HPC Track-B Raw Extraction & K-Series Construction — Methods Report (FINAL)

Scope. This report documents (1) exactly how we assembled the **Haifa Port Company (HPC)** raw financial dataset for Track B (Financial/Accounting capital), including what filings were used and which fields were extracted; and (2) the end-to-end **strategy to turn the raw extractions into a monthly K series** (and ultimately **ln(K/L)**) using reproducible code. It is aligned with the H191 Handoff Report contracts.

---

1) What we found online & how we used it

Filing types located (public sources):

- **Annual Financial Statements (2018–2019, English).** Contain: Balance Sheet (**PPE net**), **Note 8: PPE** (cost/gross, accumulated depreciation, **additions**, **disposals**), and Cash Flows (investing section and depreciation in adjustments).
- **Annual Financial Statements (2020–2021, Hebrew).** Same coverage as above; presentation in Hebrew (routines/labels mapped via Heb↔Eng crosswalk).
- **Interim FS (Q1–Q3 2022, Hebrew, YTD).** Contain YTD **Balance Sheet** and **Cash Flow** lines (e.g., depreciation add-back; net acquisition of fixed assets; purchase of intangibles; proceeds from disposal). Note 8 roll-forwards typically omitted from interims.
- **Annual Financial Statements (2024, English).** Include full **Note 8** roll-forwards for 2023–2024 and **opening 2023 balances** (which equal **closing 2022**). Also note **privatization completion on 2023-01-10**, the accounting break we will splice across.

  Why these suffice for 2022 YE: Even without a standalone HPC **2022 annual PDF**, the **opening 2023 balances** in the 2024 Note 8 provide **YE-2022 PPE gross & accumulated**. Hence we can anchor 2022 net PPE and proceed with the PPE-based K series; the **movement lines** (additions/disposals/depreciation) for 2022 are still missing and are flagged **NA** for PIM.

---

2) What we extracted (variable-by-variable)

From annuals (2018–2019 EN; 2020–2021 HE; 2024 EN): - `ppe_gross_nominal` (Note 8 – Cost)
- `accumulated_depreciation_nominal` (Note 8 – Accumulated depreciation)
- `ppe_net_nominal` (Balance Sheet; identity checked as gross − accumulated)
- `additions_purchases_note_nominal` (Note 8 – Movement)
- `disposals_note_nominal` (Note 8 – Movement; sign as reported; if cost-basis table provides, captured there)
- `depreciation_expense_note_nominal` (annual depreciation; sometimes disclosed in Note 8 or in CF adjustments)
- `acq_fixed_assets_net_cf_nominal` (Cash Flows – investing, net acquisitions of fixed assets; **outflows negative**)

- `purchase_intangibles_cf_nominal` (Cash Flows – investing; outflows negative)
- `proceeds_disposal_fixed_assets_cf` (Cash Flows – investing; inflows positive)
- `investing_grants_or_receipts_cf` (if disclosed; inflow positive)

From interims (2022 Q1–Q3, YTD): - YTD `depreciation_expense_note_nominal` (CF adjustments)
- YTD `acq_fixed_assets_net_cf_nominal` (investing)
- YTD `purchase_intangibles_cf_nominal` (investing)
- YTD `proceeds_disposal_fixed_assets_cf` (investing)

From 2024 annual Note 8 (opening 2023): - **YE-2022** `ppe_gross_nominal` and `accumulated_depreciation_nominal` (entered as a new **2022 annual row** in v8; `ppe_net_nominal` computed by identity).
- **ROU classification note:** ROU asset balances reported include **vehicles only** (recorded in comments for classification transparency; we keep Note-8 totals **as reported** in Track B).

Not available: The **2022 annual movement lines** (`additions_purchases_note_nominal`, `disposals_note_nominal`, `depreciation_expense_note_nominal`) remain **NA** until/unless the 2022 annual is recovered.

