



Timebars — Complete Reference Guide

Version: 1.2 **Last Updated:** 2026-09-10

Every help topic for **Timebars**, compiled into one document: the product guide, followed by the guides shared across Agilebars, Timebars and Costbars.

This file is **generated** from the individual guides in `/docsHelp` by `tools/buildCompleteGuides.py`. Edit those guides and rebuild — changes made here are lost on the next run.

About the cross-references. Each part keeps the links it had as a standalone guide, so a reference to another guide points at its published article online. In this document, that guide is one of the parts listed below.

Contents

1. [The Timebars Product](#)
2. [The User Interface](#)
3. [Data Model and Scheduling Engine](#)
4. [Data Synchronization, Backup, Recovery and Retention](#)
5. [Forms, Reports and Graphs](#)
6. [Risks, Issues and Change Requests](#)
7. [How to Use Ask AI](#)
8. [Supply and Demand Grids](#)
9. [Text Notifications](#)
10. [Cloud Publishing](#)
11. [The Personal Dashboard](#)
12. [The Enterprise Dashboard](#)

What changed in version 1.2 (2026-09-10)

- **Part 10 — Cloud Publishing** rebuilt from Cloud Publishing and Bulk Operations Guide v2.2. New section 11.5, "Two things that travel outside the mapping": the OpenProject project **description** and project **status** are built-ins rather than custom fields, so the Schema sheet cannot describe them and the sync now writes both itself — the description from `tbMDDescription`, the status from the project's Health (`tbMDHealth`) through a fixed seven-row table. Both are in addition to the custom fields those values already travel to, and neither needs a workbook change. Work packages are unaffected.

- One row added to the Cloud Publishing troubleshooting table, for a project status in OpenProject that does not look like the Timebar Status.

Earlier versions: 1.1 (2026-09-10), 1.0 (2026-09-04).

Part 1 — The Timebars Product

Timebars User Guide

This is the master user guide for Timebars. It is complete for everything Timebars does on its own, and is used **alongside the common guides**, which cover what all three products share — the canvas and the user interface, the data model and scheduling engine, configuration and picklists, spreadsheet sync and data control, reports, risks and issues, cloud publishing and notifications. Agilebars and Costbars each have their own master guide in the same way.

Introduction

Timebars is the traditional project management and resource scheduling edition of the Timebars product family, designed for Waterfall/Traditional Project Management practitioners who need comprehensive resource allocation, multi-level project hierarchies, and sophisticated scheduling capabilities.

While Agilebars focuses on sprint planning with Kanban boards and Costbars provides portfolio management for project selection, Timebars delivers the execution layer—helping you answer critical questions like:

- How do I allocate resources across multiple projects?
- What is my resource capacity vs. demand over the next 6 months?
- Which resources are over-allocated and need workload balancing?
- How do I schedule complex multi-level project hierarchies?
- What happens to dependent tasks when I move predecessor tasks?

What Makes Timebars Different

Timebars extends the core platform with specialized resource scheduling and traditional project management capabilities:

Advanced Resource Management

- **Resource Pool** – Centralized list of named and generic resources with roles, skills, rates, and availability
- **Resource Allocator** – Drag-and-drop resource assignment to tasks with visual availability checking
- **Allocation Bars (Gold L5)** – Visual representation of resource assignments with percent allocation
- **Supply vs. Demand Grids** – Comprehensive capacity planning showing FTE availability vs. project requirements

Flexible Project Hierarchy

- **5-Level Hierarchy** – Portfolio → Project → SubProject → Task → Allocation (color-coded: Dark Grey, Green, Orange, Blue, Gold)
- **4-Level Hierarchy** – Portfolio → Project → Task → Allocation (omit Orange SubProject level)

- **3-Level Hierarchy** – Project → Task → Allocation (omit Portfolio and SubProject levels)
- Hierarchy enforcement prevents structural violations during drag-and-drop operations

Sophisticated Scheduling Engine

- **Automatic Calculation** – Dates, duration, hours, and costs calculated from resource allocations
- **Planned/Forecast/Actual Tracking** – Distinguishes between baseline, forecast, and actual values based on Report Date
- **Task Relationships** – Predecessor-successor links that cascade schedule changes
- **Constraints** – Pin tasks to specific dates while maintaining relationship logic

Timephased Canvas

- **Better than Gantt Charts** – More screen real estate, improved productivity, visual clarity
 - **Front-load/Back-load Resources** – Drag allocation bars to schedule resources within task timeframes
 - **Weekly or Monthly Timescale** – Switch between granular weekly view and strategic monthly view
 - **Percent Allocation Graphs** – Visual indicators of resource availability and utilization
-

