Ambient Systems Documentation

Version 20261002-v924039e · ambient-core 924039e · https://www.ambientsystems.ai/docs/

Ambient Systems — Documentation (customer package)

- **Version:** 20261002-v924039e
- **Built:** 2026-10-02T01:07:42.766Z
- **ambient-core ref:** 924039e
- **Live wiki:** https://www.ambientsystems.ai/docs/index.html

This file is the full public documentation wiki as Markdown sources.

Site ops runbooks are included because they ship in the public `/docs/` tree.

---

catalog/core-metrics.md

Core metrics layer

Ambient Core defines **15 shared templates** in `catalog/shared/metrics.yaml`. At generate time they expand into every industry pack as `{industry}.core.{slug}` keys (270 expanded keys across 18 packs). Each industry page lists its core rows alongside pack-specific metrics.

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway

- Working Capital

Monthly Burn Rate

- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key pattern:** `{industry}.core.burn_rate`

Cash Conversion Cycle

- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key pattern:** `{industry}.core.ccc`

Current Ratio

- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key pattern:** `{industry}.core.current_ratio`

Days Payable Outstanding (DPO)

- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days

- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key pattern:** `{industry}.core.dpo`

Days Sales Outstanding (DSO)

- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key pattern:** `{industry}.core.dso`

EBITDA Margin

- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key pattern:** `{industry}.core.ebitda_margin`

Gross Profit Margin

- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key pattern:** `{industry}.core.gross_margin`

Headcount (FTE)

- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._

- **Catalog key pattern:** `{industry}.core.headcount`

Net Profit Margin

- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key pattern:** `{industry}.core.net_margin`

Operating Cash Flow

- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key pattern:** `{industry}.core.operating_cash_flow`

Operating Margin (EBIT)

- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue 100`**
- **Catalog key pattern:** `{industry}.core.operating_margin`

Quick Ratio (Acid Test)

- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**

- **Catalog key pattern:** `{industry}.core.quick_ratio`

Revenue Growth Rate

- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key pattern:** `{industry}.core.revenue_growth`

Cash Runway

- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key pattern:** `{industry}.core.runway`

Working Capital

- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key pattern:** `{industry}.core.working_capital`

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/index.md

Reference catalog

Eighteen vertical **analysis lenses** from the MIT ambient-core catalog. Each pack defines **KPI metrics** (methodology and machine-readable **calc** where the generator defines them) and **data options** — required dataset templates with field lists you can map to uploads and governed products.

Authoring rules and YAML layout: Catalog README and CONVENTIONS. Industry codes (ISIC Rev.4 and related): catalog-industry-coverage.md.

Industry packs

- **Aviation** — ISIC Rev.4 · 5110 · 27 metrics · 7 data options
- **Banking** — ISIC Rev.4 · 6419 · 27 metrics · 5 data options
- **Construction** — ISIC Rev.4 · 4100 · 23 metrics · 7 data options
- **Credit Granting** — ISIC Rev.4 · 6492 · 41 metrics · 8 data options
- **Financial Services** — ISIC Rev.4 · 6612 · 23 metrics · 5 data options
- **Funds** — ISIC Rev.4 · 6630 · 43 metrics · 7 data options
- **Healthcare** — ISIC Rev.4 · 8610 · 37 metrics · 19 data options
- **Insurance** — ISIC Rev.4 · 6512 · 21 metrics · 5 data options
- **Life Sciences** — ISIC Rev.4 · 7210 · 20 metrics · 6 data options
- **Manufacturing** — ISIC Rev.4 · 2819 · 21 metrics · 16 data options
- **Passenger Transit** — ISIC Rev.4 · 4922 · 20 metrics · 6 data options
- **Real Estate** — ISIC Rev.4 · 6810 · 38 metrics · 23 data options
- **Restaurants** — ISIC Rev.4 · 5610 · 28 metrics · 17 data options
- **Retail** — ISIC Rev.4 · 4711 · 20 metrics · 6 data options
- **Software (SaaS)** — ISIC Rev.4 · 5820 · 23 metrics · 8 data options
- **Transportation** — ISIC Rev.4 · 4923 · 28 metrics · 19 data options
- **Trusts** — ISIC Rev.4 · 6430 · 17 metrics · 5 data options
- **Vertical Farming** — ISIC Rev.4 · 0113 · 32 metrics · 18 data options

Shared core layer

Every pack also includes the expanded corporate-finance **core** metrics (current ratio, revenue growth, and similar). See Core metrics.

Related

- Catalog consumption — manifest, loaders, and agents
- Use cases — workspace demos — Command Centre screenshots per vertical

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

catalog/industries/aviation.md

Aviation

Pack file: `catalog/industries/aviation` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">5110</code>

27 metrics · 7 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>481111</code></dd>

<dt>NACE Rev.2</dt><dd><code>51.10</code> </dd>

<dt>GICS 2023</dt><dd> <code>203020</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (7) {#data-options}

Airport traffic and revenue summary

Catalog key <code>aviation.airport.traffic_revenue_summary</code>

Fields

<dl class="catalog-field-list"> <dt> <code>period_start_date</code></dt><dd>date</dd> <div class="catalog-field-

list__row"><dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>enplaned_passengers</code></dt> <dd>string</dd> <dt> <code>passengers_enplaned_growth</code></dt><dd>string</dd> <dt><code>aeronautical_revenue</code></dt> <dd>decimal</dd> <dt> <code>commercial_revenue</code></dt><dd>decimal</dd> <dt><code>total_revenue</code></dt> <dd>decimal</dd> </dl>

Accounting Software

Catalog key <code>aviation.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt> <dd>decimal</dd> <dt> <code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt> <dd>decimal</dd> <dt> <code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code> </dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>aviation.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt>
<dd>date</dd> </dl>

HR records

Catalog key <code>aviation.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>aviation.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt>
<dd>decimal</dd> <dt>
<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>
<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>

<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Airline operational statistics summary

Catalog key <code>aviation.network_carrier.operational_statistics_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>available_seat_kilometers</code></dt><dd>string</dd> <dt><code>revenue_passenger_kilometers</code></dt><dd>decimal</dd> <dt><code>revenue_capacity_units</code></dt><dd>decimal</dd> <dt><code>available_capacity_units</code></dt><dd>string</dd> <dt><code>total_operating_cost</code></dt><dd>decimal</dd> <dt><code>total_operating_revenue</code></dt><dd>decimal</dd> <dt><code>on_time_arrivals</code></dt><dd>string</dd> <dt><code>total_arrivals</code></dt><dd>decimal</dd> <dt><code>fuel_expense</code></dt><dd>string</dd> <dt><code>cargo_tonne_km_carried</code></dt><dd>string</dd> <dt><code>available_tonne_km</code></dt><dd>string</dd> <dt><code>cargo_revenue</code></dt><dd>decimal</dd> </dl>

Travel distribution booking summary

Catalog key <code>aviation.travel_distribution.booking_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>platform_revenue</code></dt><dd>decimal</dd> <dt><code>gross_booking_value</code></dt><dd>string</dd> <dt><code>completed_bookings</code></dt><dd>string</dd> <dt><code>qualified_sessions</code></dt><dd>string</dd> </dl>

Metrics (27) {#metrics}

- Cargo Load Factor
- Cargo Yield per Tonne-Kilometer
- Aeronautical Revenue per Passenger
- Commercial Revenue Share
- Passengers Enplaned Growth
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate

- Cash Runway
- Working Capital
- Cost per Available Seat Kilometer (CASM)
- Fuel Cost Percent of Operating Cost
- Load Factor
- On-Time Performance (OTP)
- Revenue per Available Seat Kilometer (RASM)
- Gross Booking Value Take Rate
- Look-to-Book Conversion

Cargo Load Factor

- **ID:** 1830
- **Type:** Operational
- **Unit:** %
- **Description:** Cargo tonnage carried relative to available cargo capacity (aggregated).
- **Inputs:** `cargo_tonne_km_carried`, `available_tonne_km`
- **Methodology:** Cargo Load Factor = (Cargo Tonne-Km Carried / Available Tonne-Km) × 100
- **Calculation:** *`cargo_tonne_km_carried / available_tonne_km` 100**
- **Catalog key:** `aviation.air_cargo.cargo_load_factor`

Cargo Yield per Tonne-Kilometer

- **ID:** 1831
- **Type:** Financial
- **Unit:** Currency per tonne-km
- **Description:** Cargo revenue per tonne-kilometer carried (aggregated).
- **Inputs:** `cargo_revenue`, `cargo_tonne_km_carried`
- **Methodology:** Yield = Cargo Revenue / Cargo Tonne-Kilometers Carried
- **Calculation:** **`cargo_revenue / cargo_tonne_km_carried`**
- **Catalog key:** `aviation.air_cargo.cargo_yield_per_tonne_km`

Aeronautical Revenue per Passenger

- **ID:** 1826
- **Type:** Financial
- **Unit:** Currency
- **Description:** Aeronautical charges divided by enplaned passengers (aggregated).

- **Inputs:** `aeronautical_revenue`, `enplaned_passengers`
- **Methodology:** ARPP = Aeronautical Revenue / Enplaned Passengers
- **Calculation:** **`aeronautical_revenue / enplaned_passengers`**
- **Catalog key:** `aviation.airport.aeronautical_revenue_per_passenger`

Commercial Revenue Share

- **ID:** 1827
- **Type:** Financial
- **Unit:** %
- **Description:** Non-aeronautical revenue as a share of total airport revenue (aggregated).
- **Inputs:** `commercial_revenue`, `total_revenue`
- **Methodology:** Share = (Commercial Revenue / Total Revenue) × 100
- **Calculation:** *_`commercial_revenue / total_revenue` 100_**
- **Catalog key:** `aviation.airport.commercial_revenue_share`

Passengers Enplaned Growth

- **ID:** 1825
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in enplaned passengers (aggregated).
- **Methodology:** Growth = ((Current Enplanements − Prior Enplanements) / Prior Enplanements) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `aviation.airport.passengers_enplaned_growth`

Monthly Burn Rate

- **ID:** 1812
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `aviation.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1810

- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `aviation.core.ccc`

Current Ratio

- **ID:** 1800
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `aviation.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1809
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs` days_**
- **Catalog key:** `aviation.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1808
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales` days_**

- **Catalog key:** `aviation.core.dso`

EBITDA Margin

- **ID:** 1804
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue 100`**
- **Catalog key:** `aviation.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1802
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue 100`**
- **Catalog key:** `aviation.core.gross_margin`

Headcount (FTE)

- **ID:** 1814
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `aviation.core.headcount`

Net Profit Margin

- **ID:** 1805
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.

- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` 100**
- **Catalog key:** `aviation.core.net_margin`

Operating Cash Flow

- **ID:** 1811
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `aviation.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1803
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `aviation.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1801
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `aviation.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1806
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `aviation.core.revenue_growth`

Cash Runway

- **ID:** 1813
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `aviation.core.runway`

Working Capital

- **ID:** 1807
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `aviation.core.working_capital`

Cost per Available Seat Kilometer (CASM)

- **ID:** 1821
- **Type:** Financial
- **Unit:** Currency per ASK
- **Description:** Operating cost per unit of available seat capacity (aggregated).
- **Inputs:** `total_operating_cost`, `available_seat_kilometers`
- **Methodology:** CASM = Total Operating Cost / Available Seat Kilometers

- **Calculation:** `total_operating_cost / available_seat_kilometers`
- **Catalog key:** `aviation.network_carrier.casm`

Fuel Cost Percent of Operating Cost

- **ID:** 1824
- **Type:** Financial
- **Unit:** %
- **Description:** Fuel expense as a share of total operating cost (aggregated).
- **Inputs:** `fuel_expense`, `total_operating_cost`
- **Methodology:** Fuel % = (Fuel Expense / Total Operating Cost) × 100
- **Calculation:** *`fuel_expense / total_operating_cost` 100**
- **Catalog key:** `aviation.network_carrier.fuel_cost_pct_operating`

Load Factor

- **ID:** 1820
- **Type:** Operational
- **Unit:** %
- **Description:** Revenue passenger miles or cargo tonnage carried relative to available capacity (aggregated).
- **Inputs:** `revenue_capacity_units`, `available_capacity_units`
- **Methodology:** Load Factor = (RPM or RTM / ASM or ATM) × 100
- **Calculation:** *`revenue_capacity_units / available_capacity_units` 100**
- **Catalog key:** `aviation.network_carrier.load_factor`

On-Time Performance (OTP)

- **ID:** 1823
- **Type:** Operational
- **Unit:** %
- **Description:** Share of flights arriving within regulatory on-time threshold (aggregated).
- **Inputs:** `on_time_arrivals`, `total_arrivals`
- **Methodology:** OTP = (On-Time Arrivals / Total Arrivals) × 100
- **Calculation:** *`on_time_arrivals / total_arrivals` 100**
- **Catalog key:** `aviation.network_carrier.on_time_performance`

Revenue per Available Seat Kilometer (RASM)

- **ID:** 1822
- **Type:** Financial

- **Unit:** Currency per ASK
- **Description:** Operating revenue per unit of available seat capacity (aggregated).
- **Inputs:** `total_operating_revenue`, `available_seat_kilometers`
- **Methodology:** RASM = Total Operating Revenue / Available Seat Kilometers
- **Calculation:** **`total_operating_revenue / available_seat_kilometers`**
- **Catalog key:** `aviation.network_carrier.rasm`

Gross Booking Value Take Rate

- **ID:** 1828
- **Type:** Financial
- **Unit:** %
- **Description:** Platform revenue as a share of gross bookings (aggregated).
- **Inputs:** `platform_revenue`, `gross_booking_value`
- **Methodology:** Take Rate = (Platform Revenue / Gross Booking Value) × 100
- **Calculation:** *`platform_revenue / gross_booking_value` 100**
- **Catalog key:** `aviation.travel_distribution.gbv_take_rate`

Look-to-Book Conversion

- **ID:** 1829
- **Type:** Operational
- **Unit:** %
- **Description:** Share of travel searches or sessions that complete a booking (aggregated).
- **Inputs:** `completed_bookings`, `qualified_sessions`
- **Methodology:** Conversion = (Completed Bookings / Qualified Sessions) × 100
- **Calculation:** *`completed_bookings / qualified_sessions` 100**
- **Catalog key:** `aviation.travel_distribution.look_to_book_conversion`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Airport traffic and revenue summary** — feeds Passengers Enplaned Growth, Aeronautical Revenue per Passenger, Commercial Revenue Share
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Airline operational statistics summary** — feeds Load Factor, Cost per Available Seat Kilometer (CASM), Revenue per Available Seat Kilometer (RASM), On-Time Performance (OTP), Fuel Cost Percent of Operating Cost, Cargo Load Factor, Cargo Yield per Tonne-Kilometer
- **Travel distribution booking summary** — feeds Gross Booking Value Take Rate, Look-to-Book Conversion

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/banking.md

Banking

Pack file: `catalog/industries/banking` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6419</code>

27 metrics · 5 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd>
<code>522110</code></dd>

<dt>NACE Rev.2</dt><dd><code>64.19</code>
</dd>

<dt>GICS 2023</dt><dd>
<code>401010</code></dd>

</dl>

Taxonomy tags from ambient-core
<code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification).
Coverage notes: <a href="https://github.com/Ambient-Team/ambient-
core/blob/main/docs/catalog-industry-coverage.md">catalog-industry-coverage.md.

Data options (5) {#data-options}

Accounting Software

Catalog
key <code>banking.core.accounting_software</code>

<span class="catalog-field-
label">Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_start_date</code></dt><dd>date</dd> <div class="catalog-
field-list__row"><dt><code>period_end_date</code></dt><dd>date</dd> <div
class="catalog-field-list__row"><dt><code>revenue</code></dt><dd>decimal</dd>
<dt><code>cogs</code></dt>
<dd>decimal</dd> <dt>
<code>cost_of_goods</code></dt><dd>decimal</dd> <div class="catalog-field-
list__row"><dt><code>operating_expenses</code></dt><dd>decimal</dd>
<dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>produce_revenue</code></dt><dd>decimal</dd> <div class="catalog-
field-list__row"><dt><code>total_production_cost</code></dt><dd>decimal</dd>
<dt><code>total_operational_cost</code>
</dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>banking.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>banking.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>banking.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt>

<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Aggregated regulatory and financial summary

Catalog key <code>banking.depository.regulatory_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>entity_segment</code></dt><dd>string</dd> <dt><code>net_interest_income</code></dt><dd>string</dd> <dt><code>average_earning_assets</code></dt><dd>string</dd> <dt><code>non_performing_loans</code></dt><dd>string</dd> <dt><code>total_loans</code></dt><dd>decimal</dd> <dt><code>total_deposits</code></dt><dd>decimal</dd> <dt><code>operating_expense</code></dt><dd>string</dd> <dt><code>total_revenue</code></dt><dd>decimal</dd> </dl>

Metrics (27) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Cost of Funds
- Efficiency Ratio
- Interest Rate Risk EVE Sensitivity
- Liquidity Coverage Ratio
- Loan-to-Deposit Ratio
- Net Interest Margin
- Net Stable Funding Ratio
- Non-Performing Loan Ratio
- Operational Risk Loss Ratio
- Stable Funding Ratio
- Stress Test CET1 Delta
- Tier 1 Capital Ratio

Monthly Burn Rate

- **ID:** 1512
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`

- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** `cash_out - cash_in`
- **Catalog key:** `banking.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1510
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** `dso + days_inventory_outstanding - dpo`
- **Catalog key:** `banking.core.ccc`

Current Ratio

- **ID:** 1500
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** `current_assets / current_liabilities`
- **Catalog key:** `banking.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1509
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `banking.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1508

- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` `days`**
- **Catalog key:** `banking.core.dso`

EBITDA Margin

- **ID:** 1504
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `banking.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1502
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `banking.core.gross_margin`

Headcount (FTE)

- **ID:** 1514
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._

- **Catalog key:** `banking.core.headcount`

Net Profit Margin

- **ID:** 1505
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` `100`**
- **Catalog key:** `banking.core.net_margin`

Operating Cash Flow

- **ID:** 1511
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `banking.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1503
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` `100`**
- **Catalog key:** `banking.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1501
- **Type:** Financial
- **Unit:** Ratio

- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `banking.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1506
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`*(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `banking.core.revenue_growth`

Cash Runway

- **ID:** 1513
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `banking.core.runway`

Working Capital

- **ID:** 1507
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `banking.core.working_capital`

Cost of Funds

- **ID:** 1554
- **Type:** Financial
- **Unit:** %
- **Description:** Average rate paid on interest-bearing liabilities (aggregated).
- **Methodology:** Cost of Funds = Interest Expense on Deposits and Borrowings / Average Interest-Bearing Liabilities × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.cost_of_funds`

Efficiency Ratio

- **ID:** 1533
- **Type:** Financial
- **Unit:** %
- **Description:** Non-interest expense relative to revenue (cost-to-income).
- **Inputs:** `operating_expense`, `total_revenue`
- **Methodology:** Efficiency Ratio = (Operating Expense / Total Revenue) × 100
- **Calculation:** *`operating_expense / total_revenue 100`**
- **Catalog key:** `banking.depository.efficiency_ratio`

Interest Rate Risk EVE Sensitivity

- **ID:** 1559
- **Type:** Financial
- **Unit:** %
- **Description:** Change in economic value of equity for a defined parallel rate shock (aggregated ALM).
- **Methodology:** EVE sensitivity = ΔEVE / Base EVE for standard shock
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.interest_rate_risk_eve_sensitivity`

Liquidity Coverage Ratio

- **ID:** 1557
- **Type:** Financial
- **Unit:** %
- **Description:** High-quality liquid assets relative to net cash outflows (LCR).
- **Methodology:** LCR = HQLA / Net Cash Outflows over 30 days × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.liquidity_coverage_ratio`

Loan-to-Deposit Ratio

- **ID:** 1532
- **Type:** Financial
- **Unit:** %
- **Description:** Lending funded by deposits (liquidity and balance-sheet structure).
- **Inputs:** `total_loans`, `total_deposits`
- **Methodology:** LTD = (Total Loans / Total Deposits) × 100
- **Calculation:** *`total_loans / total_deposits` 100**
- **Catalog key:** `banking.depository.loan_to_deposit_ratio`

Net Interest Margin

- **ID:** 1530
- **Type:** Financial
- **Unit:** %
- **Description:** Spread between interest earned and interest paid, relative to earning assets (aggregated regulatory summary).
- **Inputs:** `net_interest_income`, `average_earning_assets`
- **Methodology:** NIM = (Net Interest Income / Average Earning Assets) × 100
- **Calculation:** *`net_interest_income / average_earning_assets` 100**
- **Catalog key:** `banking.depository.net_interest_margin`

Net Stable Funding Ratio

- **ID:** 1558
- **Type:** Financial
- **Unit:** %
- **Description:** Available stable funding relative to required stable funding (NSFR).
- **Methodology:** NSFR = Available Stable Funding / Required Stable Funding × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.net_stable_funding_ratio`

Non-Performing Loan Ratio

- **ID:** 1531
- **Type:** Financial
- **Unit:** %
- **Description:** Share of loans past due or non-accrual (aggregated credit quality).
- **Inputs:** `non_performing_loans`, `total_loans`
- **Methodology:** NPL Ratio = (Non-Performing Loans / Total Loans) × 100
- **Calculation:** *`non_performing_loans / total_loans` 100**

- **Catalog key:** `banking.depository.non_performing_loan_ratio`

Operational Risk Loss Ratio

- **ID:** 1567
- **Type:** Financial
- **Unit:** %
- **Description:** Operational losses as a share of gross income (aggregated).
- **Inputs:** `operational_losses`, `gross_income`
- **Methodology:** Op Risk Loss Ratio = Operational Losses / Gross Income × 100
- **Calculation:** *`operational_losses / gross_income` 100**
- **Catalog key:** `banking.depository.operational_risk_loss_ratio`

Stable Funding Ratio

- **ID:** 1556
- **Type:** Financial
- **Unit:** %
- **Description:** Share of funding from stable retail and long-term sources (proxy or regulatory).
- **Methodology:** Stable Funding Ratio = Stable Funding / Total Funding × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.stable_funding_ratio`

Stress Test CET1 Delta

- **ID:** 1568
- **Type:** Financial
- **Unit:** %
- **Description:** Decline in CET1 ratio under supervisory severely adverse scenario (aggregated).
- **Methodology:** CET1 Delta = Stressed CET1 − Baseline CET1
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.stress_test_cet1_delta`

Tier 1 Capital Ratio

- **ID:** 1534
- **Type:** Financial
- **Unit:** %
- **Description:** Tier 1 capital to risk-weighted assets from aggregated regulatory filing (no account-level detail).
- **Methodology:** Tier 1 Ratio = (Tier 1 Capital / Risk-Weighted Assets) × 100

- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `banking.depository.tier1_capital_ratio`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Aggregated regulatory and financial summary** — feeds Net Interest Margin, Non-Performing Loan Ratio, Loan-to-Deposit Ratio, Efficiency Ratio, Tier 1 Capital Ratio, Cost of Funds, Stable Funding Ratio, Liquidity Coverage Ratio, Net Stable Funding Ratio, Interest Rate Risk EVE Sensitivity, Operational Risk Loss Ratio, Stress Test CET1 Delta

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

catalog/industries/construction.md

Construction

Pack file: `catalog/industries/construction` in ambient-core.

Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">4100</code>

23 metrics · 7 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>236220</code> (also <code>237310</code>, <code>237990</code>)</dd>

<dt>NACE Rev.2</dt><dd><code>41.20</code> (also <code>42.11</code>, <code>42.99</code>)</dd>

<dt>GICS 2023</dt><dd> <code>201030</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (7) {#data-options}

Job cost and schedule export

Catalog key <code>construction.building_construction.job_cost_schedule_export</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>job_id</code></dt><dd>id</dd> <dt><code>job_type</code></dt><dd>enum</dd> <dt><code>earned_value</code></dt><dd>decimal</dd> <dt><code>planned_value</code></dt><dd>decimal</dd> <dt><code>actual_cost</code></dt><dd>decimal</dd> <dt><code>civil_earned_value</code></dt><dd>decimal</dd> <dt><code>civil_planned_value</code></dt><dd>decimal</dd> <dt><code>installed_quantity</code></dt><dd>decimal</dd> <dt><code>budgeted_quantity</code></dt><dd>decimal</dd> <dt><code>approved_change_order_value</code></dt><dd>decimal</dd> <dt><code>original_contract_value</code></dt><dd>decimal</dd> </dl>

Accounting Software

Catalog key <code>construction.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>construction.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>construction.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>construction.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <div

class="catalog-field-list__row"><dt><code>headcount</code></dt>
<dd>decimal</dd> <dt>
<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>
<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd>
<dt><code>total_production_cost</code></dt>
<dd>decimal</dd> <dt>
<code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt>
<dd>decimal</dd> <dt>
<code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>
<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

Construction WIP and retention GL

Catalog key <code>construction.financial.wip_retention_gl</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>account_code</code></dt><dd>id</dd> <dt><code>work_in_progress_balance</code></dt><dd>decimal</dd>
<dt><code>contract_revenue</code></dt>
<dd>decimal</dd> <dt>
<code>retention_receivable</code></dt><dd>decimal</dd> <dt><code>days</code></dt><dd>decimal</dd>
</dl>

Safety incident log

Catalog key <code>construction.operations.safety_incident_log</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>site_id</code></dt><dd>id</dd> <dt><code>recordable_incidents</code></dt><dd>decimal</dd> <dt><code>hours_worked</code></dt><dd>decimal</dd> </dl>

Metrics (23) {#metrics}

- Cost Performance Index (CPI)
- Schedule Performance Index (SPI)
- Civil Earned Value Ratio
- Percent Complete (Physical)
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Retention Receivable Days
- Work-in-Progress Ratio

- Change Order Rate
- Total Recordable Incident Rate (TRIR)

Cost Performance Index (CPI)

- **ID:** 2101
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Earned value cost efficiency for building jobs (EV / AC).
- **Inputs:** `earned_value`, `actual_cost`
- **Methodology:** CPI = Earned Value / Actual Cost
- **Calculation:** `earned_value / actual_cost`
- **Catalog key:** `construction.building_construction.cost_performance_index`

Schedule Performance Index (SPI)

- **ID:** 2100
- **Type:** Operational
- **Unit:** Ratio
- **Description:** Earned value schedule efficiency for building jobs (EV / PV).
- **Inputs:** `earned_value`, `planned_value`
- **Methodology:** SPI = Earned Value / Planned Value
- **Calculation:** `earned_value / planned_value`
- **Catalog key:** `construction.building_construction.schedule_performance_index`

Civil Earned Value Ratio

- **ID:** 2102
- **Type:** Operational
- **Unit:** Ratio
- **Description:** Earned value relative to planned value on civil / infrastructure jobs.
- **Inputs:** `civil_earned_value`, `civil_planned_value`
- **Methodology:** Civil EV Ratio = Civil Earned Value / Civil Planned Value
- **Calculation:** `civil_earned_value / civil_planned_value`
- **Catalog key:** `construction.civil_engineering.earned_value_ratio`

Percent Complete (Physical)

- **ID:** 2103
- **Type:** Operational
- **Unit:** %

- **Description:** Physical percent complete for civil work packages (quantity-based).
- **Inputs:** `installed_quantity`, `budgeted_quantity`
- **Methodology:** Percent Complete = (Installed Quantity / Budgeted Quantity) × 100
- **Calculation:** *`installed_quantity / budgeted_quantity` 100**
- **Catalog key:** `construction.civil_engineering.percent_complete_physical`

Monthly Burn Rate

- **ID:** 2082
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `construction.core.burn_rate`

Cash Conversion Cycle

- **ID:** 2080
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `construction.core.ccc`

Current Ratio

- **ID:** 2070
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `construction.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 2079
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` `days`**
- **Catalog key:** `construction.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 2078
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` `days`**
- **Catalog key:** `construction.core.dso`

EBITDA Margin

- **ID:** 2074
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `construction.core.ebitda_margin`

Gross Profit Margin

- **ID:** 2072
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100

- **Calculation:** *`(revenue - cogs) / revenue` 100**
- **Catalog key:** `construction.core.gross_margin`

Headcount (FTE)

- **ID:** 2084
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `construction.core.headcount`

Net Profit Margin

- **ID:** 2075
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` 100**
- **Catalog key:** `construction.core.net_margin`

Operating Cash Flow

- **ID:** 2081
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `construction.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 2073
- **Type:** Financial
- **Unit:** %

- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `construction.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 2071
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `construction.core.quick_ratio`

Revenue Growth Rate

- **ID:** 2076
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `construction.core.revenue_growth`

Cash Runway

- **ID:** 2083
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `construction.core.runway`

Working Capital

- **ID:** 2077
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `construction.core.working_capital`

Retention Receivable Days

- **ID:** 2107
- **Type:** Financial
- **Unit:** Days
- **Description:** Average days retainage remains outstanding before release.
- **Inputs:** `retention_receivable`, `contract_revenue`, `days`
- **Methodology:** Retention Days = (Retention Receivable / Contract Revenue) × Number of Days
- **Calculation:** *_`retention_receivable / contract_revenue` days_**
- **Catalog key:** `construction.financial.retention_receivable_days`

Work-in-Progress Ratio

- **ID:** 2106
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Construction WIP (costs plus recognized profit less billings) relative to contract revenue.
- **Inputs:** `work_in_progress_balance`, `contract_revenue`
- **Methodology:** WIP Ratio = Work in Progress Balance / Contract Revenue
- **Calculation:** **`work_in_progress_balance / contract_revenue`**
- **Catalog key:** `construction.financial.work_in_progress_ratio`

Change Order Rate

- **ID:** 2104
- **Type:** Operational
- **Unit:** %
- **Description:** Approved change-order value as a share of original contract value.
- **Inputs:** `approved_change_order_value`, `original_contract_value`

- **Methodology:** Change Order Rate = (Approved Change Order Value / Original Contract Value) × 100
- **Calculation:** *`approved_change_order_value / original_contract_value` `100`**
- **Catalog key:** `construction.operations.change_order_rate`

Total Recordable Incident Rate (TRIR)

- **ID:** 2105
- **Type:** Operational
- **Unit:** Rate
- **Description:** OSHA-style total recordable incident rate per 200,000 hours worked.
- **Inputs:** `recordable_incidents`, `hours_worked`
- **Methodology:** TRIR = (Recordable Incidents × 200000) / Hours Worked
- **Calculation:** *`recordable_incidents` `200000 / hours_worked`**
- **Catalog key:** `construction.operations.safety_trir`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Job cost and schedule export** — feeds Schedule Performance Index (SPI), Cost Performance Index (CPI), Civil Earned Value Ratio, Percent Complete (Physical), Change Order Rate
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway

- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Construction WIP and retention GL** — feeds Work-in-Progress Ratio, Retention Receivable Days
- **Safety incident log** — feeds Total Recordable Incident Rate (TRIR)

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/credit_granting.md

Credit Granting

Pack file: `catalog/industries/credit_granting` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6492</code>

41 metrics · 8 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>522291</code> (also <code>522220</code>)</dd>

<dt>NACE Rev.2</dt><dd><code>64.92</code></dd>

<dt>GICS 2023</dt><dd> <code>402020</code> (also <code>402010</code>)</dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: <a href="https://github.com/Ambient-Team/ambient-

core/blob/main/docs/catalog-industry-coverage.md">catalog-industry-coverage.md.

Data options (8) {#data-options}

Aggregated commercial loan book summary

Catalog key
<code>credit_granting.commercial_lending.commercial_loan_book_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>entity_segment</code></dt><dd>string</dd> <dt><code>outstanding_commercial_loans</code></dt>
<dd>string</dd> <dt>
<code>interest_income</code></dt><dd>string</dd> <dt><code>average_commercial_loans</code></dt><dd>string</dd>
<dt>
<code>non_performing_commercial_loans</code></dt><dd>string</dd> <dt><code>total_commercial_loans</code></dt>
<dd>decimal</dd> <dt>
<code>net_charge_offs</code></dt><dd>string</dd> <dt><code>delinquent_balance_30_plus</code></dt><dd>string</dd>
<dt><code>origination_volume</code>
</dt><dd>decimal</dd> <dt>
<code>weighted_ltv_numerator</code></dt><dd>string</dd> <dt><code>drawn_commitments</code></dt>
<dd>string</dd> <dt>
<code>total_commitments</code></dt><dd>decimal</dd> <dt><code>sustainable_loan_balance</code></dt><dd>string</dd>
</dl>

Aggregated commercial treasury summary

Catalog key
<code>credit_granting.commercial_lending.commercial_treasury_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>entity_segment</code></dt><dd>string</dd> <dt><code>treasury_fee_revenue</code></dt><dd>decimal</dd> <dt><code>average_commercial_deposits</code></dt><dd>string</dd> <dt><code>commercial_payment_volume</code></dt><dd>decimal</dd> </dl>

Aggregated consumer loan tape summary

Catalog key <code>credit_granting.consumer_lending.loan_tape_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>product_line</code></dt><dd>string</dd> <dt><code>interest_income</code></dt><dd>string</dd> <dt><code>average_receivables</code></dt><dd>string</dd> <dt><code>net_charge_offs</code></dt><dd>string</dd> <dt><code>delinquent_balance_30_plus</code></dt><dd>string</dd> <dt><code>total_receivables</code></dt><dd>decimal</dd> <dt><code>origination_volume_growth</code></dt><dd>decimal</dd> <dt><code>payment_volume</code></dt><dd>decimal</dd> <dt><code>bnpl_origination_growth</code></dt><dd>string</dd> <dt><code>embedded_channel_revenue</code></dt><dd>enum</dd> <dt><code>total_consumer_finance_revenue</code></dt><dd>decimal</dd> <dt><code>rtp_payment_volume</code></dt><dd>decimal</dd> <dt><code>merchant_acquiring_volume</code></dt><dd>decimal</dd> </dl>

Accounting Software

Catalog key <code>credit_granting.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>credit_granting.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>credit_granting.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>credit_granting.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Aggregated residential mortgage book summary

Catalog key
<code>credit_granting.residential_mortgage.residential_mortgage_book_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>product_line</code></dt><dd>string</dd> <dt><code>interest_income</code></dt><dd>string</dd> <dt><code>average_mortgage_balances</code></dt>
<dd>string</dd> <dt>
<code>delinquent_balance_30_plus</code></dt><dd>string</dd> <dt><code>total_mortgage_balances</code></dt>
<dd>decimal</dd> <dt>
<code>weighted_ltv_numerator</code></dt><dd>string</dd> <dt><code>origination_volume</code></dt>
<dd>decimal</dd> </dl>

Metrics (41) {#metrics}

- Commercial Loan Growth
- Commercial Payment Volume
- Commercial 30+ Day Delinquency Rate
- Commercial Net Charge-Off Rate
- Commercial NPL Ratio
- Commercial Origination Volume Growth
- Portfolio DSCR
- Commercial Portfolio Yield
- Revolver Utilization Rate
- Sector Concentration Index
- Sustainable Finance Share
- Treasury Fee Yield
- Weighted Average LTV

- BNPL Origination Growth
- 30+ Day Delinquency Rate
- Embedded Finance Revenue Share
- Merchant Acquiring Volume
- Net Charge-Off Rate
- Origination Volume Growth
- Card Payment Volume
- Portfolio Yield
- RTP Payment Volume
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Residential Mortgage 30+ Day Delinquency
- Residential Mortgage Origination Growth
- Residential Mortgage Portfolio Yield
- Residential Mortgage Weighted Average LTV

Commercial Loan Growth

- **ID:** 1760
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in outstanding commercial and C&I loan balances (aggregated).
- **Methodology:** Loan Growth = ((Current Outstanding − Prior Outstanding) / Prior Outstanding) × 100

- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.commercial_lending.commercial_loan_growth`

Commercial Payment Volume

- **ID:** 1776
- **Type:** Financial
- **Unit:** Currency
- **Description:** B2B payment and wire volume processed for commercial clients (aggregated).
- **Methodology:** Directly measured payment volume in period.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.commercial_lending.commercial_payment_volume`

Commercial 30+ Day Delinquency Rate

- **ID:** 1765
- **Type:** Financial
- **Unit:** %
- **Description:** Share of commercial loan balances past due 30 days or more (aggregated).
- **Inputs:** `delinquent_balance_30_plus`, `total_commercial_loans`
- **Methodology:** Delinquency Rate = (Delinquent Balance 30+ / Total Commercial Loans) × 100
- **Calculation:** *`delinquent_balance_30_plus / total_commercial_loans` 100**
- **Catalog key:** `credit_granting.commercial_lending.delinquency_rate_30d`

Commercial Net Charge-Off Rate

- **ID:** 1763
- **Type:** Financial
- **Unit:** %
- **Description:** Commercial charge-offs net of recoveries as a share of average commercial loans.
- **Inputs:** `net_charge_offs`, `average_commercial_loans`
- **Methodology:** NCO Rate = (Net Charge-Offs / Average Commercial Loans) × 100
- **Calculation:** *`net_charge_offs / average_commercial_loans` 100**
- **Catalog key:** `credit_granting.commercial_lending.net_charge_off_rate`

Commercial NPL Ratio

- **ID:** 1762
- **Type:** Financial

- **Unit:** %
- **Description:** Share of commercial loans past due or non-accrual (aggregated credit quality).
- **Inputs:** `non_performing_commercial_loans`, `total_commercial_loans`
- **Methodology:** NPL Ratio = (Non-Performing Commercial Loans / Total Commercial Loans) × 100
- **Calculation:** *`non_performing_commercial_loans / total_commercial_loans` `100`**
- **Catalog key:** `credit_granting.commercial_lending.non_performing_loan_ratio`

Commercial Origination Volume Growth

- **ID:** 1764
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in new commercial loan commitments or fundings (aggregated).
- **Methodology:** Origination Growth = ((Current Originations − Prior Originations) / Prior Originations) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.commercial_lending.origination_volume_growth`

Portfolio DSCR

- **ID:** 1772
- **Type:** Financial
- **Unit:** Number
- **Description:** Loan-weighted average debt service coverage on commercial book (lender view).
- **Methodology:** Portfolio DSCR = Σ(DSCR × Loan Weight) on reporting book
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.commercial_lending.portfolio_dscr`

Commercial Portfolio Yield

- **ID:** 1761
- **Type:** Financial
- **Unit:** %
- **Description:** Interest income on commercial loans relative to average outstanding balance (aggregated C&I and middle market).
- **Inputs:** `interest_income`, `average_commercial_loans`
- **Methodology:** Portfolio Yield = (Interest Income / Average Commercial Loans) × 100

- **Calculation:** *`interest_income / average_commercial_loans` 100**
- **Catalog key:** `credit_granting.commercial_lending.portfolio_yield`

Revolver Utilization Rate

- **ID:** 1774
- **Type:** Financial
- **Unit:** %
- **Description:** Drawn committed lines relative to total commitments (aggregated).
- **Inputs:** `drawn_commitments`, `total_commitments`
- **Methodology:** Utilization = (Drawn Commitments / Total Commitments) × 100
- **Calculation:** *`drawn_commitments / total_commitments` 100**
- **Catalog key:** `credit_granting.commercial_lending.revolver_utilization_rate`

Sector Concentration Index

- **ID:** 1773
- **Type:** Financial
- **Unit:** Number
- **Description:** Herfindahl-style concentration of commercial exposures by industry sector (org book).
- **Methodology:** HHI = Σ(Sector Share²) × 10000 on outstanding balances
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.commercial_lending.sector_concentration_index`

Sustainable Finance Share

- **ID:** 1777
- **Type:** Financial
- **Unit:** %
- **Description:** Share of commercial outstandings tagged as sustainable or green-labeled (aggregated).
- **Inputs:** `sustainable_loan_balance`, `total_commercial_loans`
- **Methodology:** Sustainable Share = (Sustainable Loan Balance / Total Commercial Loans) × 100
- **Calculation:** *`sustainable_loan_balance / total_commercial_loans` 100**
- **Catalog key:** `credit_granting.commercial_lending.sustainable_finance_share`

Treasury Fee Yield

- **ID:** 1775
- **Type:** Financial
- **Unit:** %

- **Description:** Cash management and liquidity service fees relative to commercial deposit balances.
- **Inputs:** `treasury_fee_revenue`, `average_commercial_deposits`
- **Methodology:** Treasury Fee Yield = (Treasury Fee Revenue / Average Commercial Deposits) × 100
- **Calculation:** *`treasury_fee_revenue / average_commercial_deposits` 100**
- **Catalog key:** `credit_granting.commercial_lending.treasury_fee_yield`

Weighted Average LTV

- **ID:** 1771
- **Type:** Financial
- **Unit:** %
- **Description:** Loan-weighted LTV on commercial and CRE portfolios (lender book view).
- **Inputs:** `weighted_ltv_numerator`, `total_commercial_loans`
- **Methodology:** WA LTV = Σ(Loan Balance × LTV) / Σ Loan Balances
- **Calculation:** *`weighted_ltv_numerator / total_commercial_loans` 100**
- **Catalog key:** `credit_granting.commercial_lending.weighted_average_ltv`

BNPL Origination Growth

- **ID:** 1725
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in buy-now-pay-later originations (aggregated).
- **Methodology:** BNPL Growth = ((Current BNPL Originations − Prior) / Prior) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.consumer_lending.bnpl_origination_growth`

30+ Day Delinquency Rate

- **ID:** 1722
- **Type:** Financial
- **Unit:** %
- **Description:** Share of receivables past due 30 days or more (aggregated).
- **Inputs:** `delinquent_balance_30_plus`, `total_receivables`
- **Methodology:** Delinquency Rate = (Delinquent Balance 30+ / Total Receivables) × 100
- **Calculation:** *`delinquent_balance_30_plus / total_receivables` 100**
- **Catalog key:** `credit_granting.consumer_lending.delinquency_rate_30d`

Embedded Finance Revenue Share

- **ID:** 1726
- **Type:** Financial
- **Unit:** %
- **Description:** Share of consumer finance revenue from embedded or partner channels (aggregated).
- **Inputs:** `embedded_channel_revenue`, `total_consumer_finance_revenue`
- **Methodology:** Embedded Share = (Embedded Channel Revenue / Total Consumer Finance Revenue) × 100
- **Calculation:** *`embedded_channel_revenue / total_consumer_finance_revenue` 100**
- **Catalog key:** `credit_granting.consumer_lending.embedded_finance_revenue_share`

Merchant Acquiring Volume

- **ID:** 1728
- **Type:** Financial
- **Unit:** Currency
- **Description:** Card acquiring volume where bank is acquirer or sponsor (consumer-facing, aggregated).
- **Methodology:** Directly measured acquiring volume in period.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.consumer_lending.merchant_acquiring_volume`

Net Charge-Off Rate

- **ID:** 1721
- **Type:** Financial
- **Unit:** %
- **Description:** Charge-offs net of recoveries as a share of average receivables.
- **Inputs:** `net_charge_offs`, `average_receivables`
- **Methodology:** NCO Rate = (Net Charge-Offs / Average Receivables) × 100
- **Calculation:** *`net_charge_offs / average_receivables` 100**
- **Catalog key:** `credit_granting.consumer_lending.net_charge_off_rate`

Origination Volume Growth

- **ID:** 1723
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in new consumer loan originations (aggregated).

