular order.
2. **Platinum-first run** — Platinum orders move to the front of the queue and get a little more delivery-time flexibility.

The two results are compared, and the dispatcher picks whichever one better protects Platinum orders **without excessively harming regular orders**.

How the choice is made

1. Are there at-risk Platinum orders in the normal run?
 - **No** → use the normal run as-is. The second simulation never runs.
 - **Yes** → continue.
2. Run the Platinum-first simulation.
3. Count how many regular orders would be *newly* harmed by it.
 - **Below the impact threshold** → use the Platinum-first result.
 - **At or above the threshold** → discard it and use the normal result.

A Platinum order counts as "at risk" if it wouldn't be assigned at all, or if its estimated delivery would fall outside an acceptable refund-risk window.

What you can configure

Setting	What it controls
Enable / disable	Turns the entire layer on or off
Buffer minutes	Extra delivery-time flexibility given specifically to the Platinum-first simulation
Impact threshold	Maximum number of regular orders allowed to be newly harmed before the Platinum-first result is rejected

What it deliberately does not do

Platinum protection works **only** through the double simulation above. There is no rebalancing of existing routes to make room for a Platinum order — that limit and the others are listed in [Known system boundaries](#).

Platinum status also plays no part in ordinary reassignment ([When an order changes driver](#)) or in the Last-Option queue order ([Last-Option: the final safety net](#)).

If the layer is not enabled, Platinum orders are treated exactly like any other order.

When an order changes driver

Before every assignment attempt, the dispatcher checks whether an order already assigned to one driver should instead go to a different one — but **only** on driver priority, never on how urgent or important the order itself is.

When a reassignment is allowed

Situation	Result
Order has no driver yet	Assign freely
Previously assigned driver is no longer available	Assign freely
Same driver being considered again	No change needed
New driver has better priority	Reassign
Current driver has equal or better priority	Keep as-is

Priority is defined in [Driver classification and priority](#).

Common scenarios

Current driver	Candidate driver	Outcome
External driver	Internal driver	Reassigned — better type
Idle, at store	Idle, at store	No change — both equally good
Idle, away from store	Idle, at store	Reassigned — better location
Waiting, at store	Idle, away from store	No change — current is still better
External, away from store	External, away from store	No change — no real improvement

Passing the priority check is not a delivery guarantee

Passing it only means the dispatcher will **attempt** the assignment. It still has to verify the new driver can actually deliver on time:

Driver situation	Delivery time checked?
Idle, at store	No — always the fastest possible option
Idle, away from store	Yes
Waiting (already has orders)	Yes — must be able to deliver everything
On The Way	Yes — the full route is recalculated

If the delivery-time check fails, the assignment is reverted and the order stays available for the next driver in line. **It isn't lost.**

Rules that never change

- Orders already out for delivery — the driver has left the store — are **never** reassigned.
- Manually assigned orders, assigned by an employee rather than the dispatcher, are **completely locked** from automatic reassignment.
- Internal drivers can take an order away from an external driver. The reverse is never allowed.

Scheduling plans: how the automatic cycle works

The dispatcher runs on a schedule you define per store, or per group of stores. Scheduling and assignment behaviour are configured together, in the same plan.

How the automatic cycle works

- Each plan defines how often the dispatcher should run for a group of stores.
- You can optionally restrict execution to a daily time window — for example, only between 08:00 and 23:00.
- If a run fails, the cycle is **not** broken. The next run is still scheduled automatically, as long as the plan is active.
- If a run falls outside the configured time window it is skipped for that cycle, but the schedule keeps going and resumes normally once back inside the window.