---

3) The assembled raw dataset (HPC, FINAL v8)

Authoritative file: `/mnt/data/TrackB_Haifa_Raw_Financials_FINAL_v8.tsv`
Granularity: one row per **company × period** (annual or interim), **as reported** (thousands of NIS).
Key fields (subset): - IDs: `port`, `company`, `operator_or_owner`, `period_type`, `financial_year`, `period_end_date`, `report_year`, `language`, `currency`, `units_scale`
- Balance Sheet / Note 8 totals: `ppe_gross_nominal`, `accumulated_depreciation_nominal`, `ppe_net_nominal`
- Note 8 movements: `additions_purchases_note_nominal`, `disposals_note_nominal`, `depreciation_expense_note_nominal`
- Cash Flows (investing & adjustments): `acq_fixed_assets_net_cf_nominal`, `purchase_intangibles_cf_nominal`, `proceeds_disposal_fixed_assets_cf`, `investing_grants_or_receipts_cf`
- Provenance & flags: `page_ref_*`, `extraction_method`, `comments`, `break_or_basis_change_flag`

Conventions: amounts in **thousands NIS**; investing **outflows negative**, inflows positive; interims are **YTD**. We do **not net or deflate** at the raw stage.

Quality checks embedded in v8: 1. `qa_identity_gross_minus_accum_equals_net` (boolean): validates Note-8 identity when all three present.
2. **Interim YTD monotonicity** (diagnostic): within year, YTD **depreciation** and **|capex net|** should be non-decreasing across Q1→Q2→Q3.

---

4) Turning raw extractions into K (math + code plan)

We build two complementary K series at **port (HPC-legacy) × month**:

4.1 PPE-based K (baseline)

Idea. Use **net PPE** as the value stock; **deflate** to real; **interpolate** to monthly; **splice** across the 2023 privatization break while preserving growth.

Math. Let `PPE_net_nominal_{y}` be annual net PPE at YE. Let `D_{t}` be a monthly deflator index (base=1 in a chosen month, e.g., **2020-01**). For month `t` in year `y`:

1. **Annual to monthly path (nominal):**
   Linear interpolation within year (on **net PPE**) between YE points `y-1` and `y`:

$$\tilde{K}_t^{nom} = K_{y-1,YE}^{nom} + \frac{m(t)}{12}\left(K_{y,YE}^{nom} - K_{y-1,YE}^{nom}\right), \quad m(t) \in \{1, \dots, 12\}.$$

(Optionally weight commissioning months with a shape function `w_m` that sums to 12.)

1. **Deflation to real:**

$$K_t^{real} = \frac{\tilde{K}_t^{nom}}{D_t}.$$

2. **Break splice (Jan-2023):** If privatization causes a level break at splice month `b`, define a **level factor**

$$s = \frac{K_{b^-}^{real,pre}}{K_b^{real,post,raw}}$$

   and set $K_t^{real,post} = s \cdot K_t^{real,post,raw}$ for all $t \geq b$. This preserves **growth rates** post-break while matching the pre-break level.

Code sketch (pandas):

```
import pandas as pd, numpy as np

raw = pd.read_csv("/mnt/data/TrackB_Haifa_Raw_Financials_FINAL_v8.tsv",
sep="\t")
# Filter HPC annuals only and pivot to YE series
ann = (raw[(raw.port=="Haifa") & (raw.operator_or_owner.str.contains("Haifa-
Legacy", na=False)) &
          (raw.period_type=="annual")][["financial_year","ppe_net_nominal"]]
       .dropna())
```

```
ann = ann.sort_values("financial_year").set_index("financial_year")

# Build a monthly index from min_year-1 YE to max_year YE
years = ann.index.to_list()
start = pd.Period(f"{years[0]-1}-12", freq="M")
end   = pd.Period(f"{years[-1]}-12", freq="M")
mi = pd.period_range(start.to_timestamp(), end.to_timestamp(), freq="M")

# Interpolate nominal net PPE monthly
annual_to_month = {pd.Period(f"{y}-12", freq="M"): ann.loc[y, "ppe_net_nominal"]
for y in years}
ser = pd.Series(index=mi, dtype=float)
for y, v in annual_to_month.items():
    ser[y] = v
ser = ser.interpolate(method="time")  # linear in calendar time

# Merge deflator D_t (monthly index with base=1.0 at chosen month)
D = pd.read_csv("/mnt/data/deflators/cbs_ppe_deflator_monthly.tsv", sep="\t")
D["month"] = pd.to_datetime(D["month"])  # YYYY-MM-01
Dt = D.set_index(pd.PeriodIndex(D["month"], freq="M"))["index"]
K_real = (ser / Dt.reindex(ser.index)).rename("K_PPE_real")

# Splice at break b = 2023-01 (period)
b = pd.Period("2023-01", freq="M")
pre = K_real.loc[:b-1]
post_raw = K_real.loc[b:]
scale = pre.iloc[-1] / post_raw.iloc[0]
K_real_spliced = pd.concat([pre, post_raw*scale])
```