- **Methodology:** Origination Growth = ((Current Originations − Prior Originations) / Prior Originations) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.consumer_lending.origination_volume_growth`

Card Payment Volume

- **ID:** 1724
- **Type:** Financial
- **Unit:** Currency
- **Description:** Total payment transaction volume for card portfolios (aggregated; fee revenue proxy).
- **Methodology:** Directly measured payment volume in period.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.consumer_lending.payment_volume`

Portfolio Yield

- **ID:** 1720
- **Type:** Financial
- **Unit:** %
- **Description:** Interest income on consumer receivables relative to average outstanding balance (aggregated).
- **Inputs:** `interest_income`, `average_receivables`
- **Methodology:** Portfolio Yield = (Interest Income / Average Receivables) × 100
- **Calculation:** *`interest_income / average_receivables` `100`**
- **Catalog key:** `credit_granting.consumer_lending.portfolio_yield`

RTP Payment Volume

- **ID:** 1727
- **Type:** Financial
- **Unit:** Currency
- **Description:** Real-time payments volume for consumer-facing rails (aggregated).
- **Methodology:** Directly measured RTP volume in period.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.consumer_lending.rtp_payment_volume`

Monthly Burn Rate

- **ID:** 1712
- **Type:** Financial
- **Unit:** Currency/month

- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `credit_granting.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1710
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `credit_granting.core.ccc`

Current Ratio

- **ID:** 1700
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `credit_granting.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1709
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `credit_granting.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1708
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` `days`**
- **Catalog key:** `credit_granting.core.dso`

EBITDA Margin

- **ID:** 1704
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `credit_granting.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1702
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `credit_granting.core.gross_margin`

Headcount (FTE)

- **ID:** 1714
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.

- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.core.headcount`

Net Profit Margin

- **ID:** 1705
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `credit_granting.core.net_margin`

Operating Cash Flow

- **ID:** 1711
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `credit_granting.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1703
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue 100`**
- **Catalog key:** `credit_granting.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1701

- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `credit_granting.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1706
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `credit_granting.core.revenue_growth`

Cash Runway

- **ID:** 1713
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `credit_granting.core.runway`

Working Capital

- **ID:** 1707
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**

- **Catalog key:** `credit_granting.core.working_capital`

Residential Mortgage 30+ Day Delinquency

- **ID:** 1791
- **Type:** Financial
- **Unit:** %
- **Description:** Share of mortgage balances past due 30+ days (lender book).
- **Inputs:** `delinquent_balance_30_plus`, `total_mortgage_balances`
- **Methodology:** Delinquency Rate = (Delinquent Balance 30+ / Total Mortgage Balances) × 100
- **Calculation:** *`delinquent_balance_30_plus / total_mortgage_balances` 100**
- **Catalog key:** `credit_granting.residential_mortgage.delinquency_rate_30d`

Residential Mortgage Origination Growth

- **ID:** 1793
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in new mortgage originations (aggregated).
- **Methodology:** Origination Growth = ((Current − Prior) / Prior) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `credit_granting.residential_mortgage.origination_volume_growth`

Residential Mortgage Portfolio Yield

- **ID:** 1790
- **Type:** Financial
- **Unit:** %
- **Description:** Interest income on residential mortgage book relative to average balances (lender view).
- **Inputs:** `interest_income`, `average_mortgage_balances`
- **Methodology:** Portfolio Yield = (Interest Income / Average Mortgage Balances) × 100
- **Calculation:** *`interest_income / average_mortgage_balances` 100**
- **Catalog key:** `credit_granting.residential_mortgage.portfolio_yield`

Residential Mortgage Weighted Average LTV

- **ID:** 1792
- **Type:** Financial
- **Unit:** %
- **Description:** Loan-weighted LTV on residential mortgage portfolio (lender view).

- **Inputs:** `weighted_ltv_numerator`, `total_mortgage_balances`
- **Methodology:** WA LTV = Σ(Loan Balance × LTV) / Σ Loan Balances
- **Calculation:** *`weighted_ltv_numerator / total_mortgage_balances` 100**
- **Catalog key:** `credit_granting.residential_mortgage.weighted_average_ltv`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Aggregated commercial loan book summary** — feeds Commercial Loan Growth, Commercial Portfolio Yield, Commercial NPL Ratio, Commercial Net Charge-Off Rate, Commercial Origination Volume Growth, Commercial 30+ Day Delinquency Rate, Weighted Average LTV, Portfolio DSCR, Sector Concentration Index, Revolver Utilization Rate, Sustainable Finance Share
- **Aggregated commercial treasury summary** — feeds Treasury Fee Yield, Commercial Payment Volume
- **Aggregated consumer loan tape summary** — feeds Portfolio Yield, Net Charge-Off Rate, 30+ Day Delinquency Rate, Origination Volume Growth, Card Payment Volume, BNPL Origination Growth, Embedded Finance Revenue Share, RTP Payment Volume, Merchant Acquiring Volume
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)

- **Aggregated residential mortgage book summary** — feeds Residential Mortgage Portfolio Yield, Residential Mortgage 30+ Day Delinquency, Residential Mortgage Weighted Average LTV, Residential Mortgage Origination Growth

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/financial_services.md

Financial Services

Pack file: `catalog/industries/financial_services` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6612</code>

23 metrics · 5 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>523150</code></dd>

<dt>NACE Rev.2</dt><dd><code>66.12</code></dd>

<dt>GICS 2023</dt><dd><code>402010</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (5) {#data-options}

Accounting Software

Catalog key <code>financial_services.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>financial_services.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>financial_services.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>financial_services.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>

<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

Aggregated investment banking summary

Catalog key <code>financial_services.investment_banking.trading_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>division</code></dt><dd>string</dd> <dt><code>trading_revenue</code></dt><dd>decimal</dd> <dt><code>advisory_fee_revenue</code></dt>
<dd>decimal</dd> <dt>
<code>total_revenue</code></dt><dd>decimal</dd> <dt><code>market_making_revenue</code></dt><dd>decimal</dd>
<dt><code>value_at_risk_99</code></dt>
<dd>string</dd> <dt>
<code>stressed_var_99</code></dt><dd>string</dd> <dt><code>prime_brokerage_balance_growth</code></dt>
<dd>string</dd> <dt>
<code>trading_venue_volume_growth</code></dt><dd>decimal</dd> </dl>

Metrics (23) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin

- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Advisory Fee Ratio
- Counterparty Exposure Ratio
- Market Making Revenue Share
- Prime Brokerage Balance Growth
- Stressed VaR 99%
- Trading Revenue Ratio
- Trading Venue Volume Growth
- Value at Risk 99%

Monthly Burn Rate

- **ID:** 1612
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** `cash_out - cash_in`
- **Catalog key:** `financial_services.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1610
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** `dso + days_inventory_outstanding - dpo`
- **Catalog key:** `financial_services.core.ccc`

Current Ratio

- **ID:** 1600
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `financial_services.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1609
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs` days_**
- **Catalog key:** `financial_services.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1608
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales` days_**
- **Catalog key:** `financial_services.core.dso`

EBITDA Margin

- **ID:** 1604
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`

- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue 100`**
- **Catalog key:** `financial_services.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1602
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue 100`**
- **Catalog key:** `financial_services.core.gross_margin`

Headcount (FTE)

- **ID:** 1614
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `financial_services.core.headcount`

Net Profit Margin

- **ID:** 1605
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `financial_services.core.net_margin`

Operating Cash Flow

- **ID:** 1611
- **Type:** Financial

- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `financial_services.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1603
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `financial_services.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1601
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `financial_services.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1606
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**

- **Catalog key:** `financial_services.core.revenue_growth`

Cash Runway

- **ID:** 1613
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `financial_services.core.runway`

Working Capital

- **ID:** 1607
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `financial_services.core.working_capital`

Advisory Fee Ratio

- **ID:** 1553
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue from advisory and underwriting mandates (investment banking; no deposit book).
- **Inputs:** `advisory_fee_revenue`, `total_revenue`
- **Methodology:** Advisory Fee Ratio = (Advisory Fee Revenue / Total Revenue) × 100
- **Calculation:** *`*advisory_fee_revenue / total_revenue` 100**
- **Catalog key:** `financial_services.investment_banking.advisory_fee_ratio`

Counterparty Exposure Ratio

- **ID:** 1592
- **Type:** Financial
- **Unit:** %
- **Description:** Largest counterparty exposure relative to tier 1 capital (proxy).

- **Methodology:** Counterparty Exposure Ratio = Top Counterparty EAD / Tier 1 Capital × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `financial_services.investment_banking.counterparty_exposure_ratio`

Market Making Revenue Share

- **ID:** 1593
- **Type:** Financial
- **Unit:** %
- **Description:** Share of division revenue from principal market-making activities.
- **Inputs:** `market_making_revenue`, `total_revenue`
- **Methodology:** Market Making Share = Market Making Revenue / Total Revenue × 100
- **Calculation:** *`market_making_revenue / total_revenue` 100**
- **Catalog key:** `financial_services.investment_banking.market_making_revenue_share`

Prime Brokerage Balance Growth

- **ID:** 1594
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in prime brokerage client balances (aggregated).
- **Methodology:** Balance Growth = ((Current − Prior) / Prior) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `financial_services.investment_banking.prime_brokerage_balance_growth`

Stressed VaR 99%

- **ID:** 1591
- **Type:** Financial
- **Unit:** Currency
- **Description:** Stressed one-day 99% VaR (aggregated).
- **Methodology:** Stressed VaR per firm risk policy
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `financial_services.investment_banking.stressed_var_99`

Trading Revenue Ratio

- **ID:** 1535
- **Type:** Financial
- **Unit:** %
- **Description:** Share of firm revenue from trading activities (investment bank / markets; no deposit book).
- **Inputs:** `trading_revenue`, `total_revenue`
- **Methodology:** Trading Revenue Ratio = (Trading Revenue / Total Revenue) × 100
- **Calculation:** *`trading_revenue / total_revenue` `100`**
- **Catalog key:** `financial_services.investment_banking.trading_revenue_ratio`

Trading Venue Volume Growth

- **ID:** 1615
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in executed trading venue volume (aggregated).
- **Methodology:** Volume Growth = ((Current − Prior) / Prior) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `financial_services.investment_banking.trading_venue_volume_growth`

Value at Risk 99%

- **ID:** 1590
- **Type:** Financial
- **Unit:** Currency
- **Description:** One-day 99% VaR for trading inventory (aggregated division).
- **Methodology:** Regulatory or internal VaR at 99% confidence, one-day horizon
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `financial_services.investment_banking.value_at_risk_99`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Aggregated investment banking summary** — feeds Trading Revenue Ratio, Advisory Fee Ratio, Value at Risk 99%, Stressed VaR 99%, Counterparty Exposure Ratio, Market Making Revenue Share, Prime Brokerage Balance Growth, Trading Venue Volume Growth

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/funds.md

Funds

Pack file: `catalog/industries/funds` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6630</code>

43 metrics · 7 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>523940</code></dd>

<dt>NACE Rev.2</dt><dd><code>66.30</code></dd>

<dt>GICS 2023</dt><dd><code>402030</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (7) {#data-options}

Accounting Software

Catalog key <code>funds.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>funds.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>funds.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>funds.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <div class="catalog-

field-list__row"><dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt>
<dd>decimal</dd> <dt>
<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>
<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd>
<dt><code>total_production_cost</code></dt>
<dd>decimal</dd> <dt>
<code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt>
<dd>decimal</dd> <dt>
<code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>
<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

Aggregated fund performance summary

Catalog key <code>funds.fund.fund_performance_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>fund_vehicle_id</code></dt><dd>id</dd> <dt><code>fund_irr</code></dt><dd>string</dd> <dt><code>distributions</code></dt><dd>string</dd>
<dt><code>nav</code></dt>
<dd>string</dd> <dt>
<code>paid_in_capital</code></dt><dd>string</dd> <dt><code>called_capital</code></dt><dd>string</dd> <dt><code>committed_capital</code></dt>
<dd>string</dd> <dt>
<code>performance_fees</code></dt><dd>string</dd> <dt><code>total_fund_revenue</code></dt><dd>decimal</dd>

<dt><code>liquidity_gate_frequency</code></dt><dd>string</dd> <dt><code>strategy_return_volatility</code></dt><dd>decimal</dd> </dl>

Aggregated fund summary

Catalog key <code>funds.fund.fund_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>fund_vehicle_id</code></dt><dd>id</dd> <dt><code>book_type</code></dt><dd>enum</dd> <dt><code>assets_under_management</code></dt><dd>string</dd> <dt><code>inflows</code></dt><dd>string</dd> <dt><code>outflows</code></dt><dd>string</dd> <dt><code>beginning_aum</code></dt><dd>string</dd> <dt><code>management_fees</code></dt><dd>string</dd> <dt><code>average_aum</code></dt><dd>string</dd> <dt><code>custody_fees</code></dt><dd>string</dd> <dt><code>average_auc</code></dt><dd>string</dd> <dt><code>failed_settlements</code></dt><dd>string</dd> <dt><code>total_settlement_instructions</code></dt><dd>decimal</dd> </dl>

Aggregated investment program summary

Catalog key <code>funds.operations.investment_program_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd>

<dt><code>reporting_entity_id</code></dt><dd>id</dd> <dt><code>closed_investments</code></dt><dd>string</dd> <dt><code>qualified_opportunities</code></dt><dd>string</dd> <dt><code>deployed_capital</code></dt><dd>string</dd> <dt><code>new_investments</code></dt><dd>string</dd> <dt><code>follow_on_investments</code></dt><dd>string</dd> <dt><code>eligible_portfolio_companies</code></dt><dd>string</dd> <dt><code>impaired_carrying_value</code></dt><dd>string</dd> <dt><code>portfolio_cost_basis</code></dt><dd>decimal</dd> <dt><code>deals_with_co_investors</code></dt><dd>string</dd> <dt><code>total_closed_deals</code></dt><dd>decimal</dd> <dt><code>committed_capital</code></dt><dd>string</dd> <dt><code>fundraising_target</code></dt><dd>string</dd> <dt><code>management_fees</code></dt><dd>string</dd> <dt><code>operating_expenses</code></dt><dd>string</dd> <dt><code>program_investments</code></dt><dd>string</dd> <dt><code>program_participants</code></dt><dd>string</dd> <dt><code>inbound_qualified_opportunities</code></dt><dd>string</dd> <dt><code>total_qualified_opportunities</code></dt><dd>decimal</dd> <dt><code>active_portfolio_company_count</code></dt><dd>decimal</dd> <dt><code>new_investments_count</code></dt><dd>decimal</dd> <dt><code>qualified_opportunities_count</code></dt><dd>decimal</dd> </dl>

Metrics (43) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)

- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Assets Under Custody
- Assets Under Management
- Capital Deployment Rate
- Custody Fee Yield
- DPI
- Fund IRR
- Fund Leverage Ratio
- Liquidity Gate Frequency
- Management Fee Yield
- Net Flow Rate
- Performance Fee Share
- RVPI
- Safekeeping Asset Growth
- Settlement Fail Rate
- Strategy Return Volatility
- TVPI
- Active Portfolio Company Count
- Average Investment Size
- Co-Investment Participation Rate
- Follow-On Investment Rate
- Fundraising Commitment Progress
- Inbound Attributed Pipeline Share
- Investment Close Rate
- Management Fee Operating Coverage
- New Investments Count
- Portfolio Write-Down Rate
- Program Investment Conversion Rate
- Qualified Opportunities Count

Monthly Burn Rate

- **ID:** 1982
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `funds.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1980
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `funds.core.ccc`

Current Ratio

- **ID:** 1970
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `funds.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1979
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`

- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `funds.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1978
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `funds.core.dso`

EBITDA Margin

- **ID:** 1974
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` 100**
- **Catalog key:** `funds.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1972
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` 100**
- **Catalog key:** `funds.core.gross_margin`

Headcount (FTE)

- **ID:** 1984
- **Type:** Operational

- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.core.headcount`

Net Profit Margin

- **ID:** 1975
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `funds.core.net_margin`

Operating Cash Flow

- **ID:** 1981
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `funds.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1973
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue 100`**
- **Catalog key:** `funds.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1971
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** `(current_assets - inventory) / current_liabilities`
- **Catalog key:** `funds.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1976
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `funds.core.revenue_growth`

Cash Runway

- **ID:** 1983
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** `current_cash_balance / burn_rate`
- **Catalog key:** `funds.core.runway`

Working Capital

- **ID:** 1977
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`

- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** `current_assets - current_liabilities`
- **Catalog key:** `funds.core.working_capital`

Assets Under Custody

- **ID:** 1911
- **Type:** Financial
- **Unit:** Currency
- **Description:** Fund and client assets held in custody (aggregated).
- **Methodology:** Directly measured AUC at reporting date
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.assets_under_custody`

Assets Under Management

- **ID:** 1900
- **Type:** Financial
- **Unit:** Currency
- **Description:** Total fund assets at period end across open-end, advised, and institutional books (aggregated).
- **Methodology:** Directly measured AUM at reporting date.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.assets_under_management`

Capital Deployment Rate

- **ID:** 1907
- **Type:** Financial
- **Unit:** %
- **Description:** Called capital relative to commitments (deployment pace).
- **Inputs:** `called_capital`, `committed_capital`
- **Methodology:** Deployment Rate = (Called Capital / Committed Capital) × 100
- **Calculation:** *`called_capital / committed_capital` 100**
- **Catalog key:** `funds.fund.capital_deployment_rate`

Custody Fee Yield

- **ID:** 1912
- **Type:** Financial
- **Unit:** %
- **Description:** Custody and safekeeping fees relative to average AUC.

- **Inputs:** `custody_fees`, `average_auc`
- **Methodology:** Custody Fee Yield = Custody Fees / Average AUC × 100
- **Calculation:** *`custody_fees / average_auc` `100`**
- **Catalog key:** `funds.fund.custody_fee_yield`

DPI

- **ID:** 1906
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Distributions to paid-in capital (cash returned).
- **Inputs:** `distributions`, `paid_in_capital`
- **Methodology:** DPI = Distributions / Paid-In Capital
- **Calculation:** **`distributions / paid_in_capital`**
- **Catalog key:** `funds.fund.dpi`

Fund IRR

- **ID:** 1904
- **Type:** Financial
- **Unit:** %
- **Description:** Internal rate of return for a fund vintage (aggregated, no LP identifiers).
- **Methodology:** IRR from aggregated cash flows and NAV (directly reported).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.fund_irr`

Fund Leverage Ratio

- **ID:** 1908
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Fund-level leverage relative to net asset value (aggregated).
- **Methodology:** Leverage Ratio = Total Fund Debt / NAV
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.fund_leverage_ratio`

Liquidity Gate Frequency

- **ID:** 1909
- **Type:** Operational
- **Unit:** %

- **Description:** Share of funds with active liquidity gates or suspensions in period.
- **Methodology:** Gate Frequency = Gated Funds / Total Funds × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.liquidity_gate_frequency`

Management Fee Yield

- **ID:** 1902
- **Type:** Financial
- **Unit:** %
- **Description:** Management and advisory fees relative to average AUM.
- **Inputs:** `management_fees`, `average_aum`
- **Methodology:** Fee Yield = (Management Fees / Average AUM) × 100
- **Calculation:** *`management_fees / average_aum` 100**
- **Catalog key:** `funds.fund.management_fee_yield`

Net Flow Rate

- **ID:** 1901
- **Type:** Financial
- **Unit:** %
- **Description:** Net fund flows as a percentage of beginning AUM.
- **Inputs:** `inflows`, `outflows`, `beginning_aum`
- **Methodology:** Net Flow Rate = ((Inflows − Outflows) / Beginning AUM) × 100
- **Calculation:** *`(inflows - outflows) / beginning_aum` 100**
- **Catalog key:** `funds.fund.net_flow_rate`

Performance Fee Share

- **ID:** 1903
- **Type:** Financial
- **Unit:** %
- **Description:** Carried interest and performance fees as a share of total fund revenue (aggregated).
- **Inputs:** `performance_fees`, `total_fund_revenue`
- **Methodology:** Performance Fee Share = (Performance Fees / Total Fund Revenue) × 100
- **Calculation:** *`performance_fees / total_fund_revenue` 100**
- **Catalog key:** `funds.fund.performance_fee_share`

RVPI

- **ID:** 1922
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Residual value to paid-in capital (unrealized NAV relative to contributed capital).
- **Inputs:** `nav`, `paid_in_capital`
- **Methodology:** RVPI = NAV / Paid-In Capital
- **Calculation:** **`nav / paid_in_capital`**
- **Catalog key:** `funds.fund.rvpi`

Safekeeping Asset Growth

- **ID:** 1914
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year growth in safekeeping and custody assets.
- **Methodology:** Growth = ((Current AUC − Prior) / Prior) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.safekeeping_asset_growth`

Settlement Fail Rate

- **ID:** 1913
- **Type:** Operational
- **Unit:** %
- **Description:** Failed settlements as a share of total settlement instructions.
- **Inputs:** `failed_settlements`, `total_settlement_instructions`
- **Methodology:** Fail Rate = Failed Settlements / Total Instructions × 100
- **Calculation:** *`failed_settlements / total_settlement_instructions` 100**
- **Catalog key:** `funds.fund.settlement_fail_rate`

Strategy Return Volatility

- **ID:** 1910
- **Type:** Financial
- **Unit:** %
- **Description:** Annualized volatility of fund or strategy returns (aggregated sleeve).
- **Methodology:** Standard deviation of periodic returns annualized
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.fund.strategy_return_volatility`

TVPI

- **ID:** 1905
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Total value to paid-in capital (distributions plus NAV).
- **Inputs:** `distributions`, `nav`, `paid_in_capital`
- **Methodology:** TVPI = (Distributions + NAV) / Paid-In Capital
- **Calculation:** `(distributions + nav) / paid_in_capital`
- **Catalog key:** `funds.fund.tvpi`

Active Portfolio Company Count

- **ID:** 1927
- **Type:** Operational
- **Unit:** Count
- **Description:** Count of active portfolio companies at period end (aggregated).
- **Methodology:** Directly measured count at reporting date.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.operations.active_portfolio_company_count`

Average Investment Size

- **ID:** 1924
- **Type:** Financial
- **Unit:** Currency
- **Description:** Average capital deployed per new investment in the period (aggregated).
- **Inputs:** `deployed_capital`, `new_investments`
- **Methodology:** Average Size = Deployed Capital / New Investments
- **Calculation:** `deployed_capital / new_investments`
- **Catalog key:** `funds.operations.average_investment_size`

Co-Investment Participation Rate

- **ID:** 1930
- **Type:** Operational
- **Unit:** %
- **Description:** Share of closed deals that included external co-investors or syndicate participants.
- **Inputs:** `deals_with_co_investors`, `total_closed_deals`

- **Methodology:** Co-Investment Rate = (Deals with Co-Investors / Total Closed Deals) × 100
- **Calculation:** *`deals_with_co_investors / total_closed_deals` 100**
- **Catalog key:** `funds.operations.co_investment_participation_rate`

Follow-On Investment Rate

- **ID:** 1928
- **Type:** Operational
- **Unit:** %
- **Description:** Share of eligible portfolio companies that received follow-on capital in the period.
- **Inputs:** `follow_on_investments`, `eligible_portfolio_companies`
- **Methodology:** Follow-On Rate = (Follow-On Investments / Eligible Portfolio Companies) × 100
- **Calculation:** *`follow_on_investments / eligible_portfolio_companies` 100**
- **Catalog key:** `funds.operations.follow_on_investment_rate`

Fundraising Commitment Progress

- **ID:** 1931
- **Type:** Financial
- **Unit:** %
- **Description:** Committed capital relative to fundraising target for a vintage (aggregated).
- **Inputs:** `committed_capital`, `fundraising_target`
- **Methodology:** Progress = (Committed Capital / Fundraising Target) × 100
- **Calculation:** *`committed_capital / fundraising_target` 100**
- **Catalog key:** `funds.operations.fundraising_commitment_progress`

Inbound Attributed Pipeline Share

- **ID:** 1934
- **Type:** Operational
- **Unit:** %
- **Description:** Share of qualified opportunities attributed to inbound channels in the period.
- **Inputs:** `inbound_qualified_opportunities`, `total_qualified_opportunities`
- **Methodology:** Inbound Share = (Inbound Qualified Opportunities / Total Qualified Opportunities) × 100
- **Calculation:** *`inbound_qualified_opportunities / total_qualified_opportunities` 100**

- **Catalog key:** `funds.operations.inbound_attributed_pipeline_share`

Investment Close Rate

- **ID:** 1923
- **Type:** Operational
- **Unit:** %
- **Description:** Share of qualified opportunities that closed as new investments in the period (aggregated).
- **Inputs:** `closed_investments`, `qualified_opportunities`
- **Methodology:** Close Rate = (Closed Investments / Qualified Opportunities) × 100
- **Calculation:** *`closed_investments / qualified_opportunities` 100**
- **Catalog key:** `funds.operations.investment_close_rate`

Management Fee Operating Coverage

- **ID:** 1932
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Management fees relative to operating expenses at the manager level (aggregated).
- **Inputs:** `management_fees`, `operating_expenses`
- **Methodology:** Coverage = Management Fees / Operating Expenses
- **Calculation:** **`management_fees / operating_expenses`**
- **Catalog key:** `funds.operations.management_fee_operating_coverage`

New Investments Count

- **ID:** 1925
- **Type:** Operational
- **Unit:** Count
- **Description:** Count of new investments closed in the period (aggregated).
- **Methodology:** Directly measured count at reporting date.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.operations.new_investments_count`

Portfolio Write-Down Rate

- **ID:** 1929
- **Type:** Financial
- **Unit:** %
- **Description:** Impaired or written-down carrying value as a share of portfolio cost basis.

- **Inputs:** `impaired_carrying_value`, `portfolio_cost_basis`
- **Methodology:** Write-Down Rate = (Impaired Carrying Value / Portfolio Cost Basis) × 100
- **Calculation:** *`impaired_carrying_value / portfolio_cost_basis` 100**
- **Catalog key:** `funds.operations.portfolio_write_down_rate`

Program Investment Conversion Rate

- **ID:** 1933
- **Type:** Operational
- **Unit:** %
- **Description:** Share of program or cohort participants that became portfolio investments in the period.
- **Inputs:** `program_investments`, `program_participants`
- **Methodology:** Conversion Rate = (Program Investments / Program Participants) × 100
- **Calculation:** *`program_investments / program_participants` 100**
- **Catalog key:** `funds.operations.program_investment_conversion_rate`

Qualified Opportunities Count

- **ID:** 1926
- **Type:** Operational
- **Unit:** Count
- **Description:** Count of opportunities that passed initial screening in the period (aggregated).
- **Methodology:** Directly measured count at reporting date.
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `funds.operations.qualified_opportunities_count`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Aggregated fund performance summary** — feeds Performance Fee Share, Fund IRR, TVPI, DPI, Capital Deployment Rate, Fund Leverage Ratio, Liquidity Gate Frequency, Strategy Return Volatility, RVPI
- **Aggregated fund summary** — feeds Assets Under Management, Net Flow Rate, Management Fee Yield, Assets Under Custody, Custody Fee Yield, Settlement Fail Rate, Safekeeping Asset Growth
- **Aggregated investment program summary** — feeds Investment Close Rate, Average Investment Size, New Investments Count, Qualified Opportunities Count, Active Portfolio Company Count, Follow-On Investment Rate, Portfolio Write-Down Rate, Co-Investment Participation Rate, Fundraising Commitment Progress, Management Fee Operating Coverage, Program Investment Conversion Rate, Inbound Attributed Pipeline Share

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/healthcare.md

Healthcare

Pack file: `catalog/industries/healthcare` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">8610</code>

37 metrics · 19 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>622110</code></dd>

<dt>NACE Rev.2</dt><dd><code>86.10</code></dd>

<dt>GICS 2023</dt><dd><code>351020</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (19) {#data-options}

Accounts payable aging

Catalog key <code>Allhealthcareds-ap-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>supplier_name</code></dt><dd>string</dd> <dt><code>bill_id</code></dt><dd>id</dd> <dt><code>bill_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt><dd>string</dd> <dt><code>accounts_payable</code></dt><dd>decimal</dd> <div class="catalog-

field-list__row"><dt><code>cogs</code></dt><dd>string</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Accounts receivable aging

Catalog key <code>Allhealthcareds-ar-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_name</code></dt><dd>string</dd> <dt><code>invoice_id</code></dt><dd>id</dd> <dt><code>invoice_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt><dd>string</dd> <dt><code>accounts_receivable</code></dt><dd>decimal</dd> <dt><code>total_credit_sales</code></dt><dd>decimal</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Balance sheet

Catalog key <code>Allhealthcareds-balance-sheet</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd>

<dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>inventory</code></dt><dd>string</dd> </dl>

Bank statements

Catalog key <code>Allhealthcareds-bank-statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>transaction_date</code></dt><dd>date</dd> <dt><code>account_id</code></dt><dd>id</dd> <dt><code>description</code></dt><dd>string</dd> <dt><code>amount</code></dt><dd>decimal</dd> <dt><code>direction</code></dt><dd>string</dd> <dt><code>balance</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>net_income</code></dt><dd>string</dd> <dt><code>non_cash_charges</code></dt><dd>string</dd> <dt><code>increase_in_working_capital</code></dt><dd>string</dd> <dt><code>cash_out</code></dt><dd>string</dd> <dt><code>cash_in</code></dt><dd>string</dd> <dt><code>current_cash_balance</code></dt><dd>string</dd> </dl>

CRM / Sales pipeline

Catalog key <code>Allhealthcareds-crm-pipeline</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd>

<dt><code>opportunity_id</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>stage</code></dt><dd>enum</dd>
<dt><code>amount</code></dt>
<dd>decimal</dd> <dt>
<code>close_date</code></dt><dd>date</dd> <dt><code>owner</code></dt><dd>string</dd> <dt><code>probability</code></dt><dd>string</dd>
<dt><code>active_subscriptions</code>
</dt><dd>string</dd> <dt>
<code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>sales_marketing_spend</code></dt><dd>string</dd>
<dt><code>new_customers</code></dt>
<dd>string</dd> </dl>

General ledger

Catalog key <code>Allhealthcareds-general-ledger</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>journal_entry_id</code></dt><dd>id</dd>
<dt><code>debit</code></dt>
<dd>string</dd> <dt><code>credit</code>
</dt><dd>string</dd> <dt>
<code>period</code></dt><dd>enum</dd> <dt><code>currency</code></dt><dd>currency</dd> </dl>

Marketing spend records

Catalog key <code>Allhealthcareds-marketing-spend</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>channel</code></dt><dd>enum</dd> <div class="catalog-field-
list__row"><dt><code>campaign</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>spend</code></dt><dd>string</dd>
<dt><code>leads</code></dt>
<dd>string</dd> <dt>
<code>new_customers</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>currency</code></dt><dd>currency</dd> <div
class="catalog-field-list__row"><dt><code>sales_marketing_spend</code></dt>
<dd>string</dd> </dl>

Product usage logs

Catalog key <code>Allhealthcareds-product-usage</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>customer_id</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>active_users</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>sessions</code></dt><dd>string</dd>
<dt><code>feature_adoption</code></dt>
<dd>string</dd> <dt>
<code>usage_minutes</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>starting_mrr</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>expansion_mrr</code></dt>
<dd>string</dd> <dt>
<code>contraction_mrr</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>churned_mrr</code></dt><dd>string</dd> </dl>

Subscription billing records

Catalog key <code>Allhealthcareds-subscription-billing</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>customer_id</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>plan_name</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>mrr</code></dt><dd>string</dd>
<dt><code>billing_period</code></dt>
<dd>enum</dd> <dt><code>status</code>
</dt><dd>enum</dd> <dt>
<code>start_date</code></dt><dd>date</dd> <div class="catalog-field-
list__row"><dt><code>end_date</code></dt><dd>date</dd> <div
class="catalog-field-list__row"><dt><code>active_subscriptions</code></dt>
<dd>string</dd> <dt>
<code>avg_monthly_price</code></dt><dd>decimal</dd> <div class="catalog-
field-list__row"><dt><code>starting_mrr</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>expansion_mrr</code></dt>
<dd>string</dd> <dt>
<code>contraction_mrr</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>churned_mrr</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>avg_revenue_per_account</code></dt>
<dd>decimal</dd> <dt>
<code>churn_rate</code></dt><dd>decimal</dd> </dl>

Trial balance

Catalog key <code>Allhealthcareds-trial-balance</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>account_code</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>account_name</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>debit_balance</code></dt>
<dd>string</dd> <dt>
<code>credit_balance</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>period</code></dt><dd>enum</dd> </dl>

De-identified operational census export

Catalog key <code>HcCensus911CatalogOptionKey01</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>service_line</code></dt><dd>string</dd> <dt><code>admissions</code></dt><dd>string</dd> <dt><code>discharges</code></dt><dd>string</dd> <dt><code>patient_days</code></dt><dd>string</dd> <dt><code>occupied_beds</code></dt><dd>decimal</dd> <dt><code>available_beds</code></dt><dd>decimal</dd> </dl>

De-identified provider operations summary

Catalog key <code>HcProviderOps916CatalogOptionKey05</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>occupied_beds</code></dt><dd>decimal</dd> <dt><code>available_beds</code></dt><dd>decimal</dd> <dt><code>or_time_used</code></dt><dd>string</dd> <dt><code>or_time_available</code></dt><dd>string</dd> <dt><code>ed_wait_time</code></dt><dd>string</dd> <dt><code>case_mix_index</code></dt><dd>string</dd> <dt><code>total_operating_cost</code></dt><dd>decimal</dd> <dt><code>discharges</code></dt><dd>string</dd> <dt><code>accounts_receivable</code></dt><dd>decimal</dd> <dt><code>net_patient_revenue</code></dt><dd>decimal</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>clean_claims</code></dt><dd>string</dd> <dt><code>total_claims</code></dt><dd>decimal</dd> <div

class="catalog-field-list__row"><dt><code>inpatient_deaths</code></dt>
<dd>string</dd> <dt>
<code>infections</code></dt><dd>string</dd> <dt><code>patient_days</code></dt><dd>string</dd> </dl>

De-identified quality outcomes summary

Catalog key <code>HcQual914CatalogOptionKey04</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>cohort_id</code></dt><dd>id</dd> <dt><code>discharges</code></dt><dd>string</dd> <dt><code>readmissions_30d</code></dt>
<dd>string</dd> </dl>

Revenue cycle summary export

Catalog key <code>HcRcm913CatalogOptionKey03</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>payer_category</code></dt><dd>string</dd> <dt><code>claims_submitted</code></dt><dd>string</dd>
<dt><code>claims_denied</code></dt>
<dd>string</dd> <dt>
<code>amount_billed</code></dt><dd>decimal</dd> <dt><code>amount_collected</code></dt><dd>decimal</dd> </dl>

Scheduling utilization summary

Catalog key <code>HcSched912CatalogOptionKey02</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>department</code></dt><dd>string</dd> <dt><code>staffed_hours</code></dt><dd>decimal</dd> <dt><code>scheduled_hours</code></dt><dd>decimal</dd> <dt><code>encounter_count</code></dt><dd>decimal</dd> </dl>

Accounting Software

Catalog key <code>healthcare.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>healthcare.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>healthcare.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>healthcare.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>

<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd>
<dt><code>total_production_cost</code></dt>
<dd>decimal</dd> <dt>
<code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt>
<dd>decimal</dd> <dt>
<code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>
<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

Metrics (37) {#metrics}

- Annual Recurring Revenue (ARR)
- Customer Acquisition Cost (CAC)
- Gross Revenue Churn Rate
- Customer Lifetime Value (LTV)
- LTV : CAC Ratio
- Monthly Recurring Revenue (MRR)
- Net Revenue Retention (NRR)
- Rule of 40
- Claim Denial Rate
- Average Length of Stay
- Readmission Rate (30-day proxy)
- Staffing Ratio
- Patient Volume
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin

- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Bed Occupancy Rate
- Case Mix Index
- Cost per Discharge
- ED Wait Time
- Operating Room Utilization
- Healthcare-Associated Infection Rate
- Inpatient Mortality Rate
- AR Days
- Clean Claim Rate

Annual Recurring Revenue (ARR)

- **ID:** 1020
- **Type:** Financial
- **Unit:** Currency/year
- **Description:** Annualized run-rate of recurring subscription revenue.
- **Methodology:** ARR = MRR × 12
- **Calculation:** *`mrr 12`**
- **Catalog key:** `Allhealthcaresaas-arr`

Customer Acquisition Cost (CAC)

- **ID:** 1038
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Fully-loaded sales & marketing cost to acquire one new customer.
- **Inputs:** `sales_marketing_spend`, `new_customers`
- **Methodology:** CAC = Total Sales & Marketing Spend / New Customers Acquired
- **Calculation:** **`sales_marketing_spend / new_customers`**
- **Catalog key:** `Allhealthcaresaas-cac`

Gross Revenue Churn Rate

- **ID:** 1032
- **Type:** Operational
- **Unit:** %
- **Description:** Percentage of recurring revenue lost to cancellations and downgrades.
- **Inputs:** `churned_mrr`, `starting_mrr`
- **Methodology:** Gross Churn = (Churned MRR / Starting MRR) × 100
- **Calculation:** *`churned_mrr / starting_mrr` 100**
- **Catalog key:** `Allhealthcaresaas-gross-churn`

Customer Lifetime Value (LTV)

- **ID:** 1044
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Expected gross-margin revenue from a customer over their lifetime.
- **Inputs:** `avg_revenue_per_account`, `gross_margin_pct`, `churn_rate`
- **Methodology:** LTV = (Average Revenue per Account × Gross Margin %) / Churn Rate
- **Calculation:** *`(avg_revenue_per_account` gross_margin_pct / 100) / churn_rate**
- **Catalog key:** `Allhealthcaresaas-ltv`

LTV : CAC Ratio

- **ID:** 1050
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Ratio of customer lifetime value to acquisition cost — core SaaS unit economics.
- **Methodology:** LTV:CAC = Customer Lifetime Value / Customer Acquisition Cost
- **Calculation:** **`ltv / cac`**
- **Catalog key:** `Allhealthcaresaas-ltv-cac`

Monthly Recurring Revenue (MRR)

- **ID:** 1014
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Normalized recurring subscription revenue billed in a month.
- **Inputs:** `active_subscriptions`, `avg_monthly_price`
- **Methodology:** MRR = Σ (Active Subscriptions × Monthly Subscription Price)

- **Calculation:** *`active_subscriptions` avg_monthly_price**
- **Catalog key:** `Allhealthcaresaas-mrr`

Net Revenue Retention (NRR)

- **ID:** 1026
- **Type:** Financial
- **Unit:** %
- **Description:** Revenue retained and expanded from existing customers, net of churn and contraction.
- **Inputs:** `starting_mrr`, `expansion_mrr`, `contraction_mrr`, `churned_mrr`
- **Methodology:** NRR = ((Starting MRR + Expansion − Contraction − Churn) / Starting MRR) × 100
- **Calculation:** *`(starting_mrr + expansion_mrr - contraction_mrr - churned_mrr) / starting_mrr` 100**
- **Catalog key:** `Allhealthcaresaas-nrr`

Rule of 40

- **ID:** 1056
- **Type:** Financial
- **Unit:** %
- **Description:** Balance of growth and profitability for recurring-revenue businesses.
- **Inputs:** `revenue_growth_pct`, `profit_margin_pct`
- **Methodology:** Rule of 40 = Revenue Growth % + Profit Margin %
- **Calculation:** **`revenue_growth_pct + profit_margin_pct`**
- **Catalog key:** `Allhealthcaresaas-rule-of-40`

Claim Denial Rate

- **ID:** 915
- **Type:** Financial
- **Unit:** %
- **Description:** Share of claims denied on first submission (revenue cycle quality).
- **Inputs:** `denied_claims`, `submitted_claims`
- **Methodology:** Denied Claims / Submitted Claims × 100
- **Calculation:** *`denied_claims / submitted_claims` 100**
- **Catalog key:** `HcDenial915CatalogMetricKey05`

Average Length of Stay

- **ID:** 912
- **Type:** Operational

- **Unit:** Days
- **Description:** Mean inpatient days per discharge (aggregated by service line).
- **Inputs:** `inpatient_days`, `discharges`
- **Methodology:** ALOS = Total Inpatient Days / Discharges
- **Calculation:** **`inpatient_days / discharges`**
- **Catalog key:** `HcLos912CatalogMetricKey02`

Readmission Rate (30-day proxy)

- **ID:** 913
- **Type:** Operational
- **Unit:** %
- **Description:** Share of discharges with a related readmission within 30 days (aggregated quality signal).
- **Inputs:** `readmissions`, `eligible_discharges`
- **Methodology:** Readmissions / Eligible Discharges × 100 (de-identified cohort)
- **Calculation:** *_`readmissions / eligible_discharges` 100_**
- **Catalog key:** `HcReadmit913CatalogMetricKey03`

Staffing Ratio

- **ID:** 914
- **Type:** Operational
- **Unit:** Ratio
- **Description:** Clinical staff hours per occupied bed or per encounter (aggregated).
- **Inputs:** `staff_hours`, `occupied_beds`
- **Methodology:** Staff Hours / Occupied Beds (or / Encounters) for the period
- **Calculation:** **`staff_hours / occupied_beds`**
- **Catalog key:** `HcStaff914CatalogMetricKey04`

Patient Volume

- **ID:** 911
- **Type:** Operational
- **Unit:** Count
- **Description:** Count of patient encounters or admissions in the reporting period (aggregated, de-identified).
- **Methodology:** Sum of distinct encounter or admission records in period
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `HcVol911CatalogMetricKey01`

Monthly Burn Rate

- **ID:** 1002
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `healthcare.core.burn_rate`

Cash Conversion Cycle

- **ID:** 990
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `healthcare.core.ccc`

Current Ratio

- **ID:** 930
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `healthcare.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 984
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`

- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `healthcare.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 978
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `healthcare.core.dso`

EBITDA Margin

- **ID:** 954
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `healthcare.core.ebitda_margin`

Gross Profit Margin

- **ID:** 942
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `healthcare.core.gross_margin`

Headcount (FTE)

- **ID:** 1080
- **Type:** Operational

- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `healthcare.core.headcount`

Net Profit Margin

- **ID:** 960
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `healthcare.core.net_margin`

Operating Cash Flow

- **ID:** 996
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `healthcare.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 948
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue 100`**
- **Catalog key:** `healthcare.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 936
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** `(current_assets - inventory) / current_liabilities`
- **Catalog key:** `healthcare.core.quick_ratio`

Revenue Growth Rate

- **ID:** 966
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `healthcare.core.revenue_growth`

Cash Runway

- **ID:** 1008
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** `current_cash_balance / burn_rate`
- **Catalog key:** `healthcare.core.runway`

Working Capital

- **ID:** 972
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`

- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** `current_assets - current_liabilities`
- **Catalog key:** `healthcare.core.working_capital`

Bed Occupancy Rate

- **ID:** 1400
- **Type:** Operational
- **Unit:** %
- **Description:** Bed Occupancy Rate (aggregated, de-identified).
- **Inputs:** `occupied_beds`, `available_beds`
- **Methodology:** Bed Occupancy = (Occupied Beds / Available Beds) × 100
- **Calculation:** *`occupied_beds / available_beds` `100`**
- **Catalog key:** `healthcare.provider.bed_occupancy_rate`

Case Mix Index

- **ID:** 1403
- **Type:** Operational
- **Unit:** Index
- **Description:** Case Mix Index (aggregated, de-identified).
- **Methodology:** Average DRG weight across discharges (directly measured).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `healthcare.provider.case_mix_index`

Cost per Discharge

- **ID:** 1404
- **Type:** Financial
- **Unit:** Currency
- **Description:** Cost per Discharge (aggregated, de-identified).
- **Inputs:** `total_operating_cost`, `discharges`
- **Methodology:** Cost per Discharge = Total Operating Cost / Discharges
- **Calculation:** `total_operating_cost / discharges`
- **Catalog key:** `healthcare.provider.cost_per_discharge`

ED Wait Time

- **ID:** 1402
- **Type:** Operational
- **Unit:** Minutes
- **Description:** ED Wait Time (aggregated, de-identified).