Scheduling settings

Setting	Default	What it does
Plan name	—	A label to help you identify this plan. Optional, free text
Stores	—	The stores this plan applies to. One plan can cover several
Active	Off	Whether the automatic cycle is currently running for this plan
Run interval	60 seconds	How often the dispatcher re-runs. Allowed range: 60–300 seconds
Daily time window	None	Optional start and end time. If not set, the plan runs 24/7
Last run	—	Read-only. When this plan last actually ran

Assignment behaviour settings

Setting	Default	What it does
Assignment mode	Automatic	Forces PG Mode or Google OR-Tools, or leaves the choice automatic (recommended)
Routing strategy	Automatic	Only relevant when Google OR-Tools is used
Stop time at customer	5 min	Extra time accounted for at each customer stop
Stop time at store	3 min	Extra time accounted for at each store stop
Maximum orders per driver	3	Most orders bundled onto a single own driver. Allowed range: 1-5
Offline driver grace period	5 min	How long a briefly-offline driver is tolerated before being treated as unavailable
Apply results	On	When off, the run is a simulation only and nothing is changed
Extra time margin	10 min	Additional slack allowed. Only relevant to Google OR-Tools
Minimum orders to accept a plan	0	How many orders Google OR-Tools must successfully assign for its plan to be accepted
Priority Customers (Platinum)	Disabled	Enables and configures the Platinum layer — see Priority Customers (Platinum)
Traffic-aware estimates	Off	More accurate time estimates, at a higher cost
Traffic estimate type	—	Optimistic, pessimistic or best guess. Only used when traffic-aware estimates are on
Return-time reuse	—	Advanced. Controls how aggressively previously calculated driver-to-store times are reused instead of recalculated

Store-level delivery settings

Two further settings live at the **store level**, not inside the scheduling plan. They are configured from the store's dispatcher settings and control how much delay is tolerated before an order is treated as at risk.

Setting	Applies to	What it does
Own-fleet delay tolerance	Your own fleet — in-house and own-fleet contractors, treated equally	The delay tolerance used when checking whether your own fleet can deliver on time. If a store has no in-house drivers at all, this value is automatically replaced by the Third-Party tolerance below, so an all-contractor store isn't held to a stricter standard than it can meet
Third-Party delay tolerance	Third-Party carriers	The delay tolerance used to decide whether a carrier can be trusted to deliver on time. It always applies to carriers, whether or not the store also has in-house drivers

“ **Capacity works differently from delay tolerance.** Your own fleet has a **per-driver** limit — each vehicle carries up to the configured maximum at once. Third-Party carriers have a **single shared pool per store, per carrier company** — for example "this carrier can handle 10 concurrent orders at this store", however many individual couriers they have on the road. This is why a carrier can appear full for a store even when it looks like it has plenty of drivers: the limit is set at company level, not per courier.

Activating and pausing a plan

This is the part most likely to cause confusion.

Action	Starts the cycle?	Stops the cycle?
Create a plan with Active switched on	Yes	—
Create a plan with Active switched off	No — it stays dormant	—
Edit an existing plan and switch Active on	No effect — it only updates the stored value	—
Edit an existing plan and switch Active off	—	Yes, from the next run onwards
Use the dedicated Activate action	Yes	—
Use the dedicated Pause action	—	Yes

Action	Starts the cycle?	Stops the cycle?
Delete a plan	—	Yes

“ **Recommendation:** don't rely on editing the Active field through a general edit form. Always use the dedicated **Activate** and **Pause** actions — editing the field directly can leave a plan marked active but not actually running.

Where each setting lives

Beyond the scheduling plan, several other parts of the product hold settings that directly affect dispatcher decisions. This page lists them **by where you'd actually go to change them**, and only includes settings you can edit.

For the scheduling plan itself, see [Scheduling plans: how the automatic cycle works](#).

Zones

Edited from the zone's driver-availability settings.

Setting	What it controls
Own vs. Third-Party priority	Whether orders in this zone go straight to carriers first, or start with your own fleet as usual
Third-Party ranking rule	When more than one carrier is available for the zone, the order they're tried in: your configured order, lowest cost first, or fastest time first
Allowed Third-Party carriers	Which carriers can be offered orders from this zone
Allowed own-driver types	Which of your own driver types can deliver in this zone. Leaving this empty blocks all own drivers from the zone entirely
Extra time buffer	Additional minutes added to the expected time at the customer's location for orders in this zone — for example, to account for difficult access

“ Zones also expose a **delivery-time deviation** field, along with pricing fields such as delivery fees and minimum order amounts. These are editable on the same screen but are **not used by the dispatcher** — the delivery-time deviation is consumed by the ordering website, not by the assignment logic.

