TEMPLATE SYSTEM GUIDE

This guide explains how the reporting templates work so you can customise the output tabs without touching Apps Script code. All rendering is handled by `TemplateEngine.gs`, which reads three hidden template sheets in the workbook:

Template Sheet	Used For
ReportBodyTemplate	Individual property tabs.
ReportTotalsTemplate	Summary tab.
ReportAirbnbTemplate	Airbnb tab (only rendered when at least one property has Airbnb revenue in the period).

1. RENDERING BASICS

1. The templating engine copies the template sheet into the new report spreadsheet.
2. It replaces scalar placeholders throughout the sheet.
3. It expands any repeating row blocks and fills in row-level data.
4. It evaluates conditional flags and clears sections that should not appear.

Any placeholder that cannot be resolved is replaced with a blank and logged as a warning (visible in the Apps Script execution logs after the run).

2. SCALAR PLACEHOLDERS (`[PLACEHOLDER]`)

Use square brackets to insert single values that belong to the overall context (not a row dataset). The reporting scripts populate the following placeholders:

2.1 PROPERTY SHEET (`REPORTBODYTEMPLATE`)

Placeholder	Description
[PROPERTY NAME]	Canonical property name.
[REPORT LABEL]	Label entered in Entry Controls (B5).
[REPORT PERIOD]	Formatted start/end date range.
[TOTAL CREDITS]	Sum of credits for the property.
[TOTAL FEES]	Sum of fees for the property.
[TOTAL MARKUP]	Total markup revenue.
[TOTAL MAF]	MAF charge (credits × MAF % + unit count × 5 + admin fee when applied).
[TOTAL SECURITY DEPOSITS]	Sum of security deposits.
[TOTAL DEBITS]	Total debits (including markup and MAF).
[COMBINED CREDITS]/[TOTAL RECEIPTS]	Credits + security deposits (+ Airbnb income when applicable).
[DUE TO OWNERS]	Combined credits minus total debits.
[TOTAL TO PM]	Markup + MAF (+ Airbnb fee when applicable).
[UNIT COUNT]	Unique residential units (excludes CAM).
[ADMIN FEE APPLIED]	Yes / No.
[ADMIN FEE AMOUNT]	Flat admin fee amount (currency).
[AIRBNB TOTAL],[AIRBNB FEE]	Only set when the property is flagged *Has Airbnb*.

2.2 SUMMARY SHEET (REPORTTOTALSTEMPLATE)

In addition to [REPORT LABEL] and [REPORT PERIOD], the summary template can reference the aggregated totals computed across all properties:

Placeholder	Description
[SUMMARY PROPERTY COUNT]	Number of properties included in the run.
[SUMMARY TOTAL DUE TO OWNERS]	Sum of due-to-owner amounts.
[SUMMARY TOTAL TO PM]	Sum of totals payable to PM.
[SUMMARY TOTAL FEES]	Sum of property fee totals.
[SUMMARY TOTAL MAF]	Sum of overall MAF charges.
[SUMMARY TOTAL MARKUP]	Sum of markup revenue.
[SUMMARY TOTAL DEBITS]	Sum of total debits across properties.
[SUMMARY TOTAL CREDITS]	Sum of property credit totals.
[SUMMARY TOTAL SECURITY DEPOSITS]	Sum of security deposit totals.
[SUMMARY COMBINED CREDITS]	Sum of combined credits.
[SUMMARY TOTAL NEW LEASE FEES]	Aggregate of "New Lease Fee" debits.
[SUMMARY TOTAL RENEWAL FEES]	Aggregate of "Renewal Fee" debits.
[SUMMARY AIRBNB TOTAL]	Total Airbnb income across properties.
[SUMMARY AIRBNB FEE]	Total Airbnb collection fees.

2.3 AIRBNB SHEET (`REPORTAIRBNBTEMPLATE`)

Only two scalar placeholders are provided: `[REPORT LABEL]` and `[REPORT PERIOD]`.

