# NFL25 Agent — Millions & Survivor Tools Documentation

## Overview

This project is part of the **NFL25 Agent**, designed to support two Circa contests:

* **Millions (ATS picks)**
* **Survivor (straight‑up picks)**

The tools share common data (schedule, odds, DVOA, injuries, weather) but diverge in their planning, enrichment, and audit flows. Everything runs inside the repo nfl25-agent using PowerShell commands (Windows + PyCharm).

## Current State (Millions)

### 1. Core CSVs

* `` → team‑oriented, holds weekly ATS pick planning and enriched metadata.
* `` → game‑level schedule context (kickoff, venue, spreads, totals, DVOA, etc.).
* `→ normalized odds snapshot fromfetch\_nfl\_odds.py`.

### 2. Scripts

* ``
  + Canonical fetcher. Calls The Odds API.
  + Outputs spreads, ML, totals, implied win %.
  + **Rules:** single source, minimal edits, debug opt‑in.
* `` (A2)
  + Merges planner (team) with roadmap (game).
  + Adds spreads, totals (open/current/closing/circa), DVOA splits, context fields.
  + Optionally merges millions\_weekX\_odds.csv to backfill totals if roadmap lacks them.
  + Safe to rerun; overwrites enriched columns only.
* `` (A3)
  + Runs diagnostics on enriched planner.
  + Checks ranges (spread ±20, DVOA ±0.6).
  + Produces Markdown in picks/millions/diagnostics.

### 3. Recent Improvements

* **Totals now flow end‑to‑end.**
  + Added support for open\_total, current\_total, closing\_total, circa\_total in enrichment.
  + Planner previously showed only current\_total; now all four are aligned.
* **Cleaner merges.** --drop\_overlaps avoids duplicate \_x/\_y columns.
* **Console summaries.** Quick preview of spreads/totals non‑null counts for QA.
* **Safe backup mode.** --backup writes a .bak.csv snapshot before overwrite.

### 4. Current Flow (Millions)

fetch\_nfl\_odds.py → millions\_weekX\_odds.csv  
 ↓  
roadmap + odds → millions\_enrich\_planner.py  
 ↓  
planner enriched → millions\_audit\_week.py  
 ↓  
spotcheck md → export/dashboard (future)

Planner is now the **single source of truth** for ATS work. Enrichment pulls in live odds + roadmap context. Audit validates before export.

## Current State (Survivor)

### 1. Core CSVs

* `` → full season, byes, holiday flags, placeholder spot values.
* `` → wide heatmap (weeks × teams) with USED/LOCKED, spot\_value labels.
* `` → minimal weekly log (week, team, opponent, win\_prob, notes).

### 2. Scripts

* `` → expands schedule, adds BYE + TG/Xmas flags.
* `` → generates matrix with spot\_value placeholders, locks, holiday tags.
* `` → logs weekly pick, updates USED/LOCKED in matrix.
* `` → computes weighted spot\_value scores from win prob, DVOA, rest, injuries, etc.

### 3. Recent Improvements

* **Holiday penalty rules** integrated into spot\_value scoring.
* **DVOA normalized and aligned** (team vs opponent splits).
* **Matrix dynamically updates** with each logged pick.
* **Integration prep:** odds API feeding win probabilities for Survivor also drives Millions enrichment.

### 4. Current Flow (Survivor)

roadmap (season) → expand\_and\_flag\_schedule.py  
 ↓  
roadmap expanded → generate\_survivor\_matrix.py  
 ↓  
weekly picks → log\_survivor\_pick.py  
 ↓  
matrix w/ locks → spot\_value\_updates.py

## Cross‑Tool Interaction

* **Shared odds source:** fetch\_nfl\_odds.py feeds both tools.
* **Shared schedule meta:** Roadmap files carry kickoff, venue, rest days, byes, holidays.
* **Shared metrics:** DVOA, power ratings, injury tags.
* **Unified audit/export model:** Millions runs audit\_week; Survivor runs matrix + spot value diagnostics.

Both pipelines converge into a **Master Dashboard** (planned) with:

* Weekly pick summaries (ATS + Survivor).
* Odds vs Circa line comparisons.
* Contrarian angles (public %).
* Narrative flags (injuries, weather).

## Next Steps

### Millions

1. **Confirm Totals Flow** → Re‑run enrich + audit with Week 1, verify open\_total/current\_total/closing\_total/circa\_total appear.
2. **Export step** → Build millions\_export\_week.py to push clean planner slice to dashboard/Markdown.
3. **Pick Logging** → Simple CLI to log 5 ATS picks/week, appending to planner.
4. **Consensus Data** → Add pick % distributions for contrarian strategy.

### Survivor

1. **Finalize Spot Value Logic** → Calibrate weights vs test data.
2. **Sim Environment** → Run season simulations to test pick paths.
3. **Holiday Locks** → Ensure TG/Xmas picks flagged early.
4. **Export step** → Survivor weekly report (heatmap + pick rationale).

## Troubleshooting Guide

### Common Errors

* `→ Check script docstring; some flags differ (–odds` optional in enrich, required in fetch).
* **Duplicate** `\*\* columns\*\* → Always run with–drop\_overlaps` on enrich.
* **Totals missing in planner** → Ensure odds CSV had totals; run enrich *after* fetch.
* **Audit range errors** → Adjust --spread\_range/--dvoa\_range (must be quoted strings in PowerShell).

### PowerShell Tips

* Use backticks (`) for line continuation.
* Wrap ranges in quotes: --spread\_range "-20,20".
* Env vars: $env:ODDS\_API\_KEY="..." before fetch.

### Debugging

* Add --debug 1 to fetch for verbose API logs.
* Spot‑check outputs in diagnostics/.
* Compare planner vs odds snapshot for mismatches.

## Hand‑Off Notes

* **Principle:** single canonical odds fetcher; planner is source of truth; exporters should not recompute.
* **Audit first, export second.** Export gated on green audit.
* **Safe‑route strategy in place.** Totals baseline wired, no forks.

This documentation reflects state as of **Aug 19, 2025**. Next PM should continue with export scripts, consensus integration, and Survivor simulations.