Stores

Edited from the store's dispatcher settings.

Setting	What it controls
Own-fleet delay tolerance / Third-Party delay tolerance	See Scheduling plans: how the automatic cycle works for how these two work together
Kitchen-driver max distance	How far from the store a kitchen-staff driver can be sent. Set to zero to remove the restriction entirely

“ Stores also have opening and closing hours, per day and per shift, configured elsewhere in the product. Those are **not** used by the dispatcher.

Third-Party carriers — general settings

Edited from the carrier's general configuration. These apply company-wide, across every store.

Setting	What it controls
Enabled	Master on/off switch. While disabled, this carrier is never considered, at any store
Order type / sales channel	Restricts which order types and sales channels this carrier can be offered
Integration endpoint	The connection used to reach this carrier's platform. Normally set once, during onboarding
Default delivery cost	Used to estimate and rank this carrier when no live pricing is available
Minimum readiness buffer	How long an order must still need before it can be offered to this carrier at all
Max cash-on-delivery amount	Cash-paid orders above this amount won't be offered to this carrier
Accepts credit card	Whether this carrier can be offered card-paid orders
Excluded product families	Product categories — for example age-restricted items — this carrier will never be offered
Max distance	Farthest store-to-customer distance this carrier will be offered
Free-cancellation window	Minutes after assignment during which the order can still be pulled back without penalty. See Re-checking orders already with a carrier

Setting	What it controls
Description	Internal label for this carrier

“ A "default delivery time" setting also exists at this level, but it is no longer used. It moved to the per-store settings below so it can be tuned more precisely.

Third-Party carriers — per-store settings

Edited from the carrier's settings for a specific store. These add store-specific detail on top of the general settings above, for the same carrier.

Setting	What it controls
Max time to reach store	If this carrier's driver doesn't arrive within this time after accepting, the order is cancelled and reassigned
Estimated time to reach store (fallback)	Used when no live estimate is available for this carrier at this store
Max time to accept	If this carrier doesn't accept the order within this time, it's released for reassignment
Default delivery time	The delivery-time estimate used when no historical average is available for this carrier at this store
Clubbing allowed	Whether this carrier's driver can carry more than one order per trip at this store
Capacity at this store	A shared pool for this carrier at this store, not a per-courier limit
Operating schedule	Days and shifts during which this carrier is available at this store. Outside these hours it is skipped entirely

Quick reference rules

A one-page summary of the rules that come up most often. Each links to the page that explains it in full.

Rule	Behaviour
Orders already out for delivery	Never revalidated, never reassigned — only used to estimate when the driver returns
Orders still waiting to leave	Can be reassigned if a better driver becomes available
Reassignment checks	Run before every single assignment attempt
Reassignment criteria	Based only on driver priority — never on how urgent the order is
Manually assigned orders	Fully locked from automatic reassignment
Internal vs. external drivers	Internal can take orders from external. External can never take from internal
Attempts per cycle	Exactly one with your own fleet and one with Third-Party. The zone decides the order, not the count
Priority Customers (Platinum)	Not considered during regular reassignment — only inside its own optional evaluation layer
Order already with a carrier	Can still be pulled back to your own fleet before the carrier collects it, if inside the free-cancellation window and your fleet has room
A carrier order is never cancelled blindly	Nothing is cancelled without a confirmed replacement already lined up
Unassigned at the end of a cycle	Marked Critical or Pending, and retried from scratch on the next scheduled run

Where to read more

Topic	Page
Driver ranking	Driver classification and priority
Reassignment rules	When an order changes driver
Carrier pull-back	Re-checking orders already with a carrier
The final attempt	Last-Option: the final safety net

Topic	Page
Design limits	Known system boundaries

Known system boundaries

These are intentional design limits, not faults. Knowing them keeps expectations realistic and saves time when something looks wrong but isn't.