Getting Started with Timebars

Prerequisites

Before creating resource-loaded schedules, familiarize yourself with:

- **User Interface Guide** – Canvas navigation, filters, bar creator, editing, and general interface usage
- **Timebars Presentation** – Review the [comprehensive presentation](#) and [functional benefits](#) articles

Recommended Workflow

1. Use Spreadsheet Sync for Efficient Data Entry

- Click the **Hamburger Icon** to download Excel or LibreOffice spreadsheet template
- Populate the spreadsheet with your project data
- Drag and drop the file onto the Canvas to import

2. Populate Your Resource Pool (see Resource Pool Setup below)

- Use the Resources tab in the spreadsheet
- Define resource IDs, names, roles, skills, rates, and availability
- Import via drag-and-drop

3. Create Your Project Hierarchy

- Use the Bar Creator to add Projects (Green L2), SubProjects (Orange L3), and Tasks (Blue L4)
- Or import via spreadsheet sync

4. Allocate Resources to Tasks

- Use the Resource Allocator to assign resources
- Create Allocation bars (Gold L5) via drag-and-drop

5. Review and Optimize

- Check resource utilization with Supply vs. Demand grids
- Adjust schedules to balance resource capacity
- Create relationships and constraints as needed

⚡ Get Started Fast with Ask AI

Before building bars by hand, consider letting **Ask AI** create your project structure for you. It is the fastest way to go from an idea to a working, resource-loaded schedule.

You don't start from a blank page. When your Timebars canvas is empty, the app automatically places a starter top-level bar near the top:

L1 Get Started here using AI, click the L1 ID below

This is a ready-made **Portfolio (L1)** running from today for about six months. To begin:

1. **Click the bar's ID** to open its status pop-up form.
2. Click the ⚡ **sparkle icon** (the Ask AI button) on that form.
3. **Describe your project** in plain English — paste a Business Case, or just type a name and description — and click **Create**.

From there you work down the hierarchy, clicking the ⚡ icon on each bar:

- **Portfolio → Project (L2):** describe the project; optionally have AI fill the Project Charter and add a Resource Plan.
- **Project → Work Packages (L3), Tasks (L4) and Milestones:** AI breaks the work down; tick **Assign people** to staff each Task with the best-fit person from your Resource Pool.
- **Work Package → Tasks and Milestones:** with an option to assign people, plus a **Create Allocations** button to staff existing tasks.
- **Task → People (Allocations, L5):** type the roles you need ("add a developer and a tester") and AI assigns matching people, checking they are not over-allocated.

After AI finishes, **refresh the page (F5)** to see the new bars on your timeline.

 For full step-by-step instructions, examples and tips, see **Common_06_How_To_Use_Ask_AI** ("How to Use Ask AI") in the Common Help Topics.

The Timebars Hierarchy

Timebars conforms to three different hierarchies with built-in enforcement rules that prevent structural violations during drag-and-drop operations. Each level is color-coded for easy recognition.

Color-Coded Hierarchy Levels

Level	Color	Name	Description
L1	Dark Grey	Portfolio	Top-level grouping of related programs/projects
L2	Green	Project/Program	Always a Project in all hierarchies
L3	Orange	SubProject	Optional intermediate level for complex projects

Level	Color	Name	Description
L4	Blue	Task	Always a Task in all hierarchies (includes milestones and gates)
L5	Gold	Allocation	Always a Resource Allocation in all hierarchies

The 5-Level Hierarchy (Maximum)

When to use: Large, complex portfolios with multiple programs and sub-projects

```
L1 - Portfolio (Dark Grey)
  L2 - Project/Program (Green)
    L3 - SubProject (Orange)
      L4 - Task (Blue)
        L5 - Resource Allocation (Gold)
```

Example:

- L1: Digital Transformation Portfolio
 - L2: CRM Implementation Program
 - L3: Sales Module SubProject
 - L4: Design Sales Dashboard (Task)
 - L5: Joe Invent @ 50% (Allocation)

The 4-Level Hierarchy (Medium)

When to use: Most small and medium-sized projects where sub-projects aren't needed

```
L1 - Portfolio (Dark Grey)
  L2 - Project/Program (Green)
    L4 - Task (Blue)
      L5 - Resource Allocation (Gold)
```

Same as 5-level except Orange L3 SubProject bars are not used.

