DAI – RBC Adjusted PV Daily Updater v5.0

Prepared by: ChatGPT (GPT-5 Thinking)
Issue Date: 01 November 2025
Design Framework: DAI + RBC Framework v5.85 (October 2025)
Change Reference: Documentation – Detailed Design Creation (No behavioural change).

1. Functional Overview

This automation maintains the **Adjusted PV (Daily)** value throughout the day, using the formula Adjusted = clamp(Forecast_today + Bias_daily, 0 … max_kwh). It reacts at midnight, on HA start, and whenever either the Solcast forecast or the PV Daily bias changes. To prevent churn, it writes to the helper only when the change exceeds a small deadband. Version v5.0 removes the legacy safety-cap and end-of-day (EOD) lock so the value tracks live updates within the clamp. All calculations are in Jinja variables; no external scripts are invoked.

2. Entity Map

Entity ID / Var	Purpose	Direction
sensor.solcast_pv_forecast_forecast_today	Solcast PV forecast for today (kWh) → fc	Read
input_number.rbc_bias_pv_daily	Smoothed PV daily bias (kWh) → bias	Read
input_number.rbc_adjusted_pv_daily_kwh	Adjusted PV (kWh) output helper	Write
max_kwh (var=75)	Ceiling clamp for Adjusted PV (kWh)	Constant / Read
deadband (var=0.01)	Minimum absolute Δ required to write	Constant / Read

3. Trigger Matrix

Trigger ID	Condition	Purpose
t_midnight	Time = "00:00:10"	Initial daily refresh
t_ha_start	Home Assistant Start	Restore Adjusted PV after reboot

| t_forecast_change | State change of Solcast forecast | Reactive update |
| t_bias_change | State change of PV Daily bias | Reactive update |

4. Logic Flow

1. Read fc = sensor.solcast_pv_forecast_forecast_today and bias = input_number.rbc_bias_pv_daily.

2. Compute planned_unclamped = fc + bias.

3. Clamp to planned = min(max(planned_unclamped, 0), max_kwh).

4. Fetch prev_adj = input_number.rbc_adjusted_pv_daily_kwh and compute delta = planned – prev_adj.

5. If |delta| > deadband → write planned (rounded 2dp) to input_number.rbc_adjusted_pv_daily_kwh and logbook.log.

6. Otherwise log a no-op message (default path).

5. Guards and Safety

• No inverter writes – helper only.

• Clamp ensures Adjusted PV remains within 0 … 75 kWh.

• Deadband (0.01 kWh) avoids unnecessary churn and log spam.

• `mode: single` prevents overlap; calculations are idempotent.

• Visual-Editor-safe structure (service/target/data) in YAML actions.

6. Scheduling

Primary refresh at 00:00:10 and reactive updates on forecast or bias change. Also executes after HA restart to restore a correct Adjusted PV value.

7. Acceptance Tests

Test ID	Scenario	Expected Result
T1	Midnight run with change > deadband	Helper updated; logbook shows Planned(Forecast + Bias) summary.
T2	Forecast change during day	Adjusted PV recalculates

		and writes if $|\Delta| > 0.01$ kWh.
T3	Bias change during day	Adjusted PV recalculates and writes if $|\Delta| > 0.01$ kWh.
T4	No material change	Default log notes no write; helper unchanged.
T5	Reboot midday	HA-start branch recomputes and writes if needed within ~1 min.

8. Version Lineage & Governance

v5.0 – Removes safety-cap and end-of-day lock; Adjusted PV now tracks Forecast + Bias (clamped) with deadbanded writes.

Governance: Protected Architecture Mode v5.9; Two-Phase Code-Change Gate v2.2. This document records the live behaviour without introducing changes.

9. YAML Reference (Read-Only Excerpt)

triggers: [t_midnight @ "00:00:10", t_ha_start, t_forecast_change, t_bias_change]
variables: fc, bias, planned_unclamped, planned (clamped 0…75), prev_adj, deadband=0.01, delta, need_write
write → input_number.rbc_adjusted_pv_daily_kwh when |delta| > deadband; logbook.log with branch-specific message.

10. Compliance Checklist

☑ Entity ↔ ID alignment verified against the provided YAML.

☑ All triggers include IDs and a default no-op log path exists.

☑ Visual-Editor-safe action calls; no YAML anchors or includes.

☑ No inverter writes; Master-only rule remains unaffected.

☑ Design references present in automation description in live YAML.