No proactive rebalancing

The dispatcher never moves a regular order out of a driver's route just to make room for a Platinum order. Platinum protection works only through the double simulation described in [Priority Customers \(Platinum\)](#).

Orders out for delivery are frozen

Once a driver has left the store with an order, that order is never touched again by the dispatcher. It is still used to estimate when that driver will be free, but it cannot be reassigned, revalidated or cancelled by the algorithm.

Decisions are local, not global

Each assignment decision compares the current driver against one candidate driver. The dispatcher does not attempt to find a perfect overall solution across all drivers and all orders at once.

This is why two decisions that each look right individually can, together, look less than ideal in hindsight.

Order urgency isn't a reassignment factor

Only driver priority matters when deciding whether to reassign. A very urgent regular order will not jump the queue over a less urgent one on its own. See [When an order changes driver](#).

Last-Option can exceed the normal delay tolerance

As a last resort, an order may be placed with a slightly later estimated arrival than the normal delay tolerance would otherwise allow, rather than being left unassigned. See [Last-Option: the final safety net](#).

A store with no change is skipped

If nothing has changed for a store since the last run, it is skipped rather than re-processed. A run that appears to have "done nothing" for a quiet store is behaving correctly — see [The assignment cycle, step by step](#).

End-to-end overview

The full path a store's orders take in a single cycle, pulling together everything in this chapter.

One cycle, one store

1. **Fetch orders and drivers** for this store. Orders already sitting with a Third-Party carrier from an earlier cycle are re-examined here — see [Re-checking orders already with a carrier](#).
2. **Is this a Third-Party-first zone?**
 - **Yes** → try a Third-Party carrier first. If it takes the order, that order is done.
 - **No** → go straight to step 3.
3. **Own solver** — PG Mode or Google OR-Tools, chosen automatically unless you force one.
 - If Priority Customers (Platinum) is enabled and applicable, this step is replaced by the double simulation described in [Priority Customers \(Platinum\)](#). Everything before and after stays the same.
4. **Idle driver retry** — free store drivers the solver didn't use.
5. **Third-Party fallback**, but only for orders from a standard zone. An order from a Third-Party-first zone has already had its carrier turn and gets no second one.
6. **Post-process** — categorise results and identify orders that can be grouped onto one driver.
7. **Still unassigned?**
 - **No** → save and notify.
 - **Yes** → the final relaxed attempt, described in [Last-Option: the final safety net](#). If it places the order, save and notify. If not, the order is marked Critical or Pending and the next cycle retries from scratch.
8. **Save and notify** — assignments are written, drivers are pushed a notification, alarms are processed.

The two things worth remembering

Every order gets exactly two attempts per cycle — one with your own fleet, one with Third-Party — plus the Last-Option safety net at the end. The zone configuration changes the order of those two attempts, never the number.

Nothing is ever cancelled without a replacement lined up. An order already with a carrier is only pulled back once a real own-fleet assignment has been found and the cancellation with the carrier has succeeded.

Related pages

- [The assignment cycle, step by step](#) — the five steps in detail
- [The three assignment mechanisms](#) — how PG Mode, Google OR-Tools and carriers combine
- [Last-Option: the final safety net](#) — what happens at step 7
- [Known system boundaries](#) — what this flow deliberately does not do

Reporting a dispatcher issue

If something looks wrong — an order assigned to the "wrong" driver, delayed, or stuck unassigned — the fastest way to get it resolved is to report it with the right information the first time.

Screenshots alone, especially of unrelated screens, rarely help. Specific, matched details do.