- **Methodology:** Mean minutes from ED arrival to provider (directly measured).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `healthcare.provider.ed_wait_time`

Operating Room Utilization

- **ID:** 1401
- **Type:** Operational
- **Unit:** %
- **Description:** Operating Room Utilization (aggregated, de-identified).
- **Inputs:** `or_time_used`, `or_time_available`
- **Methodology:** OR Utilization = (OR Time Used / OR Time Available) × 100
- **Calculation:** *`or_time_used / or_time_available` 100**
- **Catalog key:** `healthcare.provider.or_utilization`

Healthcare-Associated Infection Rate

- **ID:** 1408
- **Type:** Operational
- **Unit:** Per 1000 patient-days
- **Description:** Healthcare-Associated Infection Rate (aggregated, de-identified).
- **Inputs:** `infections`, `patient_days`
- **Methodology:** HAI Rate = (Infections / Patient Days) × 1000
- **Calculation:** *`infections / patient_days` 1000**
- **Catalog key:** `healthcare.quality.infection_rate`

Inpatient Mortality Rate

- **ID:** 1407
- **Type:** Operational
- **Unit:** %
- **Description:** Inpatient Mortality Rate (aggregated, de-identified).
- **Inputs:** `inpatient_deaths`, `discharges`
- **Methodology:** Mortality Rate = (Inpatient Deaths / Discharges) × 100
- **Calculation:** *`inpatient_deaths / discharges` 100**
- **Catalog key:** `healthcare.quality.mortality_rate`

AR Days

- **ID:** 1405
- **Type:** Financial
- **Unit:** Days

- **Description:** AR Days (aggregated, de-identified).
- **Inputs:** `accounts_receivable`, `net_patient_revenue`, `days`
- **Methodology:** AR Days = (Accounts Receivable / Net Patient Revenue) × Days
- **Calculation:** *`accounts_receivable / net_patient_revenue` days**
- **Catalog key:** `healthcare.revenue_cycle.ar_days`

Clean Claim Rate

- **ID:** 1406
- **Type:** Financial
- **Unit:** %
- **Description:** Clean Claim Rate (aggregated, de-identified).
- **Inputs:** `clean_claims`, `total_claims`
- **Methodology:** Clean Claim Rate = (Clean Claims / Total Claims) × 100
- **Calculation:** *`clean_claims / total_claims` 100**
- **Catalog key:** `healthcare.revenue_cycle.clean_claim_rate`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounts payable aging** — feeds Days Payable Outstanding (DPO), Cash Conversion Cycle
- **Accounts receivable aging** — feeds Days Sales Outstanding (DSO), Cash Conversion Cycle
- **Balance sheet** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital, Cash Conversion Cycle
- **Bank statements** — feeds Operating Cash Flow, Monthly Burn Rate, Cash Runway

- **CRM / Sales pipeline** — feeds Monthly Recurring Revenue (MRR), Customer Acquisition Cost (CAC)
- **General ledger** — feeds Gross Profit Margin, Operating Margin (EBIT), EBITDA Margin, Net Profit Margin, Revenue Growth Rate, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Operating Cash Flow, Customer Acquisition Cost (CAC)
- **Marketing spend records** — feeds Customer Acquisition Cost (CAC), LTV : CAC Ratio
- **Product usage logs** — feeds Net Revenue Retention (NRR), Gross Revenue Churn Rate
- **Subscription billing records** — feeds Monthly Recurring Revenue (MRR), Annual Recurring Revenue (ARR), Net Revenue Retention (NRR), Gross Revenue Churn Rate, Customer Lifetime Value (LTV), LTV : CAC Ratio, Rule of 40
- **Trial balance** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital
- **De-identified operational census export** — feeds Patient Volume, Average Length of Stay, Bed Occupancy Rate
- **De-identified provider operations summary** — feeds Bed Occupancy Rate, Operating Room Utilization, ED Wait Time, Case Mix Index, Cost per Discharge, AR Days, Clean Claim Rate, Inpatient Mortality Rate, Healthcare-Associated Infection Rate
- **De-identified quality outcomes summary** — feeds Readmission Rate (30-day proxy)
- **Revenue cycle summary export** — feeds Claim Denial Rate
- **Scheduling utilization summary** — feeds Patient Volume, Staffing Ratio
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE), Staffing Ratio
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/index.md

Industries

Each **industry pack** is a vertical analysis lens from Ambient Core: KPI metrics, typed data options, and calc expressions you can map into uploads and governed products.

Use the docs sidebar under **Industries** to open a pack (for example Banking or Construction).

For the full pack list with ISIC codes, metric counts, and data-option counts, see the Reference catalog. Shared corporate-finance templates that expand into every pack: Core metrics layer.

---

catalog/industries/insurance.md

Insurance

Pack file: `catalog/industries/insurance` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6512</code>

21 metrics · 5 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>524126</code></dd>

<dt>NACE Rev.2</dt><dd><code>65.12</code> (also <code>65.11</code>) </dd>

<dt>GICS 2023</dt><dd> <code>403010</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (5) {#data-options}

Accounting Software

Catalog key <code>insurance.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>insurance.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>insurance.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>insurance.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <div class="catalog-field-

list__row"><dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>
<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

Aggregated insurance loss summary

Catalog key <code>insurance.property_casualty.loss_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>line_of_business</code></dt><dd>string</dd> <dt><code>entity_segment</code></dt><dd>string</dd> <dt><code>incurred_losses</code></dt>
<dd>string</dd> <dt>
<code>underwriting_expense</code></dt><dd>string</dd> <dt><code>earned_premiums</code></dt>
<dd>string</dd> <dt>
<code>premium_growth</code></dt><dd>string</dd> <dt><code>benefits_paid</code></dt><dd>string</dd> <dt><code>catastrophe_losses</code></dt>
<dd>string</dd> <dt>
<code>prior_reserve_development</code></dt><dd>string</dd> </dl>

Metrics (21) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin

- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Benefit Ratio
- Catastrophe Loss Ratio
- Combined Ratio
- Loss Ratio
- Premium Growth
- Reserve Development Ratio

Monthly Burn Rate

- **ID:** 1632
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `insurance.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1630
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `insurance.core.ccc`

Current Ratio

- **ID:** 1620
- **Type:** Financial

- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `insurance.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1629
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` `days`**
- **Catalog key:** `insurance.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1628
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` `days`**
- **Catalog key:** `insurance.core.dso`

EBITDA Margin

- **ID:** 1624
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `insurance.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1622
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `insurance.core.gross_margin`

Headcount (FTE)

- **ID:** 1634
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `insurance.core.headcount`

Net Profit Margin

- **ID:** 1625
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` `100`**
- **Catalog key:** `insurance.core.net_margin`

Operating Cash Flow

- **ID:** 1631
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`

- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** `net_income + non_cash_charges - increase_in_working_capital`
- **Catalog key:** `insurance.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1623
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `insurance.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1621
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** `(current_assets - inventory) / current_liabilities`
- **Catalog key:** `insurance.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1626
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `insurance.core.revenue_growth`

Cash Runway

- **ID:** 1633

- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `insurance.core.runway`

Working Capital

- **ID:** 1627
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `insurance.core.working_capital`

Benefit Ratio

- **ID:** 1555
- **Type:** Financial
- **Unit:** %
- **Description:** Life insurance benefits and claims relative to premiums (aggregated line).
- **Inputs:** `benefits_paid`, `earned_premiums`
- **Methodology:** Benefit Ratio = (Benefits Paid / Earned Premiums) × 100
- **Calculation:** *`benefits_paid / earned_premiums` `100`**
- **Catalog key:** `insurance.life.benefit_ratio`

Catastrophe Loss Ratio

- **ID:** 1575
- **Type:** Financial
- **Unit:** %
- **Description:** Cat losses relative to earned premiums (P&C).
- **Inputs:** `catastrophe_losses`, `earned_premiums`
- **Methodology:** Cat Loss Ratio = Catastrophe Losses / Earned Premiums × 100
- **Calculation:** *`catastrophe_losses / earned_premiums` `100`**
- **Catalog key:** `insurance.property_casualty.catastrophe_loss_ratio`

Combined Ratio

- **ID:** 1547
- **Type:** Financial
- **Unit:** %
- **Description:** Losses and expenses relative to earned premiums (P&C).
- **Inputs:** `incurred_losses`, `underwriting_expense`, `earned_premiums`
- **Methodology:** Combined Ratio = ((Incurred Losses + Underwriting Expense) / Earned Premiums) × 100
- **Calculation:** *`(incurred_losses + underwriting_expense) / earned_premiums` `100`**
- **Catalog key:** `insurance.property_casualty.combined_ratio`

Loss Ratio

- **ID:** 1548
- **Type:** Financial
- **Unit:** %
- **Description:** Incurred losses relative to earned premiums (P&C).
- **Inputs:** `incurred_losses`, `earned_premiums`
- **Methodology:** Loss Ratio = (Incurred Losses / Earned Premiums) × 100
- **Calculation:** *`incurred_losses / earned_premiums` `100`**
- **Catalog key:** `insurance.property_casualty.loss_ratio`

Premium Growth

- **ID:** 1549
- **Type:** Financial
- **Unit:** %
- **Description:** Year-over-year earned premium growth (aggregated).
- **Methodology:** Premium Growth = ((Current Premiums − Prior Premiums) / Prior Premiums) × 100
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `insurance.property_casualty.premium_growth`

Reserve Development Ratio

- **ID:** 1574
- **Type:** Financial
- **Unit:** %
- **Description:** Change in prior-year reserves relative to earned premiums (P&C).
- **Methodology:** Reserve Development = (Prior Reserve Development / Earned Premiums) × 100

- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `insurance.property_casualty.reserve_development_ratio`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Aggregated insurance loss summary** — feeds Combined Ratio, Loss Ratio, Premium Growth, Benefit Ratio, Reserve Development Ratio, Catastrophe Loss Ratio

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/life_sciences.md

Life Sciences

Pack file: `catalog/industries/life_sciences` in ambient-core.

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">7210</code>

20 metrics · 6 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>541714</code></dd>

<dt>NACE Rev.2</dt><dd><code>72.11</code></dd>

<dt>GICS 2023</dt><dd><code>352010</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (6) {#data-options}

Accounting Software

Catalog key <code>life_sciences.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt>

<code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd>
<dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd>
<dt><code>total_operational_cost</code>
</dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>life_sciences.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt>
<dd>date</dd> </dl>

HR records

Catalog key <code>life_sciences.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>life_sciences.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

De-identified clinical trial summary

Catalog key <code>life_sciences.rnd.clinical_trial_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>program_id</code></dt><dd>id</dd> <dt><code>phase</code></dt><dd>string</dd> <div

class="catalog-field-list__row"><dt><code>patients_enrolled</code></dt>
<dd>string</dd> <dt>
<code>phases_passed</code></dt><dd>string</dd> <dt><code>phases_attempted</code></dt><dd>string</dd> <dt><code>total_trial_cost</code></dt>
<dd>decimal</dd> <dt>
<code>pipeline_active_count</code></dt><dd>decimal</dd> </dl>

Production batch records

Catalog key <code>life_sciences.supply_chain.production_batch_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>batch_id</code></dt><dd>id</dd> <dt><code>good_units</code></dt><dd>string</dd> <dt><code>total_units</code></dt><dd>decimal</dd>
<dt><code>total_cogs</code></dt>
<dd>decimal</dd> <dt>
<code>units_produced</code></dt><dd>string</dd> </dl>

Metrics (20) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)

- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Phase Success Rate
- Active Pipeline Programs
- Trial Cost per Patient
- Batch Yield
- COGS per Unit

Monthly Burn Rate

- **ID:** 1312
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** `cash_out - cash_in`
- **Catalog key:** `life_sciences.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1310
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** `dso + days_inventory_outstanding - dpo`
- **Catalog key:** `life_sciences.core.ccc`

Current Ratio

- **ID:** 1300
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.

- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `life_sciences.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1309
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs` days_**
- **Catalog key:** `life_sciences.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1308
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales` days_**
- **Catalog key:** `life_sciences.core.dso`

EBITDA Margin

- **ID:** 1304
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *_`ebitda / revenue` 100_**
- **Catalog key:** `life_sciences.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1302

- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `life_sciences.core.gross_margin`

Headcount (FTE)

- **ID:** 1314
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `life_sciences.core.headcount`

Net Profit Margin

- **ID:** 1305
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` `100`**
- **Catalog key:** `life_sciences.core.net_margin`

Operating Cash Flow

- **ID:** 1311
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**

- **Catalog key:** `life_sciences.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1303
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `life_sciences.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1301
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `life_sciences.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1306
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `life_sciences.core.revenue_growth`

Cash Runway

- **ID:** 1313
- **Type:** Financial
- **Unit:** Months

- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `life_sciences.core.runway`

Working Capital

- **ID:** 1307
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `life_sciences.core.working_capital`

Phase Success Rate

- **ID:** 1332
- **Type:** Operational
- **Unit:** %
- **Description:** Phase Success Rate
- **Inputs:** `phases_passed`, `phases_attempted`
- **Methodology:** Phase Success Rate = (Phases Passed / Phases Attempted) × 100
- **Calculation:** *`*phases_passed / phases_attempted` 100**
- **Catalog key:** `life_sciences.rnd.phase_success_rate`

Active Pipeline Programs

- **ID:** 1330
- **Type:** Operational
- **Unit:** Count
- **Description:** Active Pipeline Programs
- **Methodology:** Count of active R&D programs in the period (directly measured).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `life_sciences.rnd.pipeline_active_count`

Trial Cost per Patient

- **ID:** 1331
- **Type:** Financial

- **Unit:** Currency
- **Description:** Trial Cost per Patient
- **Inputs:** `total_trial_cost`, `patients_enrolled`
- **Methodology:** Trial Cost per Patient = Total Trial Cost / Patients Enrolled
- **Calculation:** **`total_trial_cost / patients_enrolled`**
- **Catalog key:** `life_sciences.rnd.trial_cost_per_patient`

Batch Yield

- **ID:** 1333
- **Type:** Operational
- **Unit:** %
- **Description:** Batch Yield
- **Inputs:** `good_units`, `total_units`
- **Methodology:** Batch Yield = (Good Units / Total Units) × 100
- **Calculation:** *`good_units / total_units` 100**
- **Catalog key:** `life_sciences.supply_chain.batch_yield`

COGS per Unit

- **ID:** 1334
- **Type:** Financial
- **Unit:** Currency
- **Description:** COGS per Unit
- **Inputs:** `total_cogs`, `units_produced`
- **Methodology:** COGS per Unit = Total COGS / Units Produced
- **Calculation:** **`total_cogs / units_produced`**
- **Catalog key:** `life_sciences.supply_chain.cogs_per_unit`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **De-identified clinical trial summary** — feeds Active Pipeline Programs, Trial Cost per Patient, Phase Success Rate
- **Production batch records** — feeds Batch Yield, COGS per Unit

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/manufacturing.md

Manufacturing

Pack file: `catalog/industries/manufacturing` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">2819</code>

21 metrics · 16 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd>
<code>333999</code></dd>

<dt>NACE Rev.2</dt><dd><code>28.19</code>
</dd>

<dt>GICS 2023</dt><dd>
<code>201060</code></dd>

</dl>

Taxonomy tags from ambient-core
<code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification).
Coverage notes: <a href="https://github.com/Ambient-Team/ambient-
core/blob/main/docs/catalog-industry-coverage.md">catalog-industry-coverage.md.

Data options (16) {#data-options}

Accounts payable aging

Catalog
key <code>Allmanufacturingds-ap-aging</code>

<span class="catalog-field-
label">Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>supplier_name</code></dt><dd>string</dd> <div class="catalog-
field-list__row"><dt><code>bill_id</code></dt><dd>id</dd> <div
class="catalog-field-list__row"><dt><code>bill_date</code></dt><dd>date</dd>
<dt><code>due_date</code></dt>
<dd>date</dd> <dt>
<code>amount_due</code></dt><dd>decimal</dd> <div class="catalog-field-
list__row"><dt><code>days_outstanding</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>aging_bucket</code></dt>
<dd>string</dd> <dt>
<code>accounts_payable</code></dt><dd>decimal</dd> <div class="catalog-
field-list__row"><dt><code>cogs</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>days</code></dt><dd>string</dd>
<dt><code>days_inventory_outstanding</code>
</dt><dd>string</dd> </dl>

Accounts receivable aging

Catalog key <code>Allmanufacturingds-ar-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_name</code></dt><dd>string</dd> <dt><code>invoice_id</code></dt><dd>id</dd> <dt><code>invoice_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt><dd>string</dd> <dt><code>accounts_receivable</code></dt><dd>decimal</dd> <dt><code>total_credit_sales</code></dt><dd>decimal</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Balance sheet

Catalog key <code>Allmanufacturingds-balance-sheet</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>inventory</code></dt><dd>string</dd> </dl>

Bank statements

Catalog key <code>Allmanufacturingds-bank-statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>transaction_date</code></dt><dd>date</dd> <dt><code>account_id</code></dt><dd>id</dd> <dt><code>description</code></dt><dd>string</dd> <dt><code>amount</code></dt><dd>decimal</dd> <dt><code>direction</code></dt><dd>string</dd> <dt><code>balance</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>net_income</code></dt><dd>string</dd> <dt><code>non_cash_charges</code></dt><dd>string</dd> <dt><code>increase_in_working_capital</code></dt><dd>string</dd> <dt><code>cash_out</code></dt><dd>string</dd> <dt><code>cash_in</code></dt><dd>string</dd> <dt><code>current_cash_balance</code></dt><dd>string</dd> </dl>

General ledger

Catalog key <code>Allmanufacturingds-general-ledger</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>journal_entry_id</code></dt><dd>id</dd> <dt><code>debit</code></dt><dd>string</dd> <dt><code>credit</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> <dt><code>currency</code></dt><dd>currency</dd> </dl>

Inventory records

Catalog key <code>Allmanufacturingds-inventory-records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>sku</code></dt><dd>string</dd> <dt><code>warehouse</code></dt><dd>string</dd> <dt><code>quantity_on_hand</code></dt><dd>string</dd> <dt><code>unit_cost</code></dt><dd>decimal</dd> <dt><code>reorder_point</code></dt><dd>string</dd> <dt><code>inventory_value</code></dt><dd>string</dd> </dl>

Production output logs

Catalog key <code>Allmanufacturingds-production-output</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>line_id</code></dt><dd>id</dd> <dt><code>shift</code></dt><dd>string</dd> <dt><code>units_produced</code></dt><dd>string</dd> <dt><code>units_defective</code></dt><dd>string</dd> <dt><code>planned_runtime_min</code></dt><dd>string</dd> <dt><code>actual_runtime_min</code></dt><dd>string</dd> </dl>

Trial balance

Catalog key <code>Allmanufacturingds-trial-balance</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>debit_balance</code></dt><dd>string</dd> <dt><code>credit_balance</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> </dl>

CMMS maintenance logs

Catalog key <code>MfgCmms903CatalogOptionKey03</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>asset_id</code></dt><dd>id</dd> <dt><code>downtime_minutes</code></dt><dd>string</dd> <dt><code>reason_code</code></dt><dd>id</dd> <dt><code>planned_flag</code></dt><dd>string</dd> </dl>

ERP general ledger (manufacturing)

Catalog key <code>MfgErp902CatalogOptionKey02</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>account_code</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>account_name</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>debit</code></dt><dd>string</dd>
<dt><code>credit</code></dt><dd>string</dd>
<dt><code>balance</code></dt>
<dd>string</dd> <dt>
<code>cost_center</code></dt><dd>decimal</dd> </dl>

MES production export

Catalog key <code>MfgMes901CatalogOptionKey01</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>line_id</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>work_order_id</code></dt><dd>id</dd> <div
class="catalog-field-list__row"><dt><code>units_produced</code></dt>
<dd>string</dd> <dt>
<code>units_scrapped</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>run_minutes</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>planned_minutes</code></dt>
<dd>string</dd> <dt>
<code>availability</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>performance</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>quality</code></dt><dd>string</dd>
<dt><code>scrapped_units</code></dt>
<dd>string</dd> <dt>
<code>total_units</code></dt><dd>decimal</dd> <div class="catalog-field-
list__row"><dt><code>total_manufacturing_cogs</code></dt><dd>decimal</dd>
</dl>

Quality inspection logs

Catalog key <code>MfgQual904CatalogOptionKey04</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>lot_id</code></dt><dd>id</dd> <dt><code>units_inspected</code></dt><dd>string</dd> <dt><code>units_failed</code></dt><dd>string</dd> <dt><code>defect_code</code></dt><dd>id</dd> <dt><code>scrapped_units</code></dt><dd>string</dd> <dt><code>total_units</code></dt><dd>decimal</dd> </dl>

Accounting Software

Catalog key <code>manufacturing.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>manufacturing.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>manufacturing.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>manufacturing.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <div class="catalog-field-

list__row"><dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt> <dd>decimal</dd> <dt> <code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt> <dd>decimal</dd> <dt> <code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt> <dd>decimal</dd> <dt> <code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt> <dd>decimal</dd> </dl>

Metrics (21) {#metrics}

- Capacity Utilization
- Unplanned Downtime
- Inventory Turnover
- Overall Equipment Effectiveness (OEE)
- Scrap Rate
- Unit Cost of Goods Sold
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital

Capacity Utilization

- **ID:** 1073
- **Type:** Operational
- **Unit:** %
- **Description:** Share of available production capacity actually used.
- **Inputs:** `actual_output`, `max_output`
- **Methodology:** Capacity Utilization = (Actual Output / Maximum Possible Output) × 100
- **Calculation:** *`actual_output / max_output` 100**
- **Catalog key:** `Allmanufacturingmfg-capacity-utilization`

Unplanned Downtime

- **ID:** 904
- **Type:** Operational
- **Unit:** Minutes
- **Description:** Hours or minutes of production stoppage not scheduled for maintenance.
- **Methodology:** Sum of unplanned stop events from CMMS or MES downtime codes
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `MfgDownt904CatalogMetricKey04`

Inventory Turnover

- **ID:** 903
- **Type:** Financial
- **Unit:** Turns
- **Description:** How often inventory is sold or consumed and replaced over a period.
- **Inputs:** `cogs`, `average_inventory`
- **Methodology:** Inventory Turnover = COGS / Average Inventory Value
- **Calculation:** **`cogs / average_inventory`**
- **Catalog key:** `MfgInvTurn903CatalogMetricKey03`

Overall Equipment Effectiveness (OEE)

- **ID:** 901
- **Type:** Operational
- **Unit:** %
- **Description:** Composite utilization of manufacturing equipment (availability × performance × quality).

- **Inputs:** `availability`, `performance`, `quality`
- **Methodology:** OEE = Availability × Performance × Quality (each as % of ideal)
- **Calculation:** *`availability` performance `quality`*
- **Catalog key:** `MfgOee901CatalogMetricKey01`

Scrap Rate

- **ID:** 902
- **Type:** Operational
- **Unit:** %
- **Description:** Share of production output rejected or reworked due to quality defects.
- **Inputs:** `scrapped_units`, `total_units`
- **Methodology:** Scrap Rate = (Scrapped Units / Total Units Produced) × 100
- **Calculation:** *`scrapped_units / total_units` 100**
- **Catalog key:** `MfgScrap902CatalogMetricKey02`

Unit Cost of Goods Sold

- **ID:** 905
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Fully loaded cost to produce one unit of finished goods.
- **Inputs:** `total_manufacturing_cogs`, `units_produced`
- **Methodology:** Unit COGS = Total Manufacturing COGS / Units Produced
- **Calculation:** **`total_manufacturing_cogs / units_produced`**
- **Catalog key:** `MfgUnitCogs905CatalogMetricKey05`

Monthly Burn Rate

- **ID:** 1001
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `manufacturing.core.burn_rate`

Cash Conversion Cycle

- **ID:** 989

- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `manufacturing.core.ccc`

Current Ratio

- **ID:** 929
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `manufacturing.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 983
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs` days_**
- **Catalog key:** `manufacturing.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 977
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales` days_**

- **Catalog key:** `manufacturing.core.dso`

EBITDA Margin

- **ID:** 953
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `manufacturing.core.ebitda_margin`

Gross Profit Margin

- **ID:** 941
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `manufacturing.core.gross_margin`

Headcount (FTE)

- **ID:** 1081
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `manufacturing.core.headcount`

Net Profit Margin

- **ID:** 959
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.

- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` 100**
- **Catalog key:** `manufacturing.core.net_margin`

Operating Cash Flow

- **ID:** 995
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `manufacturing.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 947
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `manufacturing.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 935
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `manufacturing.core.quick_ratio`

Revenue Growth Rate

- **ID:** 965
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `manufacturing.core.revenue_growth`

Cash Runway

- **ID:** 1007
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `manufacturing.core.runway`

Working Capital

- **ID:** 971
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `manufacturing.core.working_capital`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounts payable aging** — feeds Days Payable Outstanding (DPO), Cash Conversion Cycle
- **Accounts receivable aging** — feeds Days Sales Outstanding (DSO), Cash Conversion Cycle
- **Balance sheet** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital, Cash Conversion Cycle
- **Bank statements** — feeds Operating Cash Flow, Monthly Burn Rate, Cash Runway
- **General ledger** — feeds Gross Profit Margin, Operating Margin (EBIT), EBITDA Margin, Net Profit Margin, Revenue Growth Rate, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Operating Cash Flow
- **Inventory records** — feeds Cash Conversion Cycle, Inventory Turnover
- **Production output logs** — feeds Overall Equipment Effectiveness (OEE), Capacity Utilization
- **Trial balance** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital
- **CMMS maintenance logs** — feeds Unplanned Downtime
- **ERP general ledger (manufacturing)** — feeds Inventory Turnover, Unit Cost of Goods Sold
- **MES production export** — feeds Overall Equipment Effectiveness (OEE), Scrap Rate, Unit Cost of Goods Sold
- **Quality inspection logs** — feeds Scrap Rate
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)

Further reading: Catalog consumption · Pipeline governance helpers

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/passenger_transit.md

Passenger Transit

Pack file: `catalog/industries/passenger_transit` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">4922</code>

20 metrics · 6 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>485111</code></dd>

<dt>NACE Rev.2</dt><dd><code>49.31</code> (also <code>49.10</code>) </dd>

<dt>GICS 2023</dt><dd> <code>203040</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (6) {#data-options}

Accounting Software

Catalog key <code>passenger_transit.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>passenger_transit.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>passenger_transit.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>passenger_transit.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Public transit operations summary

Catalog key <code>passenger_transit.public_transit.operations_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>operating_revenue</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>string</dd> <dt><code>unlinked_trips</code></dt><dd>string</dd> <dt><code>vehicle_revenue_hours</code></dt><dd>decimal</dd> <dt><code>on_time_trips</code></dt><dd>string</dd> <dt><code>total_trips</code></dt><dd>decimal</dd> </dl>

Rail passenger operations summary

Catalog key <code>passenger_transit.rail_passenger.operations_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>total_boardings</code></dt><dd>decimal</dd> <dt><code>train_miles_operated</code></dt><dd>decimal</dd> <dt><code>fare_revenue</code></dt><dd>decimal</dd> <dt><code>operating_cost</code></dt><dd>decimal</dd> </dl>

Metrics (20) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle

- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Transit Farebox Recovery Ratio
- On-Time Performance
- Ridership per Vehicle Hour
- Rail Farebox Recovery Ratio
- Ridership per Train-Mile

Monthly Burn Rate

- **ID:** 2032
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `passenger_transit.core.burn_rate`

Cash Conversion Cycle

- **ID:** 2030
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO

- **Calculation:** `dso + days_inventory_outstanding - dpo`
- **Catalog key:** `passenger_transit.core.ccc`

Current Ratio

- **ID:** 2020
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** `current_assets / current_liabilities`
- **Catalog key:** `passenger_transit.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 2029
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `passenger_transit.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 2028
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `passenger_transit.core.dso`

EBITDA Margin

- **ID:** 2024
- **Type:** Financial
- **Unit:** %

- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue 100`**
- **Catalog key:** `passenger_transit.core.ebitda_margin`

Gross Profit Margin

- **ID:** 2022
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue 100`**
- **Catalog key:** `passenger_transit.core.gross_margin`

Headcount (FTE)

- **ID:** 2034
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `passenger_transit.core.headcount`

Net Profit Margin

- **ID:** 2025
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `passenger_transit.core.net_margin`

Operating Cash Flow

- **ID:** 2031
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `passenger_transit.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 2023
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `passenger_transit.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 2021
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `passenger_transit.core.quick_ratio`

Revenue Growth Rate

- **ID:** 2026
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`

- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `passenger_transit.core.revenue_growth`

Cash Runway

- **ID:** 2033
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `passenger_transit.core.runway`

Working Capital

- **ID:** 2027
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `passenger_transit.core.working_capital`

Transit Farebox Recovery Ratio

- **ID:** 1868
- **Type:** Financial
- **Unit:** %
- **Description:** Fare and subsidy-eligible revenue as a share of operating expenses (aggregated).
- **Inputs:** `operating_revenue`, `operating_expenses`
- **Methodology:** Farebox Recovery = (Operating Revenue / Operating Expenses) × 100
- **Calculation:** **`operating_revenue / operating_expenses` 100**
- **Catalog key:** `passenger_transit.public_transit.farebox_recovery_ratio`

On-Time Performance

- **ID:** 1870

- **Type:** Operational
- **Unit:** %
- **Description:** Share of transit trips departing or arriving within agency on-time standard (aggregated).
- **Inputs:** `on_time_trips`, `total_trips`
- **Methodology:** OTP = (On-Time Trips / Total Trips) × 100
- **Calculation:** *`on_time_trips / total_trips` 100**
- **Catalog key:** `passenger_transit.public_transit.on_time_performance`

Ridership per Vehicle Hour

- **ID:** 1869
- **Type:** Operational
- **Unit:** Passengers per vehicle hour
- **Description:** Unlinked passenger trips per vehicle revenue hour (aggregated).
- **Inputs:** `unlinked_trips`, `vehicle_revenue_hours`
- **Methodology:** Ridership per Vehicle Hour = Unlinked Trips / Vehicle Revenue Hours
- **Calculation:** **`unlinked_trips / vehicle_revenue_hours`**
- **Catalog key:** `passenger_transit.public_transit.ridership_per_vehicle_hour`

Rail Farebox Recovery Ratio

- **ID:** 1864
- **Type:** Financial
- **Unit:** %
- **Description:** Passenger fare revenue as a share of fully allocated operating cost (aggregated).
- **Inputs:** `fare_revenue`, `operating_cost`
- **Methodology:** Farebox Recovery = (Fare Revenue / Operating Cost) × 100
- **Calculation:** *`fare_revenue / operating_cost` 100**
- **Catalog key:** `passenger_transit.rail_passenger.farebox_recovery`

Ridership per Train-Mile

- **ID:** 1863
- **Type:** Operational
- **Unit:** Passengers per train-mile
- **Description:** Passenger boardings per train-mile operated (aggregated).
- **Inputs:** `total_boardings`, `train_miles_operated`
- **Methodology:** Ridership per Train-Mile = Total Boardings / Train-Miles Operated
- **Calculation:** **`total_boardings / train_miles_operated`**

- **Catalog key:** `passenger_transit.rail_passenger.ridership_per_train_mile`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Public transit operations summary** — feeds Transit Farebox Recovery Ratio, Ridership per Vehicle Hour, On-Time Performance
- **Rail passenger operations summary** — feeds Ridership per Train-Mile, Rail Farebox Recovery Ratio

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/real_estate.md

Real Estate

Pack file: `catalog/industries/real_estate` in ambient-core.