3. REPEATING ROW BLOCKS (`[[ROW …]]`)

Row blocks allow you to duplicate one or more template rows for every item in a dataset.

```
[[ROW transactions group=unit sort=date groupTail=CAM]]
…template rows containing {field} placeholders…
[[ENDROW]]
```

- `transactions` is the dataset name supplied by the script. Property tabs expose the dataset `transactions`, the summary tab uses `summary`, and the Airbnb tab uses `airbnb`.
- Placeholders inside the row block use curly braces `{field}` to reference row fields.
- You may include formulas, formatting, and merged cells inside the block – the engine copies the range for each row before swapping values.

3.1 ROW ATTRIBUTES

Attribute	Applies To	Description
`group=<field>`	Optional	Groups consecutive rows by a field and blanks duplicate values after the first occurrence. Field names are matched case-insensitively.
`sort=<field>`	Optional	Sorts each dataset by the specified field (ascending). When `group` is also supplied, grouping is applied first and the sort happens within grouped data.
`groupTail=value1,value2,…`	Optional	After sorting, moves any rows whose group field matches one of the listed values to the end (case-insensitive). Useful for putting CAM at the bottom while keeping other values alphabetised.
`sortTail=value1,value2,…`	Optional	Similar to `groupTail` but based on the sort field instead of the group field.

3.2 DATASET FIELD REFERENCE

- **Property transactions:** {unit}, {credits}, {fees}, {debits}, {securityDeposits}, {date}, {explanation}, {markupIncluded}, {markupRevenue}, {internalNotes}.
- **Summary rows:** {property}, {dueToOwners}, {totalToPm}, {totalFees}, {totalMaf}, {totalMarkup}, {combinedCredits}, {newLeaseFees}, {renewalFees}.
- **Airbnb rows:** {property}, {income}, {collectionFee}.

If a field requested in the template does not exist, the engine clears the value and records a warning.

4. CONDITIONAL CONTENT

The templating engine supports several ways to show or hide content based on flags. Flags are booleans provided alongside placeholders (`has_airbnb`, `has_admin_fee`, etc.).

Syntax	Example	Behaviour
`[[IF flag]] … [[ENDIF]]`	`[[IF has_airbnb]] … [[ENDIF]]`	Entire rows between the markers are kept only when the flag is TRUE; otherwise they are deleted.
`[if=flag]…[/if]`	`[if=has_admin_fee]Admin fee details[/if]`	Inline block. If the flag is FALSE, the entire cell is cleared.
`['text' | if=flag]`	`['Admin fee applied' | if=has_admin_fee]`	Inserts the quoted text when the flag is TRUE, otherwise nothing. Double quotes or single quotes are both allowed.
`[PLACEHOLDER if=flag]`	`[ADMIN FEE AMOUNT if=has_admin_fee]`	Resolves the placeholder only when the flag is TRUE.

5. SCALAR PLACEHOLDER RULES

- Placeholders resolve in this order: row data → scalar placeholders → warnings.
- Placeholder names are case-sensitive.
- Use `{field}` inside row blocks and `[PLACEHOLDER]` outside (or for scalars inside a block).
- When the engine cannot resolve a placeholder, it logs `Missing value for …` in the execution logs and leaves the cell blank.

6. FLAGS SUPPLIED BY THE SCRIPTS

Context	Flag	Meaning
Property tab	`has_airbnb`	TRUE when the property is marked *Has Airbnb*.
Property tab	`has_admin_fee`	TRUE when an admin fee was applied to that property for the run.
Summary tab	*(none currently)*	
Airbnb tab	*(none currently)*	

You can introduce additional flags by modifying the Apps Script if needed, but the above are the ones available out of the box.

7. BEST PRACTICES

- Always keep the `[[ROW …]]` and `[[ENDROW]]` markers in column A with no extra content on those rows.
- When duplicating or moving template rows, ensure all markers and placeholders move together.
- Use named ranges or cell references for totals when possible; the scripts supply totals as formatted currency so you can reference those cells elsewhere on the sheet.
- Test template changes by running **Generate Report** for a narrow date range. Review the Apps Script log for warnings to catch missing placeholders or invalid fields.
- If you want to add new aggregated values, update `Reports.gs` to supply additional placeholders and

document them in this file.

With these conventions, you can safely tailor the output layout while relying on the templating engine to populate data and formatting automatically.