What to include

Detail	Why it matters
Request ID	Identifies the exact dispatcher run involved. Shown in the execution summary on the plan's detail view in the Scheduler, and looks like <code>req_1785152470621_6v3uobxk86e</code> . This is the single most useful thing you can provide — more so than the store, because it points to one exact run
Order ID / ticket number	Without it, the affected order can't be traced
Store	Which store's run is affected
Date and approximate time	Narrows down which cycle to examine
What you expected	For example: "should have gone to an idle driver at the store", "should have stayed with the original driver"
What actually happened	For example: "stayed unassigned for 12 minutes", "was reassigned to a different driver", "went to a carrier instead of our own fleet"
One-off or recurring?	A single order and a pattern across many orders point to very different causes
Was it a Priority Customer (Platinum) order?	Helps rule the optional Platinum layer in or out
Any recent configuration changes	A zone, store or carrier setting changed shortly beforehand helps separate cause from coincidence

What evidence actually helps

- A screenshot of the **specific order's detail screen**, showing timestamps and current assignment — not an unrelated dashboard or list view.
- If several orders are affected, list **all** the order IDs rather than describing them generally ("several orders around lunchtime").

- If you think a particular driver or carrier was skipped, name them, rather than saying "it didn't pick the right one".

What to avoid

- Reporting without the Request ID or the order ID. These are the two things nothing else can substitute for.
- General statements such as "the algorithm is broken" or "it's not working today", without a specific order, store and time.
- Attaching many unrelated screenshots instead of the one screen relevant to the order.

Before you report

Two behaviours are frequently mistaken for faults. Checking them first can save a round trip:

- A store where nothing has changed is **skipped**, not processed — see [The assignment cycle, step by step](#).
- Editing a plan's Active field does **not** start or stop the cycle. Use the **Activate** and **Pause** actions — see [Scheduling plans: how the automatic cycle works](#).

“ The clearer and more specific the report, the faster it can be matched against the actual run that produced the behaviour you're describing.

Dispatcher FAQ

Why did an order suddenly get reassigned to a different driver?

A driver with better priority became available — for example an idle driver at the store, or an internal driver where before only an external one was free. See [When an order changes driver](#) for the exact rules.

Why is an order still unassigned after a run?

Under normal conditions every order should end up assigned: to your own fleet, to a carrier, through a fallback pass, or via the Last-Option safety net. An order still unassigned at the end of a cycle usually points to one of these:

- **Genuine capacity shortage** — no available driver, own or Third-Party, can reach the customer within the accepted delivery window.
- **Zone / driver-type mismatch** — the order's zone doesn't allow the type of drivers currently available. Note that leaving a zone's allowed own-driver types empty blocks all own drivers from it entirely.
- **Incomplete order data** — for example missing or invalid delivery coordinates.
- **A technical problem during that run** — worth flagging if it repeats for the same store.

It will be retried automatically on the very next cycle, but if it happens often it's worth investigating rather than treating as routine.

What is Last-Option, and does it mean my order gets worse service?

Last-Option is a final, same-cycle safety net for orders the normal process couldn't place. It relaxes the usual delivery-time check to avoid leaving an order completely unassigned, so it may accept a slightly later estimated arrival — but only when no better alternative exists, and only after the normal process has had its full attempt. See [Last-Option: the final safety net](#).

Why can't I turn a plan on or off just by editing it?

Editing a plan only updates its stored configuration. It does not start or stop the automatic cycle. Always use the dedicated **Activate** and **Pause** actions. See [Scheduling plans: how the automatic cycle works](#).

What does enabling Priority Customers (Platinum) actually change?

It adds an evaluation step that tries to protect Platinum orders from being delivered late — but only if doing so doesn't harm too many regular orders at the same time. If it isn't enabled, Platinum

orders are treated exactly like any other order. See [Priority Customers \(Platinum\)](#).

How often does the dispatcher actually run?

Whatever run interval you set on the plan — minimum 60 seconds, maximum 300. If a run fails, the schedule doesn't break; the next run is still scheduled normally.

Can an order already with a Third-Party carrier move back to my own fleet?

Yes, but only before the carrier's driver has left the store with it, and only if two things are true: the order is still inside the carrier's free-cancellation window, and an own driver with capacity is available right then. Even so, the switch only happens if the cancellation with the carrier succeeds first. If any of that isn't met, the order simply stays with the carrier and nothing changes. See [Re-checking orders already with a carrier](#).

A carrier looks like it has plenty of drivers, so why is it "full" for my store?

