

Data Dictionary

How to read every column in the downloadable .xlsx — with the exact formulas that produce the weight, volume, and utilisation numbers.

1. Overview

The download is a single .xlsx workbook with six sheets. Each sheet is a different lens on the same underlying data:

- **Reports** · one row per trip date — daily rollup.
- **Summary** · one row per trip — trip-level metrics.
- **Details** · one row per hop event — driver movements between stops.
- **Invoice Level Details** · one row per order — the per-invoice audit trail.
- **Delivery Aging** · one row per order, bucketed by how long it took to close.
- **Vehicle Raw** · your fleet master, for reference.

Where you get it: `/wd/reports` → set the date range → **Download report (.xlsx)**. Everything in this document maps to columns of that file, from the same date range you picked in the UI.

2. The weight formula — one place, three sheets

Every kg you see in the workbook — trip totals in Reports/Summary, per-order weight in Invoice Level Details, hop weight in Details — resolves to the same per-SKU-line calculation. Learn this once and every weight cell becomes readable.

The per-line calculation

For each row on the order (one row per SKU on the invoice):

$\text{UOM} = \text{CFC} \rightarrow \text{invoice_qty} \times \text{sm.weight_per_cfc_kg}$

$\text{UOM} = \text{PAC} \rightarrow \text{invoice_qty} \times \text{sm.weight_per_cfc_kg} \div \text{sm.pieces_per_cfc}$

CFC = case (a full carton). PAC = individual pack (loose unit inside the case). `weight_per_cfc_kg` = master-defined weight of one full case. `pieces_per_cfc` = how many PACs fit in one CFC.

Where the values come from

- **invoice_qty** · from the Trishul CSV → stored in `order_item.invoice_qty`.
- **invoice_uom** · from the Trishul CSV → stored in `order_item.invoice_uom` as either CFC or PAC.
- **weight_per_cfc_kg** · from the SKU master → `sku_master.weight_per_cfc_kg`.
- **pieces_per_cfc** · from the SKU master → `sku_master.pieces_per_cfc`.

What gets excluded

Lines with `order_item.pending_stock = 1` — items the WD couldn't ship at import time — are not counted in any weight number. They're part of the order but not part of the load.

When a weight cell reads 0 or looks low

Almost always a gap in `sku_master`. If `weight_per_cfc_kg` is NULL or 0 for a line, that line contributes zero. Check `/admin/sku-master` (filter: Missing weight) and upload a corrections CSV.

3. The volume formula

For questions: reach out to the RouteSync team.

Volume works the same way but uses per-PAC volume as the atomic unit, then expands CFC lines into their PAC count:

UOM = PAC → $\text{invoice_qty} \times \text{sm.vol_per_pack_m3}$

UOM = CFC → $\text{invoice_qty} \times \text{sm.vol_per_pack_m3} \times \text{sm.pieces_per_cfc}$

Same pending_stock = 0 exclusion applies. Same "0 for missing master data" behaviour — if vol_per_pack_m3 is missing, that line contributes zero volume.

Unit: cubic metres (m³). To convert to litres, multiply by 1,000. The Vehicle Raw sheet stores capacity in Volumetric Capacity (L) — that's already converted.

4. Sheet 1 · Reports — daily rollup

One row per trip date. Aggregates every trip that ran that day.

Column	Label	Source / joins	How it is calculated
A	Trip Date	trip.trip_date	The scheduled date of the trip. All rows for the same date are aggregated into one line.
B	Number of Trips	COUNT(trip.*)	Trips with status IN (dispatched, in_progress, completed).
C	Number of Vehicles	COUNT DISTINCT vehicle_id	Distinct vehicles that appeared on any trip that day.
D	Total Trip (Km)	SUM(trip.actual_distance_km)	From driver-app odometer or Google-Distance-Matrix depending on trip type.
E	Delivered	stop counts	Orders on stops where status IN (delivered, partial).
F	Returned	stop counts	Orders on stops where status = undelivered.
G	Total Delivery Attempt	Delivered + Returned	Everything the driver attempted to deliver.
H	Net Sale Value	SUM(order.invoice_value_rs)	Delivered/partial orders only. Verbatim CSV Total Amount, post-discount + post-tax.
I	Return Value	SUM(order.invoice_value_rs)	Undelivered orders only. Same source column.
J	Total Net Sale	Net Sale + Return Value	Total ₹ value attempted, regardless of outcome.
K	Total Delivery Weight	weight formula → SUM	Sum of order weight across every delivered / partial stop. Undelivered stops NOT included. Rounded to 2 dp.
L	Vehicle Weight Capacity	SUM(vehicle.max_weight_kg)	Sum of max_weight_kg for the distinct vehicles used that day.

