# fetch\_nfl\_odds — Process & Change Guide (v3.3.4)

**Scope**: This guide describes how the unified odds fetcher works, how to run it safely, how to change it, and how we consume its outputs in the **Millions** Member dashboard and **Survivor** roadmap. It reflects **v3.3.4** (open\_total/open\_line wiring + planner merge fixes). If your local header shows **v3.3.3**, update from canvas to get the latest.

## 1) What the script does (high level)

* **Fetches current odds** from The Odds API (/v4/sports/americanfootball\_nfl/odds) across US books.
* **Normalizes** spreads (home/away sign), **aggregates** to medians across books, captures **Circa** when present.
* **Computes implied win probabilities** from moneylines.
* **Writes a season CSV** (e.g., data/odds/season\_all.csv).
* **Updates the Millions planner** (picks/millions/millions\_planner.csv) with current spreads, totals, Circa line/total, and (when available) **open\_total**/**open\_line**.
* **Updates the Survivor roadmap** (picks/survivor/survivor\_roadmap.csv) with team‑perspective **current moneyline** and **win probability**.

Historical (open) snapshots are **off** in this build. When we enable history/snapshots later, the same wiring will auto‑fill open\_\* fields and **line deltas**.

## 2) One‑command run (PowerShell)

$env:ODDS\_API\_KEY = "<YOUR\_API\_KEY>"  
python scripts/fetch\_nfl\_odds.py `  
 --season 2025 `  
 --regions us `  
 --out "data/odds/season\_all.csv" `  
 --update\_planner "picks/millions/millions\_planner.csv" `  
 --update\_survivor "picks/survivor/survivor\_roadmap.csv" `  
 --backup 1 `  
 --history 0

**Outputs printed:** request usage, odds rows written, planner backup & update, survivor backup & update.

## 3) Inputs, outputs & schemas

### 3.1 Inputs

* **Env**: ODDS\_API\_KEY (The Odds API key)
* **Args**: --season (infers wide Aug→Feb window), --regions us, --out, --update\_planner, --update\_survivor.

### 3.2 data/odds/season\_all.csv

Columns (core):

* home\_team, away\_team (NFL abbreviations)
* current\_spread\_home, current\_spread\_away (anti‑symmetric; home favored ⇒ negative)
* current\_total (median total)
* current\_ml\_home, current\_ml\_away (American odds, median)
* win\_prob\_home, win\_prob\_away (implied from ML)
* circa\_spread\_home, circa\_spread\_away, circa\_total (if Circa book present)

### 3.3 Planner fields populated

* current\_spread\_home, current\_spread\_away, current\_total
* circa\_line (team perspective, derived from circa\_spread\_home/away + home\_or\_away)
* circa\_total
* open\_total (when available)
* open\_line (team perspective, derived from open\_spread\_home/away + home\_or\_away)

### 3.4 Survivor fields populated

* current\_ml\_team (moneyline for the *team* in that row)
* win\_prob\_team (implied probability for that team)

## 4) How it works (under the hood)

1. **Fetch** via fetch\_odds(...) with markets h2h,spreads,totals.
2. **Extract** each event with extract\_row\_current(...):
   * Collect all book outcomes.
   * Take **median** across books for spreads/totals/ML.
   * Normalize spreads with normalize\_pair so home = −away.
   * Detect **Circa** by bookmaker title and store its line/total if present.
3. **Write** the season CSV.
4. **Merge** into planner with update\_planner(...):
   * Build a stable \_\_game\_key from the teams (order‑independent).
   * Left‑merge odds onto planner (\*\_odds suffix on the right).
   * Safely copy values into planner columns using \_safe\_get + fillna.
   * Derive circa\_line and open\_line from home/away perspective.
5. **Merge** into survivor with update\_survivor(...) mapping the team in each row to the correct home/away ML and win prob.

Defensive guards: team alias map, NaN‑safe uppercasing, duplicate‑column‑safe getters.

## 5) Beginner‑friendly “How to change X”

* **Change books/regions**: edit CLI --regions (e.g., us,eu).
* **Change sport**: pass --sport (defaults to americanfootball\_nfl).
* **Prefer a single book** (e.g., no median): inside extract\_row\_current, replace med(...) aggregation with your selection logic.
* **Include more markets** (e.g., team totals): extend markets and add extraction.
* **Add a new team alias**: add to ALIASES (e.g., "WAS": "WSH").
* **Compute line deltas / opens**: enable history (see §7) or implement a nightly snapshot logger (see §9).