Carrier capacity is a single shared pool per store, per carrier company — not a per-courier limit. See the capacity note in [Scheduling plans: how the automatic cycle works](#).

Why did a run appear to do nothing for one of my stores?

If nothing has changed for that store since the last run, it is deliberately skipped rather than re-processed. See [The assignment cycle, step by step](#).

Pausing and resuming delivery

“ **You need:** *Pause / resume delivery* — held by `admin`, `hub_manager`, `dispatcher_manager` and `branch_manager`. Viewing the status only needs store read access.

Pausing delivery stops a store taking new delivery orders for a set time, without closing the store or touching orders already in flight.

Pausing a store

1. Open the store's delivery controls.
2. Choose **Pause** and set the **duration in minutes**.
3. Confirm.

The pause is time-boxed: when the duration elapses, delivery resumes on its own. You don't have to remember to turn it back on.

“ If the store is **already paused**, pausing again doesn't stack. You'll be told it's already paused rather than silently extending it — check the current status first if you intend to extend a pause.

Resuming early

Resume ends the pause immediately, before the duration is up. Use it when the reason for the pause clears sooner than expected — a kitchen backlog that resolves, drivers coming back on shift.

Checking the current status

The delivery status view answers two questions: whether the store is paused right now, and if so what is scheduled to bring it back. Check it before pausing, so you don't mistake an existing pause for a new one.

Active pause jobs

Each pause creates a job that will resume the store when its time is up.

- **Per store** — see the active job for one store, and when it will resume.
- **Across the chain** — see every active pause job at once. Useful during an incident, when several stores were paused at different times by different people.

Emergency: cancelling all pauses

There is a chain-wide action that **cancels every active delivery pause job at once**.

“ This is an emergency control, not a routine one. It affects every store that is currently paused, including stores paused by other people for reasons you may not know about. Check the chain-wide job list first so you know exactly what you are about to resume, and tell the affected stores.

Use it when pauses were applied in bulk by mistake, or when an incident is over and you need everything back online without visiting each store.

What pausing does not do

- It does **not** stop the dispatcher for that store. Orders already accepted continue to be assigned normally — see [The assignment cycle, step by step](#).
- It does **not** change the store's opening hours, which are configured elsewhere and are not used by the dispatcher.
- It does **not** cancel or reassign existing orders.

What you can and can't see

The per-store actions are filtered by your scope: you can only pause, resume or inspect stores within it. The chain-wide job list and the emergency cancel are not store-specific, so they are limited to roles that hold delivery control without a store restriction.

Delivery zones

“ **You need:** *Delivery zones — view* to browse, *Delivery zones — edit* to change anything. Held by `admin`, `dispatcher_manager` and `polygon_manager`.

A zone is a delivery area with a map polygon and a set of rules deciding which drivers and carriers serve it. Zones are the single biggest lever on dispatcher behaviour — see [Where each setting lives](#) for what each rule does.

Browsing zones

The zones list shows every zone for the brand, with its store and whether it has a polygon drawn. A separate view shows zones alongside their **driver availability** configuration, which is where you see at a glance whether a zone is set to try carriers first.

“ The map view needs a configured Maps key for the brand. If the map is blank while the list shows zones, that's a configuration issue rather than missing zones.

Editing a zone's driver availability

This controls the rules the dispatcher actually uses: own-fleet versus carrier priority, the carrier ranking rule, which carriers are allowed, which own-driver types are allowed, and the extra time buffer.

“ **The trap worth repeating:** leaving *allowed own-driver types* empty does not mean "all types". It blocks every own driver from the zone, and orders there will only ever go to carriers. This is the most common cause of a zone whose orders all end up external.

Drawing and removing polygons

A zone's polygon defines its geographic boundary.

- **Create or update** a polygon by drawing it on the map and saving.
- **Delete** a polygon to leave the zone without a boundary.

A zone with no polygon cannot match an address, so orders won't be routed to it. Deleting a polygon is therefore effectively switching the zone off, even though the zone itself still exists.