Column	Label	Source / joins	How it is calculated
M	Weight Utilization %	$K \div L \times 100$	"How full were the trucks?" Below 40% = trips are half-empty. NA if no vehicles.
N	Time Utilization %	per-trip average	AVG(planned_duration_min ÷ actual_duration_min × 100), capped at 100 per trip.
O	Delivery Utilization %	per-trip average	AVG(actual_stops ÷ WD standard target × 100). Standard target from WD settings, default 30. NOT capped.

Column K deep-dive — Total Delivery Weight

For each trip on the date, RouteSync walks every stop whose **trip_stop.status** IN (delivered, partial), looks up the orders attached to that stop (**trip_stop.order_ids_json**), sums the weight of each order (using the formula in §2), and adds that to the date's bucket. Then rounds to two decimal places.

Rule of thumb

Pair Weight Utilization (col M) with Time Utilization (col N). A day at 40% weight but 95% time means the routes are geographically inefficient, not under-loaded — the trucks are running around lightly loaded.

5. Sheet 2 · Summary — one row per trip

For when you need to look at a specific trip — driver X ran late on the 15th, etc.

Column	Label	Source / joins	How it is calculated
A	Trip Date	trip.trip_date	The trip's scheduled date.
B	Trip Code	trip.code	AR-... for auto-route, TR-... for manual, plain text in the cell.
C	Trip Number	trip.id	Internal DB id — stable across renames.
D	Buyer Name	wd.name	Constant per WD. Duplicated on every row for readability.
E	Start Date & Time	trip.actual_start	When the driver tapped Start. NA if trip never started.
F	End Date & Time	trip.actual_end	When End trip was tapped. NA if still open.
G	Time Taken	End – Start	Formatted HH:MM:SS. NA if either endpoint missing.
H	Total Travel Time	Time Taken – Unloading	Time spent driving vs. time at stops.
I	Total Unloading Time	SUM per stop	Σ (actual_depart_at – actual_arrival_at) across all stops.
J	Total Trip (Km)	trip.actual_distance_km	Odometer-based when populated; auto-route planner-based otherwise.
K	Total Delivered	stop counts	Orders on delivered/partial stops.

Column	Label	Source / joins	How it is calculated
L	Total Returned	stop counts	Orders on undelivered stops.
M	Total Delivery Attempt	K + L	What the driver attempted.
N	Net Sale Value	SUM(order.invoice_value_rs)	Delivered/partial only.
O	Return Value	SUM(order.invoice_value_rs)	Undelivered only.
P	Delivered Total Weight	weight formula → SUM	Sum of order weight for delivered/partial stops on THIS trip. Rounded to 2 dp.
Q	Returned Total Weight	weight formula → SUM	Sum of order weight for undelivered stops on THIS trip.
R	Total Net Sale	N + O	Value handled by this trip regardless of outcome.
S	Total Delivery Weight	P + Q	Delivered + Returned weight. Everything the truck carried. Do NOT confuse with column P.
T	Vehicle	vehicle.reg_no	The truck used, e.g. TS-30-T-1264.
U	Driver	user.name	Driver who ran the trip.

Column P vs Column Q vs Column S

- **P Delivered Total Weight** · the weight that actually made it to the retailer (delivered + partial stops).
- **Q Returned Total Weight** · the weight that came back on the truck (undelivered stops).
- **S Total Delivery Weight** · P + Q. What was physically on the truck when it left. Use this for capacity/ utilisation reasoning.

Cross-check invariant

Σ (Summary column P for all trips on a date) $\approx$ Reports sheet · column K for that date. If the two differ, look for a trip with a missing SKU master row.