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6810</code>

38 metrics · 23 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>531120</code> (also <code>531390</code>)</dd>

<dt>NACE Rev.2</dt><dd><code>68.10</code> (also <code>68.20</code>) </dd>

<dt>GICS 2023</dt><dd> <code>601020</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (23) {#data-options}

Rental income records

Catalog key <code>2j9pysT31hgn2DQcejtz</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>property_id</code></dt><dd>id</dd> <dt><code>rent_collection_date</code></dt><dd>date</dd> <dt><code>rent_amount</code></dt>

<dd>decimal</dd> <dt>
<code>tenant_name</code></dt><dd>string</dd> <dt><code>payment_method</code></dt><dd>enum</dd> <dt><code>payment_status</code></dt>
<dd>enum</dd> <dt>
<code>rental_income</code></dt><dd>string</dd> <dt><code>operating_expenses</code></dt><dd>string</dd> <dt><code>debt_service</code></dt>
<dd>string</dd> <dt>
<code>property_price</code></dt><dd>decimal</dd> <dt><code>gross_rental_income</code></dt><dd>string</dd> </dl>

Accounts payable aging

Catalog key <code>Allrealestateds-ap-aging</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>supplier_name</code></dt><dd>string</dd> <dt><code>bill_id</code></dt><dd>id</dd> <dt><code>bill_date</code></dt><dd>date</dd>
<dt><code>due_date</code></dt>
<dd>date</dd> <dt>
<code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt>
<dd>string</dd> <dt>
<code>accounts_payable</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>string</dd> <dt><code>days</code></dt><dd>string</dd>
<dt><code>days_inventory_outstanding</code>
</dt><dd>string</dd> </dl>

Accounts receivable aging

Catalog key <code>Allrealestateds-ar-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_name</code></dt><dd>string</dd> <dt><code>invoice_id</code></dt><dd>id</dd> <dt><code>invoice_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt><dd>string</dd> <dt><code>accounts_receivable</code></dt><dd>decimal</dd> <dt><code>total_credit_sales</code></dt><dd>decimal</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Balance sheet

Catalog key <code>Allrealestateds-balance-sheet</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>inventory</code></dt><dd>string</dd> </dl>

Bank statements

Catalog key <code>Allrealestateds-bank-statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>transaction_date</code></dt><dd>date</dd> <div class="catalog-
field-list__row"><dt><code>account_id</code></dt><dd>id</dd> <div
class="catalog-field-list__row"><dt><code>description</code></dt><dd>string</dd>
<dt><code>amount</code></dt>
<dd>decimal</dd> <dt>
<code>direction</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>balance</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>currency</code></dt><dd>currency</dd>
<dt><code>net_income</code></dt>
<dd>string</dd> <dt>
<code>non_cash_charges</code></dt><dd>string</dd> <div class="catalog-
field-list__row"><dt><code>increase_in_working_capital</code></dt><dd>string</dd>
<dt><code>cash_out</code></dt>
<dd>string</dd> <dt>
<code>cash_in</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>current_cash_balance</code></dt><dd>string</dd>
</dl>

CRM / Sales pipeline

Catalog key <code>Allrealestateds-crm-pipeline</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>opportunity_id</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>account_name</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>stage</code></dt><dd>enum</dd>
<dt><code>amount</code></dt>
<dd>decimal</dd> <dt>
<code>close_date</code></dt><dd>date</dd> <div class="catalog-field-
list__row"><dt><code>owner</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>probability</code></dt><dd>string</dd>
<dt><code>active_subscriptions</code>
</dt><dd>string</dd> <dt>
<code>avg_monthly_price</code></dt><dd>decimal</dd> <div class="catalog-
field-list__row"><dt><code>sales_marketing_spend</code></dt><dd>string</dd>
<dt><code>new_customers</code></dt>
<dd>string</dd> </dl>

General ledger

Catalog key <code>Allrealestateds-general-ledger</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>journal_entry_id</code></dt><dd>id</dd> <dt><code>debit</code></dt><dd>string</dd> <dt><code>credit</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> <dt><code>currency</code></dt><dd>currency</dd> </dl>

Marketing spend records

Catalog key <code>Allrealestateds-marketing-spend</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>channel</code></dt><dd>enum</dd> <dt><code>campaign</code></dt><dd>string</dd> <dt><code>spend</code></dt><dd>string</dd> <dt><code>leads</code></dt><dd>string</dd> <dt><code>new_customers</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>sales_marketing_spend</code></dt><dd>string</dd> </dl>

Product usage logs

Catalog key <code>Allrealestateds-product-usage</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_id</code></dt><dd>id</dd> <dt><code>active_users</code></dt><dd>string</dd> <dt><code>sessions</code></dt><dd>string</dd> <dt><code>feature_adoption</code></dt><dd>string</dd> <dt><code>usage_minutes</code></dt><dd>string</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt><dd>string</dd> <dt><code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> </dl>

Subscription billing records

Catalog key <code>Allrealestateds-subscription-billing</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_id</code></dt><dd>id</dd> <dt><code>plan_name</code></dt><dd>string</dd> <dt><code>mrr</code></dt><dd>string</dd> <dt><code>billing_period</code></dt><dd>enum</dd> <dt><code>status</code></dt><dd>enum</dd> <dt><code>start_date</code></dt><dd>date</dd> <dt><code>end_date</code></dt><dd>date</dd> <dt><code>active_subscriptions</code></dt><dd>string</dd> <dt><code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <div

class="catalog-field-list__row"><dt><code>expansion_mrr</code></dt>
<dd>string</dd> <dt>
<code>contraction_mrr</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>churned_mrr</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>avg_revenue_per_account</code></dt>
<dd>decimal</dd> <dt>
<code>churn_rate</code></dt><dd>decimal</dd> </dl>

Trial balance

Catalog
key <code>Allrealestateds-trial-balance</code>

<span class="catalog-field-
label">Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>account_code</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>account_name</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>debit_balance</code></dt>
<dd>string</dd> <dt>
<code>credit_balance</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>period</code></dt><dd>enum</dd> </dl>

Energy bills

Catalog
key <code>Grtv4vjRFWRmrDSFF8cS</code>

<span class="catalog-field-
label">Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>property_id</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>bill_date</code></dt><dd>date</dd> <div
class="catalog-field-list__row"><dt><code>energy_consumed</code></dt>
<dd>string</dd> <dt>
<code>energy_cost</code></dt><dd>decimal</dd> <div class="catalog-field-
list__row"><dt><code>total_energy_cost</code></dt><dd>decimal</dd> <div

class="catalog-field-list__row"><dt><code>rentable_square_feet</code></dt>
<dd>string</dd> <dt>
<code>gross_rental_income</code></dt><dd>string</dd> <dt><code>operating_expenses</code></dt><dd>string</dd>
<dt><code>property_value</code></dt>
<dd>string</dd> <dt>
<code>provider_name</code></dt><dd>string</dd> </dl>

Mortgage documents

Catalog key <code>TPRNcAtqtiY387QTkah7</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>property_id</code></dt><dd>id</dd> <dt><code>mortgage_lender</code></dt><dd>string</dd> <dt><code>loan_amount</code></dt>
<dd>decimal</dd> <dt>
<code>interest_rate</code></dt><dd>decimal</dd> <dt><code>loan_term</code></dt><dd>string</dd> <dt><code>monthly_payment</code></dt>
<dd>string</dd> <dt>
<code>start_date</code></dt><dd>date</dd> <dt><code>end_date</code></dt><dd>date</dd> </dl>

Property purchase records

Catalog key <code>UVuuRwg8ksAOiP3WlKKr</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>property_id</code></dt><dd>id</dd> <dt><code>purchase_date</code></dt><dd>date</dd> <div

class="catalog-field-list__row"><dt><code>purchase_price</code></dt>
<dd>decimal</dd> <dt>
<code>seller_name</code></dt><dd>string</dd> <dt><code>property_location</code></dt><dd>string</dd> <dt><code>property_type</code></dt>
<dd>enum</dd> <dt>
<code>mortgage_details</code></dt><dd>string</dd> </dl>

Sale records

Catalog key <code>bs0gsbE2YDGDHyqRtkx8</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>property_id</code></dt><dd>id</dd> <dt><code>sale_date</code></dt><dd>date</dd> <dt><code>sale_price</code></dt><dd>decimal</dd>
<dt><code>buyer_name</code></dt>
<dd>string</dd> <dt>
<code>property_location</code></dt><dd>string</dd> <dt><code>property_type</code></dt><dd>enum</dd> <dt><code>agent_details</code></dt>
<dd>string</dd> </dl>

Property management systems

Catalog key <code>gKEmKLl2XRjSbTQGztuy</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>property_id</code></dt><dd>id</dd> <dt><code>tenant_name</code></dt><dd>string</dd> <dt><code>lease_start_date</code></dt>

<dd>date</dd> <dt>
<code>lease_end_date</code></dt><dd>date</dd> <dt><code>monthly_rent</code></dt><dd>string</dd> <dt><code>maintenance_records</code></dt>
<dd>string</dd> <dt>
<code>payment_status</code></dt><dd>enum</dd> <dt><code>same_store_noi_growth</code></dt><dd>string</dd>
<dt><code>operating_expenses</code></dt>
<dd>string</dd> <dt>
<code>effective_gross_income</code></dt><dd>string</dd> <dt><code>rentable_square_feet</code></dt>
<dd>string</dd> <dt>
<code>gross_rental_income</code></dt><dd>string</dd> <dt><code>vacant_units</code></dt><dd>string</dd> <dt><code>total_units</code></dt><dd>decimal</dd>
<dt><code>total_energy_cost</code></dt>
<dd>decimal</dd> </dl>

Property valuation reports

Catalog key <code>jYYkOyU5EQfa40iMf9x8</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>property_id</code></dt><dd>id</dd> <dt><code>valuation_date</code></dt><dd>date</dd> <dt><code>valuation_amount</code></dt>
<dd>decimal</dd> <dt>
<code>valuer_name</code></dt><dd>string</dd> <dt><code>valuation_method</code></dt><dd>enum</dd> <dt><code>property_location</code></dt>
<dd>string</dd> </dl>

Loan documents

Catalog key <code>ohvSqlJVvLde3k1gqYZh</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>property_id</code></dt><dd>id</dd> <dt><code>lender_name</code></dt><dd>string</dd> <dt><code>loan_amount</code></dt><dd>decimal</dd> <dt><code>interest_rate</code></dt><dd>decimal</dd> <dt><code>loan_term_months</code></dt><dd>decimal</dd> <dt><code>maturity_date</code></dt><dd>date</dd> <dt><code>outstanding_balance</code></dt><dd>decimal</dd> <dt><code>annual_debt_service</code></dt><dd>string</dd> </dl>

Accounting Software

Catalog key <code>real_estate.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>real_estate.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>real_estate.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>real_estate.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt>

<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

REIT supplemental reporting

Catalog key <code>real_estate.reit.reit_supplemental</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>trust_vehicle_id</code></dt><dd>id</dd> <dt><code>vehicle_type</code></dt><dd>enum</dd> <dt><code>funds_from_operations</code></dt><dd>decimal</dd> <dt><code>ffo</code></dt><dd>string</dd> <dt><code>distributions</code></dt><dd>string</dd> <dt><code>current_noi</code></dt><dd>string</dd> <dt><code>prior_noi</code></dt><dd>string</dd> <dt><code>trust_assets_under_administration</code></dt><dd>decimal</dd> <dt><code>trustee_fees</code></dt><dd>string</dd> <dt><code>average_trust_aua</code></dt><dd>string</dd> </dl>

Metrics (38) {#metrics}

- Debt Service Coverage Ratio (DSCR)
- Annual Recurring Revenue (ARR)
- Customer Acquisition Cost (CAC)
- Gross Revenue Churn Rate
- Customer Lifetime Value (LTV)
- LTV : CAC Ratio
- Monthly Recurring Revenue (MRR)
- Net Revenue Retention (NRR)
- Rule of 40
- Vacancy Rate
- Loan-to-Value Ratio (LTV)
- Gross Rent Multiplier (GRM)
- Net Operating Income (NOI)
- Return on Investment (ROI)
- Cash Flow
- Capitalization Rate (Cap Rate)
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Energy Cost per Square Foot
- Operating Expense Ratio

- Same-Store NOI Growth
- FFO Payout Ratio
- Funds From Operations
- REIT Same-Store NOI Growth
- Cash-on-Cash Return

Debt Service Coverage Ratio (DSCR)

- **ID:** 8
- **Type:** Financial
- **Unit:** Number
- **Description:** Ratio of net operating income to annual debt payments.
- **Inputs:** `annual_debt_service`
- **Methodology:** DSCR = Net Operating Income / Annual Debt Service
- **Calculation:** **`noi / annual_debt_service`**
- **Catalog key:** `1cvcTcHNt1VYX867UUG7`

Annual Recurring Revenue (ARR)

- **ID:** 1016
- **Type:** Financial
- **Unit:** Currency/year
- **Description:** Annualized run-rate of recurring subscription revenue.
- **Methodology:** ARR = MRR × 12
- **Calculation:** *`mrr` 12**
- **Catalog key:** `Allrealestatesaas-arr`

Customer Acquisition Cost (CAC)

- **ID:** 1034
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Fully-loaded sales & marketing cost to acquire one new customer.
- **Inputs:** `sales_marketing_spend`, `new_customers`
- **Methodology:** CAC = Total Sales & Marketing Spend / New Customers Acquired
- **Calculation:** **`sales_marketing_spend / new_customers`**
- **Catalog key:** `Allrealestatesaas-cac`

Gross Revenue Churn Rate

- **ID:** 1028

- **Type:** Operational
- **Unit:** %
- **Description:** Percentage of recurring revenue lost to cancellations and downgrades.
- **Inputs:** `churned_mrr`, `starting_mrr`
- **Methodology:** Gross Churn = (Churned MRR / Starting MRR) × 100
- **Calculation:** *`churned_mrr / starting_mrr` `100`**
- **Catalog key:** `Allrealestatesaas-gross-churn`

Customer Lifetime Value (LTV)

- **ID:** 1040
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Expected gross-margin revenue from a customer over their lifetime.
- **Inputs:** `avg_revenue_per_account`, `gross_margin_pct`, `churn_rate`
- **Methodology:** LTV = (Average Revenue per Account × Gross Margin %) / Churn Rate
- **Calculation:** *`(avg_revenue_per_account` `gross_margin_pct / 100) / churn_rate`**
- **Catalog key:** `Allrealestatesaas-ltv`

LTV : CAC Ratio

- **ID:** 1046
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Ratio of customer lifetime value to acquisition cost — core SaaS unit economics.
- **Methodology:** LTV:CAC = Customer Lifetime Value / Customer Acquisition Cost
- **Calculation:** **`ltv / cac`**
- **Catalog key:** `Allrealestatesaas-ltv-cac`

Monthly Recurring Revenue (MRR)

- **ID:** 1010
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Normalized recurring subscription revenue billed in a month.
- **Inputs:** `active_subscriptions`, `avg_monthly_price`
- **Methodology:** MRR = Σ (Active Subscriptions × Monthly Subscription Price)
- **Calculation:** *`active_subscriptions` `avg_monthly_price`**
- **Catalog key:** `Allrealestatesaas-mrr`

Net Revenue Retention (NRR)

- **ID:** 1022
- **Type:** Financial
- **Unit:** %
- **Description:** Revenue retained and expanded from existing customers, net of churn and contraction.
- **Inputs:** `starting_mrr`, `expansion_mrr`, `contraction_mrr`, `churned_mrr`
- **Methodology:** NRR = ((Starting MRR + Expansion − Contraction − Churn) / Starting MRR) × 100
- **Calculation:** *`(starting_mrr + expansion_mrr - contraction_mrr - churned_mrr) / starting_mrr` `100`**
- **Catalog key:** `Allrealestatesaas-nrr`

Rule of 40

- **ID:** 1052
- **Type:** Financial
- **Unit:** %
- **Description:** Balance of growth and profitability for recurring-revenue businesses.
- **Inputs:** `revenue_growth_pct`, `profit_margin_pct`
- **Methodology:** Rule of 40 = Revenue Growth % + Profit Margin %
- **Calculation:** **`revenue_growth_pct + profit_margin_pct`**
- **Catalog key:** `Allrealestatesaas-rule-of-40`

Vacancy Rate

- **ID:** 9
- **Type:** Operational
- **Unit:** %
- **Description:** Percentage of rental units that are unoccupied.
- **Inputs:** `vacant_units`, `total_units`
- **Methodology:** Vacancy Rate = (Number of Vacant Units / Total Units) * 100
- **Calculation:** *`vacant_units / total_units` `100`**
- **Catalog key:** `EDCn8khPJg44trzt4JTA`

Loan-to-Value Ratio (LTV)

- **ID:** 7
- **Type:** Financial
- **Unit:** %
- **Description:** Ratio of mortgage loan amount to the property's appraised value.

- **Inputs:** `mortgage_amount`, `appraised_value`
- **Methodology:** LTV = (Mortgage Amount / Property Appraised Value) * 100
- **Calculation:** *`mortgage_amount / appraised_value` 100**
- **Catalog key:** `Ob4nc825yNcKAA91kCua`

Gross Rent Multiplier (GRM)

- **ID:** 6
- **Type:** Financial
- **Unit:** Number
- **Description:** Property price divided by gross rental income, used for quick valuation.
- **Inputs:** `property_price`, `gross_rental_income`
- **Methodology:** GRM = Property Price / Gross Rental Income
- **Calculation:** **`property_price / gross_rental_income`**
- **Catalog key:** `VLDD4ijNDixuJ7EDJtw3`

Net Operating Income (NOI)

- **ID:** 2
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Income generated from a property after deducting operating expenses.
- **Inputs:** `gross_rental_income`, `operating_expenses`
- **Methodology:** NOI = Gross Rental Income - Operating Expenses
- **Calculation:** **`gross_rental_income - operating_expenses`**
- **Catalog key:** `YsoV0l2DLsptgTldBESm`

Return on Investment (ROI)

- **ID:** 1
- **Type:** Financial
- **Unit:** Percentage
- **Description:** Percentage return on investment relative to its cost.
- **Inputs:** `net_profit`, `cost_of_investment`
- **Methodology:** ROI = ((Net Profit / Cost of Investment) * 100)
- **Calculation:** *`net_profit / cost_of_investment` 100**
- **Catalog key:** `o4HkbNgvPm6RAPEcAzHV`

Cash Flow

- **ID:** 4
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Net income from a property after mortgage and operating expenses.
- **Inputs:** `rental_income`, `operating_expenses`, `debt_service`
- **Methodology:** Cash Flow = Rental Income - Operating Expenses - Debt Service
- **Calculation:** **`rental_income - operating_expenses - debt_service`**
- **Catalog key:** `o5PSXR0Vw3Jy35QyRrJ8`

Capitalization Rate (Cap Rate)

- **ID:** 3
- **Type:** Financial
- **Unit:** Percentage
- **Description:** Ratio of NOI to property value, used to estimate potential return.
- **Inputs:** `property_value`
- **Methodology:** Cap Rate = (NOI / Property Value) * 100
- **Calculation:** *`noi / property_value` 100**
- **Catalog key:** `rbIvVbYWRuUUr4rzZGRH`

Monthly Burn Rate

- **ID:** 998
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `real_estate.core.burn_rate`

Cash Conversion Cycle

- **ID:** 986
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO

- **Calculation:** `dso + days_inventory_outstanding - dpo`
- **Catalog key:** `real_estate.core.ccc`

Current Ratio

- **ID:** 926
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** `current_assets / current_liabilities`
- **Catalog key:** `real_estate.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 980
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `real_estate.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 974
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `real_estate.core.dso`

EBITDA Margin

- **ID:** 950
- **Type:** Financial
- **Unit:** %

- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue 100`**
- **Catalog key:** `real_estate.core.ebitda_margin`

Gross Profit Margin

- **ID:** 938
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue 100`**
- **Catalog key:** `real_estate.core.gross_margin`

Headcount (FTE)

- **ID:** 1082
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `real_estate.core.headcount`

Net Profit Margin

- **ID:** 956
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `real_estate.core.net_margin`

Operating Cash Flow

- **ID:** 992
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `real_estate.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 944
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `real_estate.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 932
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `real_estate.core.quick_ratio`

Revenue Growth Rate

- **ID:** 962
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`

- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `real_estate.core.revenue_growth`

Cash Runway

- **ID:** 1004
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `real_estate.core.runway`

Working Capital

- **ID:** 968
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `real_estate.core.working_capital`

Energy Cost per Square Foot

- **ID:** 1060
- **Type:** Operational
- **Unit:** Currency
- **Description:** Energy spend per rentable square foot (aggregated portfolio).
- **Inputs:** `total_energy_cost`, `rentable_square_feet`
- **Methodology:** Energy Cost per SF = Total Energy Cost / Rentable Square Feet
- **Calculation:** **`total_energy_cost / rentable_square_feet`**
- **Catalog key:** `real_estate.operations.energy_cost_per_sf`

Operating Expense Ratio

- **ID:** 1059
- **Type:** Financial

- **Unit:** %
- **Description:** Operating expenses relative to effective gross income (property operations).
- **Inputs:** `operating_expenses`, `effective_gross_income`
- **Methodology:** OER = (Operating Expenses / Effective Gross Income) × 100
- **Calculation:** *`operating_expenses / effective_gross_income` 100**
- **Catalog key:** `real_estate.operations.operating_expense_ratio`

Same-Store NOI Growth

- **ID:** 1058
- **Type:** Financial
- **Unit:** %
- **Description:** Year-over-year NOI growth on stabilized comparable properties (property-operator view).
- **Methodology:** Same-Store NOI Growth = ((Current NOI − Prior NOI) / Prior NOI) × 100 on comparable set
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `real_estate.operations.same_store_noi_growth`

FFO Payout Ratio

- **ID:** 1941
- **Type:** Financial
- **Unit:** %
- **Description:** Distributions paid relative to FFO.
- **Inputs:** `distributions`, `ffo`
- **Methodology:** FFO Payout = (Distributions / FFO) × 100
- **Calculation:** *`distributions / ffo` 100**
- **Catalog key:** `real_estate.reit.ffo_payout_ratio`

Funds From Operations

- **ID:** 1940
- **Type:** Financial
- **Unit:** Currency
- **Description:** REIT FFO for the period (vehicle reporting, aggregated).
- **Methodology:** FFO per NAREIT or trust reporting definition (directly reported aggregate).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `real_estate.reit.funds_from_operations`

REIT Same-Store NOI Growth

- **ID:** 1942
- **Type:** Financial
- **Unit:** %
- **Description:** Year-over-year NOI growth on a same-store basis at the trust vehicle level.
- **Inputs:** `current_noi`, `prior_noi`
- **Methodology:** Same-Store NOI Growth = ((Current NOI − Prior NOI) / Prior NOI) × 100
- **Calculation:** *`(current_noi - prior_noi) / prior_noi` 100**
- **Catalog key:** `real_estate.reit.same_store_noi_growth`

Cash-on-Cash Return

- **ID:** 5
- **Type:** Financial
- **Unit:** Percentage
- **Description:** Annual return on invested cash.
- **Inputs:** `net_cash_flow`, `initial_cash_investment`
- **Methodology:** Cash-on-Cash Return = (Net Cash Flow / Initial Cash Investment) * 100
- **Calculation:** *`net_cash_flow / initial_cash_investment` 100**
- **Catalog key:** `wCpkOtGniudQu8pSh22k`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Rental income records** — feeds Cash Flow, Gross Rent Multiplier (GRM)

- **Accounts payable aging** — feeds Days Payable Outstanding (DPO), Cash Conversion Cycle
- **Accounts receivable aging** — feeds Days Sales Outstanding (DSO), Cash Conversion Cycle
- **Balance sheet** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital, Cash Conversion Cycle
- **Bank statements** — feeds Operating Cash Flow, Monthly Burn Rate, Cash Runway
- **CRM / Sales pipeline** — feeds Monthly Recurring Revenue (MRR), Customer Acquisition Cost (CAC)
- **General ledger** — feeds Gross Profit Margin, Operating Margin (EBIT), EBITDA Margin, Net Profit Margin, Revenue Growth Rate, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Operating Cash Flow, Customer Acquisition Cost (CAC)
- **Marketing spend records** — feeds Customer Acquisition Cost (CAC), LTV : CAC Ratio
- **Product usage logs** — feeds Net Revenue Retention (NRR), Gross Revenue Churn Rate
- **Subscription billing records** — feeds Monthly Recurring Revenue (MRR), Annual Recurring Revenue (ARR), Net Revenue Retention (NRR), Gross Revenue Churn Rate, Customer Lifetime Value (LTV), LTV : CAC Ratio, Rule of 40
- **Trial balance** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital
- **Energy bills** — feeds Net Operating Income (NOI), Capitalization Rate (Cap Rate), Energy Cost per Square Foot
- **Mortgage documents** — feeds Loan-to-Value Ratio (LTV)
- **Property purchase records** — feeds Return on Investment (ROI)
- **Sale records** — feeds Return on Investment (ROI)
- **Property management systems** — feeds Net Operating Income (NOI), Vacancy Rate, Same-Store NOI Growth, Operating Expense Ratio, Energy Cost per Square Foot
- **Property valuation reports** — feeds Capitalization Rate (Cap Rate), Gross Rent Multiplier (GRM)
- **Loan documents** — feeds Debt Service Coverage Ratio (DSCR)
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway, Return on Investment (ROI), Net Operating Income (NOI), Capitalization Rate (Cap Rate), Cash Flow, Cash-on-Cash Return, Debt Service Coverage Ratio (DSCR)
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)

- **REIT supplemental reporting** — feeds Funds From Operations, FFO Payout Ratio, REIT Same-Store NOI Growth

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/restaurants.md

Restaurants

Pack file: `catalog/industries/restaurants` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">5610</code>

28 metrics · 17 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>722511</code></dd>

<dt>NACE Rev.2</dt><dd><code>56.10</code> </dd>

<dt>GICS 2023</dt><dd> <code>253010</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (17) {#data-options}

Accounts payable aging

Catalog key <code>Allrestaurantsds-ap-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>supplier_name</code></dt><dd>string</dd> <dt><code>bill_id</code></dt><dd>id</dd> <dt><code>bill_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt><dd>string</dd> <dt><code>accounts_payable</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>string</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Accounts receivable aging

Catalog key <code>Allrestaurantsds-ar-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_name</code></dt><dd>string</dd> <dt><code>invoice_id</code></dt><dd>id</dd> <dt><code>invoice_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <div

class="catalog-field-list__row"><dt><code>aging_bucket</code></dt>
<dd>string</dd> <dt>
<code>accounts_receivable</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>total_credit_sales</code></dt>
<dd>decimal</dd> <dt>
<code>days</code></dt><dd>string</dd>
<dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Balance sheet

Catalog key <code>Allrestaurantsds-balance-sheet</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>inventory</code></dt><dd>string</dd> </dl>

Bank statements

Catalog key <code>Allrestaurantsds-bank-statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>transaction_date</code></dt><dd>date</dd> <dt><code>account_id</code></dt><dd>id</dd> <div
class="catalog-field-list__row"><dt><code>description</code></dt><dd>string</dd>
<dt><code>amount</code></dt>
<dd>decimal</dd> <dt>
<code>direction</code></dt><dd>string</dd> <dt><code>balance</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>currency</code></dt><dd>currency</dd>
<dt><code>net_income</code></dt>
<dd>string</dd> <dt>

<code>non_cash_charges</code></dt><dd>string</dd> <dt><code>increase_in_working_capital</code></dt><dd>string</dd> <dt><code>cash_out</code></dt><dd>string</dd> <dt><code>cash_in</code></dt><dd>string</dd> <dt><code>current_cash_balance</code></dt><dd>string</dd> </dl>

CRM / Sales pipeline

Catalog key <code>Allrestaurantsds-crm-pipeline</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>opportunity_id</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>stage</code></dt><dd>enum</dd> <dt><code>amount</code></dt><dd>decimal</dd> <dt><code>close_date</code></dt><dd>date</dd> <dt><code>owner</code></dt><dd>string</dd> <dt><code>probability</code></dt><dd>string</dd> <dt><code>active_subscriptions</code></dt><dd>string</dd> <dt><code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>sales_marketing_spend</code></dt><dd>string</dd> <dt><code>new_customers</code></dt><dd>string</dd> </dl>

General ledger

Catalog key <code>Allrestaurantsds-general-ledger</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>account_code</code></dt><dd>id</dd> <div class="catalog-field-
list__row"><dt><code>account_name</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>journal_entry_id</code></dt><dd>id</dd>
<dt><code>debit</code></dt>
<dd>string</dd> <dt><code>credit</code>
</dt><dd>string</dd> <dt>
<code>period</code></dt><dd>enum</dd> <div class="catalog-field-
list__row"><dt><code>currency</code></dt><dd>currency</dd> </dl>

Marketing spend records

Catalog key <code>Allrestaurantsds-marketing-spend</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>channel</code></dt><dd>enum</dd> <div class="catalog-field-
list__row"><dt><code>campaign</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>spend</code></dt><dd>string</dd>
<dt><code>leads</code></dt>
<dd>string</dd> <dt>
<code>new_customers</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>currency</code></dt><dd>currency</dd> <div
class="catalog-field-list__row"><dt><code>sales_marketing_spend</code></dt>
<dd>string</dd> </dl>

Product usage logs

Catalog key <code>Allrestaurantsds-product-usage</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>

<dt><code>customer_id</code></dt><dd>id</dd> <dt><code>active_users</code></dt><dd>string</dd> <dt><code>sessions</code></dt><dd>string</dd> <dt><code>feature_adoption</code></dt><dd>string</dd> <dt><code>usage_minutes</code></dt><dd>string</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt><dd>string</dd> <dt><code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> </dl>

Subscription billing records

Catalog key <code>Allrestaurantsds-subscription-billing</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_id</code></dt><dd>id</dd> <dt><code>plan_name</code></dt><dd>string</dd> <dt><code>mrr</code></dt><dd>string</dd> <dt><code>billing_period</code></dt><dd>enum</dd> <dt><code>status</code></dt><dd>enum</dd> <dt><code>start_date</code></dt><dd>date</dd> <dt><code>end_date</code></dt><dd>date</dd> <dt><code>active_subscriptions</code></dt><dd>string</dd> <dt><code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt><dd>string</dd> <dt><code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> <dt><code>avg_revenue_per_account</code></dt><dd>decimal</dd> <dt><code>churn_rate</code></dt><dd>decimal</dd> </dl>

Trial balance

Catalog key <code>Allrestaurantsds-trial-balance</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>debit_balance</code></dt><dd>string</dd> <dt><code>credit_balance</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> </dl>

Inventory count export

Catalog key <code>RestInv922CatalogOptionKey02</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>location_id</code></dt><dd>id</dd> <dt><code>category</code></dt><dd>string</dd> <dt><code>quantity_on_hand</code></dt><dd>string</dd> <dt><code>usage_value</code></dt><dd>string</dd> <dt><code>waste_value</code></dt><dd>string</dd> <dt><code>spoilage_value</code></dt><dd>string</dd> </dl>

Labor scheduling export

Catalog key <code>RestLabor923CatalogOptionKey03</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>location_id</code></dt><dd>id</dd> <dt><code>role</code></dt><dd>string</dd> <dt><code>scheduled_hours</code></dt><dd>decimal</dd> <dt><code>actual_hours</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> </dl>

POS daily close

Catalog key <code>RestPos921CatalogOptionKey01</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>location_id</code></dt><dd>id</dd> <dt><code>gross_sales</code></dt><dd>string</dd> <dt><code>net_sales</code></dt><dd>string</dd> <dt><code>food_sales</code></dt><dd>string</dd> <dt><code>covers</code></dt><dd>string</dd> <dt><code>avg_check_minutes</code></dt><dd>string</dd> <dt><code>labor_hours</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>food_cogs</code></dt><dd>string</dd> <dt><code>food_revenue</code></dt><dd>decimal</dd> <dt><code>total_labor</code></dt><dd>decimal</dd> <dt><code>total_revenue</code></dt><dd>decimal</dd> <dt><code>table_turnover</code></dt><dd>string</dd> <dt><code>current_sss</code></dt><dd>string</dd> <dt><code>prior_sss</code></dt><dd>string</dd> </dl>

Accounting Software

Catalog key <code>restaurants.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>restaurants.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>restaurants.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>restaurants.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Metrics (28) {#metrics}

- Annual Recurring Revenue (ARR)
- Customer Acquisition Cost (CAC)
- Gross Revenue Churn Rate
- Customer Lifetime Value (LTV)
- LTV : CAC Ratio
- Monthly Recurring Revenue (MRR)
- Net Revenue Retention (NRR)
- Rule of 40
- Food Cost Percentage
- Labor Cost Percentage
- Same-Store Sales Growth
- Table Turn Time
- Food Waste and Spoilage
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital

Annual Recurring Revenue (ARR)

- **ID:** 1021
- **Type:** Financial

- **Unit:** Currency/year
- **Description:** Annualized run-rate of recurring subscription revenue.
- **Methodology:** ARR = MRR × 12
- **Calculation:** *`mrr 12`**
- **Catalog key:** `Allrestaurantssaas-arr`

Customer Acquisition Cost (CAC)

- **ID:** 1039
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Fully-loaded sales & marketing cost to acquire one new customer.
- **Inputs:** `sales_marketing_spend`, `new_customers`
- **Methodology:** CAC = Total Sales & Marketing Spend / New Customers Acquired
- **Calculation:** **`sales_marketing_spend / new_customers`**
- **Catalog key:** `Allrestaurantssaas-cac`

Gross Revenue Churn Rate

- **ID:** 1033
- **Type:** Operational
- **Unit:** %
- **Description:** Percentage of recurring revenue lost to cancellations and downgrades.
- **Inputs:** `churned_mrr`, `starting_mrr`
- **Methodology:** Gross Churn = (Churned MRR / Starting MRR) × 100
- **Calculation:** *`churned_mrr / starting_mrr 100`**
- **Catalog key:** `Allrestaurantssaas-gross-churn`

Customer Lifetime Value (LTV)

- **ID:** 1045
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Expected gross-margin revenue from a customer over their lifetime.
- **Inputs:** `avg_revenue_per_account`, `gross_margin_pct`, `churn_rate`
- **Methodology:** LTV = (Average Revenue per Account × Gross Margin %) / Churn Rate
- **Calculation:** *`(avg_revenue_per_account gross_margin_pct / 100) / churn_rate`**
- **Catalog key:** `Allrestaurantssaas-ltv`

LTV : CAC Ratio

- **ID:** 1051
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Ratio of customer lifetime value to acquisition cost — core SaaS unit economics.
- **Methodology:** LTV:CAC = Customer Lifetime Value / Customer Acquisition Cost
- **Calculation:** `ltv / cac`
- **Catalog key:** `Allrestaurantssaas-ltv-cac`

Monthly Recurring Revenue (MRR)

- **ID:** 1015
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Normalized recurring subscription revenue billed in a month.
- **Inputs:** `active_subscriptions`, `avg_monthly_price`
- **Methodology:** MRR = Σ (Active Subscriptions × Monthly Subscription Price)
- **Calculation:** *`active_subscriptions` `avg_monthly_price`**
- **Catalog key:** `Allrestaurantssaas-mrr`

Net Revenue Retention (NRR)

- **ID:** 1027
- **Type:** Financial
- **Unit:** %
- **Description:** Revenue retained and expanded from existing customers, net of churn and contraction.
- **Inputs:** `starting_mrr`, `expansion_mrr`, `contraction_mrr`, `churned_mrr`
- **Methodology:** NRR = ((Starting MRR + Expansion − Contraction − Churn) / Starting MRR) × 100
- **Calculation:** *`(starting_mrr + expansion_mrr - contraction_mrr - churned_mrr) / starting_mrr` `100`**
- **Catalog key:** `Allrestaurantssaas-nrr`

Rule of 40

- **ID:** 1057
- **Type:** Financial
- **Unit:** %
- **Description:** Balance of growth and profitability for recurring-revenue businesses.

- **Inputs:** `revenue_growth_pct`, `profit_margin_pct`
- **Methodology:** Rule of 40 = Revenue Growth % + Profit Margin %
- **Calculation:** **`revenue_growth_pct + profit_margin_pct`**
- **Catalog key:** `Allrestaurantssaas-rule-of-40`

Food Cost Percentage

- **ID:** 921
- **Type:** Financial
- **Unit:** %
- **Description:** Cost of food sold as a percentage of restaurant revenue.
- **Inputs:** `food_cogs`, `food_revenue`
- **Methodology:** Food Cost % = Food COGS / Food Revenue × 100
- **Calculation:** *`food_cogs / food_revenue` `100`**
- **Catalog key:** `RestFood921CatalogMetricKey01`

Labor Cost Percentage

- **ID:** 922
- **Type:** Financial
- **Unit:** %
- **Description:** Total labor cost as a percentage of revenue for the period.
- **Inputs:** `total_labor`, `total_revenue`
- **Methodology:** Labor Cost % = Total Labor / Total Revenue × 100
- **Calculation:** *`total_labor / total_revenue` `100`**
- **Catalog key:** `RestLabor922CatalogMetricKey02`

Same-Store Sales Growth

- **ID:** 925
- **Type:** Financial
- **Unit:** %
- **Description:** Year-over-year revenue change for locations open in both periods.
- **Inputs:** `current_sss`, `prior_sss`
- **Methodology:** (Current Period SSS Revenue − Prior Year SSS Revenue) / Prior Year SSS Revenue × 100
- **Calculation:** *`(current_sss - prior_sss) / prior_sss` `100`**
- **Catalog key:** `RestSss925CatalogMetricKey05`

Table Turn Time

- **ID:** 923

- **Type:** Operational
- **Unit:** Minutes
- **Description:** Average minutes from seat to payment for dine-in service.
- **Methodology:** Mean elapsed time per closed check for dine-in dayparts
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `RestTurn923CatalogMetricKey03`

Food Waste and Spoilage

- **ID:** 924
- **Type:** Operational
- **Unit:** Currency (e.g., USD)
- **Description:** Value or weight of inventory written off as waste or spoilage.
- **Methodology:** Sum of waste/spoilage adjustments from inventory counts
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `RestWaste924CatalogMetricKey04`

Monthly Burn Rate

- **ID:** 1003
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `restaurants.core.burn_rate`

Cash Conversion Cycle

- **ID:** 991
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `restaurants.core.ccc`

Current Ratio

- **ID:** 931
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `restaurants.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 985
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs`_ `days`**
- **Catalog key:** `restaurants.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 979
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales`_ `days`**
- **Catalog key:** `restaurants.core.dso`

EBITDA Margin

- **ID:** 955
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`

- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue 100`**
- **Catalog key:** `restaurants.core.ebitda_margin`

Gross Profit Margin

- **ID:** 943
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue 100`**
- **Catalog key:** `restaurants.core.gross_margin`

Headcount (FTE)

- **ID:** 1083
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `restaurants.core.headcount`

Net Profit Margin

- **ID:** 961
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `restaurants.core.net_margin`

Operating Cash Flow

- **ID:** 997
- **Type:** Financial

- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `restaurants.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 949
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` `100`**
- **Catalog key:** `restaurants.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 937
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `restaurants.core.quick_ratio`

Revenue Growth Rate

- **ID:** 967
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` `100`**

- **Catalog key:** `restaurants.core.revenue_growth`

Cash Runway

- **ID:** 1009
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** `current_cash_balance / burn_rate`
- **Catalog key:** `restaurants.core.runway`

Working Capital

- **ID:** 973
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** `current_assets - current_liabilities`
- **Catalog key:** `restaurants.core.working_capital`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounts payable aging** — feeds Days Payable Outstanding (DPO), Cash Conversion Cycle

- **Accounts receivable aging** — feeds Days Sales Outstanding (DSO), Cash Conversion Cycle
- **Balance sheet** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital, Cash Conversion Cycle
- **Bank statements** — feeds Operating Cash Flow, Monthly Burn Rate, Cash Runway
- **CRM / Sales pipeline** — feeds Monthly Recurring Revenue (MRR), Customer Acquisition Cost (CAC)
- **General ledger** — feeds Gross Profit Margin, Operating Margin (EBIT), EBITDA Margin, Net Profit Margin, Revenue Growth Rate, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Operating Cash Flow, Customer Acquisition Cost (CAC)
- **Marketing spend records** — feeds Customer Acquisition Cost (CAC), LTV : CAC Ratio
- **Product usage logs** — feeds Net Revenue Retention (NRR), Gross Revenue Churn Rate
- **Subscription billing records** — feeds Monthly Recurring Revenue (MRR), Annual Recurring Revenue (ARR), Net Revenue Retention (NRR), Gross Revenue Churn Rate, Customer Lifetime Value (LTV), LTV : CAC Ratio, Rule of 40
- **Trial balance** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital
- **Inventory count export** — feeds Food Cost Percentage, Food Waste and Spoilage
- **Labor scheduling export** — feeds Labor Cost Percentage
- **POS daily close** — feeds Food Cost Percentage, Labor Cost Percentage, Table Turn Time, Same-Store Sales Growth
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE), Labor Cost Percentage
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/retail.md

Retail

Pack file: `catalog/industries/retail` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">4711</code>

20 metrics · 6 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>455219</code></dd>

<dt>NACE Rev.2</dt><dd><code>47.11</code> </dd>

<dt>GICS 2023</dt><dd> <code>255040</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (6) {#data-options}

Accounting Software

Catalog key <code>retail.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt>

<dd>decimal</dd> <dt>
<code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd>
<dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd>
<dt><code>total_operational_cost</code>
</dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>retail.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt>
<dd>date</dd> </dl>

HR records

Catalog key <code>retail.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>retail.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Inventory valuation export

Catalog key <code>retail.financial.inventory_summary</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>banner</code></dt><dd>string</dd> <dt><code>cogs</code></dt><dd>string</dd> <div class="catalog-

field-list__row"><dt><code>average_inventory_value</code></dt><dd>string</dd>
<dt><code>shrinkage_value</code></dt>
<dd>string</dd> <dt><code>sales</code>
</dt><dd>string</dd> </dl>

Store sales summary export

Catalog key <code>retail.operations.store_sales_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>store_id</code></dt><dd>id</dd> <dt><code>current_period_sales</code></dt><dd>enum</dd> <dt><code>prior_period_sales</code></dt>
<dd>enum</dd> <dt>
<code>net_sales</code></dt><dd>string</dd> <dt><code>selling_square_footage</code></dt><dd>string</dd>
<dt><code>transactions</code></dt>
<dd>string</dd> <dt>
<code>visitors</code></dt><dd>string</dd> </dl>

Metrics (20) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)

- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Inventory Turnover
- Sales per Square Foot
- Conversion Rate
- Same-Store Sales Growth
- Shrink Rate

Monthly Burn Rate

- **ID:** 1652
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `retail.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1650
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `retail.core.ccc`

Current Ratio

- **ID:** 1640
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.

- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `retail.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1649
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs` days_**
- **Catalog key:** `retail.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1648
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales` days_**
- **Catalog key:** `retail.core.dso`

EBITDA Margin

- **ID:** 1644
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *_`ebitda / revenue` 100_**
- **Catalog key:** `retail.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1642

- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `retail.core.gross_margin`

Headcount (FTE)

- **ID:** 1654
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `retail.core.headcount`

Net Profit Margin

- **ID:** 1645
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` `100`**
- **Catalog key:** `retail.core.net_margin`

Operating Cash Flow

- **ID:** 1651
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**

- **Catalog key:** `retail.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1643
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `retail.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1641
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `retail.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1646
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `retail.core.revenue_growth`

Cash Runway

- **ID:** 1653
- **Type:** Financial
- **Unit:** Months

- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `retail.core.runway`

Working Capital

- **ID:** 1647
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `retail.core.working_capital`

Inventory Turnover

- **ID:** 1671
- **Type:** Financial
- **Unit:** Turns
- **Description:** How often retail inventory is sold and replaced over a period.
- **Inputs:** `cogs`, `average_inventory_value`
- **Methodology:** Inventory Turnover = COGS / Average Inventory Value
- **Calculation:** **`cogs / average_inventory_value`**
- **Catalog key:** `retail.financial.inventory_turnover`

Sales per Square Foot

- **ID:** 1673
- **Type:** Financial
- **Unit:** Currency
- **Description:** Revenue productivity per selling square foot (aggregated by store or banner).
- **Inputs:** `net_sales`, `selling_square_footage`
- **Methodology:** Sales per Sq Ft = Net Sales / Selling Square Footage
- **Calculation:** **`net_sales / selling_square_footage`**
- **Catalog key:** `retail.financial.sales_per_square_foot`

Conversion Rate

- **ID:** 1674
- **Type:** Operational
- **Unit:** %
- **Description:** Share of store or site visitors who complete a purchase (omnichannel where available).
- **Inputs:** `transactions`, `visitors`
- **Methodology:** Conversion Rate = (Transactions / Visitors) × 100
- **Calculation:** *`transactions / visitors` 100**
- **Catalog key:** `retail.operations.conversion_rate`

Same-Store Sales Growth

- **ID:** 1670
- **Type:** Operational
- **Unit:** %
- **Description:** Year-over-year revenue growth for stores open in both periods (aggregated).
- **Inputs:** `current_period_sales`, `prior_period_sales`
- **Methodology:** SSS Growth = ((Current Period Sales − Prior Period Sales) / Prior Period Sales) × 100
- **Calculation:** *`(current_period_sales - prior_period_sales) / prior_period_sales` 100**
- **Catalog key:** `retail.operations.same_store_sales_growth`

Shrink Rate

- **ID:** 1672
- **Type:** Operational
- **Unit:** %
- **Description:** Inventory loss from theft, damage, or administrative error (aggregated).
- **Inputs:** `shrinkage_value`, `sales`
- **Methodology:** Shrink Rate = (Shrinkage Value / Sales) × 100
- **Calculation:** *`shrinkage_value / sales` 100**
- **Catalog key:** `retail.operations.shrink_rate`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Inventory valuation export** — feeds Inventory Turnover, Shrink Rate
- **Store sales summary export** — feeds Same-Store Sales Growth, Sales per Square Foot, Conversion Rate

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/software_saas.md

Software (SaaS)

Pack file: `catalog/industries/software_saas` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">5820</code>

23 metrics · 8 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>511210</code></dd>

<dt>NACE Rev.2</dt><dd><code>58.29</code></dd>

<dt>GICS 2023</dt><dd><code>451030</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (8) {#data-options}

Accounting Software

Catalog key <code>software_saas.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>software_saas.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>software_saas.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>software_saas.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <div class="catalog-

field-list__row"><dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt>
<dd>decimal</dd> <dt>
<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>
<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd>
<dt><code>total_production_cost</code></dt>
<dd>decimal</dd> <dt>
<code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt>
<dd>decimal</dd> <dt>
<code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>
<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

CRM / Sales pipeline

Catalog key <code>software_saas.subscription.crm_pipeline</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>opportunity_id</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>stage</code></dt><dd>enum</dd>
<dt><code>amount</code></dt>
<dd>decimal</dd> <dt>
<code>close_date</code></dt><dd>date</dd> <dt><code>owner</code></dt><dd>string</dd> <dt><code>probability</code></dt><dd>string</dd>
<dt><code>active_subscriptions</code>
</dt><dd>string</dd> <dt>
<code>avg_monthly_price</code></dt><dd>decimal</dd> <div class="catalog-

field-list__row"><dt><code>sales_marketing_spend</code></dt><dd>string</dd>
<dt><code>new_customers</code></dt>
<dd>string</dd> </dl>

Marketing spend records

Catalog key <code>software_saas.subscription.marketing_spend</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>channel</code></dt><dd>enum</dd> <dt><code>campaign</code></dt><dd>string</dd> <dt><code>spend</code></dt><dd>string</dd>
<dt><code>leads</code></dt>
<dd>string</dd> <dt>
<code>new_customers</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>sales_marketing_spend</code></dt>
<dd>string</dd> <dt>
<code>avg_revenue_per_account</code></dt><dd>decimal</dd> <dt><code>churn_rate</code></dt>
<dd>decimal</dd> </dl>

Product usage logs

Catalog key <code>software_saas.subscription.product_usage</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>customer_id</code></dt><dd>id</dd> <dt><code>active_users</code></dt><dd>string</dd> <dt><code>sessions</code></dt><dd>string</dd>
<dt><code>feature_adoption</code></dt>

<dd>string</dd> <dt>
<code>usage_minutes</code></dt><dd>string</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt>
<dd>string</dd> <dt>
<code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> </dl>

Subscription billing records

Catalog key <code>software_saas.subscription.subscription_billing</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>customer_id</code></dt><dd>id</dd> <dt><code>plan_name</code></dt><dd>string</dd> <dt><code>mrr</code></dt><dd>string</dd>
<dt><code>billing_period</code></dt>
<dd>enum</dd> <dt><code>status</code>
</dt><dd>enum</dd> <dt>
<code>start_date</code></dt><dd>date</dd> <dt><code>end_date</code></dt><dd>date</dd> <dt><code>active_subscriptions</code></dt>
<dd>string</dd> <dt>
<code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt>
<dd>string</dd> <dt>
<code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> </dl>

Metrics (23) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio

- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Annual Recurring Revenue (ARR)
- Customer Acquisition Cost (CAC)
- Gross Revenue Churn Rate
- Customer Lifetime Value (LTV)
- LTV : CAC Ratio
- Monthly Recurring Revenue (MRR)
- Net Revenue Retention (NRR)
- Rule of 40

Monthly Burn Rate

- **ID:** 2052
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `software_saas.core.burn_rate`

Cash Conversion Cycle

- **ID:** 2050
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.

- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `software_saas.core.ccc`

Current Ratio

- **ID:** 2040
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `software_saas.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 2049
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `software_saas.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 2048
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `software_saas.core.dso`

EBITDA Margin

- **ID:** 2044

- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue 100`**
- **Catalog key:** `software_saas.core.ebitda_margin`

Gross Profit Margin

- **ID:** 2042
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue 100`**
- **Catalog key:** `software_saas.core.gross_margin`

Headcount (FTE)

- **ID:** 2054
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `software_saas.core.headcount`

Net Profit Margin

- **ID:** 2045
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**

- **Catalog key:** `software_saas.core.net_margin`

Operating Cash Flow

- **ID:** 2051
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `software_saas.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 2043
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` `100`**
- **Catalog key:** `software_saas.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 2041
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** **`(current_assets - inventory) / current_liabilities`**
- **Catalog key:** `software_saas.core.quick_ratio`

Revenue Growth Rate

- **ID:** 2046
- **Type:** Financial
- **Unit:** %

- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `software_saas.core.revenue_growth`

Cash Runway

- **ID:** 2053
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `software_saas.core.runway`

Working Capital

- **ID:** 2047
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `software_saas.core.working_capital`

Annual Recurring Revenue (ARR)

- **ID:** 1581
- **Type:** Financial
- **Unit:** Currency/year
- **Description:** Annualized run-rate of recurring subscription revenue.
- **Methodology:** ARR = MRR × 12
- **Calculation:** *`mrr` 12**
- **Catalog key:** `software_saas.subscription.arr`

Customer Acquisition Cost (CAC)

- **ID:** 1584

- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Fully-loaded sales and marketing cost to acquire one new customer.
- **Inputs:** `sales_marketing_spend`, `new_customers`
- **Methodology:** CAC = Total Sales & Marketing Spend / New Customers Acquired
- **Calculation:** **`sales_marketing_spend / new_customers`**
- **Catalog key:** `software_saas.subscription.cac`

Gross Revenue Churn Rate

- **ID:** 1583
- **Type:** Operational
- **Unit:** %
- **Description:** Percentage of recurring revenue lost to cancellations and downgrades.
- **Inputs:** `churned_mrr`, `starting_mrr`
- **Methodology:** Gross Churn = (Churned MRR / Starting MRR) × 100
- **Calculation:** *`churned_mrr / starting_mrr` `100`**
- **Catalog key:** `software_saas.subscription.gross_churn`

Customer Lifetime Value (LTV)

- **ID:** 1585
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Expected gross-margin revenue from a customer over their lifetime.
- **Inputs:** `avg_revenue_per_account`, `gross_margin_pct`, `churn_rate`
- **Methodology:** LTV = (Average Revenue per Account × Gross Margin %) / Churn Rate
- **Calculation:** *`(avg_revenue_per_account` `gross_margin_pct / 100) / churn_rate`**
- **Catalog key:** `software_saas.subscription.ltv`

LTV : CAC Ratio

- **ID:** 1586
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Ratio of customer lifetime value to acquisition cost.
- **Methodology:** LTV:CAC = Customer Lifetime Value / Customer Acquisition Cost
- **Calculation:** **`ltv / cac`**
- **Catalog key:** `software_saas.subscription.ltv_cac`

Monthly Recurring Revenue (MRR)

- **ID:** 1580
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Normalized recurring subscription revenue billed in a month.
- **Inputs:** `active_subscriptions`, `avg_monthly_price`
- **Methodology:** MRR = Σ (Active Subscriptions × Monthly Subscription Price)
- **Calculation:** *`active_subscriptions` `avg_monthly_price`**
- **Catalog key:** `software_saas.subscription.mrr`

Net Revenue Retention (NRR)

- **ID:** 1582
- **Type:** Financial
- **Unit:** %
- **Description:** Revenue retained and expanded from existing customers, net of churn and contraction.
- **Inputs:** `starting_mrr`, `expansion_mrr`, `contraction_mrr`, `churned_mrr`
- **Methodology:** NRR = ((Starting MRR + Expansion − Contraction − Churn) / Starting MRR) × 100
- **Calculation:** *`(starting_mrr + expansion_mrr - contraction_mrr - churned_mrr) / starting_mrr` `100`**
- **Catalog key:** `software_saas.subscription.nrr`

Rule of 40

- **ID:** 1587
- **Type:** Financial
- **Unit:** %
- **Description:** Balance of growth and profitability for recurring-revenue businesses.
- **Inputs:** `revenue_growth_pct`, `profit_margin_pct`
- **Methodology:** Rule of 40 = Revenue Growth % + Profit Margin %
- **Calculation:** **`revenue_growth_pct + profit_margin_pct`**
- **Catalog key:** `software_saas.subscription.rule_of_40`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **CRM / Sales pipeline** — feeds Monthly Recurring Revenue (MRR), Customer Acquisition Cost (CAC)
- **Marketing spend records** — feeds Customer Acquisition Cost (CAC), Customer Lifetime Value (LTV)
- **Product usage logs** — feeds Net Revenue Retention (NRR), Gross Revenue Churn Rate
- **Subscription billing records** — feeds Monthly Recurring Revenue (MRR), Annual Recurring Revenue (ARR), Net Revenue Retention (NRR), Gross Revenue Churn Rate, Rule of 40

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/transportation.md

Transportation

Pack file: `catalog/industries/transportation` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">4923</code>

28 metrics · 19 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd> <code>484110</code> (also <code>484122</code>, <code>483111</code>)</dd>

<dt>NACE Rev.2</dt><dd><code>49.41</code> (also <code>50.20</code>) </dd>

<dt>GICS 2023</dt><dd> <code>203040</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (19) {#data-options}

Vehicle telematics data

Catalog key <code>5tZAB2WLZbRDQdj6l7X6</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>vehicle_id</code></dt><dd>id</dd> <dt><code>telematics_date</code></dt><dd>date</dd> <dt><code>engine_performance</code></dt> <dd>string</dd> <dt> <code>fuel_efficiency</code></dt><dd>string</dd> <dt><code>maintenance_alerts</code></dt><dd>string</dd> <div

class="catalog-field-list__row"><dt><code>driver_behavior</code></dt>
<dd>string</dd> </dl>

Accounts payable aging

Catalog key <code>Alltransportationds-ap-aging</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>supplier_name</code></dt><dd>string</dd> <dt><code>bill_id</code></dt><dd>id</dd> <dt><code>bill_date</code></dt><dd>date</dd>
<dt><code>due_date</code></dt>
<dd>date</dd> <dt>
<code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt>
<dd>string</dd> <dt>
<code>accounts_payable</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>string</dd> <dt><code>days</code></dt><dd>string</dd>
<dt><code>days_inventory_outstanding</code>
</dt><dd>string</dd> </dl>

Accounts receivable aging

Catalog key <code>Alltransportationds-ar-aging</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>customer_name</code></dt><dd>string</dd> <dt><code>invoice_id</code></dt><dd>id</dd> <dt><code>invoice_date</code></dt><dd>date</dd>

<dt><code>due_date</code></dt>
<dd>date</dd> <dt>
<code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt>
<dd>string</dd> <dt>
<code>accounts_receivable</code></dt><dd>decimal</dd> <dt><code>total_credit_sales</code></dt>
<dd>decimal</dd> <dt>
<code>days</code></dt><dd>string</dd>
<dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Balance sheet

Catalog key <code>Alltransportationds-balance-sheet</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>inventory</code></dt><dd>string</dd> </dl>

Bank statements

Catalog key <code>Alltransportationds-bank-statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>transaction_date</code></dt><dd>date</dd> <dt><code>account_id</code></dt><dd>id</dd> <dt><code>description</code></dt><dd>string</dd>
<dt><code>amount</code></dt>
<dd>decimal</dd> <dt>
<code>direction</code></dt><dd>string</dd> <div class="catalog-field-

list__row"><dt><code>balance</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>net_income</code></dt><dd>string</dd> <dt><code>non_cash_charges</code></dt><dd>string</dd> <dt><code>increase_in_working_capital</code></dt><dd>string</dd> <dt><code>cash_out</code></dt><dd>string</dd> <dt><code>cash_in</code></dt><dd>string</dd> <dt><code>current_cash_balance</code></dt><dd>string</dd> </dl>

General ledger

Catalog key <code>Alltransportationds-general-ledger</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>journal_entry_id</code></dt><dd>id</dd> <dt><code>debit</code></dt><dd>string</dd> <dt><code>credit</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> <dt><code>currency</code></dt><dd>currency</dd> </dl>

Trial balance

Catalog key <code>Alltransportationds-trial-balance</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd>

<dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>debit_balance</code></dt>
<dd>string</dd> <dt>
<code>credit_balance</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> </dl>

Maintenance logs

Catalog key <code>DclVZaLbyz6Pwxt5Ctr1</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>vehicle_id</code></dt><dd>id</dd> <dt><code>maintenance_date</code></dt><dd>date</dd> <dt><code>maintenance_type</code></dt>
<dd>enum</dd> <dt>
<code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt>
<dd>decimal</dd> <dt>
<code>odometer_reading</code></dt><dd>decimal</dd> <dt><code>work_order_id</code></dt><dd>id</dd> </dl>

Fleet management systems

Catalog key <code>Ta6fK4eo9bymVVQGYEpl</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>fleet_id</code></dt><dd>id</dd> <dt><code>vehicle_ids</code></dt><dd>string</dd> <dt><code>maintenance_schedule</code></dt>
<dd>string</dd> <dt>

<code>fuel_logs</code></dt><dd>string</dd> <dt><code>insurance_details</code></dt><dd>string</dd> <dt><code>fleet_time_in_use</code></dt><dd>string</dd> <dt><code>fleet_available_time</code></dt><dd>string</dd> </dl>

Delivery logs

Catalog key <code>i0N19s7GpHi9QTyG5dhM</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>delivery_date</code></dt><dd>date</dd> <dt><code>vehicle_id</code></dt><dd>id</dd> <dt><code>route_id</code></dt><dd>id</dd> <dt><code>delivery_status</code></dt><dd>enum</dd> <dt><code>delivery_address</code></dt><dd>string</dd> <dt><code>customer_name</code></dt><dd>string</dd> <dt><code>delivery_time</code></dt><dd>string</dd> </dl>

GPS tracking data

Catalog key <code>jHscrlnM67xOUF1IXfos</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>vehicle_id</code></dt><dd>id</dd> <dt><code>tracking_date</code></dt><dd>date</dd> <dt><code>start_location</code></dt><dd>string</dd> <dt><code>end_location</code></dt><dd>string</dd> <div class="catalog-field-

list__row"><dt><code>distance_travelled</code></dt><dd>string</dd> <dt><code>average_speed</code></dt> <dd>string</dd> <dt> <code>route_taken</code></dt><dd>string</dd> </dl>

Accounting Software

Catalog key <code>transportation.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt> <dd>decimal</dd> <dt> <code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt> <dd>decimal</dd> <dt> <code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code> </dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>transportation.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <div

class="catalog-field-list__row"><dt><code>period_end_date</code></dt>
<dd>date</dd> </dl>

HR records

Catalog key <code>transportation.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>transportation.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt>
<dd>decimal</dd> <dt>
<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>
<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd>
<dt><code>total_production_cost</code></dt>
<dd>decimal</dd> <dt>

<code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt> <dd>decimal</dd> <dt> <code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt> <dd>decimal</dd> </dl>

Freight forwarding operations summary

Catalog key <code>transportation.freight_forwarding.operations_summary</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>shipments_handled</code></dt><dd>string</dd> <dt><code>on_time_shipments</code></dt> <dd>string</dd> <dt> <code>total_shipments</code></dt><dd>decimal</dd> </dl>

Maritime fleet operations summary

Catalog key <code>transportation.maritime.operations_summary</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>operating_days</code></dt><dd>string</dd> <dt><code>available_days</code></dt><dd>string</dd> <dt><code>voyage_revenue</code></dt> <dd>decimal</dd> <dt> <code>voyage_costs</code></dt><dd>decimal</dd> <dt><code>on_hire_days</code></dt><dd>string</dd> <dt><code>on_schedule_voyages</code></dt>

<dd>string</dd> <dt>
<code>total_voyages</code></dt><dd>decimal</dd> </dl>

Rail freight operations summary

Catalog key <code>transportation.rail_freight.operations_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>loaded_train_miles</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>available_train_miles</code></dt>
<dd>string</dd> <dt>
<code>total_operating_cost</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>ton_miles_moved</code></dt>
<dd>string</dd> <dt>
<code>on_time_shipments</code></dt><dd>string</dd> <dt><code>total_shipments</code></dt><dd>decimal</dd>
</dl>

Fuel logs

Catalog key <code>uRTkFZUmJC2BLwoRtJxL</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>vehicle_id</code></dt><dd>id</dd> <dt><code>fuel_type</code></dt><dd>enum</dd> <div
class="catalog-field-list__row"><dt><code>fuel_date</code></dt><dd>date</dd>
<dt><code>fuel_quantity</code></dt>
<dd>string</dd> <dt>
<code>fuel_cost</code></dt><dd>decimal</dd> <dt><code>odometer_reading</code></dt><dd>string</dd> </dl>

Metrics (28) {#metrics}

- Driver Turnover Rate
- Fuel Efficiency
- On-Time Delivery Rate
- Fleet Utilization Rate
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Gross Margin per Shipment
- On-Time Delivery (Shipper SLA)
- Schedule Reliability
- Time Charter Equivalent (TCE)
- Vessel Utilization
- Cost per Ton-Mile
- On-Time Train Performance
- Train Utilization
- Maintenance Cost per Mile

Driver Turnover Rate

- **ID:** 14
- **Type:** Operational

- **Unit:** %
- **Description:** The percentage of drivers that leave the company within a specific period.
- **Inputs:** `drivers_left`, `total_drivers`
- **Methodology:** (Number of drivers who left / Total number of drivers) * 100
- **Calculation:** *`drivers_left / total_drivers` 100**
- **Catalog key:** `DtBqsvp9S9HBlcxsvAgd`

Fuel Efficiency

- **ID:** 10
- **Type:** Operational
- **Unit:** Miles per Gallon (MPG)
- **Description:** Average miles per gallon for the fleet.
- **Inputs:** `total_miles`, `total_fuel`
- **Methodology:** Total miles driven / Total fuel used
- **Calculation:** **`total_miles / total_fuel`**
- **Catalog key:** `SAZkxRxYFtf1EIxBXWMk`

On-Time Delivery Rate

- **ID:** 12
- **Type:** Operational
- **Unit:** %
- **Description:** The percentage of deliveries made on or before the scheduled delivery time.
- **Inputs:** `on_time_deliveries`, `total_deliveries`
- **Methodology:** (Number of on-time deliveries / Total deliveries) * 100
- **Calculation:** *`on_time_deliveries / total_deliveries` 100**
- **Catalog key:** `biA3oIK816fwfInBKg2e`

Fleet Utilization Rate

- **ID:** 11
- **Type:** Operational
- **Unit:** Percentage
- **Description:** The percentage of time the fleet is actively in use.
- **Inputs:** `fleet_time_in_use`, `fleet_available_time`
- **Methodology:** (Total time fleet in use / Total available fleet time) * 100
- **Calculation:** *`fleet_time_in_use / fleet_available_time` 100**
- **Catalog key:** `iIaKhCCf0BGvHX1W1i7n`

Monthly Burn Rate

- **ID:** 1000
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `transportation.core.burn_rate`

Cash Conversion Cycle

- **ID:** 988
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `transportation.core.ccc`

Current Ratio

- **ID:** 928
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `transportation.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 982
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`

- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `transportation.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 976
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `transportation.core.dso`

EBITDA Margin

- **ID:** 952
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `transportation.core.ebitda_margin`

Gross Profit Margin

- **ID:** 940
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `transportation.core.gross_margin`

Headcount (FTE)

- **ID:** 1084
- **Type:** Operational

- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `transportation.core.headcount`

Net Profit Margin

- **ID:** 958
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `transportation.core.net_margin`

Operating Cash Flow

- **ID:** 994
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `transportation.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 946
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue 100`**
- **Catalog key:** `transportation.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 934
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** `(current_assets - inventory) / current_liabilities`
- **Catalog key:** `transportation.core.quick_ratio`

Revenue Growth Rate

- **ID:** 964
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `transportation.core.revenue_growth`

Cash Runway

- **ID:** 1006
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** `current_cash_balance / burn_rate`
- **Catalog key:** `transportation.core.runway`

Working Capital

- **ID:** 970
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`

- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** `current_assets - current_liabilities`
- **Catalog key:** `transportation.core.working_capital`

Gross Margin per Shipment

- **ID:** 1871
- **Type:** Financial
- **Unit:** Currency
- **Description:** Average gross profit per shipment handled by the forwarder (aggregated).
- **Inputs:** `gross_profit`, `shipments_handled`
- **Methodology:** Margin per Shipment = Gross Profit / Shipments Handled
- **Calculation:** `gross_profit / shipments_handled`
- **Catalog key:** `transportation.freight_forwarding.gross_margin_per_shipment`

On-Time Delivery (Shipper SLA)

- **ID:** 1872
- **Type:** Operational
- **Unit:** %
- **Description:** Share of forwarder-managed shipments delivered within customer SLA (aggregated).
- **Inputs:** `on_time_shipments`, `total_shipments`
- **Methodology:** OTD = (On-Time Shipments / Total Shipments) × 100
- **Calculation:** *`on_time_shipments / total_shipments` 100**
- **Catalog key:** `transportation.freight_forwarding.on_time_delivery_shipper`

Schedule Reliability

- **ID:** 1867
- **Type:** Operational
- **Unit:** %
- **Description:** Share of port calls or voyages completed within published schedule window (aggregated).
- **Inputs:** `on_schedule_voyages`, `total_voyages`
- **Methodology:** Reliability = (On-Schedule Voyages / Total Voyages) × 100
- **Calculation:** *`on_schedule_voyages / total_voyages` 100**
- **Catalog key:** `transportation.maritime_shipping.schedule_reliability`

Time Charter Equivalent (TCE)

- **ID:** 1866

- **Type:** Financial
- **Unit:** Currency per day
- **Description:** Daily revenue net of voyage costs per day chartered (aggregated).
- **Inputs:** `voyage_revenue`, `voyage_costs`, `on_hire_days`
- **Methodology:** TCE = (Voyage Revenue − Voyage Costs) / On-Hire Days
- **Calculation:** `(voyage_revenue - voyage_costs) / on_hire_days`
- **Catalog key:** `transportation.maritime_shipping.time_charter_equivalent`

Vessel Utilization

- **ID:** 1865
- **Type:** Operational
- **Unit:** %
- **Description:** Operating days at sea or laden days relative to available vessel days (aggregated).
- **Inputs:** `operating_days`, `available_days`
- **Methodology:** Vessel Utilization = (Operating Days / Available Days) × 100
- **Calculation:** *`operating_days / available_days` 100**
- **Catalog key:** `transportation.maritime_shipping.vessel_utilization`

Cost per Ton-Mile

- **ID:** 1861
- **Type:** Financial
- **Unit:** USD per ton-mile
- **Description:** Total rail freight operating cost per ton-mile moved (aggregated).
- **Inputs:** `total_operating_cost`, `ton_miles_moved`
- **Methodology:** Cost per Ton-Mile = Total Operating Cost / Ton-Miles Moved
- **Calculation:** `total_operating_cost / ton_miles_moved`
- **Catalog key:** `transportation.rail_freight.cost_per_ton_mile`

On-Time Train Performance

- **ID:** 1862
- **Type:** Operational
- **Unit:** %
- **Description:** Share of rail shipments arriving within contractual or service window (aggregated).
- **Inputs:** `on_time_shipments`, `total_shipments`
- **Methodology:** OTP = (On-Time Shipments / Total Shipments) × 100
- **Calculation:** *`on_time_shipments / total_shipments` 100**

- **Catalog key:** `transportation.rail_freight.on_time_train_performance`

Train Utilization

- **ID:** 1860
- **Type:** Operational
- **Unit:** %
- **Description:** Loaded train-miles or tonnage moved relative to available train capacity (aggregated).
- **Inputs:** `loaded_train_miles`, `available_train_miles`
- **Methodology:** Train Utilization = (Loaded Train-Miles / Available Train-Miles) × 100
- **Calculation:** *`loaded_train_miles / available_train_miles` 100**
- **Catalog key:** `transportation.rail_freight.train_utilization`

Maintenance Cost per Mile

- **ID:** 13
- **Type:** Operational
- **Unit:** USD/mile
- **Description:** Total maintenance costs divided by total miles driven.
- **Inputs:** `total_maintenance_costs`, `total_miles`
- **Methodology:** Total maintenance costs / Total miles driven
- **Calculation:** **`total_maintenance_costs / total_miles`**
- **Catalog key:** `vXlB5QC3mZJw1auWaIfK`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Vehicle telematics data** — feeds Fuel Efficiency

- **Accounts payable aging** — feeds Days Payable Outstanding (DPO), Cash Conversion Cycle
- **Accounts receivable aging** — feeds Days Sales Outstanding (DSO), Cash Conversion Cycle
- **Balance sheet** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital, Cash Conversion Cycle
- **Bank statements** — feeds Operating Cash Flow, Monthly Burn Rate, Cash Runway
- **General ledger** — feeds Gross Profit Margin, Operating Margin (EBIT), EBITDA Margin, Net Profit Margin, Revenue Growth Rate, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Operating Cash Flow
- **Trial balance** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital
- **Maintenance logs** — feeds Maintenance Cost per Mile
- **Fleet management systems** — feeds Fleet Utilization Rate
- **Delivery logs** — feeds On-Time Delivery Rate
- **GPS tracking data** — feeds Fuel Efficiency, Fleet Utilization Rate, On-Time Delivery Rate, Maintenance Cost per Mile
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE), Driver Turnover Rate
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT), Driver Turnover Rate, Maintenance Cost per Mile
- **Freight forwarding operations summary** — feeds Gross Margin per Shipment, On-Time Delivery (Shipper SLA)
- **Maritime fleet operations summary** — feeds Vessel Utilization, Time Charter Equivalent (TCE), Schedule Reliability
- **Rail freight operations summary** — feeds Train Utilization, Cost per Ton-Mile, On-Time Train Performance
- **Fuel logs** — feeds Fuel Efficiency

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

catalog/industries/trusts.md

Trusts

Pack file: `catalog/industries/trusts` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">6430</code>

17 metrics · 5 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>525120</code></dd>

<dt>NACE Rev.2</dt><dd><code>64.30</code></dd>

<dt>GICS 2023</dt><dd><code>402010</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (5) {#data-options}

Accounting Software

Catalog key <code>trusts.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd>

<dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>trusts.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>trusts.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>trusts.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt><dd>decimal</dd> <dt><code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt><dd>decimal</dd> <dt><code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt><dd>decimal</dd> <dt><code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd> <dt><code>total_miles</code></dt><dd>decimal</dd> </dl>

Aggregated trust vehicle summary

Catalog key <code>trusts.trust.trust_vehicle_summary</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>trust_vehicle_id</code></dt><dd>id</dd> <div class="catalog-
field-list__row"><dt><code>vehicle_type</code></dt><dd>enum</dd> <div
class="catalog-field-list__row"><dt><code>funds_from_operations</code></dt>
<dd>decimal</dd> <dt><code>ffo</code>
</dt><dd>string</dd> <dt>
<code>distributions</code></dt><dd>string</dd> <div class="catalog-field-
list__row"><dt><code>current_noi</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>prior_noi</code></dt><dd>string</dd>
<dt>
<code>trust_assets_under_administration</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>trustee_fees</code></dt><dd>string</dd>
<dt><code>average_trust_aua</code>
</dt><dd>string</dd> </dl>

Metrics (17) {#metrics}

- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Trust Assets Under Administration
- Trustee Fee Yield

Monthly Burn Rate

- **ID:** 1997
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** **`cash_out - cash_in`**
- **Catalog key:** `trusts.core.burn_rate`

Cash Conversion Cycle

- **ID:** 1995
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** **`dso + days_inventory_outstanding - dpo`**
- **Catalog key:** `trusts.core.ccc`

Current Ratio

- **ID:** 1985
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `trusts.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 1994
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`

- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *`accounts_payable / cogs` days**
- **Catalog key:** `trusts.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 1993
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *`accounts_receivable / total_credit_sales` days**
- **Catalog key:** `trusts.core.dso`

EBITDA Margin

- **ID:** 1989
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *`ebitda / revenue` `100`**
- **Catalog key:** `trusts.core.ebitda_margin`

Gross Profit Margin

- **ID:** 1987
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` `100`**
- **Catalog key:** `trusts.core.gross_margin`

Headcount (FTE)

- **ID:** 1999
- **Type:** Operational

- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `trusts.core.headcount`

Net Profit Margin

- **ID:** 1990
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue 100`**
- **Catalog key:** `trusts.core.net_margin`

Operating Cash Flow

- **ID:** 1996
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`
- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** **`net_income + non_cash_charges - increase_in_working_capital`**
- **Catalog key:** `trusts.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 1988
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue 100`**
- **Catalog key:** `trusts.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 1986
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** `(current_assets - inventory) / current_liabilities`
- **Catalog key:** `trusts.core.quick_ratio`

Revenue Growth Rate

- **ID:** 1991
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `trusts.core.revenue_growth`

Cash Runway

- **ID:** 1998
- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** `current_cash_balance / burn_rate`
- **Catalog key:** `trusts.core.runway`

Working Capital

- **ID:** 1992
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`

- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** `current_assets - current_liabilities`
- **Catalog key:** `trusts.core.working_capital`

Trust Assets Under Administration

- **ID:** 1943
- **Type:** Financial
- **Unit:** Currency
- **Description:** Assets administered under trust mandates at period end (aggregated).
- **Methodology:** Directly measured AUA at reporting date
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `trusts.trust.trust_assets_under_administration`

Trustee Fee Yield

- **ID:** 1944
- **Type:** Financial
- **Unit:** %
- **Description:** Trustee and administration fees relative to average trust AUA.
- **Inputs:** `trustee_fees`, `average_trust_aua`
- **Methodology:** Trustee Fee Yield = Trustee Fees / Average Trust AUA × 100
- **Calculation:** *`trustee_fees / average_trust_aua 100`**
- **Catalog key:** `trusts.trust.trustee_fee_yield`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE)
- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT)
- **Aggregated trust vehicle summary** — feeds Trust Assets Under Administration, Trustee Fee Yield

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

catalog/industries/vertical_farming.md

Vertical Farming

Pack file: `catalog/industries/vertical_farming` in ambient-core.

Workspace demo (screenshots) · Catalog consumption

Industry summary

Industry code

ISIC Rev.4 <code class="industry-summary__isic-code">0113</code>

32 metrics · 18 data options

<dl class="industry-summary__codes">

<dt>NAICS 2022</dt><dd><code>111219</code></dd>

<dt>NACE Rev.2</dt><dd><code>01.13</code></dd>

<dt>GICS 2023</dt><dd><code>302020</code></dd>

</dl>

Taxonomy tags from ambient-core <code>pack.yaml</code> / manifest v3 (analysis lens, not legal-entity self-classification). Coverage notes: catalog-industry-coverage.md.

Data options (18) {#data-options}

Accounts payable aging

Catalog key <code>Allverticalfarmingds-ap-aging</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>supplier_name</code></dt><dd>string</dd> <dt><code>bill_id</code></dt><dd>id</dd> <dt><code>bill_date</code></dt><dd>date</dd> <dt><code>due_date</code></dt><dd>date</dd> <dt><code>amount_due</code></dt><dd>decimal</dd> <dt><code>days_outstanding</code></dt><dd>string</dd> <dt><code>aging_bucket</code></dt><dd>string</dd> <dt><code>accounts_payable</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>string</dd> <dt><code>days</code></dt><dd>string</dd> <dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Accounts receivable aging

Catalog key <code>Allverticalfarmingds-ar-aging</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>customer_name</code></dt><dd>string</dd> <div class="catalog-
field-list__row"><dt><code>invoice_id</code></dt><dd>id</dd> <div
class="catalog-field-list__row"><dt><code>invoice_date</code></dt><dd>date</dd>
<dt><code>due_date</code></dt>
<dd>date</dd> <dt>
<code>amount_due</code></dt><dd>decimal</dd> <div class="catalog-field-
list__row"><dt><code>days_outstanding</code></dt><dd>string</dd> <div
class="catalog-field-list__row"><dt><code>aging_bucket</code></dt>
<dd>string</dd> <dt>
<code>accounts_receivable</code></dt><dd>decimal</dd> <div
class="catalog-field-list__row"><dt><code>total_credit_sales</code></dt>
<dd>decimal</dd> <dt>
<code>days</code></dt><dd>string</dd>
<dt><code>days_inventory_outstanding</code></dt><dd>string</dd> </dl>

Balance sheet

Catalog key <code>Allverticalfarmingds-balance-sheet</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_end_date</code></dt><dd>date</dd> <div class="catalog-
field-list__row"><dt><code>inventory</code></dt><dd>string</dd> </dl>

Bank statements

Catalog key <code>Allverticalfarmingds-bank-statements</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>transaction_date</code></dt><dd>date</dd> <div class="catalog-

field-list__row"><dt><code>account_id</code></dt><dd>id</dd> <dt><code>description</code></dt><dd>string</dd> <dt><code>amount</code></dt><dd>decimal</dd> <dt><code>direction</code></dt><dd>string</dd> <dt><code>balance</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>net_income</code></dt><dd>string</dd> <dt><code>non_cash_charges</code></dt><dd>string</dd> <dt><code>increase_in_working_capital</code></dt><dd>string</dd> <dt><code>cash_out</code></dt><dd>string</dd> <dt><code>cash_in</code></dt><dd>string</dd> <dt><code>current_cash_balance</code></dt><dd>string</dd> </dl>

CRM / Sales pipeline

Catalog key <code>Allverticalfarmingds-crm-pipeline</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>opportunity_id</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>stage</code></dt><dd>enum</dd> <dt><code>amount</code></dt><dd>decimal</dd> <dt><code>close_date</code></dt><dd>date</dd> <dt><code>owner</code></dt><dd>string</dd> <dt><code>probability</code></dt><dd>string</dd> <dt><code>active_subscriptions</code></dt><dd>string</dd> <dt><code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>sales_marketing_spend</code></dt><dd>string</dd> <dt><code>new_customers</code></dt><dd>string</dd> </dl>

General ledger

Catalog key <code>Allverticalfarmingds-general-ledger</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>journal_entry_id</code></dt><dd>id</dd> <dt><code>debit</code></dt><dd>string</dd> <dt><code>credit</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> <dt><code>currency</code></dt><dd>currency</dd> </dl>

Marketing spend records

Catalog key <code>Allverticalfarmingds-marketing-spend</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>channel</code></dt><dd>enum</dd> <dt><code>campaign</code></dt><dd>string</dd> <dt><code>spend</code></dt><dd>string</dd> <dt><code>leads</code></dt><dd>string</dd> <dt><code>new_customers</code></dt><dd>string</dd> <dt><code>currency</code></dt><dd>currency</dd> <dt><code>sales_marketing_spend</code></dt><dd>string</dd> </dl>

Product usage logs

Catalog key <code>Allverticalfarmingds-product-usage</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_id</code></dt><dd>id</dd> <dt><code>active_users</code></dt><dd>string</dd> <dt><code>sessions</code></dt><dd>string</dd> <dt><code>feature_adoption</code></dt><dd>string</dd> <dt><code>usage_minutes</code></dt><dd>string</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt><dd>string</dd> <dt><code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> </dl>

Subscription billing records

Catalog key <code>Allverticalfarmingds-subscription-billing</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>customer_id</code></dt><dd>id</dd> <dt><code>plan_name</code></dt><dd>string</dd> <dt><code>mrr</code></dt><dd>string</dd> <dt><code>billing_period</code></dt><dd>enum</dd> <dt><code>status</code></dt><dd>enum</dd> <dt><code>start_date</code></dt><dd>date</dd> <dt><code>end_date</code></dt><dd>date</dd> <dt><code>active_subscriptions</code></dt><dd>string</dd> <dt><code>avg_monthly_price</code></dt><dd>decimal</dd> <dt><code>starting_mrr</code></dt><dd>string</dd> <dt><code>expansion_mrr</code></dt><dd>string</dd> <dt>

<code>contraction_mrr</code></dt><dd>string</dd> <dt><code>churned_mrr</code></dt><dd>string</dd> <dt><code>avg_revenue_per_account</code></dt><dd>decimal</dd> <dt><code>churn_rate</code></dt><dd>decimal</dd> </dl>

Trial balance

Catalog key <code>Allverticalfarmingds-trial-balance</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>account_code</code></dt><dd>id</dd> <dt><code>account_name</code></dt><dd>decimal</dd> <dt><code>debit_balance</code></dt><dd>string</dd> <dt><code>credit_balance</code></dt><dd>string</dd> <dt><code>period</code></dt><dd>enum</dd> </dl>

Irrigation logs

Catalog key <code>DRHUJoXTe9TO3vlfY8bK</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>farm_id</code></dt><dd>id</dd> <dt><code>zone_id</code></dt><dd>id</dd> <dt><code>water_volume_liters</code></dt><dd>decimal</dd> <dt><code>irrigation_duration_minutes</code></dt><dd>decimal</dd> <dt><code>nutrient_ec</code></dt><dd>decimal</dd> <dt><code>ph_level</code></dt><dd>decimal</dd> <div class="catalog-field-

list__row"><dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_water_liters</code></dt> <dd>decimal</dd> </dl>

Farm management systems

Catalog key <code>QKVboc2mj2QQ8u2RFOF1</code>

Fields

<dl class="catalog-field-list"> <dt> <code>date</code></dt><dd>date</dd> <dt><code>farm_id</code></dt><dd>id</dd> <dt><code>crop_planted</code></dt><dd>string</dd> <dt><code>total_harvested_kg</code></dt> <dd>decimal</dd> <dt> <code>total_growing_area</code></dt><dd>decimal</dd> <dt><code>total_water_liters</code></dt><dd>decimal</dd> <dt><code>total_energy_kwh</code></dt> <dd>decimal</dd> <dt> <code>sum_health_scores</code></dt><dd>string</dd> <dt><code>total_crops</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt> <dd>decimal</dd> <dt> <code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt> <dd>decimal</dd> <dt> <code>energy_consumed</code></dt><dd>string</dd> <dt><code>total_light_delivered</code></dt><dd>decimal</dd> <dt><code>repair_cost_avoided</code> </dt><dd>decimal</dd> <dt> <code>predictive_maintenance_cost</code></dt><dd>decimal</dd> </dl>

Harvest logs

Catalog key <code>SuYpCOlDxv3Q0Sf2xq9L</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>farm_id</code></dt><dd>id</dd> <dt><code>crop_type</code></dt><dd>enum</dd> <dt><code>harvest_date</code></dt><dd>date</dd> <dt><code>quantity_harvested</code></dt><dd>string</dd> <dt><code>market_price</code></dt><dd>decimal</dd> <dt><code>quality_grade</code></dt><dd>enum</dd> </dl>

Crop monitoring systems

Catalog key <code>mZgKaZV2P6tbi7msxL3y</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>farm_id</code></dt><dd>id</dd> <dt><code>crop_id</code></dt><dd>id</dd> <dt><code>health_score</code></dt><dd>decimal</dd> <dt><code>growth_stage</code></dt><dd>enum</dd> <dt><code>pest_pressure_index</code></dt><dd>decimal</dd> <dt><code>sensor_reading_count</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd> <dt><code>total_growing_area</code></dt><dd>decimal</dd> <dt><code>sum_health_scores</code></dt><dd>string</dd> <dt><code>total_crops</code></dt><dd>decimal</dd> </dl>

Accounting Software

Catalog key <code>vertical_farming.core.accounting_software</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>cogs</code></dt><dd>decimal</dd> <dt><code>cost_of_goods</code></dt><dd>decimal</dd> <dt><code>operating_expenses</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt><dd>decimal</dd> <dt><code>produce_revenue</code></dt><dd>decimal</dd> <dt><code>total_production_cost</code></dt><dd>decimal</dd> <dt><code>total_operational_cost</code></dt><dd>decimal</dd> </dl>