Moving a zone to another store

A zone — together with its polygon — can be migrated to a different store. Use this when a store relocates or when an area is reassigned between stores, rather than deleting the zone and redrawing it, which loses the boundary.

Bulk changes by file

For anything beyond a handful of zones, use the file-based tools.

Action	What it does
Export zones	Downloads the current zone configuration as a CSV. Always export before a bulk change — it is your rollback
Import zones	Applies zone configuration from a CSV
Import polygons from KML	Bulk-loads zone boundaries from a KML file, the format mapping tools export

KML import is partial by design: valid rows are imported even when others fail. You get what worked plus a report of what didn't, rather than an all-or-nothing rejection. Read the report — a partially successful import leaves the rest of your zones untouched and still needing attention.

Time-based zone switching

Zones can switch on a schedule, so an area is served differently at different times of day. These rules are loaded from a CSV.

Unlike the KML import, **this import is all-or-nothing**: if any row is invalid, nothing is applied. Fix the file and re-import.

Before and after a bulk change

1. **Export first.** The export is the only rollback you have.
2. Apply the import.
3. Read the result carefully — especially for KML, where partial success is normal.
4. Check a sample zone on the map.
5. Confirm with a real order in that area, if you can.

Zone backups are kept and can be restored — see [Backups and restoring configuration](#).

External carriers

“ **You need:** *Third-party — view* to browse, *Third-party — edit* to change. Held by `admin`, `dispatcher_manager` and `third_party_manager`. Per-store carrier settings are also reachable by `hub_manager`.

Carrier configuration exists at two levels, and knowing which one you're editing is most of the battle.

Level	Applies to	Typical settings
Brand	Every store in the brand	Whether a carrier is enabled at all, payment rules, product exclusions, maximum distance, free-cancellation window
Store	One carrier, at one store	Timing limits, capacity at that store, operating schedule, clubbing, default delivery time

Both levels are listed field by field in [Where each setting lives](#).

Brand-level configuration

The brand view lists every carrier type available to you and its current configuration. Editing here changes that carrier **everywhere**.

“ The master **Enabled** switch is brand-wide. Turning a carrier off here removes it from every store immediately, whatever the per-store settings say. If you only want to stop using a carrier at one location, change that store's settings instead.

Store-level configuration

The per-store view shows the same carriers with their settings for that one store. This is where you tune the things that genuinely differ by location: how long a carrier gets to accept, how long they

get to reach the store, how many concurrent orders they can hold there, and the hours they operate.

Two of these drive the behaviour described in [Re-checking orders already with a carrier](#) — *Max time to accept* and *Max time to reach store*. They look similar and compare completely different things; that page has the comparison.

Capacity is per company, not per courier

A carrier's capacity at a store is a **single shared pool for that carrier company**. "Ten concurrent orders at this store" means ten, however many couriers that company has nearby.

This is the usual explanation when a carrier looks busy on their own app but appears full to the dispatcher.

Bulk changes by file

For chain-wide carrier work, use the file tools rather than editing store by store.

Action	What it does
Export per-store carrier configuration	Downloads current per-store settings as CSV. Export before every bulk change
Import per-store carrier configuration	Applies per-store settings from CSV

Backups of carrier configuration are kept and restorable — see [Backups and restoring configuration](#).

When a carrier is never offered orders

Work through this before reporting a problem. A carrier is skipped entirely if any of these is true:

- It is disabled at brand level.

- The current time is outside its operating schedule for that store.
- Its capacity pool at that store is full.
- The order's zone doesn't list it among the allowed carriers — see [Delivery zones](#).
- The order fails one of its structural checks: payment type, an excluded product family, or distance.

The full ordered list of those checks is in [When does an alternative exist?](#)

Store delivery settings

“ **You need:** *Store config* — *view* to browse, *Store config* — *edit* to change. Held by `admin`, `hub_manager`, `dispatcher_manager` and `branch_manager`.

Each store carries its own dispatcher settings, separate from the scheduling plan that governs when the dispatcher runs. Those two are configured in different places and are easy to confuse — the plan decides *when and how* the dispatcher runs, the store decides *how tolerant it is* for that location.