Outputs: `K_B_monthly_port.tsv` with columns `[port, month, K_PPE_real_spliced, K_PPE_real_raw, splice_flag]`.

4.2 Perpetual Inventory Method (PIM) K (sensitivity)

Idea. Accumulate a capital stock from **additions (I)** and **disposals** with **depreciation** at rate δ , using annual flows when available and a principled allocation to months. Build **low/central/high** bands for δ by asset class.

Math. Monthly PIM:

$$K_t = (1-\delta_m)K_{t-1} + I_t - \text{Disposals}_t, \qquad \delta_m = 1-(1-\delta)^{1/12}.$$

Where annual movement lines exist (2018–2021, 2023–2024): - Let annual **cost additions** and **disposals** be I_y^c and D_y^c from Note 8 – Cost table.
- Allocate to months with weights w_m (uniform by default; commissioning-weighted optional) s.t.

$\sum_m w_m = 1$.
- Annual **depreciation** Dep_y informs a check against $\delta \cdot \bar{K}_y$ (not imposed in raw PIM; used for QA).

Where 2022 annual movements are missing: - Use **YE-2022 gross/accumulated anchors** and **2021 YE** to compute $\Delta\mathrm{Cost}$ and $\Delta\mathrm{Accum}$ over 2022; allocate monthly with weights w_m . Mark 2022 as **imputed**.

Capital services (optional): $S_t = (r + \delta)K_t$, with real rate $r \approx 0.03$.

Code sketch (pandas):

```
# Assume we have annual movement tables from v8 for years with Note 8 movements
mov = (raw[(raw.port=="Haifa") & (raw.period_type=="annual")]

[["financial_year","additions_purchases_note_nominal","disposals_note_nominal"]]
       ).dropna(how="all",
subset=["additions_purchases_note_nominal","disposals_note_nominal"]).set_index("financial_year")

# Build monthly weights (uniform here; replace with commissioning-aware as
needed)
w = np.repeat(1/12, 12)

# Parameters
delta_central = 0.08  # illustrative for mixed yard/STS; we will run bands per
asset class in practice
r = 0.03

def delta_m(delta):
    return 1 - (1-delta)**(1/12)

# Initialize K at 2017-12 by backcasting from 2018 net PPE if needed, or set
K_2017 to net PPE 2017 YE
K = pd.Series(index=mi, dtype=float)
K.iloc[0] = ser.iloc[0]  # seed with first nominal net PPE as rough initial
level (could be improved)

for i in range(1, len(mi)):
    t = mi[i]
    y = t.year
    I_t, D_t = 0.0, 0.0
    # allocate annual additions/disposals to month
    if y in mov.index:
        m_idx = (t.month-1)
        I_year = mov.loc[y, "additions_purchases_note_nominal"] or 0.0
        D_year = mov.loc[y, "disposals_note_nominal"] or 0.0
        I_t = I_year * w[m_idx]
        D_t = D_year * w[m_idx]
    K.iloc[i] = (1 - delta_m(delta_central)) * K.iloc[i-1] + I_t - D_t

S = (r + delta_central) * K
```

Outputs: `K_B_monthly_port.tsv` with additional columns `[K_PIM_central, K_PIM_low, K_PIM_high, K_services_central]` and flags `[imputed_2022]`.