The 3-Level Hierarchy (Minimum)

When to use: Simple projects or single-project scheduling

```
L2 - Project/Program (Green)
  L4 - Task (Blue)
    L5 - Resource Allocation (Gold)
```

Same as 5-level except Dark Grey L1 Portfolio and Orange L3 SubProject bars are not used.

Hierarchy Enforcement

The system automatically enforces hierarchy rules:

- You cannot place a Task (Blue L4) directly under a Portfolio (Grey L1) – must go under a Project (Green L2) or SubProject (Orange L3)
 - You cannot place an Allocation (Gold L5) anywhere except under a Task (Blue L4)
 - Drag-and-drop operations that violate hierarchy rules are prevented
-

The Scheduling Engine

Timebars uses a sophisticated scheduling engine that distinguishes itself from Agilebars through its unique approach to calculating dates, durations, hours, costs, and percent complete.

How the Scheduling Engine Works

The engine behavior changes based on a task's position relative to the **Report Date** (today's date or a baseline date you set):

1. Future Tasks (Start Date > Report Date)

When: You move or resize a bar that starts to the right of the Report Date

Calculations:

- Computes **forecast dates** (start and finish)
- Calculates **forecast duration**
- Calculates **forecast work** (hours)
- Calculates **forecast cost**

Use Case: Planning future work

2. In-Progress Tasks (Task Straddles Report Date)

When: You move a bar so it straddles the Report Date (starts before, finishes after)

Calculations:

- **Start date** becomes an **actual date** (locked in the past)
- **Finish date** remains a **forecast date** (still in the future)
- Calculates **remaining duration** from Report Date to finish
- Calculates **actual work** (hours consumed before Report Date)
- Calculates **actual cost** (costs incurred before Report Date)
- Calculates **percentage completion** based on actual vs. total work

Use Case: Tracking active tasks

3. Completed Tasks (Finish Date < Report Date)

When: You move a bar so its finish is before the Report Date

Calculations:

- **Start and finish dates** become **actual dates**

- **Remaining duration** becomes zero
- Calculates **actual work** (total hours for completed task)
- Calculates **actual cost** (total cost for completed task)
- Sets **percentage completion to 100%**

Use Case: Recording completed work

Work Hours Calculation Formula

By default, the scheduling engine calculates hours using:

$$\text{Hours} = \text{Percent Allocated} \times \text{Work Day Hours} \times \text{Remaining Duration}$$

Example:

- Resource is assigned to a task at **50% allocation**
- Resource works an **8-hour day** (from resource calendar)
- Task is **10 days long**
- **Calculation:** $8 \times 0.5 \times 10 = 40$ hours

Percent Allocated

Set during assignment:

- Determined by the **Percent Available** value in the Resource Pool
- Default is typically 50% (resource splits time between this task and other work)
- Can be 100% for dedicated resources

Changing percent allocated:

1. Click the **Edit link** at the bottom left of an Allocation bar
2. The **Hours Calculator popup** launches
3. Enter a new **Percent Allocated** value
4. Optionally change the **Workday** hours
5. Hours and costs recalculate automatically

Cost Calculation

Costs are calculated as:

$$\text{Cost} = \text{Calculated Hours} \times \text{Pay Rate}$$

- **Pay Rate** comes from the Resource Pool (set per resource in the Resources spreadsheet tab)
- Costs update automatically when hours or pay rates change
- Supports planned, forecast, and actual cost tracking

Reviewing Dates, Hours, and Costs

Cost Schedule Popup:

1. Click on the lower left portion of any bar (e.g., "T:3773...")
2. The **Cost Schedule Popup** launches
3. View complete details:
 - **Planned** dates, costs, hours (from baseline)
 - **Forecast** dates, costs, hours (current projection)
 - **Actual** dates, costs, hours (completed work)
4. Traffic light indicators show status (Green/Yellow/Red)
5. Links to Burndown Chart and Metadata View

Hours Calculator Popup (Gold Allocation bars only):

1. Click on the Allocation bar second line (e.g., "L5:2101")
 2. The popup displays the **Hours Calculator** link
 3. Click to see dynamic hours and cost calculations
 4. Values update in real-time as you drag the bar
 5. Changes save automatically when you drop the bar
-

Resource Pool Setup

The Resource Pool is your centralized list of resources (people or generic roles) who perform work on tasks. This pool maintains metadata such as role, skill, pay rate, and availability.