Financial statements

Catalog key <code>vertical_farming.core.financial_statements</code>

Fields

<dl class="catalog-field-list"> <dt><code>date</code></dt><dd>date</dd> <dt><code>statement_type</code></dt><dd>enum</dd> <dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> </dl>

HR records

Catalog key <code>vertical_farming.core.hr_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>headcount</code></dt><dd>decimal</dd> </dl>

Payroll records

Catalog key <code>vertical_farming.core.payroll_records</code>

Fields

<dl class="catalog-field-list"> <dt>
<code>date</code></dt><dd>date</dd>
<dt><code>period_start_date</code></dt><dd>date</dd> <dt><code>period_end_date</code></dt><dd>date</dd> <dt><code>headcount</code></dt>
<dd>decimal</dd> <dt>
<code>gross_pay</code></dt><dd>decimal</dd> <dt><code>net_pay</code></dt><dd>decimal</dd> <dt><code>employer_taxes</code></dt>
<dd>decimal</dd> <dt>
<code>labor_cost</code></dt><dd>decimal</dd> <dt><code>revenue</code></dt><dd>decimal</dd> <dt><code>net_income</code></dt>
<dd>decimal</dd> <dt>
<code>total_labor_costs</code></dt><dd>decimal</dd> <dt><code>total_harvested_kg</code></dt><dd>decimal</dd>
<dt><code>total_production_cost</code></dt>
<dd>decimal</dd> <dt>
<code>total_operational_cost</code></dt><dd>decimal</dd> <dt><code>drivers_left</code></dt>
<dd>decimal</dd> <dt>
<code>total_drivers</code></dt><dd>decimal</dd> <dt><code>total_maintenance_costs</code></dt><dd>decimal</dd>
<dt><code>total_miles</code></dt>
<dd>decimal</dd> </dl>

Metrics (32) {#metrics}

- Predictive Maintenance Cost Savings
- Cost per Kilogram of Produce
- Energy Cost per Unit of Light Delivered
- Annual Recurring Revenue (ARR)
- Customer Acquisition Cost (CAC)
- Gross Revenue Churn Rate
- Customer Lifetime Value (LTV)
- LTV : CAC Ratio
- Monthly Recurring Revenue (MRR)
- Net Revenue Retention (NRR)
- Rule of 40
- Crop Health Index
- Profit Margin
- Labor Cost per Kilogram of Produce
- Water Usage Efficiency
- Yield per Square Meter
- Monthly Burn Rate
- Cash Conversion Cycle
- Current Ratio
- Days Payable Outstanding (DPO)
- Days Sales Outstanding (DSO)
- EBITDA Margin
- Gross Profit Margin
- Headcount (FTE)
- Net Profit Margin
- Operating Cash Flow
- Operating Margin (EBIT)
- Quick Ratio (Acid Test)
- Revenue Growth Rate
- Cash Runway
- Working Capital
- Energy Consumption per Square Meter

Predictive Maintenance Cost Savings

- **ID:** 23
- **Type:** Financial

- **Unit:** Currency (e.g., USD)
- **Description:** Cost savings achieved by preventing equipment failures through predictive maintenance.
- **Inputs:** `repair_cost_avoided`, `predictive_maintenance_cost`
- **Methodology:** Potential Repair Cost Avoided - Predictive Maintenance Cost
- **Calculation:** **`repair_cost_avoided - predictive_maintenance_cost`**
- **Catalog key:** `0MJiYgAPcgkzE3xEgOFE`

Cost per Kilogram of Produce

- **ID:** 20
- **Type:** Operational
- **Unit:** $/kg
- **Description:** Total cost associated with producing one kilogram of crops.
- **Inputs:** `total_production_cost`, `total_operational_cost`
- **Methodology:** Total Production Cost (Seeds, Fertilizers, Nutrients) + Total Operational Cost (Labor, Energy, Water, Maintenance) / Total Yield (kg)
- **Calculation:** **`total_production_cost + total_operational_cost`**
- **Catalog key:** `6IfrKxlTi9pbGzpuLL5u`

Energy Cost per Unit of Light Delivered

- **ID:** 22
- **Type:** Operational
- **Unit:** kWh/lux-hour
- **Description:** Energy consumption required to deliver the desired light intensity.
- **Inputs:** `energy_consumed`, `total_light_delivered`
- **Methodology:** Energy Consumed (kWh) / Total Light Delivered (lux-hours or similar unit)
- **Calculation:** **`energy_consumed / total_light_delivered`**
- **Catalog key:** `8B9bFHS6GSsjzHx9ppKu`

Annual Recurring Revenue (ARR)

- **ID:** 1017
- **Type:** Financial
- **Unit:** Currency/year
- **Description:** Annualized run-rate of recurring subscription revenue.
- **Methodology:** ARR = MRR × 12
- **Calculation:** *`mrr` 12**
- **Catalog key:** `Allverticalfarmingsaas-arr`

Customer Acquisition Cost (CAC)

- **ID:** 1035
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Fully-loaded sales & marketing cost to acquire one new customer.
- **Inputs:** `sales_marketing_spend`, `new_customers`
- **Methodology:** CAC = Total Sales & Marketing Spend / New Customers Acquired
- **Calculation:** **`sales_marketing_spend / new_customers`**
- **Catalog key:** `Allverticalfarmingsaas-cac`

Gross Revenue Churn Rate

- **ID:** 1029
- **Type:** Operational
- **Unit:** %
- **Description:** Percentage of recurring revenue lost to cancellations and downgrades.
- **Inputs:** `churned_mrr`, `starting_mrr`
- **Methodology:** Gross Churn = (Churned MRR / Starting MRR) × 100
- **Calculation:** *_churned_mrr / starting_mrr_ `100`**
- **Catalog key:** `Allverticalfarmingsaas-gross-churn`

Customer Lifetime Value (LTV)

- **ID:** 1041
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Expected gross-margin revenue from a customer over their lifetime.
- **Inputs:** `avg_revenue_per_account`, `gross_margin_pct`, `churn_rate`
- **Methodology:** LTV = (Average Revenue per Account × Gross Margin %) / Churn Rate
- **Calculation:** *_(avg_revenue_per_account_ `gross_margin_pct / 100) / churn_rate`**
- **Catalog key:** `Allverticalfarmingsaas-ltv`

LTV : CAC Ratio

- **ID:** 1047
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Ratio of customer lifetime value to acquisition cost — core SaaS unit economics.
- **Methodology:** LTV:CAC = Customer Lifetime Value / Customer Acquisition Cost

- **Calculation:** `ltv / cac`
- **Catalog key:** `Allverticalfarmingsaas-ltv-cac`

Monthly Recurring Revenue (MRR)

- **ID:** 1011
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Normalized recurring subscription revenue billed in a month.
- **Inputs:** `active_subscriptions`, `avg_monthly_price`
- **Methodology:** MRR = Σ (Active Subscriptions × Monthly Subscription Price)
- **Calculation:** *`active_subscriptions` `avg_monthly_price`**
- **Catalog key:** `Allverticalfarmingsaas-mrr`

Net Revenue Retention (NRR)

- **ID:** 1023
- **Type:** Financial
- **Unit:** %
- **Description:** Revenue retained and expanded from existing customers, net of churn and contraction.
- **Inputs:** `starting_mrr`, `expansion_mrr`, `contraction_mrr`, `churned_mrr`
- **Methodology:** NRR = ((Starting MRR + Expansion − Contraction − Churn) / Starting MRR) × 100
- **Calculation:** *`(starting_mrr + expansion_mrr - contraction_mrr - churned_mrr) / starting_mrr` `100`**
- **Catalog key:** `Allverticalfarmingsaas-nrr`

Rule of 40

- **ID:** 1053
- **Type:** Financial
- **Unit:** %
- **Description:** Balance of growth and profitability for recurring-revenue businesses.
- **Inputs:** `revenue_growth_pct`, `profit_margin_pct`
- **Methodology:** Rule of 40 = Revenue Growth % + Profit Margin %
- **Calculation:** **`revenue_growth_pct + profit_margin_pct`**
- **Catalog key:** `Allverticalfarmingsaas-rule-of-40`

Crop Health Index

- **ID:** 19
- **Type:** Operational

- **Unit:** Index
- **Description:** An index indicating the overall health of crops based on various parameters.
- **Inputs:** `sum_health_scores`, `total_crops`
- **Methodology:** Crop Health Index = (Sum of Health Scores / Total Number of Crops) * 100
- **Calculation:** *`sum_health_scores / total_crops` 100**
- **Catalog key:** `E5khIDv93onvwtXBUsJH`

Profit Margin

- **ID:** 21
- **Type:** Financial
- **Unit:** %
- **Description:** Difference between revenue and total costs.
- **Inputs:** `produce_revenue`, `total_production_cost`, `total_operational_cost`
- **Methodology:** Revenue from Produce Sales - Total Production Cost - Total Operational Cost
- **Calculation:** **`produce_revenue - total_production_cost - total_operational_cost`**
- **Catalog key:** `F6IlgQNV6qNtUhk3BVtT`

Labor Cost per Kilogram of Produce

- **ID:** 18
- **Type:** Operational
- **Unit:** $/kg
- **Description:** Average labor cost associated with producing one kilogram of produce.
- **Inputs:** `total_labor_costs`, `total_harvested_kg`
- **Methodology:** Labor Cost per Kilogram of Produce = Total Labor Costs / Total Harvested Produce (kg)
- **Calculation:** **`total_labor_costs / total_harvested_kg`**
- **Catalog key:** `WeTkxvZyeWX51XfBGsuR`

Water Usage Efficiency

- **ID:** 16
- **Type:** Operational
- **Unit:** kg/liter
- **Description:** Amount of produce harvested per liter of water used.
- **Inputs:** `total_harvested_kg`, `total_water_liters`
- **Methodology:** Water Usage Efficiency = Total Harvested Produce (kg) / Total Water Used (liters)

- **Calculation:** `total_harvested_kg / total_water_liters`
- **Catalog key:** `f3yEk5mEWwKhAFFQFyje`

Yield per Square Meter

- **ID:** 15
- **Type:** Operational
- **Unit:** kg/m²
- **Description:** Amount of produce harvested per square meter of growing area.
- **Inputs:** `total_harvested_kg`, `total_growing_area`
- **Methodology:** Yield per Square Meter = Total Harvested Produce (kg) / Total Growing Area (m²)
- **Calculation:** `total_harvested_kg / total_growing_area`
- **Catalog key:** `jnQw1pD0motTKBZ9csPD`

Monthly Burn Rate

- **ID:** 999
- **Type:** Financial
- **Unit:** Currency/month
- **Description:** Net cash consumed per month (negative net operating cash flow).
- **Inputs:** `cash_out`, `cash_in`
- **Methodology:** Net Burn = Cash Out − Cash In (per month)
- **Calculation:** `cash_out - cash_in`
- **Catalog key:** `vertical_farming.core.burn_rate`

Cash Conversion Cycle

- **ID:** 987
- **Type:** Financial
- **Unit:** Days
- **Description:** Days to convert investments in inventory and receivables back into cash.
- **Inputs:** `days_inventory_outstanding`
- **Methodology:** CCC = DSO + Days Inventory Outstanding − DPO
- **Calculation:** `dso + days_inventory_outstanding - dpo`
- **Catalog key:** `vertical_farming.core.ccc`

Current Ratio

- **ID:** 927
- **Type:** Financial

- **Unit:** Ratio
- **Description:** Short-term liquidity: ability to cover current liabilities with current assets.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Current Ratio = Current Assets / Current Liabilities
- **Calculation:** **`current_assets / current_liabilities`**
- **Catalog key:** `vertical_farming.core.current_ratio`

Days Payable Outstanding (DPO)

- **ID:** 981
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days the business takes to pay its suppliers.
- **Inputs:** `accounts_payable`, `cogs`, `days`
- **Methodology:** DPO = (Accounts Payable / COGS) × Number of Days
- **Calculation:** *_`accounts_payable / cogs`_ `days`**
- **Catalog key:** `vertical_farming.core.dpo`

Days Sales Outstanding (DSO)

- **ID:** 975
- **Type:** Financial
- **Unit:** Days
- **Description:** Average number of days to collect cash after a sale.
- **Inputs:** `accounts_receivable`, `total_credit_sales`, `days`
- **Methodology:** DSO = (Accounts Receivable / Total Credit Sales) × Number of Days
- **Calculation:** *_`accounts_receivable / total_credit_sales`_ `days`**
- **Catalog key:** `vertical_farming.core.dso`

EBITDA Margin

- **ID:** 951
- **Type:** Financial
- **Unit:** %
- **Description:** Core operating cash profitability before interest, tax, depreciation and amortization.
- **Inputs:** `ebitda`, `revenue`
- **Methodology:** EBITDA Margin = (EBITDA / Revenue) × 100
- **Calculation:** *_`ebitda / revenue`_ `100`**
- **Catalog key:** `vertical_farming.core.ebitda_margin`

Gross Profit Margin

- **ID:** 939
- **Type:** Financial
- **Unit:** %
- **Description:** Share of revenue retained after the direct cost of goods sold.
- **Inputs:** `revenue`, `cogs`
- **Methodology:** Gross Margin = ((Revenue − COGS) / Revenue) × 100
- **Calculation:** *`(revenue - cogs) / revenue` 100**
- **Catalog key:** `vertical_farming.core.gross_margin`

Headcount (FTE)

- **ID:** 1085
- **Type:** Operational
- **Unit:** Count
- **Description:** Period-end count of active full-time-equivalent employees. A directly measured input, not a derived ratio.
- **Methodology:** Period-end count of active employees expressed as full-time equivalents (directly measured, not calculated).
- **Calculation:** _No machine-readable calc — directly reported or qualitative._
- **Catalog key:** `vertical_farming.core.headcount`

Net Profit Margin

- **ID:** 957
- **Type:** Financial
- **Unit:** %
- **Description:** Bottom-line profitability after all costs, interest and tax.
- **Inputs:** `net_income`, `revenue`
- **Methodology:** Net Profit Margin = (Net Income / Revenue) × 100
- **Calculation:** *`net_income / revenue` 100**
- **Catalog key:** `vertical_farming.core.net_margin`

Operating Cash Flow

- **ID:** 993
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Cash generated by core operations during the period.
- **Inputs:** `net_income`, `non_cash_charges`, `increase_in_working_capital`

- **Methodology:** Operating Cash Flow = Net Income + Non-Cash Charges − Increase in Working Capital
- **Calculation:** `net_income + non_cash_charges - increase_in_working_capital`
- **Catalog key:** `vertical_farming.core.operating_cash_flow`

Operating Margin (EBIT)

- **ID:** 945
- **Type:** Financial
- **Unit:** %
- **Description:** Operating profitability after operating expenses (EBIT margin).
- **Inputs:** `operating_income`, `revenue`
- **Methodology:** Operating Margin = (Operating Income / Revenue) × 100
- **Calculation:** *`operating_income / revenue` 100**
- **Catalog key:** `vertical_farming.core.operating_margin`

Quick Ratio (Acid Test)

- **ID:** 933
- **Type:** Financial
- **Unit:** Ratio
- **Description:** Acid-test liquidity excluding inventory — most conservative short-term solvency view.
- **Inputs:** `current_assets`, `inventory`, `current_liabilities`
- **Methodology:** Quick Ratio = (Current Assets − Inventory) / Current Liabilities
- **Calculation:** `(current_assets - inventory) / current_liabilities`
- **Catalog key:** `vertical_farming.core.quick_ratio`

Revenue Growth Rate

- **ID:** 963
- **Type:** Financial
- **Unit:** %
- **Description:** Period-over-period revenue growth rate.
- **Inputs:** `revenue`, `revenue_prior`
- **Methodology:** Revenue Growth = ((Current Period Revenue − Prior Period Revenue) / Prior Period Revenue) × 100
- **Calculation:** *`(revenue - revenue_prior) / revenue_prior` 100**
- **Catalog key:** `vertical_farming.core.revenue_growth`

Cash Runway

- **ID:** 1005

- **Type:** Financial
- **Unit:** Months
- **Description:** Number of months of cash remaining at the current net burn rate.
- **Inputs:** `current_cash_balance`
- **Methodology:** Runway (months) = Current Cash Balance / Monthly Net Burn
- **Calculation:** **`current_cash_balance / burn_rate`**
- **Catalog key:** `vertical_farming.core.runway`

Working Capital

- **ID:** 969
- **Type:** Financial
- **Unit:** Currency (e.g., USD)
- **Description:** Operating liquidity available to fund day-to-day operations.
- **Inputs:** `current_assets`, `current_liabilities`
- **Methodology:** Working Capital = Current Assets − Current Liabilities
- **Calculation:** **`current_assets - current_liabilities`**
- **Catalog key:** `vertical_farming.core.working_capital`

Energy Consumption per Square Meter

- **ID:** 17
- **Type:** Operational
- **Unit:** kWh/m²
- **Description:** Amount of energy consumed per square meter of growing area.
- **Inputs:** `total_energy_kwh`, `total_growing_area`
- **Methodology:** Energy Consumption per Square Meter = Total Energy Consumed (kWh) / Total Growing Area (m²)
- **Calculation:** **`total_energy_kwh / total_growing_area`**
- **Catalog key:** `yGoqLDC3Mrk1OeH36jkB`

Data engineering pipeline

Ambient Core ships the **catalog** (typed data options and metrics) and `ambient_pipeline` helpers. Lakehouse jobs and schedules live in the application repository that pins a core tag.

```
flowchart LR
  upload[Bronze CSV upload]
  option[Pack data option fields]
  silver[Silver tenant-metrics]
  calc[ambient_calc]
  gold[Gold products]
  upload --> option --> silver --> calc --> gold
```

Canonical order: map columns → coerce types → unpivot → stamp lineage → (optional Silver validate) → contract assert → write. Gold vertical products evaluate catalog formulas with `ambient_calc` after Silver.

How this pack feeds the pipeline

- **Accounts payable aging** — feeds Days Payable Outstanding (DPO), Cash Conversion Cycle
- **Accounts receivable aging** — feeds Days Sales Outstanding (DSO), Cash Conversion Cycle
- **Balance sheet** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital, Cash Conversion Cycle
- **Bank statements** — feeds Operating Cash Flow, Monthly Burn Rate, Cash Runway
- **CRM / Sales pipeline** — feeds Monthly Recurring Revenue (MRR), Customer Acquisition Cost (CAC)
- **General ledger** — feeds Gross Profit Margin, Operating Margin (EBIT), EBITDA Margin, Net Profit Margin, Revenue Growth Rate, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Operating Cash Flow, Customer Acquisition Cost (CAC)
- **Marketing spend records** — feeds Customer Acquisition Cost (CAC), LTV : CAC Ratio
- **Product usage logs** — feeds Net Revenue Retention (NRR), Gross Revenue Churn Rate
- **Subscription billing records** — feeds Monthly Recurring Revenue (MRR), Annual Recurring Revenue (ARR), Net Revenue Retention (NRR), Gross Revenue Churn Rate, Customer Lifetime Value (LTV), LTV : CAC Ratio, Rule of 40
- **Trial balance** — feeds Current Ratio, Quick Ratio (Acid Test), Working Capital
- **Irrigation logs** — feeds Water Usage Efficiency
- **Farm management systems** — feeds Yield per Square Meter, Water Usage Efficiency, Energy Consumption per Square Meter, Crop Health Index, Cost per Kilogram of Produce, Profit Margin, Energy Cost per Unit of Light Delivered, Predictive Maintenance Cost Savings
- **Harvest logs** — feeds Yield per Square Meter, Labor Cost per Kilogram of Produce
- **Crop monitoring systems** — feeds Yield per Square Meter, Crop Health Index
- **Accounting Software** — feeds Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, Cost per Kilogram of Produce, Profit Margin
- **Financial statements** — feeds Current Ratio, Quick Ratio (Acid Test), Gross Profit Margin, Operating Margin (EBIT), Net Profit Margin, EBITDA Margin, Operating Cash Flow, Revenue Growth Rate, Working Capital, Days Sales Outstanding (DSO), Days Payable Outstanding (DPO), Cash Conversion Cycle, Monthly Burn Rate, Cash Runway
- **HR records** — feeds Headcount (FTE), Labor Cost per Kilogram of Produce

- **Payroll records** — feeds Net Profit Margin, Operating Margin (EBIT), Labor Cost per Kilogram of Produce, Cost per Kilogram of Produce

Further reading: Catalog consumption · Pipeline governance helpers

---

Generated from ambient-core ref `924039e` (digest `2fa4ebc79e445a83`). Regenerate: `npm run sync:catalog-wiki`.

---

index.md

Start Here

Public technical wiki for **ambient-core** (open source) and the Ambient Systems **Command Centre** (platform). Read the orientation pages under this section when a topic spans both. Use Integrator or Operator only when the content belongs to one lane.

Choose your path

- **Orientation (this section)** - Compare open-source core and hosted platform without mixing ownership. Start with Foundation vs full product, then the Work cycles pages, then Ambient Core in the platform.
- **Open Source Integrator** - Pin ambient-core, validate contracts, and embed catalog and pipeline libraries in your own stack. Start with Usage and Integrating. Metric equations and data-option definitions live only under Integrator (open-source core and reference catalog).
- **Platform Subscriber** - Hosted Command Centre guides for invitation-only or local setup (not a self-serve signup funnel). Start with Using the platform and Platform features. Operator pages show product behavior and demos; they link to Integrator for formulas and pack definitions.
- **Site ops** - Marketing-site capture and outbound runbooks (Wix Forms → Contacts). Start with Site ops then Outbound capture.

Open-source core and catalog pages are synced from ambient-core at pinned refs recorded during site build. Operator platform, workspace demo, and site-ops pages are maintained in this wiki.

Download full Docs

Every documentation page includes a bottom download bar for the versioned customer package (Markdown and PDF). The site footer also lists Downloads.

- Package version includes the build date and the pinned ambient-core ref (see the download bar).

- Files: `/docs/downloads/ambient-systems-docs-latest.md` and `/docs/downloads/ambient-systems-docs-latest.pdf` (also versioned filenames beside them).

---

open-core/assurance-lifecycle.md

Assurance lifecycle: core vs platform

Assurance is the work cycle that asks whether a metric can be **defended**: correct under a fixed definition, supported by evidence, traceable through lineage, and aligned between operational signals and financial outcomes. It is distinct from benchmarking (who is better than whom) and from investor disclosure (whether mandates for capital access are met)—though disclosed and benchmarked numbers should ideally pass the same assurance bar.

ambient-core defines **what** is measured and **how** Gold products are shaped; a **paid platform** runs control workflows, sign-off, and attestation packaging.

Index of all cycles: work-cycles.md.

End-to-end flow

```
flowchart LR
 subgraph core [ambient_core_OSS]
 Catalog[catalog_methodology_calc]
 QualityContract[quality_v1]
 BridgeContract[operational_financial_bridge_v1]
 Lineage[pipeline_lineage_rules]
 end
 subgraph platform [paid_platform]
 Controls[control_packs]
 Evidence[signoff_and_evidence]
 DQViews[dq_and_variance_UI]
 Attest[assurance_exports]
 end
 Catalog --> Controls
 Lineage --> Evidence
 QualityContract --> DQViews
 BridgeContract --> DQViews
 Evidence --> Attest
```

Phase mapping

1. Control scope and definitions

- **Core** — Catalog `methodology`, `industry.segment.slug` keys, and CONVENTIONS.md; contract schemas that define required columns and consumption rules.
- **Platform** — Control objectives per org and lens; mapping catalog metrics and contracts to control assertions; `assurance_framework` or `control_pack_id` in run metadata.

2. Evidence and ingestion

- **Core** — Bronze lineage enforcement (`enforce_bronze_lineage`), catalog input field policy, PHI denylist; plain-text extract boundary per governed-data.md.
- **Platform** — Upload approvals, immutable evidence store, who certified which period, restatement policy.

3. Data quality measurement

- **Core** — quality-v1.yaml shape: completeness, accuracy, anomaly counts, overall score, BCBS 239–oriented lineage fields on the governed scorecard and pipeline audit products.
- **Platform** — DQ dashboards, thresholds, escalation when grades fall below policy.

4. Operational–financial reconciliation

- **Core** — operational-financial-bridge-v1.yaml (alignment flag, variance counts, bridge mapping records); bridge_rules.yaml and `metricBridgeHints.js` for narrative links between operational names and financial metrics (energy, NOI, DSCR themes).
- **Platform** — Reconciliation UI, ticketed variances, covenant “watermelon KPI” guards described in contract consumption notes.

5. Attestation and audit trail

- **Core** — observability-pipeline-v1.yaml audit and performance event shapes for pipeline observability.
- **Platform** — Assurance packs for external reviewers, read-only auditor portals, period close sign-off matrices.

What core will not do

- Issue audit opinions or SOC reports.
- Replace GRC, SOX, or IT general-control tools.
- Store control testing workflows or sample selections.

The **`auditor` agent profile** in ../AGENTS.md validates contract YAML and summarizes governance observations in a plan-execute run—it is **not** the assurance product described here.

REIT illustration (alignment, not ranking)

For a real-estate or listed REIT org, assurance often requires showing that **energy and property operations** tie to **financial outcomes**: energy consumption data options in real_estate.yaml, NOI and cap-rate semantics in the same pack, and bridge hints for energy versus opex and NOI. Assurance succeeds when operational extracts and financial close reconcile under one definition—not when the REIT beats a peer on NOI margin (benchmarking-lifecycle.md).

Related

- work-cycles.md
- benchmarking-lifecycle.md
- investor-disclosure-lifecycle.md
- CANONICAL_SCOPE.md
- pipeline.md
- quality-v1.yaml

Canonical source: `ambient-core` at `docs/assurance-lifecycle.md` (sync ref `924039e`).

---

open-core/benchmarking-lifecycle.md

Benchmarking lifecycle: core vs platform

Ambient Core is designed to support **benchmarking-style** FP&A and operational intelligence: fixed KPI definitions, peer-appropriate metrics within an analysis lens, and guardrail bands—not the full commercial loop from peer comparison through gap decomposition to managed improvement. That second half lives in a **paid multi-tenant platform** (for example ambient-systems-platform) on top of governed Gold data.

Practitioner benchmarking separates two phases that must not be conflated: a **performance-comparison** phase (levels, gaps, pace-setters) and a **practice-improvement** phase (what to change, whether it is transferable, and how to sustain gains). Core ships the measurement vocabulary and contracts; the platform ships population comparison, normalization policy, structural-vs-improvable decomposition, and action workflows.

End-to-end flow

```
flowchart LR
  subgraph core [ambient_core_OSS]
    Catalog[catalog_metrics_calc]
    Bench[benchmarks_yaml]
    Codes[industryCodes_ISIC_NAICS]
    Calc[ambient_calc]
    BridgeContract[operational_financial_bridge_v1]
  end
  subgraph platform [paid_platform]
    Peers[peer_cohort_actuals]
    Norm[normalization_policy]
    Gap[gap_and_waterfall_UI]
    Improve[opportunity_workflows]
  end
  Catalog --> Calc
  Bench --> Gap
  Codes --> Peers
  Calc --> BridgeContract
  Peers --> Gap
  Norm --> Gap
  Gap --> Improve
```

Phase mapping

1. Scope and comparability

- **Core** — Industry packs in catalog/packs.yaml; analysis lens terminology; official **ISIC / NAICS / NACE / GICS** tags on each pack (`industryCodes` in `pack.yaml`, exported in manifest v3). Coverage gaps: catalog-industry-coverage.md.
- **Platform** — `peer_group_id` and org matching rules (sector, size band, geography, capital structure); optional `reporting_group_id` for holding-company rollup UI only. See recommended metadata in catalog-consumption.md.

2. Metric selection

- **Core** — Metrics carry `type` (Financial / Operational), `segment`, and `fpaWorkflow`; operating drivers sit beside *`.core.`* close metrics in each pack (for example Real Estate NOI and cap rate in real_estate.yaml; listed REIT investor metrics under `financial_services.reits.*` in financial_services.yaml).
- **Platform** — Curated benchmark packs per tenant org; effectiveness versus efficiency groupings in the product UI.

3. Operational definitions

- **Core** — `methodology`, stable keys `industry.segment.slug`, and machine-readable `calc` blocks (`expr` + `inputs`) or `input: true` per CONVENTIONS.md. lib/ambient_calc evaluates declared formulas deterministically; core does not run ad-hoc peer comparison math.

- **Platform** — Upload mapping enforcement, harmonized recomputation when filers use different FFO or NOI boundaries, versioned study or disclosure files.

4. Normalization

- **Core** — Intensity-style metrics where authored (per available seat-km, per megawatt, per share, per ton-mile). Normalization **policy** (FX, mix indices, unlevered views) is not computed in this repository.
- **Platform** — Applies normalization before gaps are asserted; frontier methods such as DEA for large peer sets are platform analytics, not OSS.

5. Gap analysis (pace-setter vs focal)

- **Core** — Per-vertical `catalog/industries/<pack>/benchmarks.yaml` supplies **healthy bands** linked via `benchmarkKey` on metrics (guardrails for planning, not peer rankings). Sector profiles indicate which metrics typify a business model for comparison design.
- **Platform** — Stores peer actuals, computes gap versus pace-setter, ranked displays, optional composite scores with declared weights and sensitivity analysis.

6. Decomposition: improvable vs structural

This is the main **paid** insight after benchmarking: users need to see which parts of a gap they can influence versus which reflect structure they should not misread as poor management.

Core enablers (not the product):

- **`calc.inputs` and metric dependencies** — Show which levers are mathematically part of a KPI (numerator, denominator, and upstream metrics resolved by `ambient_calc`). The platform can walk this graph when building bridges; core does not label slices “structural” or “improvable.”
- **operational-financial-bridge-v1.yaml** — Governed alignment between operational telemetry and financial metrics (variance counts, bridge mapping records). This guards “watermelon KPIs”; it is not a full waterfall chart product.
- **bridge_rules.yaml** and generated `metricBridgeHints.js` — Narrative hints linking operational names to financial outcomes.

Platform (paid):

- Bridge or waterfall charts that split a headline gap into **structural** slices (documented portfolio mix, geography, asset age, climate, one-offs) and an **improvable** residual tied to operating or financing discipline.
- Linking improvable residuals to actions, owners, and monitoring.

7. Improvement and practice transfer

- **Core** — `fpaWorkflow` strings on catalog metrics guide where improvement effort maps.

- **Platform** — Post-benchmark improvement plans, ROI estimates, workflows, ticketing, and sustainment—commercial product scope.

How the platform should use catalog mechanics

When showing “what is improvable,” the platform should:

1. Resolve the focal metric’s `calc` block and dependency chain from the manifest or YAML.
2. Compare focal and pace-setter values **per input or sub-metric** where data exists, using the same operational definitions.
3. Attribute remaining gap to **documented structural factors** stored in platform metadata (not invented in core).
4. Surface `fpaWorkflow` and bridge hints for interpretation when recommendations are generated.

Core will **not** store peer time series, draw waterfall UI, or classify gap slices without platform logic.

Illustrative REIT paired comparison (sketch)

This sketch mirrors a rigorous paired benchmark design (comparability screen, fixed definitions, normalization, gap, decomposition). **Empirical figures and primary sources belong in external research materials**; ambient-core supplies the metric definitions and methodology strings below.

Setup. Two comparable listed data-centre REITs in the same market and disclosure regime, compared under a Real Estate / REIT investor lens. Each is a **tenant org** on the platform with the same `peer_group_id`, not a single legal entity picker.

Definitions (core).

- Occupancy — portfolio rate with a fixed numerator/denominator boundary in upload mapping (platform enforces parity across filers).
- NOI margin — aligns with Real Estate pack semantics (net operating income relative to revenue); see Real Estate NOI methodology in the catalog.
- Investor metrics — `financial_services.reits.funds_from_operations`, `financial_services.reits.ffo_payout_ratio`, `financial_services.reits.same_store_noi_growth` where the org is modeled on the REIT investor segment.
- Leverage and coverage — use *`.core.`* and REIT-appropriate financial metrics; platform reads disclosed gearing and interest coverage under harmonized definitions.

Normalization (platform). Common currency for cross-listing comparisons; per-megawatt energy intensity when MW and cost lines are harmonized; mix index for colocation versus hyperscale weighting.

Gap (platform). Pace-setter identified per metric; absolute and percentage gaps computed after normalization.

Decomposition (platform). Example narrative for a large NOI margin gap: portion attributed to **structural** portfolio mix or non-comparable cost boundaries; residual attributed to **improvable** operating discipline (for example power usage effectiveness and procurement) where operational inputs exist. A distribution-per-unit gap may be decomposed through occupancy and same-store NOI, operating cost intensity, and financing structure—each sub-metric tied to catalog keys or declared inputs.

Improvement (platform). Improvable slices become opportunities with confidence scores; practices are assessed for fit before transfer, consistent with theory-aware benchmarking (visible results depend on underlying systems, not copy-paste KPIs).

Future use cases (same lifecycle)

- Widen REIT compare to a **peer set** with frontier efficiency methods at platform scale.
- Insurance LOB segments (`property_casualty`, `life`, …) with official industry codes and peer cohorts per org lens.
- Aviation network carriers on CASM, RASM, and load factor versus `aviation_network_carrier` profile peers.
- Multimodal transportation orgs on `road_freight` versus `maritime_shipping` profiles without mixing modal definitions.

Optional future core enhancements (not required for the lifecycle doc): export calc dependency graphs in `manifest.json`; a neutral `benchmark-gap-v1` contract if gap tables should be governed in git.

Related

- work-cycles.md — hub for benchmarking, assurance, and disclosure cycles
- governed-data.md — catalog vs contracts; analysis lens and tenancy
- catalog-consumption.md — manifest, segment filtering, metadata conventions
- CORE_VS_PLATFORM.md — what is not in OSS
- crosswalk.md — metric to Gold product links
- AGENTS.md — one analysis lens per agent run
- catalog/README.md — benchmarks.yaml and terminology

Canonical source: `ambient-core` at `docs/benchmarking-lifecycle.md` (sync ref `924039e`).

---

open-core/catalog-consumption.md

Catalog consumption (integrators)

Authoring and generator commands live in catalog/README.md. This page explains **how to read** catalog output in apps, jobs, and agents. For contracts and shared env vars, see governed-data.md. When **authoring** YAML, use catalogue keys and ids per CONVENTIONS.md.

When to use what

- **List or resolve KPI definitions** — generated **manifest** via `ambient_contracts.catalog_manifest.load_manifest()` → `catalog/manifest.json`. The loader parses JSON once and returns a typed **`CatalogManifest`** (`ambient_contracts.manifest_models`); use `manifest.metrics`, `manifest.resolve_metric(id)`, or `CatalogMetric.to_tool_dict()` when you need the same camelCase objects as the raw file.
- **Agent tools `catalog_list_metrics` / `catalog_resolve_metric`** — same manifest; see lib/ambient_cli/tools.py
- **Upload column mapping, data-option templates, bronze field rules** — industry YAML under `catalog/industries/<pack>/` via `ambient_pipeline.catalog_loader.load_data_option()` (falls back to `manifest.json` for expanded shared options)
- **Front-end industry picker, metric browser, enrichment** — generated JS in `catalog/runtime/*.js` (e.g. `catalogIndustries.js`)
- **Link a catalog metric to a governed Gold contract** — catalog/crosswalk.yaml; see crosswalk.md

The manifest is the **stable JSON export** for machines; industry YAML is the **editable source** for vertical-specific metrics and `dataOptions`. Generated *`catalog/runtime/`*`.js`** mirrors the same semantics for front-end bundlers — see CONVENTIONS.md. After YAML edits, run `ambient-catalog-generate` so `manifest.json` and `runtime/` stay in sync.

Path resolution

Set `AMBIENT_CATALOG_DIR` to a directory that contains `manifest.json`, or run from a full ambient-core checkout. `AMBIENT_CORE_ROOT` also enables discovery via lib/ambient_contracts/paths.py.

Wheel-only pip installs include bundled **contracts**, not the full catalog tree — submodule or `AMBIENT_CATALOG_DIR` is required for manifest-driven features.

Manifest glossary

Top-level keys in catalog/manifest.json:

- **`version`** — manifest format version
- **`industries`** — registered verticals (`value`, `label`, `pack` filename)
- **`metrics`** — expanded metric records (one row per vertical metric)

Common fields on each **metric** object:

- **`id`** — numeric metric id (stringified when passed to `catalog_resolve_metric`)
- **`catalogMetricKey`** — stable string key from the generator (legacy keys may appear until retired; new YAML uses `industry.segment.slug` per CONVENTIONS.md)
- **`name`** — display name
- **`industry`** — vertical label (e.g. `Real Estate`)
- **`methodology`** — human-readable definition
- **`type`** — e.g. `Financial`, `Operational`
- **`unit`** — display unit
- **`frequency`** — calculation / reporting cadence for the KPI (for example `monthly`, `quarterly`)
- **`requiredSources`** — list of data options with `catalogOptionKey`, `name`, `fields` (field names)
- **`calc`** — the machine-readable formula (`expr` + declared `inputs`), or `null` for directly-reported metrics
- **`directlyReported`** — `true` when the metric value is uploaded directly (`input: true`, no `calc`)
- **`inputs`** — the input-coverage recipe (see below)
- **`inputCoverage`** — one-word summary of how feedable the metric is (see below)

Use `id` for agent resolve and OLAP joins; use `catalogMetricKey` for crosswalk entries and stable keys across regenerations when ids are reassigned only through controlled generator changes.

From data options to KPIs (input coverage)

Definitions tell you *what* a metric is; the **input-coverage recipe** tells you *what to upload so `calc` actually runs*. For every metric the generator resolves each required input to the data option(s) that supply it and records the result on the metric object.

Each entry in **`inputs`** has:

- **`name`** — the input variable used by `calc.expr` (or the metric's own slug for directly-reported metrics)
- **`kind`** — `measured` (a value you supply) or `derived` (computed from another catalog metric)
- **`covered`** — whether a linked data option can supply it
- **`satisfiedBy`** — the supplying data option(s), each with `catalogOptionKey`, `name`, and a `via`:

- **`field`** — the input is an explicit, mappable column on that data option (map this column)

- **`upload`** — the input comes from that option's document/template contents rather than an enumerated field (upload that document). Accounting aggregates and ratios are intentionally not enumerated as fields — see catalog/input_field_policy.yaml.

The metric-level **`inputCoverage`** summarizes feedability:

- **`complete`** — every measured input is an explicit mappable field
- **`upload`** — feedable, but at least one input comes from a linked upload's document/template
- **`none`** — no data option references this metric; it cannot be fed as currently wired
- **`derived`** — computed only from other metrics (nothing to upload directly)
- **`unspecified`** — the metric declares neither `calc` nor `input`

Operator flow, then, is: pick the metric, read its `inputs`, upload each `satisfiedBy` data option (mapping `via: field` columns and attaching `via: upload` documents), run the pipeline, and the KPI computes. Run scripts/check_metric_inputs.py for a per-industry coverage report (which metrics are `complete` / `upload` / `none`); it is informational and does not block CI. The reverse direction (**which metrics a data option feeds**) is on each `dataOptions[].metricIds`.

Data options: typed fields and two frequencies

Each **`dataOptions[]`** row in manifest version **3** includes:

- **`fields`** — objects `{ name, type, … }` where `type` is a catalog enum (`decimal`, `date`, `string`, …). Optional **`unit`** (for example `USD`, `seat-km`) and **`description`** (authoring hint for and integrators). YAML may still list legacy string field names; the generator normalizes them at export time. **`dummyFields` is retired** — typed `fields` is the only schema surface for uploads and Bronze mapping.
- **`fieldCoverage`** — `upload` (document/financial statement path; measured inputs via upload policy) or `enumerated` (column-mapped exports; strict validation applies). Inferred at generate time when omitted; persisted by scripts/harden_catalog_data_options.py.
- **`collectionFrequency`** — how often the org should refresh or upload data for this template (for example `monthly`). Same concept as *source frequency* or *required upload cadence*; the YAML key is `collectionFrequency` for manifest stability.
- **`grain`** — row-level expectation for Bronze tables (`transaction`, `day`, `week`, `month`, `quarter`).

Metric **`frequency`** is the **calculation / reporting** cadence. Collection can be finer than calculation (daily feeds, monthly KPIs); the generator warns when `collectionFrequency` is coarser than the strictest linked metric `frequency`.

Generator **measured-input** checks default to **warnings** on legacy packs. **`fieldCoverage: enumerated`** options are validated strictly when you pass `--strict-data-option-inputs` (CI uses this after hardening). Run `python scripts/harden_catalog_data_options.py --write` across packs before enabling strict locally.

Bronze coercion reads manifest field types when mapping uploads — see pipeline.md. Refresh the gap report with scripts/audit_data_option_fields.py (`--markdown --report docs/catalog-input-field-gaps.md`).