Store list

The list shows every store within your scope. A `branch_manager` sees their own store; a chain-wide role sees all of them. If a store you expect is missing, that is a scope question rather than a configuration one — see [How scope works in practice](#).

Dispatcher settings for a store

Two settings matter most, and both are explained in full in [Scheduling plans: how the automatic cycle works](#):

- **Own-fleet delay tolerance** — how much delay is accepted when checking whether your own drivers can deliver on time.
- **Third-party delay tolerance** — the same, for carriers.

“ There is a rule here that surprises people: if a store has **no in-house drivers at all**, the own-fleet tolerance is automatically replaced by the third-party value. An all-contractor store is not held to a stricter standard than it can meet. You don't configure this — it just happens.

A third setting, **kitchen-driver max distance**, caps how far a kitchen-staff driver can be sent from the store. Setting it to zero removes the restriction entirely rather than blocking all kitchen

drivers.

Promised delivery times

Each store exposes the delivery times it can promise a customer. These feed the time a customer is quoted at ordering, and the promised time the dispatcher then works against when deciding what is late — see the definitions of Critical and Pending in the [Dispatcher glossary](#).

Products and promotions per store

A store's view also lists the products and promotions currently active there. This is a read-only cross-check: useful when an order behaves unexpectedly and you want to confirm what that store actually offers, rather than what the brand catalogue says.

What belongs elsewhere

Looking for	Actually lives in
How often the dispatcher runs for this store	The scheduling plan — Scheduling plans: how the automatic cycle works
Which drivers and carriers serve an area	The zone — Delivery zones
Carrier timing and capacity at this store	Carrier per-store settings — External carriers
Pausing deliveries temporarily	Pausing and resuming delivery
Store opening hours	The general store configuration. Not used by the dispatcher

Backups and restoring configuration

“ **You need:** the view or edit permission for whatever the backup contains — not a separate backup permission. A zones backup needs delivery-zone access; a carrier backup needs third-party access; an employee backup needs employee access.

Every time you export configuration, a backup is kept. You can list them, download them, restore one, or delete it.

What gets backed up

Backup type	Covers	Permission it follows
Zones	Delivery zone configuration	Delivery zones
Schedule zones	Time-based zone switching rules	Delivery zones
External carriers	Per-store carrier configuration	Third-party
Employees	Employee and own-driver configuration	Employees

Because each backup follows the permission of its own content, someone who can restore a zones backup cannot necessarily restore an employees one. Access is decided by what the backup *contains*, checked when you open it.

Browsing backups

The list can be filtered by type. Each entry shows what it covers and when it was taken. Filenames follow the pattern `backup-<type>--<date>.csv`, so a downloaded file stays identifiable after it leaves the console.

Downloading

Download gives you the CSV snapshot exactly as it was captured. Two good reasons to download rather than restore:

- **To inspect before restoring.** Open the CSV and confirm it holds what you expect, especially if the backup is old.
- **To compare.** Diffing yesterday's export against today's answers "what actually changed?" faster than reading screens.

Restoring

Restoring re-applies a captured snapshot.

“ **Read this before restoring.** A restore reinstates the configuration as it was at capture time. Anything changed since — by you or anyone else — is overwritten for the entities in that snapshot. If two people have been editing zones this week, restoring last week's backup discards both sets of changes, not just the one you meant to undo.

The safe sequence:

1. **Export now**, creating a fresh backup of the current state. This is your way back.
2. **Download the backup you intend to restore** and read it.
3. Restore.
4. Verify on a real zone, store or carrier that the result is what you wanted.

Deleting

Deleting removes a backup permanently. Do it to clear out noise, not to tidy up after a change — the older backups are exactly what you need when a problem is noticed late.

Before deleting, check it isn't the last known-good snapshot from before a significant change.

Related pages

- [Delivery zones](#) — where zone exports and imports are done
- [External carriers](#) — carrier configuration exports
- [Where each setting lives](#) — what each restored field actually controls