How to Populate the Resource Pool

Step 1: Download the Spreadsheet

1. Click **Hamburger Icon** > Download Excel or LibreOffice SS
2. Choose the appropriate format for your environment

Step 2: Open the Resources Tab

1. Open the downloaded spreadsheet
2. Navigate to the **Resources** tab
3. You'll see column headers for resource metadata

Step 3: Enter Your Data

- Type directly into the spreadsheet
- Or paste data from your enterprise system (HR database, ERP, etc.)
- Fill in the required fields (see table below)

Step 4: Import the Data

1. Save the spreadsheet
2. Drag and drop the file onto the Canvas
3. The system imports your resource pool

Step 5: Verify the Import

1. Click the **Resources icon** on the main menu

2. The Resource Allocator popup opens
3. Verify your resources are listed correctly

Resource Pool Fields

Field	Required	Description	Example
ResID	Yes	Unique resource identifier	700, 701, 702
Name	Yes	Full resource name	Joe Invent, Ally HR, Julia Finance
Workday Calendar	Yes	Hours per work day	8
Qty Resources (Supply)	Yes	Number of FTE (typically 1 for named resources)	1
Pay Rate	No	Hourly or daily rate for cost calculations	50
Primary Role	Recommended	Job function or role	R&D, HR SME, Finance SME
Primary Skill	Recommended	Core competency	Testing, IT Architecture, Procurement
Name Short	Recommended	Initials or abbreviation	Jl, AH, JF
Labour Type	Yes	Human or Generic	Human, Generic
Default Percent Available	Recommended	Default allocation percentage	50 (means 50% available)

Example Resources Spreadsheet Tab

ResID	Workday Calendar	Qty Resources	Pay Rate	Name	Primary Role	Primary Skill	Name Short	Labour Type	Default Percent Available
700	8	1	50	Joe Invent	R&D	Testing	Jl	Human	50
701	8	1	50	Ally HR	HR SME	IT Architecture	AH	Human	50
702	8	1	50	Julia Finance	Finance SME	Procurement	JF	Human	50

Resource Allocator

The Resource Allocator is a movable popup that displays your Resource Pool and enables drag-and-drop resource assignment to tasks.

Opening the Resource Allocator

Method 1: Main Menu

1. Click **Main Menu > Resources**
2. The Resource Allocator popup opens

Method 2: Keyboard Shortcut (if configured)

What You See:

- List of all resources from your Resource Pool
- Metadata columns (Name, Role, Skill, Rate, Availability)
- Search box for filtering
- Check boxes to filter by Human/Generic resources
- Width Icon to show/hide additional metadata columns
- Refresh Data button to update usage calculations

How to Create an Allocation Bar (Gold L5)

Step 1: Open the Resource Allocator

- Click **Main Menu > Resources**

Step 2: Find the Desired Resource

- Use the search box if needed
- Click column headings to sort (e.g., sort by Role to find all Developers)

Step 3: Drag the Resource

- Click and hold on a resource row
- Drag it out of the allocator popup

Step 4: Drop on a Task

- Drop the resource onto a Task (Blue L4 bar)
- A new **Gold L5 Allocation bar** appears below the task

Step 5: Position the Allocation

- The allocation bar spans the same timeframe as the parent task by default
- Drag the allocation bar left/right to front-load or back-load the resource within the task timeframe
- Resize the allocation bar to shorten the assignment duration

Resource Allocator Features

Search:

- Not case-sensitive
- Searches all fields (Name, Role, Skill, etc.)
- Returns matching rows instantly

Filtering:

- ☒ **Human Resources** – Show only named people

- ☒ **Generic Resources** – Show only role-based placeholders
- ☒ **All Resources** – Show both types

Sorting:

- Click any column header to sort by that field
- Click again to reverse sort order
- Useful for grouping by Role, Department, or Skill

Width Icon:

- Click to show additional metadata columns
- View Department, Location, Secondary Skills, etc.
- Click again to collapse back to default view

Refresh Data:

- Click to recalculate resource demand
- Ensures Resource Demand Charts include latest allocations
- Run before generating usage reports

Checking Resource Availability

Method 1: Double-Click for Availability Graph

1. Ensure Canvas is in **Weekly timescale mode**
2. Double-click a resource row in the Resource Allocator
3. Yellow vertical bars appear at the top of the Canvas (below weekly date values)
4. Each bar represents **percent allocated** for that week

Interpreting the Availability Graph:

- **Short yellow bar (e.g., 25% high)** – Resource has 75% available capacity
- **Medium yellow bar (e.g., 50% high)** – Resource at 50% utilization
- **Tall yellow bar (e.g., 100% high)** – Resource fully allocated
- **Bar exceeding 100%** – Resource over-allocated (needs rebalancing)