6. Sheet 3 · Details — per hop event

Every driver action on the road, in chronological order. A "hop" is a leg of the trip: WH → Stop 1, Stop 1 → Stop 2, etc.

Column	Label	Source / joins	How it is calculated
A	Trip Date	trip.trip_date	
B	Trip Number	trip.id	
C	Buyer Name	wd.name	
D	Hop Type	derived	Warehouse-Departure / Retailer / Warehouse-Return.
E	Delivery Status	trip_stop.status	pending / arrived / delivered / partial / undelivered.
F	Timestamp	actual_arrival_at OR actual_depart_at	GPS-stamped by the driver PWA.
G	Total Distance (km)	cumulative distance_from_prev_km	Cumulative from trip start. NA on manual trips where per-leg km wasn't populated.
H	Hop Name	retailer.name	Which shop this hop is for. Warehouse for the depart/return hops.
I	Travel Time	this_arrival – prev_depart	Time between leaving the previous stop and arriving here.
J	Unloading Time	this_depart – this_arrival	Time spent at this stop.
K	Time taken	Travel + Unloading	End-to-end time for this hop.
L	Hop Weight (kg)	weight formula → SUM per stop	Sum of order weight for orders attached to THIS stop. Delivered here or attempted here.
M	Net Invoice Value	SUM(order.invoice_value_rs) per stop	Order value attempted at this stop.
N	Vehicle	vehicle.reg_no	
O	Driver	user.name	
P	Actual Lat	trip_stop.arrival_lat / depart_lat	GPS at the moment the driver tapped Arrive / Delivered.
Q	Actual Long	trip_stop.arrival_lng / depart_lng	

7. Sheet 4 · Invoice Level Details — one row per order

Your per-order audit trail. Everything that happened to an individual invoice — from creation to close.

Column	Label	Source / joins	How it is calculated
A	WD Name	wd.name	
B	WD Dest Code	wd.code	
C	Order ID	orders.order_no	Text-typed cell so leading zeros survive Excel.
D	Invoice ID	orders.invoice_no	Text-typed. From Trishul.

For questions: reach out to the RouteSync team.

Column	Label	Source / joins	How it is calculated
E	Creation Date	orders.created_at	When the row was inserted by CSV import.
F	Billing Date	orders.order_date	Trishul-side billing date.
G	Delivery Date	delivery action ts	When the stop was marked delivered / partial / undelivered.
H	Store Name	retailer.name	
I	Status	orders.status	Delivered / Partial / Undelivered / Cancelled.
J	Net Invoice Value	orders.invoice_value_rs	Verbatim CSV Total Amount.
K	Return Value	derived from Status	Delivered/Partial → 0. Undelivered/Cancelled → Net Invoice Value.
L	Net Sale Value	derived from Status	Delivered/Partial → Net Invoice Value. Else → 0.
M	Weight	weight formula → SUM	Order weight (Σ of every SKU line on this invoice). Rounded to 3 dp.
N	Order Type	not captured	Always NA in v1 — reserved for a future field.
O	Sales Beat Name	beat.name	The beat this order was originally tagged to.
P	Sales Person Name	orders.salesman_name	From Trishul → the salesman who booked the order.
Q	Return Remark	trip_stop.remarks	For undelivered orders — the reason the driver gave.
R	Trip Code	trip.code	Last trip that touched this order.
S	Delivery Attempts	derived	How many times the order was on a dispatched trip.

Column M deep-dive — Weight

For the order, RouteSync walks every `order_item` row (`WHERE order_id = <this order> AND pending_stock = 0`), joins `sku_master`, applies the per-line formula from §2, sums, rounds to 3 decimal places (finer than trip-level columns because individual orders are smaller and 2 dp flattens differences).

Invoice weight vs Trip weight

Σ (Invoice column M for orders on a trip) is NOT necessarily equal to Summary column S for that trip. Invoice M counts every order attached to the trip; Summary S counts only the stops the driver engaged with (delivered + returned). Orders released on End Shift, or cancelled, are in Invoice but not in Summary S.