JavaScript runtime vs JSON only

- **JSON / Python backends** — `manifest.json` or `load_manifest()` (typed `CatalogManifest`) is enough for list/resolve and server-side mapping.
- **Browser or Node UI** — import generated modules from `catalog/runtime/` so industry resolution and metric keys match the generator (single bridge module in your app re-exporting from the pinned core submodule is a common pattern). Prefer JSON manifest for APIs; prefer generated `.js` when the app already bundles ES modules from core.

Hand-edit only `catalog/runtime/catalogEnrichment.js` when documented in catalog README; other `runtime/` files are generator output.

Analysis lens and metric filtering

The core agent tool **`catalog_list_metrics`** filters manifest rows by **`industry`** (the pack display label, for example `Banking`), not by metric `segment`. For a business-line tenant org, your platform should further filter the manifest (or industry YAML) by **segment** and by **`industryCodes`** from each pack's `pack.yaml` / manifest industry row so users only see KPIs for that org's lens. Peer comparison should use the same official code cohort, not the legal entity's marketing name. See catalog-industry-coverage.md.

Recommended `AgentRunContext.metadata` keys (platform convention)

Core does not validate these; document them in your worker so synthesis and logging stay consistent:

- **`org_id`** — tenant organization (required for production attribution).
- **`catalog_industry`** — resolved pack label for this org's lens.
- **`industry_code_system`** — optional: `isic` | `naics` | `nace` | `gics` (which taxonomy the tenant treats as primary for cohorting).
- **`industry_code_primary`** — optional primary code from the pack's manifest `industryCodes` block for that system.
- **`peer_group_id`** — platform-defined cohort for benchmarking.
- **`assurance_framework`** or **`control_pack_id`** — assurance scope.
- **`disclosure_mandate_id`** — investor or exchange mandate set.
- **`investor_audience`** — coarse audience (listed, LP, lender, internal).
- **`plan_version_id`** — budget or forecast version for variance.
- **`scenario_id`** — plan scenario label.
- **`reporting_group_id`** — optional link to a legal or group holding company for rollup UI only; does not change catalog resolution in core.

See governed-data.md, work-cycles.md, and catalog/README.md.

Related

- crosswalk.md — metric → contract links
- CONVENTIONS.md — catalogue keys and formats
- pipeline.md — bronze mapping using catalog loaders
- INTEGRATING.md — pin and env vars in monorepos

Canonical source: `ambient-core` at `docs/catalog-consumption.md` (sync ref `924039e`).

---

open-core/core-vs-platform.md

Foundation vs full product

What you get from **this repository alone** versus what belongs in a **separate application repository** (UI, deploy, tenancy, operator secrets). Ecosystem context: ECOSYSTEM.md.

With only this repository

- Validate and load **data-product contracts** (`validate-contracts`, `ambient_contracts`).
- Author and generate the **reference catalog** (`ambient-catalog-generate`, `catalog/manifest.json`, `catalog/runtime/` JS).
- Run the **free Colab / pandas** path (Colab entry, `ambient_pipeline/colab_smoke.py`) with samples from the website zip (ambient-core-demo-samples-v1.zip) — **storage = none** in this repo (`data/raw/.gitkeep` only). Drive remains a runnable SSOT for development. This pack is not for scale testing; enterprise deployments run scalable tests separately.
- Optionally use **Spark/Delta** helpers under `lib/ambient_pipeline/` when you install `.[pipeline]` and provide the same external CSVs.
- Fork or integrate contracts and catalog semantics without shipping a full SaaS stack.

Not in this repository (typical downstream / commercial platform)

- **Consumption UI** (web or mobile clients), API gateways tied to a specific vendor OLTP.
- **Managed lakehouse deploy** (Databricks Asset Bundles, scheduled workspace jobs, Unity Catalog materialization).
- **App-specific pipeline glue** (Firestore sync, tenant sync IDs, cloud storage conventions)—may live beside imported `ambient_pipeline` in the consumer repo.

- **Customer-facing AI / inference / model hosting** — not shipped from this repository. Future proprietary AI/ML, if any, belongs only in the commercial platform application.
- **Business-line organizations** — splitting a conglomerate into multiple tenant orgs, each with its own catalog industry, contracts, uploads, and peer set; optional holding-company rollup is platform analytics, not a second catalog industry in core.
- **Benchmark gap decomposition and improvement workflows** — peer actuals storage, structural versus improvable waterfall/bridge UI, normalization policy, and post-benchmark action tracking on top of core metrics and contracts; see benchmarking-lifecycle.md.
- **Assurance and attestation workflows** — control packs, evidence sign-off, DQ and bridge reconciliation UI, external reviewer exports; see assurance-lifecycle.md.
- **Investor disclosure and fundraising workflows** — mandate mapping, gap-to-requirement, data-room and remediation narratives; see investor-disclosure-lifecycle.md.
- **Planning and variance workflows** — plan versions, scenarios, and variance bridges vs actuals; see planning-variance-lifecycle.md.
- Operator secrets, billing UI, and end-to-end tests against a live multi-tenant deployment.

Source of truth

- **Contract YAML** — `contracts/` here; bundled under `lib/ambient_contracts/bundled/` for wheels.
- **Catalog YAML** — `catalog/` here; generated JSON manifest and JS via `ambient-catalog-generate`.
- **Plain text first** — definitions and semantics live in git as YAML (and generated JSON); binary uploads and live databases are precursors or forward stores, not a second SSOT (CONVENTIONS.md).
- Consumers should **not** maintain a second editable copy of `contracts/` or `catalog/`; pin a tag and use submodule checkout or `AMBIENT_CORE_ROOT` (INTEGRATING.md).
- **Platform deployments** wire precursor OLTP/Bronze and forward lakehouse **instances**; core still owns the YAML definitions those stores implement.

`commercial-usage-v1.yaml`

A **data contract** describing metering and usage shapes for BI or finance exports—not a statement that this OSS repo is a commercial product SKU.

Read next

- ECOSYSTEM.md — components and priorities
- USAGE.md — quick start

- INTEGRATING.md — pin a release in another repo
- CONTRIBUTING.md — releases and consumer follow-up
- CONVENTIONS.md — naming, formats, and storage

Canonical source: `ambient-core` at `docs/CORE_VS_PLATFORM.md` (sync ref `924039e`).

---

open-core/examples-pipeline.md

Pipeline examples

minimal_governed_data.py

Validates all contracts, loads the catalog manifest, and prints a sample metric plus the tenant-metrics contract product name and table. No Spark required.

```
cd ambient-core
pip install -e .
# Optional but recommended when core is a submodule:
# export AMBIENT_CORE_ROOT="$PWD"
python examples/pipeline/minimal_governed_data.py
```

For bronze mapping, provenance, and Spark validation, see docs/pipeline.md, docs/governed-data.md, contracts/README.md, and docs/CONVENTIONS.md. Tests live under `tests/pipeline/`.

Canonical source: `ambient-core` at `examples/pipeline/README.md` (sync ref `924039e`).

---

open-core/governed-data.md

Governed data: catalog and contracts

This guide answers how **catalog** and **contracts** are used in practice and how to build on top of them without duplicating SSOT in your application repository.

Naming, integer ids, serialization formats, and where databases sit relative to git are defined in CONVENTIONS.md. This page focuses on consumption and the end-to-end data flow.

For catalog YAML authoring, see catalog/README.md. For contract file layout in this repo, see contracts/README.md.

Plain text SSOT and data flow

Ambient Core owns the **middle** layer: human-authored **YAML** for contracts and catalog semantics, and machine-generated **JSON** (`catalog/manifest.json`, JSON Schemas, bronze

mapping payloads) and **JavaScript** (`catalog/runtime/*.js` from `ambient-catalog-generate`). Do not maintain a second editable copy of those trees in a consumer repo — pin a tagged checkout or submodule (INTEGRATING.md, CANONICAL_SCOPE.md).

Operational databases and lakehouse Bronze are **precursors**: data is extracted to plain text (typically **CSV/TSV** or **JSON**) at the upload boundary before governance runs. Silver → Gold and similar **forward** stores are Parquet/Delta or service SQL in deployment; their shapes are still defined by YAML in `contracts/`. Details: CONVENTIONS.md — Data formats and storage.

```
flowchart LR
  precursor[Precursor_OLTP_or_Bronze]
  extract[Plain_text_extract_CSV_or_JSON]
  yaml[YAML_contracts_and_catalog_in_git]
  forward[Forward_Parquet_Delta]
  precursor --> extract --> yaml --> forward
```

Medallion job steps that implement this path are in pipeline.md.

Two layers

- **Catalog** — reference KPIs, industries, data-source templates, benchmarks (semantic intent, not physical tables). SSOT: authored **YAML** under `catalog/` → generated **`manifest.json`** (JSON) and `catalog/runtime/*.js`.
- **Contracts** — governed **data-product** interfaces: schema, lineage, quality, consumption rules. SSOT: **YAML** in `contracts/` (bundled in wheels as `ambient_contracts.bundled`).

Catalog metrics do **not** replace contracts. A metric may exist in the catalog long before a Gold product is defined. Optional links are recorded in crosswalk.yaml — see crosswalk.md.

Catalog fields → contracts

Manifest **v3** typed `fields` and data-option `fieldCoverage` govern **what uploads may contain** and how Bronze mapping coerces columns. **Contracts** govern **what may be written** to Silver and Gold: required columns, bronze lineage, and product governance. Typical path: catalog-mapped Bronze → tenant-metrics-v1.yaml (Silver) → vertical Gold products (`finance-*-v1`, healthcare, life sciences). Diagram and product groupings: contracts/README.md — How catalog and contracts connect.

Analysis lens and multi-org tenancy

Catalog industry and segment describe an **analysis lens** (which KPIs and peer comparisons apply), not the legal entity's self-classification. **ambient-core** supplies the taxonomy: industry packs in catalog/packs.yaml with `industryCodes` on each pack, metrics, benchmarks, and contracts. A **multi-tenant platform** (not in this repo) supplies `org_id`, entitlements, which catalog pack and Gold contract bind to each org, and peer cohort membership.

A single banking group on a paid platform is often modeled as **several tenant organizations**—for example owned branch real estate (Real Estate lens), consumer depository banking (Banking lens), C&I and commercial CRE lending (Commercial Finance lens), residential mortgage lender book (Consumer Finance `residential_mortgage` segment), unsecured cards (Consumer Finance `consumer_lending`), pooled or advised fund books (**Funds** lens), listed REIT or trust **vehicle** reporting (**Trusts** lens), and investment banking or fintech SaaS (Financial Services lens). Each org is compared to peers in **that** lens. Consolidated equity may move because real estate is revalued; that driver should be analyzed with real estate KPIs on the RE org, not by stretching banking metrics across the whole group. Do **not** use a legacy “REITs” industry class or `reits` segment—vehicle FFO and payout belong on **Trusts**; property NOI and vacancy belong on **Real Estate**.

Risk and payments are not separate catalog industries or standalone `finance-risk-v1` / payments contracts. Credit, liquidity, market, and payment-volume metrics live on the existing pack and matching `finance-*-v1` Gold product for that economic engine (for example NPL on Commercial Finance, VaR on investment banking, card volume on Consumer Finance). Institution-wide liquidity and stress metrics stay on Banking. Macro peer share and HHI across the market belong in platform benchmarking, not per-org Gold schema.

Upload fields such as `entity_segment` on some catalog data options support **segmented extracts** within a lens; they do not assert that the tenant org “is” a bank or insurer in the legal sense. Core does **not** validate tenancy or lens choice. Platform boundaries: CORE_VS_PLATFORM.md. Terminology: catalog/README.md.

For **product work cycles** on the same governed metrics—**benchmarking**, **assurance**, **investor disclosure**, and **planning and variance**—core supplies definitions and contracts; the paid platform supplies workflows and UI. See work-cycles.md and the child lifecycle docs linked from that hub.

Path resolution

Python and CLIs resolve paths via lib/ambient_contracts/paths.py:

- **`AMBIENT_CORE_ROOT`** — repo root (submodule or clone)
- **`AMBIENT_CONTRACTS_DIR`** — override contracts directory (default: `contracts/` under root, or wheel bundle)
- **`AMBIENT_CATALOG_DIR`** — catalog tree containing `manifest.json`

Wheel-only installs ship bundled contracts; catalog tools and `load_manifest()` need a full checkout or `AMBIENT_CATALOG_DIR` pointing at a tree with `manifest.json`.

Set these in CI before `validate-contracts` and `ambient-catalog-generate --check` when core is a submodule — see INTEGRATING.md.

How each consumer uses the same SSOT

```
flowchart TB
  subgraph ssot [ambient-core SSOT]
    C[contracts/*.yaml]
    M[manifest.json]
  end
  UI[Web app / JS]
  Jobs[Spark / lakehouse jobs]
  UI --> M
  UI --> runtime[catalog/runtime/*.js]
  Jobs --> C
  Jobs --> M
```

UI and planning (JavaScript or JSON)

- Import generated modules from `catalog/runtime/` (industries, metrics, enrichment) or read `catalog/manifest.json` directly.
- Use catalog for **labels, methodology, required sources** — not for asserting warehouse table shapes.
- Field-level manifest guidance: catalog-consumption.md.

Pipelines and notebooks (Python)

- **Contracts:** `ContractLoader` from `ambient_contracts` — load YAML, `assert_required_columns()` on Spark DataFrames, `enforce_bronze_lineage()` before Silver/Gold writes.
- **Catalog:** `load_manifest()` for metric lists; `ambient_pipeline.catalog_loader.load_data_option()` for upload-mapping rules tied to industry YAML; manifest v3 field types feed `coerce_mapped_columns()` in bronze mapping (pipeline.md).
- **Governance helpers** (git checkout, not full wheel): `ambient_pipeline` — provenance stamping, PII pseudonymization, bronze→tenant-metrics mapping. See pipeline.md.

Default execution is local Spark/Delta via `notebooks/` or `scripts/run_oss_bronze_silver_smoke.py`. Managed schedules, Databricks jobs, and Firestore sync live in **your application repository**; core supplies contracts, catalog semantics, reusable helpers, and the OSS demo.

CI gates

From a core checkout:

```
validate-contracts
python scripts/harden_catalog_data_options.py --check
python scripts/generate_reference_catalog.py --check --strict-data-option-inputs
```

Data-product inventory

Silver tenant snapshots (tenant-metrics-v1), **Gold** vertical rollups (`finance-*-v1`, healthcare, life sciences, org-kpi), and **platform** products (quality, operational–financial bridge, commercial usage, pipeline observability) are defined as YAML under `contracts/`.

Full grouped inventory with scope notes: contracts/README.md — Current data products. Open each YAML for table names, required columns, and consumption rules. `validate-contracts` checks structural keys (`product`, `schema`, `lineage`, `governance`); deeper semantics are enforced in pipeline code and tests.

Crosswalk

See crosswalk.md for field definitions, the in-repo DSCR example, and maintainer workflow.

Build on top (checklist)

1. Pin a tagged `ambient-core` release — INTEGRATING.md.
2. Set `AMBIENT_CORE_ROOT` / contract and catalog dirs in CI.
3. Gate merges with `validate-contracts`, catalog hardening `--check`, and `ambient-catalog-generate --check` (with strict enumerated-field validation in CI).
4. Jobs: load contracts and catalog rules in Python; do not copy YAML into app-only trees.
5. UI: consume `manifest.json` or `catalog/runtime/` from the pinned checkout.

Try the minimal Python walkthrough: examples/pipeline/minimal_governed_data.py.

Anti-patterns

- Maintaining a second editable `contracts/` or generated `manifest.json` outside the pinned core checkout.
- Encoding KPI definitions only in ad hoc prompts instead of catalog + contracts.
- Using `ContractLoader` with user-supplied paths without basename checks in **your** API layer.

Related

- catalog-consumption.md — manifest vs YAML vs JS runtime
- crosswalk.md — metric → contract links
- USAGE.md — install recipes
- pipeline.md — bronze → contract flow with `ambient_pipeline`
- CONVENTIONS.md — catalogue keys, contract versions, formats and storage
- CANONICAL_SCOPE.md — what must change only here
- CORE_VS_PLATFORM.md — foundation vs full product repo

Canonical source: `ambient-core` at `docs/governed-data.md` (sync ref `924039e`).

---

open-core/integrating.md

Integrating a tagged release

Use this when you embed **ambient-core** in a monorepo, fork, or product application. There is **one** canonical tree: a tagged release of **this** repository. All updates to contracts, catalog, and shared pipeline code happen here first.

Scope: CANONICAL_SCOPE.md.

1. Pick a version

Choose a release tag `vX.Y.Z` from this repo's **Releases** (or tags). Record the same tag everywhere you depend on core (pip, submodule, container build args).

2. Python install (required for CLIs and libraries)

```
pip install "ambient-core[pipeline,postgres] @ git+<repository-url>@vX.Y.Z"
```

Replace `<repository-url>` with the HTTPS git URL of **this** repository (or your fork if you maintain one). Extras: `pipeline`, `postgres`, `dev`, `all` — see USAGE.md.

Editable install for core development:

```
git clone <repository-url>
cd ambient-core
pip install -e ".[all]"
```

3. Full tree (catalog JS, on-disk contracts, pipeline tests)

For repos that import `catalog/runtime/*.js` or validate against checkout paths:

```
git submodule add <repository-url> ambient-core
cd ambient-core
git fetch --tags
git checkout vX.Y.Z   # detached HEAD at release tag
cd ..
git add ambient-core
```

Pin the submodule SHA in the parent commit; avoid tracking `branch = main` if you need reproducible builds.

4. Environment variables (monorepo)

When the parent repo is not the core repo root:

```
export AMBIENT_CORE_ROOT="$PWD/ambient-core"
export AMBIENT_CONTRACTS_DIR="$PWD/ambient-core/contracts"
export AMBIENT_CATALOG_DIR="$PWD/ambient-core/catalog"
```

Set these in CI before `validate-contracts` and `ambient-catalog-generate --check` so validation uses the submodule tree, not only wheel bundled data.

5. Verify after import or upgrade

From the core checkout (submodule or clone):

```
validate-contracts
ambient-catalog-generate --check
pytest   # optional; Spark tests need Java and AMBIENT_SPARK_TESTS=1
```

Run your application repository's CI as well.

6. Upgrade loop

1. Merge changes in **this** repository; tag `vX.Y.Z` on `main`.
2. In each consumer repo:

- Bump `ambient-core @ git+…@vX.Y.Z` in dependency files.
- `cd ambient-core && git fetch --tags && git checkout vX.Y.Z`.
- Set `AMBIENT_CORE_ROOT` in CI if used.

1. Regenerate catalog if YAML changed: `ambient-catalog-generate` (or a thin wrapper script in the consumer).
2. Run consumer CI; merge.

Appendix A — Consumer checklist (after a core release)

- [ ] Dependency pin — `ambient-core[…] @ git+…@vX.Y.Z` matches the new tag.
- [ ] Submodule — `ambient-core/` checked out at the **same** tag SHA; parent commit records the pointer.
- [ ] CI — `AMBIENT_CORE_ROOT` (and catalog/contracts dirs) set before contract/catalog validation.
- [ ] No mirrors — no `contracts/*.yaml` or `catalog/manifest.json` outside the `ambient-core/` checkout (stub README pointers are fine).
- [ ] Authoring rules — read CONVENTIONS.md before changing catalog keys, contract major versions, or ingestion formats; do not use binary spreadsheets or PDFs as SSOT in the consumer repo (extract to CSV/JSON/YAML at the boundary).
- [ ] Catalog JS — import from `ambient-core/catalog/runtime/` (single bridge module in the app is a common pattern).
- [ ] Notebooks and jobs — resolve contracts under `ambient-core/contracts/`.

- [ ] Pipeline — shared modules from `ambient-core/lib`; app-only glue stays in the consumer repo.

Follow-up details: CONTRIBUTING.md.

Product work cycles (platform)

If you ship workflows on governed KPIs, pin ambient-core for catalog metrics, per-pack `benchmarks.yaml`, industry packs (`industryCodes`), `ambient_calc`, and contracts (`quality-v1`, `operational-financial-bridge-v1`, and others). Implement tenancy, UI, and cycle-specific logic in your application repository:

- **Benchmarking** — peer actuals, normalization, waterfalls, improvement; benchmarking-lifecycle.md.
- **Assurance** — control packs, evidence, DQ and bridge reconciliation, attestation exports; assurance-lifecycle.md.
- **Investor disclosure / fundraising** — mandate mapping, gap-to-requirement, data room; investor-disclosure-lifecycle.md.
- **Planning and variance** — budgets, scenarios, variance bridges; planning-variance-lifecycle.md.

Overview and metadata conventions: work-cycles.md.

Canonical source: `ambient-core` at `docs/INTEGRATING.md` (sync ref `924039e`).

---

open-core/investor-disclosure-lifecycle.md

Investor disclosure and fundraising lifecycle: core vs platform

Investor disclosure (and related **fundraising** readiness) is the work cycle that asks whether an organization meets **stakeholder mandates** for reporting—so capital remains available from listed markets, limited partners, sponsors, or lenders with climate or sustainability expectations. It is not the same as benchmarking (peer performance) or assurance (defensibility of numbers)—but published metrics should be traceable through assurance, and operational definitions should match those used in benchmarks.

ambient-core supplies **metric vocabulary** and contract shapes; the **paid platform** maps mandates, computes gap-to-requirement, and supports remediation narratives and data rooms.

Index of all cycles: work-cycles.md.

This document is **not legal or securities advice**; it describes product architecture only.

End-to-end flow

```
flowchart LR
  subgraph core [ambient_core_OSS]
    Catalog[catalog_RE_and_REIT_metrics]
    EnergyOpts[energy_data_options]
    OpportunityShape[opportunity_v1]
  end
  subgraph platform [paid_platform]
    Mandates[disclosure_mandate_mapping]
    Boundary[GHG_and_scope_policy]
    ReqGap[gap_vs_requirement]
    DataRoom[fundraising_and_data_room]
  end
  Catalog --> Mandates
  EnergyOpts --> Boundary
  Mandates --> ReqGap
  ReqGap --> DataRoom
  OpportunityShape --> DataRoom
```

Phase mapping

1. Audience and mandate mapping

- **Core** — Analysis lens per org (governed-data.md); industry packs and benchmark guardrails for comparison **design**, not legal obligation lists.
- **Platform** — `disclosure_mandate_id`, `investor_audience` (listed exchange, LP, sponsor, lender); checklist per jurisdiction or fund policy (for example exchange sustainability reporting expectations for listed real estate, portfolio-level carbon disclosure demanded by institutional LPs). Map mandates to catalog metric ids and contract products.

2. Metric pack selection

- **Core** — Real Estate operating metrics (NOI, cap rate, occupancy where defined); listed REIT investor metrics under `financial_services.reits.*` in financial_services.yaml; energy consumption fields on Real Estate data options (`energy_consumed`, `energy_cost`). Bridge rules link energy signals to financial impact narratives.
- **Platform** — Curated “investor pack” per mandate; effectiveness metrics (growth, distributions) versus efficiency and environmental intensity grouped for disclosure schedules.

Catalog gap (v1): There is no dedicated `sustainability` segment or native GHG metrics in core yet. Intensity-style environmental KPIs (for example per megawatt, PUE) are expected to be platform-normalized or added in a future catalog band—not invented in prompts.

3. Boundaries and harmonization

- **Core** — Documented methodologies and `calc` boundaries per metric; intensity metrics where authored in catalog.
- **Platform** — GHG scope boundaries, tenant versus landlord energy, currency and period alignment, restatement and versioned disclosure files.

4. Gap versus requirement

- **Core** — Per-industry `catalog/industries/<pack>/benchmarks.yaml` healthy bands are **planning guardrails**, not exchange or LP thresholds.
- **Platform** — Required versus actual versus prior period; optional peer percentile for **narrative** only when mandate references market practice. Success criterion is **compliance or mandate gap**, not pace-setter rank.

5. Remediation, fundraising, and data room

- **Core** — `fpaWorkflow` hints on metrics guide where remediation recommendations map.
- **Platform** — CapEx and retrofit roadmaps, sustainability action owners, data-room exports, equity or debt storytelling tied to catalog metric keys.

Overlap: benchmark cycle vs disclosure cycle on the same org

Run **benchmarking** when the user needs peer context and improvable decomposition (for example NOI margin versus another data-centre REIT). Run **disclosure** when the user must satisfy a **named mandate** (exchange, fund, lender) even if peers are strong or weak. The same tenant org may run both in one quarter; use separate metadata (`peer_group_id` vs `disclosure_mandate_id`) and do not treat mandate compliance as "winning" a benchmark.

Listed issuers may face **regulatory or exchange** sustainability disclosure pressure; **private** capital may impose parallel portfolio-level reporting on the same assets through LP agreements. Platform modeling as separate mandates on the same or sibling orgs under one `reporting_group_id` matches analysis lens patterns.

Illustrative data-centre REIT sketch

Setup. A listed data-centre REIT org under Real Estate / REIT investor lenses; a related mandate from an LP portfolio policy on energy and carbon transparency.

Definitions (core). Occupancy and NOI margin semantics from Real Estate pack; FFO and payout from `financial_services.reits.*`; energy uploads from Real Estate data options where available.

Mandates (platform). Exchange-style sustainability schedule plus fund policy fields not fully enumerated in core.

Gap (platform). Missing intensity metrics or undisclosed PUE where mandate requires them—gap to **requirement**, not to a peer REIT.

Remediation (platform). Efficiency and procurement initiatives tracked in the platform. Empirical filings and figures remain in external research or issuer disclosures.

Related

- work-cycles.md
- benchmarking-lifecycle.md
- assurance-lifecycle.md
- governed-data.md
- catalog-consumption.md

Canonical source: `ambient-core` at `docs/investor-disclosure-lifecycle.md` (sync ref `924039e`).

---

open-core/overview.md

Integrator overview

Docs for embedding ambient-core (MIT): contracts, catalog metrics and data options, and pipeline helpers. Product Command Centre how-to lives under **Operator**.

In this section

- Using ambient-core - install, extras, and quick recipes (start here for a clone or pin).
- Integrating a tagged release - pin a core tag in another repository.
- Governed data: catalog and contracts - how catalog YAML and contracts relate.
- Pipeline governance helpers - `ambient_pipeline` helpers and Silver/Gold order.
- Catalog consumption - load packs and map data options in code.
- Pipeline examples - worked examples.

Reference catalog

- Reference catalog - all eighteen industry packs (ISIC, metric and data-option counts).
- Core metrics layer - shared templates expanded into every pack.
- Industries - pack landing; open a vertical from the sidebar.

Foundation vs full product orientation: Core vs platform (under **Start Here**).

---

open-core/pipeline.md

Pipeline governance helpers

Ambient Core does **not** ship a medallion job runner or Databricks bundle. It ships **contracts**, **catalog** semantics, and Python modules under `lib/ambient_pipeline/` that your lakehouse jobs import. Schedules, notebooks, and deploy glue belong in your application repository.

For catalog vs contracts and path env vars, see governed-data.md. Product inventory and catalog linkage: contracts/README.md. For pinning core in a monorepo, see INTEGRATING.md. Bronze uploads use **CSV/TSV** at the ingestion boundary; governed Gold output is typically **Parquet/Delta** in your lakehouse — see CONVENTIONS.md.

Boundary: the **catalog** (manifest v3 and data options) defines upload field names, types, and mapping rules; **contracts** define the publishable Silver/Gold product shape that jobs must satisfy before write.

What lives where

In core: `contracts/*.yaml`; `catalog/` + `manifest.json`; `ambient_pipeline` helpers; `validate-contracts` and the catalog generator CLIs.

In your app repo: job definitions, DABs, orchestration; tenant upload UX and entitlements; Firestore sync, OLAP queries, commercial APIs; CI that calls the same CLIs.

observability-pipeline-v1.yaml describes pipeline health products and may reference SQL, bundles, or notebooks that are maintained **downstream** — treat it as the interface contract, not as code in this repo.

Install and test

```
git clone <repository-url>
cd ambient-core
pip install -e ".[pipeline,dev]"
set AMBIENT_SPARK_TESTS=1   # Windows; use export on Unix
pytest tests/pipeline/
```

Java 17+ is required for Spark tests.

Packaging note: The published wheel includes `ambient_contracts`, `ambient_cli`, and `ambient_calc`. **`ambient_pipeline` requires a git checkout** (tests use `pythonpath = lib` in `pyproject.toml`). Import it from `lib/ambient_pipeline` in notebooks and jobs pinned to the same core tag.

Typical job flow (Bronze → Silver tenant-metrics)

The steps below implement precursor → plain-text extract → YAML-governed → forward store for **Silver** `tenant-metrics-v1.yaml`. Gold vertical products are a separate step — see Silver → Gold.

```
flowchart LR
  Upload[Bronze CSV upload]
  Map[map and coerce]
  Unpivot[unpivot long metrics]
  Stamp[lineage stamp]
  Validate[SilverValidator optional]
  Contract[ContractLoader assert]
  Gold[Gold vertical tables]
  Upload --> Map
  Map --> Unpivot
  Unpivot --> Stamp
  Stamp --> Validate
  Validate --> Contract
  Contract --> Gold
```

Canonical order: map → coerce → **unpivot** → stamp → (optional validate) → contract assert → write Silver.

Anti-pattern: stamping provenance on **wide** mapped rows **before** unpivot. `unpivot_to_tenant_metrics` uses `stack()` and does not carry `_bronze_*` columns from a wide frame — stamp **after** unpivot (or call `bronze_to_tenant_metrics` which encodes this order).

1. **Resolve contract** — `ContractLoader.load("tenant-metrics-v1.yaml")`. Call `enforce_bronze_lineage()` before writes that depend on bronze provenance columns.
2. **Map uploads to catalog shape** — `apply_column_mapping`, then `coerce_mapped_columns` with manifest v3 types from `catalog_types_for_option()` (catalog_field_rules.py).
3. **Unpivot to long metrics** — `unpivot_to_tenant_metrics` turns numeric catalog fields into `name` / `value` rows with stable `metric_id`.
4. **Stamp lineage** — `stamp_lineage_columns` (or `BronzeProvenanceStamper` on the long DataFrame) sets `_bronze_org_id` from `org_id`, `_bronze_row_hash`, and mapping metadata (provenance.py).
5. **Validate Silver quality (optional)** — `SilverValidator` applies ISO 8000–style checks (validation.py).
6. **Assert schema** — `ContractLoader.assert_required_columns()` on the column set before publish.

Orchestration helper: `bronze_to_tenant_metrics(...)` runs steps 2–4 in order. See tests/pipeline/test_bronze_to_tenant_metrics_contract.py.

Platform-only (optional): `fetch_firestore_data_source` / `append_firestore_lineage_event` in `bronze_catalog_map.py` require Firebase secrets — not part of the OSS default path.

Silver → Gold (catalog, calc, and vertical contracts)

Open-Source Core **does not** ship Spark jobs that write `finance-*-v1` or healthcare Gold tables. Recommended platform pattern:

1. Read Silver long metrics from `tenant-metrics-v1` shape (or query your lakehouse Silver table).
2. Evaluate catalog formulas with `ambient_calc` (`compute_all` on manifest metrics + measured inputs).
3. Use `catalog/crosswalk.yaml` and `gold_contract_map` to relate manifest metric keys to Gold contract columns.
4. Pivot or join into the target product schema (for example `finance-banking-v1.yaml`), then `ContractLoader.assert_required_columns()` on that product before write.

Details: contracts/README.md, crosswalk.md.

Public re-exports from `ambient_pipeline`: `ContractLoader`, `BronzeProvenanceStamper`, `PiiPseudonymizer`, `SilverValidator`, plus `bronze_to_tenant_metrics`, `MappingSpec`, and mapping helpers (`__init__.py`).

Walkthrough by test module

Use these as executable recipes (run with `AMBIENT_SPARK_TESTS=1` where Spark is involved):

- test_contracts.py — load tenant-metrics contract; bronze lineage enforcement; required columns
- test_bronze_catalog_map.py — mapping JSON, CSV header validation, stable metric ids, coercion
- test_bronze_to_tenant_metrics_contract.py — map → unpivot → stamp → tenant-metrics contract assert
- test_catalog_sample_csv_bronze.py — end-to-end bronze CSV against catalog field rules
- test_catalog_manifest.py — manifest loading in pipeline context
- test_validation.py — Silver validation rules

Minimal example (no Spark)

From a core clone with `pip install -e .`:

```
set AMBIENT_CORE_ROOT=%CD%
python examples/pipeline/minimal_governed_data.py
```

See examples/pipeline/README.md.

Agents and pipelines

Core agents read **metadata** via `catalog_` *and* *`contracts_`* tools; they do not run Spark jobs. To let an agent trigger a pipeline or read live Gold, register tools in your worker — AGENTS.md.

Related

- USAGE.md — recipe 4 (pipeline pytest)
- governed-data.md — catalog + contracts consumption
- contracts/README.md — SSOT products and catalog → contract flow
- catalog-consumption.md — manifest field rules for Bronze mapping
- CONVENTIONS.md — formats, bronze boundary, forward stores
- CORE_VS_PLATFORM.md — scope split

Canonical source: `ambient-core` at `docs/pipeline.md` (sync ref `003ceb1`).

---

open-core/planning-variance-lifecycle.md

Planning and variance lifecycle: core vs platform

Planning and variance asks how **actuals** compare to the organization's own **plan**—budget, forecast, or board case—and what explains the gap. The baseline is internal intent, not a peer (benchmarking-lifecycle.md).

FP&A close on work-cycles.md is the recurring rhythm (close calendar, board packs); this cycle is the **plan-vs-actual analytical product** built on the same catalog metrics.

Index: work-cycles.md.

End-to-end flow

```
flowchart LR
  subgraph core [ambient_core_OSS]
    Catalog[fpaWorkflow_and_core_metrics]
    Calc[ambient_calc_inputs]
  end
  subgraph platform [paid_platform]
    Plans[plan_versions_and_scenarios]
    Variance[variance_bridges]
    Accountability[owners_and_commentary]
  end
  Catalog --> Plans
  Calc --> Variance
  Plans --> Variance
  Variance --> Accountability
```

Phase mapping

1. Metric and calendar alignment

- **Core** — *`.core.`* close metrics (margins, growth, working capital, cash flow); segment operational metrics per analysis lens; `fpaWorkflow` strings describing typical FP&A use on each row in catalog YAML.
- **Platform** — Fiscal calendar, plan version locking, mapping actuals period to plan period; metadata `plan_version_id`, optional `scenario_id` (base, upside, downside).

2. Plan storage and scenarios

- **Core** — No plan SSOT in ambient-core; plans are tenant data in the platform OLTP or lakehouse.
- **Platform** — Versioned budgets and forecasts at catalog metric id granularity (or mapped account trees); scenario comparison UI.

3. Variance computation and decomposition

- **Core** — `calc.expr` and `calc.inputs` define how a KPI is built from components; platform uses the same graph for variance bridges (price vs volume vs mix) where inputs exist.
- **Platform** — Variance waterfalls, commentary, accountability by department; not peer-ranked.

4. Close integration

- **Core** — Assurance and bridge products may explain why actuals moved before variance is finalized.
- **Platform** — Tie variance cycle to close status, reforecast triggers, board deck export.

Distinct from benchmarking

Benchmarking diagnoses performance versus **external** pace-setters and improvable structural gaps. Planning variance diagnoses performance versus **yesterday's plan**. The same metric (for example `operating_margin`) can run through both cycles in one quarter with different metadata (`peer_group_id` vs `plan_version_id`).

Examples

- **Any lens:** Quarterly actual `revenue_growth` and `ebitda_margin` vs annual budget using `industry.core.*` metrics after catalog expansion.
- **Aviation network carrier org:** Actual CASM/RASM vs operating plan using `aviation.network_carrier.*` definitions; variance on fuel and load factor inputs.

- **Retail org:** Same-store sales growth actual vs plan using `retail.operations.same_store_sales_growth`.

Related

- work-cycles.md
- benchmarking-lifecycle.md
- catalog/README.md
- lib/ambient_calc

Canonical source: `ambient-core` at `docs/planning-variance-lifecycle.md` (sync ref `924039e`).

---

open-core/usage.md

Using ambient-core

Quick start for contracts, catalog, and governance helpers. For embedding in another repo, see INTEGRATING.md. Foundation vs full product: CORE_VS_PLATFORM.md.

Install

From a git clone (recommended for catalog + pipeline tests):

```
git clone <repository-url>
cd ambient-core
py -3.12 -m venv .venv
# Windows: .venv\Scripts\activate
pip install -e ".[all]"
```

From a pinned git URL (downstream apps):

```
pip install "ambient-core[dev] @ git+<repository-url>@v0.3.0"
```

PyPI is **not** the documented distribution path unless publishing is added later.

Extras: `.[pipeline,dev]` (Spark tests, Java 17) · `.[postgres]` · `.[all]` (pipeline + postgres + dev)

Recipes

1. Contracts only — validate and load data-product YAML.

```
pip install -e .
validate-contracts
```

- Edit `contracts/`; sync `lib/ambient_contracts/bundled/` before release (CI enforces).
- Wheel-only installs use `ambient_contracts.bundled`; override with `AMBIENT_CONTRACTS_DIR` or `AMBIENT_CORE_ROOT` (`lib/ambient_contracts/paths.py`).

2. Catalog — author YAML and regenerate manifest/runtime.

```
pip install -e .
ambient-catalog-generate
ambient-catalog-generate --check
```

- Edit `catalog/shared/` and `catalog/industries/<pack>/` (catalog/README.md).
- Use `manifest.json` for pipelines; `catalog/runtime/` for JS apps.
- Set `AMBIENT_CATALOG_DIR` or run from a full checkout (`resolve_catalog_root()` in `paths.py`).

3. Pipeline governance (Spark) — pytest with governance modules.

```
pip install -e ".[pipeline,dev]"
set AMBIENT_SPARK_TESTS=1
pytest
```

- Needs Java 17+ for Spark tests.
- End-to-end narrative: pipeline.md.

Contributing and releases

CONTRIBUTING.md — scope, workflows, tagging, consumer follow-up.

Canonical source: `ambient-core` at `docs/USAGE.md` (sync ref `924039e`).

---

ops/crm-wix-dashboard.md

CRM mutations: use Wix Dashboard

Contact CSV import, promote-from-CSV (add contacts), and member invite are **not** available on the ambientsystems.ai members CRM (`/members/crm/`). That page is a read/update board for INTERNAL operators (left contacts list with Funnel/source/STP/sequence filters, middle funnel columns, outreach tracking via the CRM BFF).

Do these in the **Wix Dashboard** instead:

- **Contacts** — import CSV / add contacts
- **Members** — invite members; set-password / roles HITL as needed
- **Roles & permissions** — Admin vs Member access (never grant Admin from the site CRM)

Site forms (Lead / Contribute) still capture into Contacts with Funnel: Leads. See Outbound capture runbook and CRM operator SOP.

Account (Company) backfill (optional)

The members CRM resolves Account on load/save from native `company`, then `as_company_norm` / `as_account_domain`, then work-email host when STP left company empty. Saving the drawer writes the resolved value into native Company. A Dashboard pass is only needed if you want every empty Company filled without opening contacts in the CRM.

Repo mirror: `docs/ops/crm-wix-dashboard.md`.

---

ops/index.md

Site ops

Operator runbooks for the **ambientsystems.ai** marketing site (Wix Headless Forms + Contacts). These pages are wiki-authored under `wiki/docs/ops/`. Matching SSOT prose also lives in the repo under `docs/` for git-facing edits.

Command Centre product guides stay under Operator → platform. Do not mix product UI how-to with site capture wiring.

In this section

- Outbound sequence SOP — tool roles, STP 142 filter, Funnel labels, ATP gate
- Outbound capture runbook — form upsert → Funnel: Leads, smoke scripts, thank-you + Make webhook HITL
- CRM — Wix Dashboard — CSV import, promote, and member invite (not on site CRM)

Related (repo, not wiki)

- CRM operator SOP — INTERNAL `/members/crm/` + Wix-hosted CRM BFF
- Inbox and outreach — forms, UTM, outreach glue
- docs/ README — full operator file index

---

ops/outbound-capture-runbook.md

Outbound capture runbook (Wix Forms → Contacts)

Operator SSOT for wiring Lead ("Leave a note") and Contribute form submits into Wix Contacts with **Funnel: Leads**, then ATP-gated handoff to Make / lemlist / Instantly.

Related: Outbound sequence SOP, CRM operator SOP, Inbox and outreach.

Repo mirror: `docs/OUTBOUND-CAPTURE-RUNBOOK.md`.

Capture path (smallest reliable)

1. Browser submits via Headless Forms SDK (`ContactForm.astro` → `submitLeadToWixForms`).
2. Form `postSubmissionTriggers.upsertContact` creates/updates the Wix Contact (email + name mapping).
3. Upsert applies labels:

- Lead form: `website-note` + **Funnel: Leads**
- Contribute form: `website-contribute` + **Funnel: Leads**

1. INTERNAL board at `/members/crm/` shows the contact under Leads (CRM BFF).
2. Optional enrich: set **AS source channel** = `web-form` (form upsert does not write AS_* extended fields).
3. Ability to pay: operator judgment in **AS outreach notes** (and Funnel move to Qualified when cleared). Do not invent ATP from the form.