Method 2: Click Usage Icon

1. Click the **Usage icon** on the Resource Allocator
2. Opens detailed **Resource Usage** reports
3. Shows Supply vs. Demand grids and charts
4. See [Resource Supply vs. Demand Analysis](#) below for complete details

Method 3: Resource Usage Reports

1. Click **Main Menu > Reports > Resource > Resource Usage**
2. View demand by hours or FTE
3. Choose Weekly or Monthly view
4. Group by Resource Name, Project, or Role
5. See tabular grids and colorful bar graphs

Resource Supply vs. Demand Analysis

The Supply vs. Demand grid is a strategic resource management tool that provides visibility into resource capacity planning and allocation across projects.

What Are Supply and Demand?

Supply represents the available capacity of resources in your organization:

- Total FTE (Full-Time Equivalent) or hours that resources can work
- Based on employment contracts, calendars, and availability settings
- Sourced from the Resource Pool (tbResources store)

Demand represents the actual resource requirements from project allocations:

- How much capacity is being consumed by tasks and projects
- Calculated from resource assignments across all active projects
- Shows where resources are committed

The Gap: The difference between supply and demand reveals critical resource management issues:

- **Over-allocation:** When demand exceeds supply (resources over-committed, projects at risk)
- **Under-utilization:** When supply exceeds demand (unused capacity available)
- **Future capacity planning:** Identifying when you'll need to hire, train, or contract additional resources
- **Portfolio decisions:** Understanding if you have capacity to take on new projects

Who Uses These Grids?

Resource Managers:

- Balance workloads across the organization
- Identify hiring needs 3-6 months in advance
- Resolve resource conflicts and over-allocations
- Optimize resource utilization rates

Project Managers:

- Understand if sufficient resources are available
- Plan schedules based on resource availability
- Request additional resources when gaps identified
- Adjust timelines when resources over-committed

Portfolio Managers:

- Assess organizational capacity for new work
- Make go/no-go decisions on project proposals
- Prioritize projects based on resource constraints
- Plan strategic initiatives around availability

Executive Leadership:

- Make hiring and budgeting decisions
- Understand organizational capacity at high level

- Evaluate departmental utilization and productivity
- Align business strategy with resource capacity

The Five Summary Rows

At the top of the grid, five critical rows provide aggregated metrics:

1. Supply (FTE) Row – Blue Background

What it shows: Total available resource capacity across all resources in the pool

Data source:

- Sourced from **tbResources** store (your Resource Pool)
- Fields: **tbResMonth1** through **tbResMonth24**
- Each resource has monthly FTE availability defined (typically 1.0 for full-time)
- Sums ALL resources regardless of the groupBy selection

Example: If you have 6 full-time resources (1.0 FTE each), the supply shows 6.0 FTE per month

Timing considerations:

- Resource start dates (**tbResStart**) are factored in – new hires only contribute from their start month
- Resource finish dates (**tbResFinish**) are factored in – departing resources stop contributing
- Allows forward planning for known hiring or departures

2. Demand (FTE) Row – Green Background

What it shows: Total resource requirements from all project allocations

Data source:

- Calculated from **tbResCalcsUsage** store
- Generated by aggregating task allocations across all projects
- Fields: **tbResCalcMonth1** through **tbResCalcMonth32**
- Respects the groupBy filter – only sums demand for filtered resources/projects/roles

Calculation process:

1. Task allocations split into monthly buckets
2. Each task's total work distributed across its duration
3. Work hours per day calculated: $\text{totalWork} / \text{numberOfWorkDays}$
4. Monthly hours summed for each resource
5. Converted to FTE: $\text{monthlyHours} / (8 \text{ hours/day} \times 20 \text{ work days/month})$

3. Variance (S-D) Row – Yellow Background

What it shows: The difference between Supply and Demand (Supply minus Demand)

Color coding:

- **Green text (positive variance):** Excess capacity – resources under-utilized
- **Red text (negative variance):** Capacity deficit – resources over-allocated

Strategic interpretation:

Variance	Meaning	Action
Small negative (< 0.2 FTE)	Minor over-allocation	Manageable with overtime
Large negative (> 0.5 FTE)	Serious capacity problem	Hire or delay projects
Consistent positive	Opportunity for more work	Take on new projects or reduce headcount

4. Month Header Row – Gray Background

What it shows: The calendar months for each column

Format: "MMM YYYY" (e.g., "Sep 2025", "Oct 2025")