4.3 Build ln(K/L) mediator

Join keys & caveats. Use **port×month** keys. For **Haifa-Legacy**, join to terminal-level **L hours** from `Data/L_proxy/L_Proxy.tsv` filtered to the legacy terminal. Do **not** divide HPC-only `K` by **port-total L** after Bayport's entry. Port-level K (if needed) will later be `K_legacy (Track B) + K_bayport (Track A/C)`.

Math. $M_t = \ln\left(K_t/L_t\right)$ for each K variant. Drop months where `L_t` is NA (pre-opening) or where K is spliced across the break if continuity is required for a given spec.

Code sketch:

```
L = pd.read_csv("/mnt/data/Data/L_proxy/L_Proxy.tsv", sep="\t")
L["month"] = pd.to_datetime(L["month"])  # ensure monthly key
# Filter to Haifa-Legacy terminal hours and aggregate to port if needed
L_leg = L[(L.port=="Haifa") & (L.terminal=="Haifa-Legacy")][["month","hours"]]
K_df = pd.read_csv("/mnt/data/K_B_monthly_port.tsv", sep="\t")
K_df["month"] = pd.to_datetime(K_df["month"])

merged = K_df.merge(L_leg, on="month", how="left", validate="1:1")
for col in ["K_PPE_real_spliced","K_PIM_central","K_PPE_real_raw"]:
    merged[f"lnK_over_L__{col}"] = np.log(merged[col] / merged["hours"])

merged.to_csv("/mnt/data/Mediator_K_over_L.tsv", sep="\t", index=False)
```

---

5) Deflators, parameters & metadata

- **Deflator** `D_t`**:** Prefer CBS monthly indices relevant to port fixed assets (construction inputs / capital goods). If only annual indices available, convert to monthly via intra-year interpolation. **Base** set to 1.0 at a reference month (e.g., 2020-01). Store as `/mnt/data/deflators/cbs_ppe_deflator_monthly.tsv` with columns `[month, index]`.
- **Depreciation** δ **:** Use asset-class bands for sensitivity: civil works 0.03–0.04, cranes 0.06–0.10, yard equipment 0.08–0.12; central value chosen per HPC's asset mix (document assumptions in META).
- **Rate** r **:** 0.03 real for capital services.
- **Splice meta:** Record splice date (2023-01), scale factor, and rationale in `_meta_K_B.json`.

---

6) QA after K build

1. **Continuity check** around 2023 splice: visually inspect and compute the one-month log change; confirm no jump remains in the spliced series.
2. **Price-level sanity:** deflation removes inflation humps; compare PPE-real trend to port commissioning timeline.
3. **PIM vs PPE cross-check:** Trends should be broadly consistent; divergences flagged and explained (e.g., revaluations, disposals bursts, grants).
4. **Mediator stability:** $\ln(K/L)$ should respond positively after commissioning/privatization instruments in first-stage regressions.

---

7) Reproducible outputs

- `TrackB_Haifa_Raw_Financials_FINAL_v8.tsv` (this report's input)
- `K_B_monthly_port.tsv` (PPE-real raw & spliced; PIM bands; services; flags)
- `Mediator_K_over_L.tsv` (join to L)
- `_meta_K_B.json` (deflators, parameters, splice factor, data lineage)
- `qa_K_B_report.tsv` (numerical QA: identities, splice continuity, summary stats)

---

8) Open items & decisions

- **Open:** 2022 annual **movement lines** missing (additions, disposals, depreciation). We proceed with PPE-based K and mark PIM-2022 as **imputed** if constructed.
- **Decisions (Track B):** Raw stays **as reported**; no reclass of ROU; CF signs as in statements; interims are YTD; **splice** at 2023-01; mediator built at **legacy-terminal × month** (port aggregation later when Bayport K is available from Track A/C).

End of report.