

Team Per-Registrant Fee Precedence

Updated: Nov 28, 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:

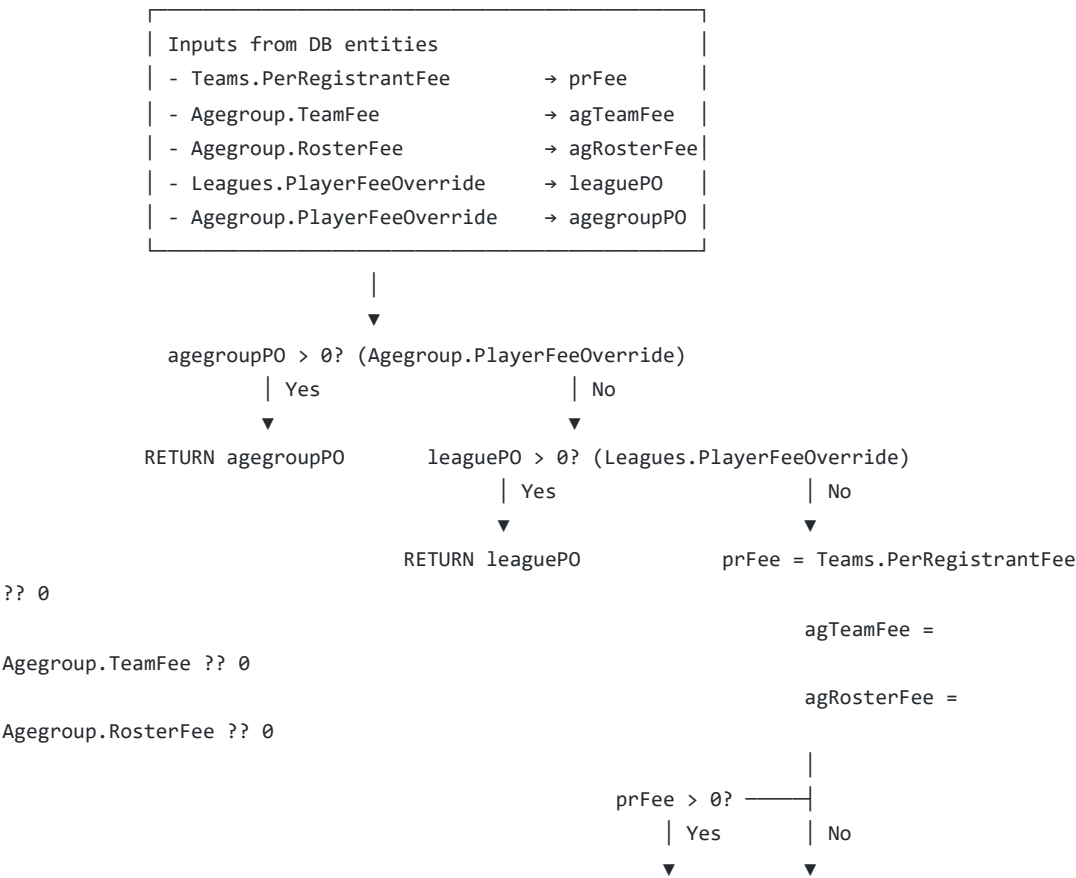
- File: `src/backend/TSIC.API/Services/TeamLookupService.cs`
- Method: `private static decimal ComputePerRegistrantFee(decimal? prFee, decimal? agTeamFee, decimal? agRosterFee, decimal? leaguePlayerFeeOverride, decimal? agegroupPlayerFeeOverride)`

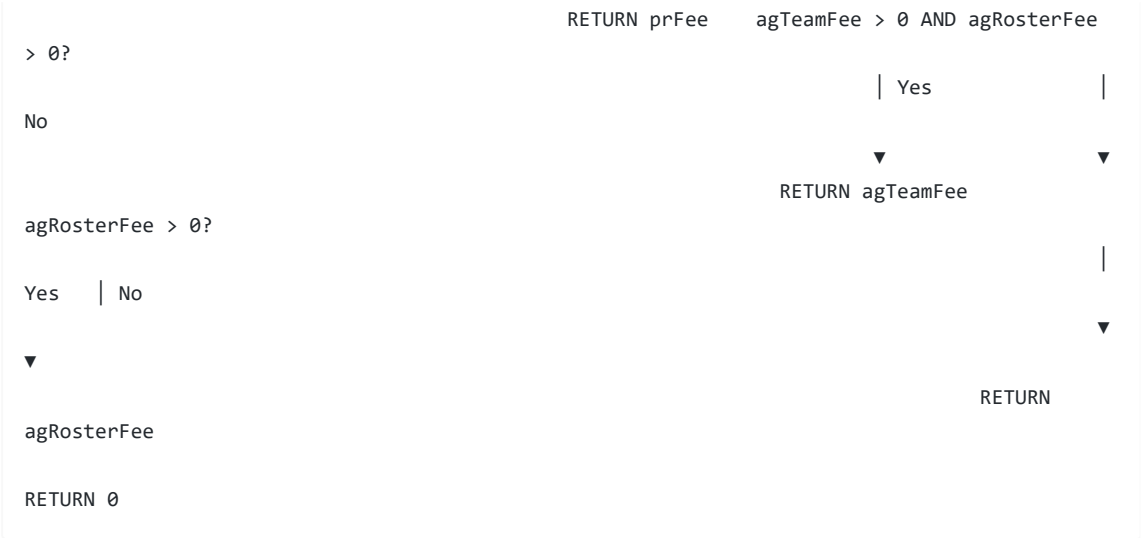
Inputs and Entity Origins

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

Flow Diagram

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





Precedence Summary (Highest → Lowest)

- 1. `Agegroup.PlayerFeeOverride` (when > 0)
- 2. `Leagues.PlayerFeeOverride` (when > 0)
- 3. `Teams.PerRegistrantFee` (when > 0)
- 4. `Agegroup.TeamFee` (only when `Agegroup.RosterFee` is also > 0)
- 5. `Agegroup.RosterFee` (when > 0)
- 6. 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 agegroup-level override provides mid-tier control, allowing specific age groups to override the league default without affecting team-specific pricing.
- The league-level override provides a top-down mechanism to unify fees across teams within a league.