Canonical form IDs (also in `src/config/wix.ts` / `scripts/wix-site-policy.mjs`):

- Lead: `eff382b4-4245-4ce7-aee6-39c9b25db6dc`
- Contribute: `e10a9a3e-6ecb-4986-b4ef-dbf3c34fa574`

Site forms must stay **enabled**. Site env: `PUBLIC_LEAD_FORM_ENABLED=true` (never ship disabled commercial forms without an explicit operator decision). Never put `WIX_API_KEY` in `PUBLIC_*`.

Scripts (this repo)

Auth: GitHub / Cursor secret `WIX_API_KEY` (preferred), or `wix login` then CLI token. Never put the key in `PUBLIC_*` or commit it to `.env`.

```
# Idempotent: ensure Funnel + website-* labels exist; patch both forms' upsertContact
npm run wix:outbound-capture:ensure

# Dry-run
node scripts/wix-outbound-capture-ensure.mjs --dry-run

# Verify form wiring (enabled + required labels)
npm run wix:outbound-capture:smoke

# After a real test submit, confirm the contact
node scripts/wix-outbound-capture-smoke.mjs --email=you+test@example.com

# Set AS source channel = web-form when empty (dry-run default)
npm run wix:outbound-capture:enrich
node scripts/wix-outbound-capture-enrich.mjs --apply --limit=25

# Pure helper self-test (no Wix auth)
npm run test:outbound-capture
```

Form create scripts (`wix-create-lead-form.mjs`, `wix-create-opensource-form.mjs`) should leave upsertContact labels non-empty after ensure; re-run ensure after recreating a form.

Ivan HITL — when Cursor / agent lacks `WIX_API_KEY`

Agents without the secret cannot call Forms/Contacts admin APIs. **Do not** compensate by blasting many production form submits. Prefer the API smoke path below on a machine that has the key (local shell, Actions, or Cursor env secret for site `0b498e2e-d8ea-4bec-9d56-1d4b74845393`).

```
cd /path/to/ambientsystems.ai
export WIX_API_KEY='…'   # same secret as CI Deploy Wix / CRM BFF bake
npm run wix:outbound-capture:ensure
npm run wix:outbound-capture:smoke
# Expect both forms: enabled true, ok true, Funnel: Leads label present
# Optional — one controlled UI submit, then:
node scripts/wix-outbound-capture-smoke.mjs --email=you+capture-test@example.com
# Expect hasFunnelLeads: true
```

Also add `WIX_API_KEY` to the Cursor Cloud environment secrets for this repo so future agents can run ensure/smoke without HITL.

Smoke checklist (confirm Funnel: Leads)

1. Confirm home still exposes the form: `data-use-wix-forms="true"` and a non-empty `data-wix-form-id` on `https://www.ambientsystems.ai/`.
2. Run `npm run wix:outbound-capture:smoke` — both forms `ok: true`, `enabled: true`, labels include `custom.funnel-leads` (or the live Funnel: Leads key).
3. Submit a unique test email on home (Leave a note) or product Contribute. Wait until the UI shows success. **Do not spam** production forms — one controlled submit per verification.

4. Run `node scripts/wix-outbound-capture-smoke.mjs --email=<that-email>` — expect `hasFunnelLeads: true`.
5. Optional: Dashboard Contacts or INTERNAL `/members/crm/` — contact appears under **Leads**.
6. Optional: `node scripts/wix-outbound-capture-enrich.mjs --email=<that-email> --apply` — **AS source channel** becomes `web-form`.
7. ATP: leave blank / note "ATP unclear" until an operator clears ability to pay; only then move to Qualified and allow sequencer export.

Local browser submit (forms only, not label check): `npm run test:contact-form` against a running `astro` / `astro dev`.

Wix Automations HITL — thank-you + Make webhook

No `wix:outbound-capture:ensure` for thank-you mail. Automations V2 REST exists (`/automations-service/v2/automations`), but wiring a send-email thank-you needs verified sender, email template/body mapping, `SCOPE.CRM.SETUP-AUTOMATIONS`, and Validate-before-activate — none of that is in the existing form/label scripts. **Do not** add a half-baked ensure script here. Author thank-you + Make in the Wix Dashboard (**Automations**) until an operator deliberately scopes an Automations ensure. Keep scenario URLs and secrets in hub local notes (`commercial/local/`) / 1Password, never in `PUBLIC_*` or marketing copy.

Form schema `automationId` (printed by smoke) is the Forms app association — it is **not** proof a thank-you email automation exists.

Dashboard: https://manage.wix.com/dashboard/0b498e2e-d8ea-4bec-9d56-1d4b74845393/triggers

A — Thank-you email (submitter confirmation)

Do this first. It does **not** require ATP and must not enroll sequencers.

1. Wix Dashboard → **Automations** (link above) → create or edit automation for **Wix Forms** submission on Lead and/or Contribute (form IDs above). Prefer one automation per form if the UI forces a single form trigger.
2. **Trigger:** Form submitted / `wix_form_app-form_submitted` filtered to the Lead or Contribute form id.
3. **Action:** Send email (Wix Emails / verified sender domain) to the submitter.
4. **Copy:** short confirmation only (received + next step is human review). No pricing, no pilot claims, no "we will enroll you in outreach."
5. **From:** company verified sender (Settings → Emails). Confirm domain authentication before relying on deliverability.
6. **Test:** one disposable submit → inbox receive. Check spam once. Confirm no second automation also sends a duplicate thank-you.
7. **Record:** automation name + ID in hub local notes.

B — Form submission → Make (capture fan-out, HITL)

1. Create Make scenario in the **company** workspace (not a personal throwaway). Webhook module first; store URL in 1Password / hub local.
2. Wix Automations → trigger on the same form submit (or sibling automation).
3. **Action:** Webhook POST to Make. Map payload: email, form id / name, submission id, contact id if present, label keys, timestamp.
4. **Make filter (required):** continue only if contact has **Funnel: Leads** (or `website-note` / `website-contribute`). Drop honeypot / empty email / obvious spam.
5. **Downstream (default off):** log to a sheet or notify Slack/email for INTERNAL review. Do **not** auto-create lemlist/Instantly leads from Stub B until ATP is cleared.
6. **HITL gate:** operator reviews Make (or CRM board) before any sequencer action. Document who owns the review in hub local.
7. **Test:** one disposable email; confirm Make receives payload; confirm sequencer remains empty.

C — ATP gate → lemlist or Instantly

- **Trigger:** Make schedule or Wix contact label change to **Funnel: Qualified** (ATP cleared), not raw form submit.
- **Filter:** AS source / STP bucket acceptable; **AS outreach notes** show ATP cleared; cooldown empty or past.
- **Action:** Create/update lead in lemlist **or** Instantly campaign (company account only). Sequencer is not the funnel SSOT — mirror stage on Wix Funnel labels.
- **Do not:** auto-push every website-note into cold sequences.

D — Stage change fan-out (optional)

- **Trigger:** Contact label updated (Funnel:*).
- **Action:** Make webhook → update sequencer status / pause on Parked / Closed.

Operator checklist (thank-you + Make)

Use this as the close-out list after form capture smoke is green:

1. [ ] Thank-you automation exists for Lead form; verified sender; one test receive OK
2. [ ] Thank-you automation exists for Contribute form (or shared rule covers both); no duplicate sends
3. [ ] Make scenario created in company workspace; webhook URL stored in hub local / 1Password (not git)
4. [ ] Wix Automations webhook attached to form submit; payload maps email + form id + labels
5. [ ] Make filter requires Funnel: Leads (or website-* labels); spam path drops
6. [ ] Default Make path stops at notify / log — **no** lemlist/Instantly enroll until Qualified + ATP note

7. [ ] Qualified → sequencer path (Stub C) documented separately; ATP still operator judgment
8. [ ] Concrete automation + Make scenario IDs recorded only in hub local notes
9. [ ] Forms remain **enabled** on production home / product after testing

Anti-patterns

- Cloudflare Workers CRM path (retired)
- `WIX_API_KEY` in `PUBLIC_*` or browser bundles
- Auto-enrolling form leads into lemlist/Instantly without ATP
- Disabling home Lead / Contribute forms without an explicit operator decision
- Inventing weekly reply metrics or throwaway demo accounts
- Shipping thank-you copy that promises outreach enrollment or pricing

---

ops/outbound-sequence-sop.md

Outbound sequence SOP

Operator SSOT for Ambient Systems outreach on the marketing site stack. No fake weekly metrics. No invented account creation. Ability to pay before launch is the commercial filter.

Related: Outbound capture runbook, CRM operator SOP, Inbox and outreach. Hub pack: `commercial/contacts-stp/` (live A = 142 finance FP&A overlap; people_master ~7k is not the refill pool).

Repo mirror: `docs/OUTBOUND-SEQUENCE-SOP.md`.

Tool roles (do not blur)

- **Wix (Headless site + Contacts + Forms)** — Capture, CRM board, warm email/call task labels, webhook fan-out when Automations are wired. Browser never gets `WIX_API_KEY` in `PUBLIC_`. *Same-origin CRM BFF (`/api/crm/`)* is the Contacts path for INTERNAL members.
- **lemlist or Instantly + Make** — ATP-qualified cold/warm email and LinkedIn sequences only after ability-to-pay / STP qualification. Export or sync from Wix Contacts (or hub STP CSVs) into the sequencer; do not treat the sequencer as the system of record for funnel stage.
- **X (Twitter)** — Manual only. Track handle on `as_x_handle` when known; no bulk API automation required for v1.
- **Hub local** — Pre-lead staging, STP CSVs, and operator notes stay outside public git (hub `commercial/local/`, `contacts-stp/`).

Do not create throwaway Wix / lemlist / Instantly accounts for demos. Use the existing company accounts and Cursor env secrets only.

Who to work

1. **Primary:** STP A overlap (HubSpot ∩ campaign) — live **A = 142** finance FP&A contacts.
2. **Filter:** Ability to pay before launch (operator judgment + notes). Park or nurture when ATP is unclear.
3. **Do not:** Refill sequences from the ~7k people_master just to keep volume up.
4. **Pilot cadence:** Run **10–20 of the 142** at a time for STP sequencing. Expand only after reply / meeting quality is visible.

Funnel label mapping (Wix Contacts)

INTERNAL board columns use Contact labels:

- **Funnel: Leads** — In pool; not yet in active validation conversation
- **Funnel: Validation** — Discovery / validation thread live
- **Funnel: Qualified** — ATP cleared for a bounded next step
- **Funnel: Closed** — Won / customer / subscriber outcome
- **Funnel: Parked** — Drop-off, cooldown after Professional Breakup, or churned

Cadence days (1 / 3 / 4 / 6 / 8 / 10 / 14) live on **AS sequence step**, not as extra funnel columns. After breakup / exit: set **AS cooldown until** (last touch + 90 days), then re-engagement or nurture — never the same Day 1–14 without a new hook.

Admin-labeled contacts are operators, not outreach rows.

Sequence sketch (ATP-qualified only)

1. Confirm STP source / bucket and ATP notes on the Wix contact.
2. Push or sync into lemlist|Instantly (+ Make if bridging LinkedIn tasks).
3. Run Day 1–14 steps; stop on reply or meeting (`AS outcome`).
4. Mirror stage moves on the INTERNAL Funnel board.
5. X touches stay manual; log channel on **AS last channel** when used.
6. Warm email / call tasks that must stay inside Wix: use Automations or operator HITL — do not invent a second CRM.

Day-by-day copy templates (paste into lemlist / Instantly)

Draft text for ATP-qualified STP contacts only. Personalize `{{firstName}}`, `{{company}}`, and one concrete FP&A pain from research. Do not invent metrics, customers, or pricing. Stop the sequence on reply or meeting. Do not create throwaway sequencer accounts - paste into the company lemlist / Instantly workspace after HITL login.

Day 1 - Email

- **Subject:** Contract-backed Gold for FP&A at {{company}}
- **Body:**

```
{{firstName}} -

Reworking board and close numbers after the lakehouse was supposed to be authoritative is the
failure mode we fix.

Ambient ships contract-backed Gold for FP&A and ops: ODCS-shaped contracts, Unity Catalog
governance, consumption adapters. Gold does not leave without a contract. Ambient Core (MIT) is
inspectable on GitHub.

If ability to pay before launch is real for a bounded validation, reply with a work email and the
metric set you defend today. If not a fit, say so and I will close the thread.

Ivan Damnjanovic
Ambient Systems
https://www.ambientsystems.ai/
```

Day 3 - LinkedIn (connection or note)

```
{{firstName}} - short note on contract-backed Gold for FP&A (lineage + ODCS contracts, not
another dashboard). Happy to share the public Ambient Core contracts if useful. No pitch deck
required.
```

Day 4 - Email (bump)

- **Subject:** Re: Contract-backed Gold for FP&A at {{company}}
- **Body:**

```
{{firstName}} -

Quick bump. If you already have a governed Gold path you trust under audit, ignore this.

If rework after close is still eating calendar, the next step is a short validation conversation
- ability to pay before launch is the filter, not a price sheet. Ambient Core contracts stay
public either way: https://github.com/Ambient-Team/ambient-core

Ivan
```

Day 6 - Call task (Wix / operator)

- **Goal:** 5-minute ATP screen, not a demo dump.
- **Opener:** "Calling about FP&A Gold contracts - is rework after close still a live problem for {{company}}?"
- **If yes:** Confirm buyer / budget path and book a validation note (Members for approved accounts).
- **If unclear ATP:** Park / nurture. Do not push into lemlist Day 8+.

- **If no:** Professional Breakup path; set **AS cooldown until** (+90 days).
- **Log:** AS last channel = call; update Funnel label.

Day 8 - LinkedIn (value, no ask spam)

```
{{firstName}} - one concrete detail: we keep Agency / Advisory (cash, controls, TCO, audit
review) as a separate SKU from product Gold. If your need is advisory-only, say so and I will
route that instead of a platform validation.
```

Day 10 - Email (breakup foreshadow)

- **Subject:** Close the loop?
- **Body:**

```
{{firstName}} -

I will close this thread after one more note unless validation is useful.

What we offer: invitation-only review of contract-backed Gold for FP&A, scoped to ability to pay
before launch. What we do not offer: public list pricing, Canada/Serbia/memoir framing, or
dashboard-agency staffing pitches.

Reply "interested" with context, or "not now" and I will stop.

Ivan
```

Day 14 - Email (Professional Breakup)

- **Subject:** Closing the thread
- **Body:**

```
{{firstName}} -

Closing the outreach thread on my side. If FP&A Gold contracts become relevant later, leave a
note at https://www.ambientsystems.ai/contact/index.html or reply here.

Ivan Damnjanovic
Ambient Systems
```

After Day 14: set **Funnel: Parked**, **AS cooldown until** = last touch + 90 days. Never restart Day 1–14 without a new hook.

Webhooks

Prefer Wix Automations → Make (or equivalent) for form capture and stage change fan-out. Keep secrets in CI / Cursor env, never in `PUBLIC_*`. Document concrete Make scenarios in hub local notes — not in public marketing copy.

Operational capture (form → Contacts **Funnel: Leads**, thank-you + Make HITL, smoke checklist): Outbound capture runbook. Run `npm run wix:outbound-capture:ensure` then `npm run wix:outbound-capture:smoke`.

Anti-patterns

- Merging Agency / Advisory consulting into the product Gold hero
- Canada / Serbia / memoir framing on company surfaces
- Fake weekly dashboards or invented reply rates
- Cloudflare Workers CRM path (retired; Wix-hosted `dist-server` only)
- Shipping empty form IDs or disabling commercial forms without an explicit operator decision
- Putting `WIX_API_KEY` in `PUBLIC_*` or auto-pushing form leads into lemlist/Instantly before ATP

Post-deploy checks (site)

```
curl -sI -L https://www.ambientsystems.ai/agency | head -20
curl -sI -L https://www.ambientsystems.ai/agency/ | head -20
curl -sI -L https://www.ambientsystems.ai/advisory | head -20
curl -sI -L https://www.ambientsystems.ai/advisory/ | head -20
curl -sI -L https://www.ambientsystems.ai/product | head -20
curl -sI -L https://www.ambientsystems.ai/enterprise | head -20
curl -sI -L https://www.ambientsystems.ai/zh-hk/ | head -20
curl -sI -L https://www.ambientsystems.ai/docs/ops/outbound-sequence-sop/ | head -20
curl -sI -L https://www.ambientsystems.ai/docs/ops/outbound-capture-runbook/ | head -20
curl -sL https://www.ambientsystems.ai/sitemap-index.xml | head -40
curl -sL https://www.ambientsystems.ai/robots.txt | head -40
```

Expect HTML (pretty URL 302 → …`/index.html` via dist-server when CDN strip-slash applies), same pattern as `/product`. `/enterprise` and `/zh-hk` must resolve to real HTML (not JSON `not_found`). Ops wiki pages should return **200**.

If `/robots.txt` or `/sitemap.xml` still show Wix SEO Tools "pricing-plans" stubs, that is HITL in the Wix Robots.txt / Sitemap editor — code-owned discovery remains `/sitemap-index.xml` and `/sitemap-0.xml`.

---

platform/agents-in-workspace.md

Agents in the workspace

Ambient Systems does **not** ship customer-facing AI or agent panels in the workspace today. Internal engineering may use Cursor Agent / IDE for development only.

Future patented proprietary AI/ML, if any, would ship in the commercial platform application — not in MIT ambient-core, and not as resale of third-party models.

Related

- Using the platform
- Platform features

platform/ambient-core.md

Ambient Core in the platform

The Command Centre is built on a **tag-pinned** ambient-core release: contracts, reference catalog, and shared pipeline libraries. The platform adds multi-tenant OLTP, lakehouse automation, and hosted services—it does not fork or mirror core trees in duplicate.

Metric equations and pack definitions are documented only under **Integrator** (Reference catalog, Core metrics). This Operator page describes how the product consumes that pin.

What comes from open-source core

- **Catalog** — Industry packs, metrics, data options—onboarding and Data library (definitions: Reference catalog)
- **Contracts** — Governed data product schemas validated in CI and at runtime
- **Pipeline libraries** — Shared medallion and contract-loading helpers in analytical jobs

Customer-facing AI is not shipped from MIT ambient-core or as a current platform product feature.

Upgrade discipline

Category 1 operators coordinate core tag bumps with contract validation, catalog regeneration, and deploy. Integrators follow Integrating in their own repositories.

Related

- Core vs platform
- Catalog consumption
- Using the platform

---

platform/architecture.md

Using the platform

Guides for **operators and power users** with access to the Ambient Command Centre. Metric equations live under **Integrator**; this section covers product behavior.

In this section

- Using the platform - this overview (after-login map and data flow below).

- Platform features - feature index and guide status.
- Organizations and member roles
- Hosted pipeline
- Overview dashboards
- FP&A workspace
- Data hub
- Managed connectors
- Governed metrics in the app
- Reporting and exports
- Usage gates
- Bridge and optimization
- Lineage and compliance
- Agents in the workspace
- Workspace demos - screenshot walkthroughs by industry.

Open-Source Core pin in the product: Ambient Core in the platform (under **Start Here**).

What you see after login

Each **organization** (tenant) has its own governed workspace. During onboarding you choose an **industry pack** from the Ambient Core catalog - that vertical sets default KPIs, suggested **data options** (dataset templates), and FP&A workspace lenses.

Typical areas in the Command Centre:

- **Onboarding** - confirm organization profile, industry pack, and initial data connections.
- **Overview** - executive snapshot of tracked metrics and recent activity.
- **FP&A Workspace** - planning, variance, and industry-specific KPI views for your pack.
- **Data hub** - map catalog **data options** to uploads, track fulfillment status, and open source detail.

From data options to KPIs

Catalog **metrics** and **data options** are defined under **Integrator** (Reference catalog). Operator docs do not restate equations. In the app, metrics label what Overview and FP&A show; data options are the dataset templates you fulfill in Data hub.

Typical path for a derived KPI:

1. Pick the **data option** your pack expects (see Reference catalog and the pack Data options section).
2. Upload or enter data in the **Data hub** and complete field mapping for your organization.

3. The **hosted pipeline** validates against governed products and publishes **Gold** outputs for your org.
4. **Overview** and **FP&A Workspace** read those governed results (Core calc expressions run in the pipeline - not redefined here).

If a KPI stays empty, confirm the linked data option is fulfilled and pipeline status is healthy - see Data hub and Hosted pipeline.

How data flows (conceptual)

```
flowchart LR
  uploads[Your uploads and connectors]
  processing[Governed processing]
  gold[Governed Gold products]
  mirror[In-app mirror]
  ui[Command Centre]
  uploads --> processing --> gold --> mirror --> ui
```

1. **Upload or connect** - files and connector feeds are scoped to your organization.
2. **Processing** - governed jobs validate and transform data into analytical products.
3. **Gold products** - curated metrics and tables your organization is allowed to consume.
4. **In-app mirror** - low-latency views for Overview, FP&A, and Data hub.
5. **External BI** (optional) - approved sharing paths for Tableau, Power BI, or similar.

Three planes (do not conflate them)

- **Reference catalog** - KPI definitions, methodologies, calc expressions, and data options (Integrator SSOT). Touch via onboarding, Data hub, and Reference catalog.
- **Contracts** - Governed data products (schemas, lineage, SLAs). Touch via pipeline status, org data panels, assurance workflows.
- **Gold** - Computed analytical outputs. Touch via Overview, FP&A Workspace, external BI when enabled.

Changing a catalog metric definition is not the same as changing a Gold value - catalog drives templates and labels; Gold holds computed results for your org.

Open-Source Core vs this product

Integrators embed ambient-core (MIT) in their own stacks. The commercial Command Centre adds multi-tenant hosting, the medallion pipeline, and workspace features on top of the same catalog and contracts story. Comparison: Core vs platform.

---

platform/bridge-and-optimization.md

Bridge and optimization

Operational–financial bridge and **optimization** Gold products connect operational signals to financial KPIs and advanced planning scenarios. Availability depends on your industry pack and commercial entitlements—not the limited hosted evaluation tier.

Who can use it

Finance and operations leaders on full platform tiers; not included in severely scoped open evaluation.

How to

:::note[Instructions coming soon]

Bridge dashboards and optimization workflows will be documented from demos.

:::

- [ ] Confirm entitlements in organization settings when hosted limits apply
- [ ] Open bridge or optimization views when enabled
- [ ] Relate outputs to Optimization lifecycle concepts

Related

- FP&A workspace
- Using the platform
- Core vs platform

---

platform/data-hub.md

Data hub

The **Data hub** connects catalog **data options** to what your organization actually supplies: CSV uploads, manual entry, mapping, and **fulfillment status** before Gold metrics appear.

Who can use it

Members who manage operational data; admins may configure which data options are in scope for the org.

How to

:::note[Instructions coming soon]

Upload flows, field mapping, and fulfillment badges will be documented from demos.

:::

- [ ] Identify required data options for your pack (Reference catalog)
- [ ] Upload or enter data and complete mapping
- [ ] Confirm fulfillment before expecting KPI updates

Related

- Using the platform
- Hosted pipeline
- Managed connectors

---

platform/features.md

Platform features

These guides explain **how to use** hosted Command Centre capabilities. Pages marked as coming soon will be filled from product demos.

| Feature | Guide |

| --- | --- |

| Organizations and member roles | Organizations and roles |

| Hosted Bronze–Silver–Gold pipeline | Hosted pipeline |

| Overview dashboards | Overview dashboards |

| FP&A workspace | FP&A workspace |

| Data hub | Data hub |

| Managed connector setup | Managed connectors |

| Governed metrics in the app | Governed metrics in the app |

| Reporting and exports | Reporting and exports |

| Usage gates and pipeline triggers | Usage gates |

| Operational–financial bridge and optimization | Bridge and optimization |

| Lineage audit and compliance pack | Lineage and compliance |

| Agents in the workspace | Agents in the workspace |

Start with Using the platform for the end-to-end data and KPI flow.

---

platform/fpa-workspace.md

FP&A workspace

The **FP&A Workspace** extends Overview with planning-oriented views: KPI tracking, goals, variance context, and early-access forecast features for your vertical pack.

Who can use it

Workspace users in finance and operations roles; some forecast features may be gated by product tier.

How to

:::note[Instructions coming soon]

Goals, forecast panels, and pack-specific lenses will be documented from demos.

:::

- [ ] Open FP&A Workspace from the main navigation
- [ ] Align KPIs with your industry pack from onboarding
- [ ] Set or review goals where enabled

Related

- Overview dashboards
- Planning and variance lifecycle
- Using the platform

---

platform/governed-metrics-in-app.md

Governed metrics in the app

Workspace screens read **contract-backed Gold** through an in-app consumption layer—low-latency mirrors of analytical products your org is entitled to see. Values are not ad hoc queries against raw warehouse tables.

Who can use it

All workspace users consume governed metrics through Overview, FP&A, and Data hub; entitlements follow your commercial pack.

How to

:::note[Instructions coming soon]

Refresh behavior, stale indicators, and lineage shortcuts will be documented from demos.

:::

- [ ] Confirm pipeline published Gold for the metric
- [ ] Compare catalog definition (Reference catalog) with displayed values
- [ ] Use reporting or exports where enabled

Related

- Hosted pipeline
- Governed data
- Using the platform

---

platform/hosted-pipeline.md

Hosted pipeline

The commercial platform runs a **hosted medallion pipeline** that validates uploads against governed contracts and publishes **Gold** products your workspace can read. You trigger and monitor runs from the Command Centre—not from a separate analytics console.

Who can use it

Organization members with data permissions can see pipeline status tied to their org. Admins typically manage triggers and usage limits.

How to

:::note[Instructions coming soon]

Run history, retry, and status panels will be documented from demos.

:::

- [ ] Confirm data hub fulfillment for required data options
- [ ] Trigger or wait for scheduled pipeline processing
- [ ] Verify Gold availability in Overview and FP&A Workspace

Related

- Using the platform
- Data hub
- Governed metrics in the app
- Governed data

platform/lineage-and-compliance.md

Lineage and compliance

The **enterprise compliance pack** adds **full lineage audit** visibility from catalog definitions through pipeline stages to Gold consumption—supporting assurance and disclosure workflows.

Who can use it

Enterprise-tier organizations; assurance and compliance roles. Not part of minimal evaluation scopes.

How to

:::note[Instructions coming soon]

Lineage explorers and compliance exports will be documented from demos.

:::

- [ ] Open lineage views for a metric or data product
- [ ] Export audit evidence where enabled
- [ ] Pair with Assurance lifecycle playbooks

Related

- Governed metrics in the app
- Governed data

---

platform/managed-connectors.md

Managed connectors

Managed connector setup is request-based: your team asks Ambient to configure governed feeds into Bronze. The Command Centre does **not** offer live OAuth connector self-serve for all sources.

Who can use it

Organization admins and data owners typically submit connector requests; members may see resulting connection status in Data hub.

How to

:::note[Instructions coming soon]

Request forms, expected timelines, and status tracking will be documented from demos.

:::

- [ ] List data options that need a managed connector
- [ ] Submit a setup request with required metadata
- [ ] Track fulfillment alongside CSV and manual paths

Related

- Data hub
- Using the platform

---

platform/organizations-and-roles.md

Organizations and member roles

Multi-tenant **organizations** isolate governed data, pipeline runs, and workspace views. Members join an org with roles that control who can manage billing, data uploads, and admin settings.

Who can use it

Organization **admins** invite members and manage commercial usage. **Members** use workspace features allowed for their role. Exact permission labels appear in the product profile and organization settings.

How to

:::note[Instructions coming soon]

Step-by-step invites, role changes, and org switching will be added from product demos.

:::

- [ ] Create or select your organization after access is granted
- [ ] Invite members and assign roles
- [ ] Confirm pipeline and usage visibility for admins vs members

Related

- Using the platform
- Usage gates
- Platform features

platform/overview-dashboards.md

Overview dashboards

Overview is the Command Centre landing view: location-filtered **financial and operational metrics** sourced from governed Gold products for your industry pack.

Who can use it

All organization members with workspace access can use Overview; metric visibility may depend on role and entitlements.

How to

:::note[Instructions coming soon]

Filters, drill-down, and empty-state troubleshooting will be added from demos.

:::

- [ ] Select organization and location context
- [ ] Review KPI cards against your catalog pack
- [ ] Follow links to Data hub when metrics are missing inputs

Related

- Using the platform
- FP&A workspace
- Reference catalog

---

platform/reporting-and-exports.md

Reporting and exports

Standard reporting and exports let you download **PDF** and **CSV** views of governed metrics and tables you can access in the workspace.

Who can use it

Members with export permissions for their org; some reports may require admin or enterprise entitlements.

How to

:::note[Instructions coming soon]

Export entry points, formats, and limits will be documented from demos.

:::

- [ ] Open the view or report to export
- [ ] Choose PDF or CSV
- [ ] Store exports according to your org’s data policy

Related

- Overview dashboards
- FP&A workspace

---

platform/usage-gates.md

Usage gates

Commercial usage gates tie hosted pipeline runs and certain features to **credits** and plan entitlements. Admins see usage and limits in organization settings.

Who can use it

Organization **admins** manage billing-related usage; members may see when actions are blocked by limits.

How to

:::note[Instructions coming soon]

Credit balances, trigger rules, and upgrade paths will be documented from demos.

:::

- [ ] Review usage in organization admin panels
- [ ] Understand which actions consume credits or triggers
- [ ] Adjust entitlements via organization admin or invitation terms when hosted limits apply

Related

- Organizations and roles
- Hosted pipeline

---

use-cases/demo-provenance.md

Demo provenance

This page records how industry demo screenshots under Use cases were produced. **ambientsystems.ai** owns demo scenarios, mock/marketing fixtures, screenshots, and recordings used on the public site and docs.

Capture source

- **Owner:** ambientsystems.ai (`content/industry-demos.manifest.json` is SSOT)
- **Hosted App URL:** https://ambient-fintech-platform.web.app
- **Last capture from this repo (UTC):** not yet re-captured from this repo
- **Last wiki/public sync (UTC):** 2026-09-02T22:04:19.824Z
- **Viewport:** 1440×900

Capture environment

- Signed in as a real demo user against the **hosted** Command Centre (not a local emulator seed)
- Credentials: `DEMO_APP_EMAIL` / `DEMO_APP_PASSWORD` (operator secrets; never invent users against prod)
- Refresh from this repo: `npm run capture:demos` then `npm run import:demos`

Imported files

- **Binary count (PNG + WebM):** 68
- **Content fingerprint (SHA-256 prefix):** `312314587a8c7d6f`

Static files live under `wiki/static/img/platform-demos/` (also mirrored to `public/collateral/demos/`) and are served at `/docs/img/platform-demos/` on the public site.

`npm run import:demos` only embeds screenshot/video markdown for binaries that exist on disk. Industries without captures keep a catalog link and a pending-capture note (no invented PNGs).

Legacy platform import (historical)

Earlier binaries were imported from ambient-systems-platform before site ownership of demos.

- **Legacy platform git ref:** `5ec8949`
- **Legacy platform MANIFEST date:** 2026-07-03

How to refresh

1. Ensure `DEMO_APP_EMAIL` and `DEMO_APP_PASSWORD` are set (see `.env.example` / docs/DEMO-CAPTURE.md).
2. Run `npm run capture:demos` — Playwright captures against `PUBLIC_APP_URL` or the Hosting fallback and writes PNG/WebM under `public/collateral/demos/`.
3. Run `npm run import:demos` — mirrors into wiki static, regenerates industry pages and this provenance block from the site manifest.
4. Review the diff, then commit updated binaries, wiki pages, and `content/industry-demos.manifest.json`.

`ambient-systems-platform` is commercial-only for demos. Do **not** clone it or set `PLATFORM_ROOT` to refresh marketing fixtures. Doc markdown sync (`npm run sync:docs`) remains orthogonal and may still use a platform checkout when configured.

---

use-cases/index.md

Use cases - workspace demos

Command Centre screenshots for each Ambient Core catalog vertical (hosted App demo). This section is the demo gallery; pack definitions (metrics, data options, and calc logic) live once in the Reference catalog. Demo provenance records the hosted App capture metadata and fingerprint.

Industry workspace demos

- Banking (depository)
- Financial Services
- Funds
- Trusts
- Insurance
- Credit Granting
- Software (SaaS)
- Retail
- Real Estate (commercial)
- Manufacturing
- Transportation (freight)
- Passenger Transit
- Aviation
- Healthcare
- Life Sciences

- Restaurants
- Vertical Farming

Related

- Reference catalog - pack definitions, metrics, and data options.
- Catalog consumption - data options to KPIs.
- Platform features - hosted Command Centre capabilities.
- Foundation vs full product - orientation under Start Here.

---

use-cases/industries/aviation.md

Aviation

Screenshots show the Ambient Systems Command Centre after onboarding with the **Aviation** catalog pack. The illustrative organization **SkyBridge Airways** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Aviation

Overview

!Overview dashboard for Aviation

FP&A Workspace

!FP&A Workspace for Aviation

Data library

!Data library for Aviation

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/aviation.yaml

---

use-cases/industries/banking.md

Banking (depository)

Screenshots show the Ambient Systems Command Centre after onboarding with the **Banking (depository)** catalog pack. The illustrative organization **First Meridian Bank** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Banking (depository)

Overview

!Overview dashboard for Banking (depository)

FP&A Workspace

!FP&A Workspace for Banking (depository)

Data library

!Data library for Banking (depository)

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/banking.yaml

---

use-cases/industries/credit-granting.md

Credit Granting

The illustrative organization **SwiftLend Finance** is a demo persona only, not a customer reference. Pack definitions live in the Reference catalog; workspace screenshots for this vertical are pending site-owned capture.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Screenshots

Command Centre screenshots for this vertical are not in site collateral yet (no invented placeholders). Capture them with `npm run capture:demos` once `DEMO_APP_*` secrets are set — see Demo provenance. Until then, see a captured vertical such as Banking or the Reference catalog.

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/credit_granting.yaml

---

use-cases/industries/financial-services.md

Financial Services

Screenshots show the Ambient Systems Command Centre after onboarding with the **Financial Services** catalog pack. The illustrative organization **Atlas Capital Partners** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Financial Services

Overview

!Overview dashboard for Financial Services

FP&A Workspace

!FP&A Workspace for Financial Services

Data library

!Data library for Financial Services

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.

- Pack source (ambient-core): catalog/industries/financial_services.yaml

---

use-cases/industries/funds.md

Funds

Screenshots show the Ambient Systems Command Centre after onboarding with the **Funds** catalog pack. The illustrative organization **Horizon Growth Fund** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Funds

Overview

!Overview dashboard for Funds

FP&A Workspace

!FP&A Workspace for Funds

Data library

!Data library for Funds

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/funds.yaml

---

use-cases/industries/healthcare.md

Healthcare

Screenshots show the Ambient Systems Command Centre after onboarding with the **Healthcare** catalog pack. The illustrative organization **Riverdale Health System** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Healthcare

Overview

!Overview dashboard for Healthcare

FP&A Workspace

!FP&A Workspace for Healthcare

Data library

!Data library for Healthcare

Screen recording

<video controls width="100%" poster="/docs/img/platform-demos/screenshots/overview-healthcare.png">

<source src="/docs/img/platform-demos/videos/workspace-healthcare.webm" type="video/webm" />

</video>

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/healthcare.yaml

---

use-cases/industries/index.md

Industry demos

Command Centre screenshots for each Ambient Core catalog vertical (hosted App demo). Pack definitions (metrics, data options, and calc) live in the Reference catalog.

See Use cases for the full demo gallery and Demo provenance for capture metadata and fingerprint.

---

use-cases/industries/insurance.md

Insurance

Screenshots show the Ambient Systems Command Centre after onboarding with the **Insurance** catalog pack. The illustrative organization **Guardian Mutual Insurance** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Insurance

Overview

!Overview dashboard for Insurance

FP&A Workspace

!FP&A Workspace for Insurance

Data library

!Data library for Insurance

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/insurance.yaml

---

use-cases/industries/life-sciences.md

Life Sciences

Screenshots show the Ambient Systems Command Centre after onboarding with the **Life Sciences** catalog pack. The illustrative organization **NovaGen Therapeutics** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

Overview

FP&A Workspace

Data library

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/life_sciences.yaml

---

use-cases/industries/manufacturing.md

Manufacturing

Screenshots show the Ambient Systems Command Centre after onboarding with the **Manufacturing** catalog pack. The illustrative organization **Precision Parts Co** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

Overview

FP&A Workspace

Data library

!Data library for Manufacturing

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/manufacturing.yaml

---

use-cases/industries/passenger-transit.md

Passenger Transit

The illustrative organization **Metro Transit Authority** is a demo persona only, not a customer reference. Pack definitions live in the Reference catalog; workspace screenshots for this vertical are pending site-owned capture.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Screenshots

Command Centre screenshots for this vertical are not in site collateral yet (no invented placeholders). Capture them with `npm run capture:demos` once `DEMO_APP_*` secrets are set — see Demo provenance. Until then, see a captured vertical such as Banking or the Reference catalog.

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/passenger_transit.yaml

---

use-cases/industries/real-estate.md

Real Estate (commercial)

Screenshots show the Ambient Systems Command Centre after onboarding with the **Real Estate (commercial)** catalog pack. The illustrative organization **Summit Properties Group** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Real Estate (commercial)

Overview

!Overview dashboard for Real Estate (commercial)

FP&A Workspace

!FP&A Workspace for Real Estate (commercial)

Data library

!Data library for Real Estate (commercial)

Screen recording

<video controls width="100%" poster="/docs/img/platform-demos/screenshots/overview-real-estate.png">

<source src="/docs/img/platform-demos/videos/onboarding-real-estate.webm" type="video/webm" />

</video>

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/real_estate.yaml

---

use-cases/industries/restaurants.md

Restaurants

Screenshots show the Ambient Systems Command Centre after onboarding with the **Restaurants** catalog pack. The illustrative organization **Harbor Bistro Group** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Restaurants

Overview

!Overview dashboard for Restaurants

FP&A Workspace

!FP&A Workspace for Restaurants

Data library

!Data library for Restaurants

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/restaurants.yaml

---

use-cases/industries/retail.md

Retail

Screenshots show the Ambient Systems Command Centre after onboarding with the **Retail** catalog pack. The illustrative organization **Urban Retail Collective** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Retail

Overview

!Overview dashboard for Retail

FP&A Workspace

!FP&A Workspace for Retail

Data library

!Data library for Retail

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/retail.yaml

---

use-cases/industries/software-saas.md

Software (SaaS)

The illustrative organization **Nimbus Apps** is a demo persona only, not a customer reference. Pack definitions live in the Reference catalog; workspace screenshots for this vertical are pending site-owned capture.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Screenshots

Command Centre screenshots for this vertical are not in site collateral yet (no invented placeholders). Capture them with `npm run capture:demos` once `DEMO_APP_*` secrets are set — see Demo provenance. Until then, see a captured vertical such as Banking or the Reference catalog.

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/software_saas.yaml

---

use-cases/industries/transportation.md

Transportation (freight)

Screenshots show the Ambient Systems Command Centre after onboarding with the **Transportation (freight)** catalog pack. The illustrative organization **Northstar Logistics** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Transportation (freight)

Overview

!Overview dashboard for Transportation (freight)

FP&A Workspace

!FP&A Workspace for Transportation (freight)

Data library

!Data library for Transportation (freight)

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/transportation.yaml

---

use-cases/industries/trusts.md

Trusts

Screenshots show the Ambient Systems Command Centre after onboarding with the **Trusts** catalog pack. The illustrative organization **Legacy Family Trust** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Trusts

Overview

!Overview dashboard for Trusts

FP&A Workspace

!FP&A Workspace for Trusts

Data library

!Data library for Trusts

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.
- Pack source (ambient-core): catalog/industries/trusts.yaml

---

use-cases/industries/vertical-farming.md

Vertical Farming

Screenshots show the Ambient Systems Command Centre after onboarding with the **Vertical Farming** catalog pack. The illustrative organization **GreenTower Farms** is a demo persona only, not a customer reference.

Capture uses the **hosted** Command Centre at 1440×900 as a signed-in demo user. See Demo provenance for App URL and fingerprint.

Full metric definitions · Catalog consumption

Onboarding

!Onboarding industry selection for Vertical Farming

Overview

!Overview dashboard for Vertical Farming

FP&A Workspace

!FP&A Workspace for Vertical Farming

Data library

!Data library for Vertical Farming

Catalog reference

- Full metric definitions and calc logic — generated from ambient-core YAML.
- Catalog consumption — how industry YAML and manifest relate to the workspace.

- Pack source (ambient-core): catalog/industries/vertical_farming.yaml

---