Start date: Begins with the `apStatusDate` from Admin Panel configuration

- Typically set to current month or project baseline date
- Allows historical analysis by setting status date in the past
- Enables future planning by projecting 24+ months forward

5. Resource/Project/Role Header Row – White Background

What it shows: The first column header, which changes based on `groupBy`:

- "Resource ID/Owner" when grouping by `ResourceName`
- "Project Name" when grouping by `Project`
- "Role" when grouping by `Role`

View Modes and Configurations

FTE vs Hours Toggle

FTE (Full-Time Equivalent):

- Standard view for strategic planning
- 1.0 FTE = one full-time resource for one month
- Easier to understand capacity at a glance
- Recommended for executive reporting and high-level planning

Hours:

- Detailed view for tactical planning
- Shows actual work hours (e.g., 160 hours/month for 1.0 FTE)
- Useful for detailed project scheduling
- Helps with billing and time-tracking analysis

Conversion formula:

$$\text{FTE} = \text{Hours} / (\text{Calendar hours per day} \times \text{Working days per period})$$

For monthly:

$$\text{FTE} = \text{Hours} / (8 \text{ hours/day} \times 20 \text{ days/month}) = \text{Hours} / 160$$

For weekly:

$$\text{FTE} = \text{Hours} / (8 \text{ hours/day} \times 5 \text{ days/week}) = \text{Hours} / 40$$

Monthly vs Weekly Toggle

Monthly View:

- Displays resource capacity and demand by calendar month
- Better for long-term strategic planning (6-24 months out)
- Smooths out weekly variations
- Recommended for portfolio planning and hiring decisions
- Supply and demand totals are shown

Weekly View:

- Displays resource capacity and demand by work week (Monday-Friday)
- Better for short-term tactical planning (1-3 months out)
- Shows week-to-week fluctuations
- Useful for identifying specific bottleneck weeks
- Currently shows demand data only (no supply totals)

Why monthly is preferred: Resource supply data is typically managed monthly (hiring, contracts), monthly aligns with financial cycles, and weekly granularity creates noise in capacity planning.

Group By Options

The groupBy selection controls how demand rows are organized (does NOT affect supply totals).

Group By: Project

Use case: Understanding capacity allocated to each project

What you see:

- One row per project showing demand
- Supply row shows total capacity across ALL resources
- Variance shows if organization has capacity for all projects combined

Strategic questions answered:

- Which projects are consuming the most resources?
- Do we have enough capacity for all active projects?
- Can we take on a new project without hiring?

Example scenario: You have 3 projects (CRM Migration, ERP Upgrade, Website Redesign). The grid shows how much FTE each project demands per month, and whether your total resource pool can support all three simultaneously.

Group By: Resource Name

Use case: Monitoring individual resource utilization

What you see:

- One row per named resource (e.g., "Joe Invent", "Ally HR")
- Each row shows that resource's project allocations
- Supply row still shows organizational total (all resources)

Strategic questions answered:

- Which individuals are over-allocated?
- Who has available capacity for new work?
- Are workloads balanced across the team?

Example scenario: Joe Invent shows 1.2 FTE demand in March (over-allocated by 20%), while Julia Finance shows 0.6 FTE (has 40% available capacity). You can rebalance by moving some of Joe's work to Julia.

Group By: Role

Use case: Understanding capacity by job function or skillset

What you see:

- One row per role (e.g., "Developer", "Engineer", "Finance SME")
- Shows aggregated demand for all resources in that role
- Critical for skills-based capacity planning

Strategic questions answered:

- Do we have enough developers vs. testers?
- Which role is the bottleneck in our organization?
- Should we hire for a specific skillset?

Example scenario: Your "Developer" role shows 3.5 FTE demand but you only have 3.0 FTE supply of developers. You need to hire an additional 0.5 FTE developer (potentially a contractor) or delay some development work.

Named vs. Generic Resources

The system supports two types of resources that work together for flexible capacity planning:

Named Resources (Humans)

What they are: Individual people with specific identities

Characteristics:

- **tbResLabourType:** "Human"
- Unique identifier: **tbResID** (e.g., "700", "701", "702")
- Named individuals: "Joe Invent", "Ally HR", "Julia Finance"
- Assigned a Primary Role: "R&D", "HR SME", "Finance SME"
- Quantity is typically 1

Why we need them:

- Track specific people's workloads and allocations
- Manage vacation, training, and individual availability
- Support performance management and utilization reporting
- Enable skills-based assignment to tasks

Primary Role assignment: Each named resource is assigned one Primary Role (e.g., Joe Invent has Primary Role = "R&D"). This categorizes the resource by function for reporting and role-based filtering.

