

Team Per-Registrant Fee Precedence

Updated: Nov 24, 2025

This document captures the effective precedence for the per-registrant fee used when listing available teams and when resolving fees for a specific team. The logic lives in:

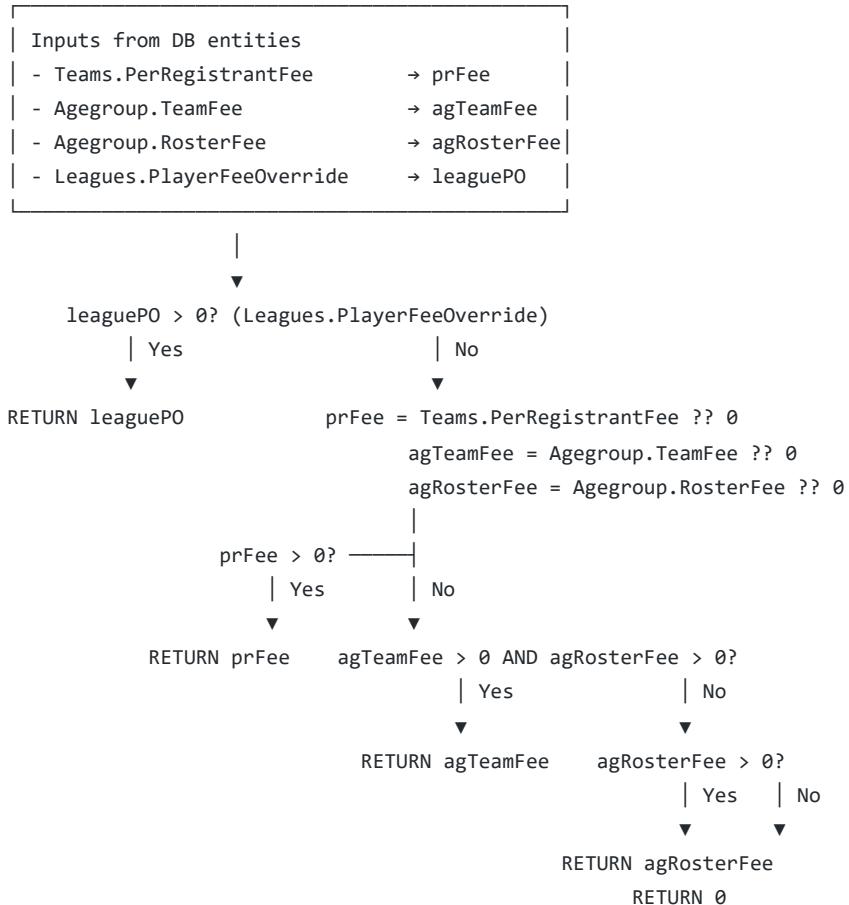
- File: `src/backend/TSIC.API/Services/TeamLookupService.cs`
- Method: `private static decimal ComputePerRegistrantFee(decimal? prFee, decimal? agTeamFee, decimal? agRosterFee, decimal? leaguePlayerFeeOverride)`

Inputs and Entity Origins

- `prFee` → `Teams.PerRegistrantFee`
- `agTeamFee` → `Agegroup.TeamFee`
- `agRosterFee` → `Agegroup.RosterFee`
- `leaguePlayerFeeOverride` → `Leagues.PlayerFeeOverride`

Flow Diagram

```
ComputePerRegistrantFee(prFee, agTeamFee, agRosterFee, leaguePlayerFeeOverride)
```



Precedence Summary (Highest → Lowest)

1. `Leagues.PlayerFeeOverride` (when > 0)
2. `Teams.PerRegistrantFee` (when > 0)
3. `Agegroup.TeamFee` (only when `Agegroup.RosterFee` is also > 0)
4. `Agegroup.RosterFee` (when > 0)
5. Otherwise 0

Related Calls

- Used when building team list: `GetAvailableTeamsForJobAsync` (computes `PerRegistrantFee` for each team)
- Used when resolving a single team: `ResolvePerRegistrantAsync`

Notes:

- All inputs are treated as nullable; nulls are normalized to 0 before comparisons.
- The league-level override provides a top-down mechanism to unify fees across teams within a league.