## 6) Quick QA after a run (copy/paste)

**Odds sanity:** spreads anti‑symmetric, totals in 35–60 range (varies preseason)

python - << 'PY'  
import pandas as pd  
odds = pd.read\_csv('data/odds/season\_all.csv')  
print(odds[['home\_team','away\_team','current\_spread\_home','current\_spread\_away','current\_total']].head(8))  
PY

**Planner has fields:**

python - << 'PY'  
import pandas as pd  
pl = pd.read\_csv('picks/millions/millions\_planner.csv')  
need = ['current\_spread\_home','current\_spread\_away','current\_total','circa\_line','circa\_total','open\_line','open\_total']  
print({c:(c in pl.columns) for c in need})  
PY

**Survivor has ML & win %:**

python - << 'PY'  
import pandas as pd  
sv = pd.read\_csv('picks/survivor/survivor\_roadmap.csv')  
print(sv[['team','opponent','current\_ml\_team','win\_prob\_team']].head(10))  
PY

## 7) About history / opens / line deltas

* The Odds API **historical** endpoint requires a paid plan. With history enabled we’d:
  + Pull a snapshot near open (open\_offsets like 21, 10, 5 days pre‑kick).
  + Populate open\_spread\_\*, open\_total and compute line\_delta\_home.
* In **v3.3.4**, planner wiring is already in place. When history is available, these fields fill in automatically.

**If you don’t upgrade yet** → use our **nightly snapshot logger** (see §9) to compute deltas prospectively.

## 8) Troubleshooting (common errors)

* **401 Unauthorized (historical)**: key/plan doesn’t include /historical. Disable history or upgrade plan.
* `: fixed in v3.3.3+ by avoidingSeries.where` on ambiguous columns.
* ``: fixed by NaN‑safe uppercasing in survivor merge.
* **Planner totals all NaN**: odds file has totals but planner didn’t copy → ensure you’re on **v3.3.4** and re‑run. If odds also lacks totals for a game, Member Dashboard shows **TBD** by design.

## 9) Nightly snapshot logger (lightweight alternative to historical)

**Idea**: Save data/odds/snapshots/<YYYY‑MM‑DD>.csv nightly. Compute open\_\* as the **first** snapshot seen for an event; later runs compute **deltas**.

Minimal cron/Task Scheduler action:

* Command: same as main run, but --out data/odds/snapshots/%DATE%.csv and no planner/survivor update.
* A tiny helper can then merge the earliest snapshot per game into the main odds file to fill open\_\*.

If you want this, I’ll drop a ready‑to‑run scripts/odds\_snapshot\_logger.py + a scripts/compute\_line\_deltas.py helper.

## 10) Safety & rollback

* Every planner/survivor update writes a .bak alongside the CSV. To restore, replace the current file with the .bak and re‑run the exporter.
* The fetcher **never deletes rows**; it only fills/updates columns by key.

## 11) How this feeds the UI

* **Member dashboard exporter** (scripts/millions\_export\_member\_html.py) reads the planner and shows:
  + Favorite (open/current), Circa Line, **Total** (TBD if missing), **Auto‑Notes** chips (line move/injuries/weather).
* **Survivor roadmap** uses win\_prob\_team for sorting/short‑listing.

## 12) Change map (where to edit)

* **Markets / extraction**: extract\_row\_current (add/modify outcomes)
* **Aggregation**: med(...) inside extract\_row\_current
* **Team aliases**: ALIASES and TEAM\_NAME\_TO\_ABBR
* **Planner merge rules**: update\_planner (copy list + perspective logic)
* **Survivor mapping**: update\_survivor → choose(...)

### Example: add a bookmaker preference

# Replace median with priority order  
def pick(vals, preferred=None):  
 if preferred: # a list of (book, value)  
 for b,v in preferred:  
 if v is not None:  
 return v  
 return pd.Series(vals).median() if vals else None

## 13) Quick audit snapshot (from latest runs)

* **Odds**: ~272 rows written (season window)
* **Planner update**: rows updated, Circa/Current filled; open fields will populate when history/snapshots are available.
* **Survivor update**: ~800+ rows updated with current\_ml\_team + win\_prob\_team.

## 14) Next steps

* Decide on **history** vs **nightly snapshots** for opens/line deltas.
* (Optional) Add fallback to show current\_total as a temporary stand‑in for open\_total when opens are missing.
* Wire a tiny **Survivor Week short‑list exporter** (top teams by win prob with constraints).

If anything’s unclear, paste the command/output you ran and I’ll annotate exactly which section of this doc applies.