Generic Resources

What they are: Placeholder resources representing a pool of similar workers

Characteristics:

- **tbResLabourType:** "Generic"
- Role-based identifier: **tbResID** (e.g., "800", "818", "830")
- Generic names matching the role: "R&D", "HR SME", "Finance SME"
- Assigned a Primary Role that matches the resource name
- Quantity can be > 1 (e.g., "2" means 2 FTE of this role)

Why we need them:

- Early project planning before specific people are assigned
- Representing contractor pools or offshore teams
- Capacity planning when you don't know who specifically will do the work
- Future hiring – you know you need "3 developers" but haven't hired them yet

How quantity works: If Generic Resource "Developer" has **tbResQuantity:** "2" and **tbResMonth1:** "2.00", this means you have 2.0 FTE of developer capacity available (could be 2 full-time developers, 4 half-time developers, or any combination totaling 2.0 FTE).

Why Have Both Named and Generic?

This dual approach supports the full project lifecycle:

Early Planning Phase:

- Use generic resources ("we need 2 developers and 1 tester")
- Estimate capacity needs without knowing specific people
- Justify hiring or budget requests

Assignment Phase:

- Replace generic allocations with named resources
- Assign Joe, Chris, and John to specific tasks
- Track actual people doing the work

Capacity Planning:

- Generic resources fill the gap between demand and available named resources
- If named resources show 4.5 FTE demand but only 4.0 FTE supply, the 0.5 FTE gap might be filled by generic "Developer" resource (representing a contractor)

Reporting Flexibility:

- Group by Resource Name: See named resources only
- Group by Role: See combined named + generic resources by function
- Supply totals: Includes both named and generic for total capacity

Best Practices for Resource Analysis

For Accurate Supply Data:

1. Keep resource pool current – update start/finish dates when people join or leave
2. Set realistic availability – use monthly values to reflect part-time schedules or known vacations
3. Maintain generic resources – keep quantities aligned with actual contractor pools or hiring plans

For Accurate Demand Data:

1. Assign resources to all tasks – unassigned tasks don't appear in demand calculations
2. Set realistic task durations – overly long tasks spread demand too thin; overly short tasks create spikes
3. Use appropriate work estimates – review work values to ensure they reflect actual effort
4. Assign Primary Roles – ensure all resources have roles for accurate role-based reporting

For Effective Analysis:

1. Review monthly in FTE first – start with monthly FTE view for strategic planning
2. Identify persistent negative variances – months showing red variance consistently indicate structural capacity problems
3. Look ahead 3-6 months – use the grid to identify future capacity needs before they become urgent
4. Compare groupBy views:
 - Project view: Find projects causing over-allocation
 - Resource view: Find specific individuals who are over-committed
 - Role view: Find which job functions need hiring
5. Set the right status date – use apStatusDate to focus on relevant time periods

Common Patterns and What They Mean

Pattern	Meaning	Action
Supply and demand equal (variance ≈ 0.0)	Optimal utilization	Monitor closely to maintain balance
Consistent negative variance across multiple months	Structural under-capacity	Hire, contract, or reduce project scope
Spiky negative variance in specific months	Temporary bottleneck or project crunch time	Shift tasks to other months or bring in temporary contractors
Large positive variance	Under-utilization or over- capacity	Take on new projects, reduce headcount, or invest in training/innovation
Variance switches from positive to negative	Capacity consumed by project ramp-up	Normal pattern, but verify project schedules are achievable

Creating Relationships and Constraints

Task Relationships (Predecessor-Successor)

Relationships instruct the scheduling engine to reschedule the successor task by the same amount of time that the predecessor task was moved on the Canvas.

How to Create a Relationship:

1. Grab the **beginning** of the desired successor task
2. Drag it over the **ending** of the target predecessor task
3. Wait for the **red dashed box** to appear
4. Drop to create the relationship
5. A visual line connects the two tasks (if relationship lines are enabled)

How to Remove a Relationship:

- Repeat the same operation (drag successor beginning over predecessor end and drop)
- The relationship breaks

What Happens When You Move a Predecessor:

- All successor tasks automatically reschedule
- The time offset is preserved
- Cascading updates flow through chains of relationships
- Hours and costs recalculate for affected tasks

Constraints (Date Pinning)

Constraints ensure that tasks start or finish on specific dates regardless of predecessor relationships attempting to reschedule them.