8. Sheet 5 · Delivery Aging — one row per order + bucket

Same per-order granularity as Invoice Level Details, but focused on how long the order took to close. Useful for finding chronic delays.

Column	Label	Source / joins	How it is calculated
A	Order ID	orders.order_no	Text-typed.
B	Invoice ID	orders.invoice_no	Text-typed.
C	Creation Date	orders.created_at	
D	Billing Date	orders.order_date	
E	Delivery Date	delivery action ts	
F	Store Name	retailer.name	
G	Status	orders.status	
H	Net Invoice Value	orders.invoice_value_rs	
I	Weight	weight formula → SUM	Same as Invoice sheet column M. Rounded to 3 dp.
J	Order Type	not captured	Always NA.
K	Sales Beat Name	beat.name	
L	Sales Person Name	orders.salesman_name	
M	Delivery Aging Bucket	derived	Same day / D+1 / D+2 / D+3+ / Still undelivered / Today's orders.

9. Sheet 6 · Vehicle Raw — fleet reference

Your fleet master, exactly as stored. Not filtered by date range. Useful for reconciling the Weight Utilization denominator on the Reports sheet.

Column	Label	Source / joins	How it is calculated
A	ID	vehicle.id	Internal DB id.
B	Name	derived	WD-<reg_no> for display.
C	Registration Number	vehicle.reg_no	Text-typed.
D	Ownership Type	not captured	Always NA in v1.
E	Max Weight (kg)	vehicle.max_weight_kg	THE denominator used in Weight Utilization (Reports col M).
F	Volumetric Capacity (L)	vehicle.max_volume_m3 × 1000	The DB stores cubic metres; the sheet shows litres for readability.
G	Status	vehicle.status	active / paused. Only active vehicles are eligible for auto-route.

10. Cross-sheet invariants — sanity checks

These are the equalities you can use to spot bad data:

- **Reports col K** $\approx$ **Σ Summary col P** for the same date. If not equal, look for a trip with missing SKU master rows.
- **Summary col S** = **P** + **Q** for every row. Always. If not, it's a bug — report it.

- **Σ Invoice col M for a trip $\geq$ Summary col S for that trip.** The Invoice sheet counts every order attached; Summary S counts only what the driver actually engaged with.
- **Reports col L = Σ Vehicle Raw col E** for the distinct vehicles used that day. Confirms capacity is picking up the right trucks.

11. Edge cases + common mistakes

Missing SKU master data

If `sku_master.weight_per_cfc_kg` or `sku_master.pieces_per_cfc` is NULL or 0 for a SKU, every `order_item` line for that SKU contributes zero weight. The order is not skipped — it just weighs less than it should. This is the #1 reason a report reads "too low" against physical fact.

Fix: go to `/admin/sku-master` → filter Missing weight → download the gaps CSV → fill it in Excel → upload back.

pending_stock lines

Order items marked `pending_stock = 1` at import (the WD couldn't ship them) are excluded from every weight/volume/value roll-up. The order still appears in the Invoice sheet but its weight will not include those items.

Weight Utilization > 100%

If Reports col M shows over 100%, the trucks used that day carried more weight than their combined `max_weight_kg`. Either overloaded (bad) or master data is wrong (more common — check the vehicle row in Vehicle Raw).

Time Utilization $\leq$ 100%

Deliberately capped at 100% per trip before averaging. A driver who beat the plan is 100% efficient, not 120% — this keeps a couple of fast trips from masking a slow day.

Delivery Utilization uncapped

By contrast, Delivery Utilization (Reports col O) is not capped. A trip running 45 stops against a 30 target shows 150% — that's useful signal, not clipping.

Rounding

- Trip- and date-level weights (Reports col K, Summary cols P/Q/S): **2 decimal places**.
- Order-level weight (Invoice col M, Delivery Aging col I): **3 decimal places**.
- Values (₹): **2 decimal places**.
- Latitudes / longitudes: **7 decimal places** (~1 cm precision).

CSV alternative

If Excel doesn't open the .xlsx (older versions on older machines), request the CSV-in-a-zip export from your super-admin. Same data, one CSV per sheet, all UTF-8 BOM-prefixed so retailer names with non-ASCII characters render correctly.