When to Use Constraints:

- Fixed deadline dates (regulatory, contractual, or business-critical)
- External dependencies (vendor delivery, conference dates, fiscal year-end)
- Milestones that must not move

How to Add a Constraint:

1. Locate the **red push pin** on the left-hand side of the Canvas
2. Drag the push pin
3. Drop it near the end of the desired task
4. The task is now constrained to that date

How to Remove a Constraint:

- Double-click on the push pin
- The constraint is removed

Constraint Behavior:

- If the project start date changes, the task will adjust accordingly (constraint overrides relationships but respects project-level changes)
 - If a predecessor tries to reschedule a constrained task, the constraint wins
 - Use sparingly – too many constraints reduce scheduling flexibility
-

Reporting

Timebars provides comprehensive reporting capabilities for tracking project status, resource utilization, and portfolio health.

Accessing Reports

Main Menu > Reports

The Reports menu launches editable HTML reports that you can:

- Search and filter
- Sort by any column
- Export to Excel or PDF
- Print for distribution

Available Reports

General Reports

Report	Description
General Tabular View	Tabular view with search and filtering for all bars
General Card View	Card view with search and filtering
Print WBS	Work Breakdown Structure in markdown format (copy/paste into markdown editor) – includes number, name, and description

Portfolio Reports

Report	Description
Drilldown from Portfolio	Portfolio Report with Color Hierarchy – visual drill-down by portfolio/program/project
PPM Tabular Report	PPM Tabular view with search, filtering, and project selection features
PPM Cards Report	PPM Cards view with search and filtering

Project Reports

Report	Description
Project Status	Lists all Projects with Green/Yellow/Red health indicators
All Tabular	One large table with all columns and all rows available – comprehensive data export

Task Reports

Report	Description
Items Overdue	Milestones that are past the baseline by one or more days
Milestone Horizon	Approaching milestones to watch for in the near future

Resource Reports

Report	Description
Resource Usage	Resource Allocation Supply and Demand Grids and Charts – comprehensive capacity analysis
Shared Resource Pool	All Resources in the tbResources Store – complete list with metadata

For detailed local reporting, see [Forms, Reports and Graphs Guide](#).

For cloud reporting, see [Common Enterprise Dashboard Guide](#).

Scheduling Tips

1. **Plan ahead:** Take advantage of Timebars' forecasting capabilities by creating long-term schedules based on future resource availability. This will help you anticipate potential conflicts or bottlenecks before they arise.
2. **Collaborate effectively:** Share schedules with team members or stakeholders using Timebars' collaborative features. This ensures everyone stays on the same page and can make adjustments if necessary.
3. **Optimize resource allocation:** Use Timebars' analytics tools to identify patterns in resource usage over time. By analyzing data such as workload distribution or peak usage periods, you can optimize your resource allocation for maximum efficiency.
4. **Automate repetitive tasks:** Save time by utilizing Timebars' automation features. Set up recurring tasks or automate notifications for upcoming deadlines, ensuring nothing falls through the cracks.
5. **Monitor progress in real-time:** Stay updated on project status with Timebars' real-time tracking capabilities. Visualize progress through interactive dashboards or generate reports to keep stakeholders informed.

References to Common Guides

Timebars shares many features with Agilebars and Costbars. For guidance on these common capabilities, refer to these user guides:

Core Functionality

- [Common User Interface Guide](#) – Canvas navigation, filters, hierarchy display, bar creator, editing bars, baseline management, view controls
- [Data Synchronization, Backup, Recovery and Retention Guide](#) – How data is organized, IndexedDB details, data hierarchy, backup/restore procedures
- [Data Synchronization, Backup, Recovery and Retention Guide](#) – 6-step sync cycle, CSV file export/import, template configuration, bidirectional data exchange

Project Management Features

- [Common Risks Issues Change Requests User Guide](#) – Creating and managing RIC items, visual indicators, card/tabular views, progress workflows

Reporting and Publishing

- [Forms, Reports and Graphs Guide](#) – General Reports, Portfolio Reports, Project Reports, Task Reports, Resource Reports
- [Common Cloud Reports and Dashboard Guide](#) – Executive Portfolio Reports, Card-Based Drilldown, Interactive dashboards with 7-dimension health tracking
- [Common Cloud Publishing Guide](#) – Publishing to Timebars Cloud, PubSets, Re-Publish, Re-Hydrate, cross-device sync

Quick Reference

Hierarchy Color Guide

Level	Color	Type	Always Present?
L1	Dark Grey	Portfolio	No (only in 4-level and 5-level hierarchies)
L2	Green	Project/Program	Yes (all hierarchies)
L3	Orange	SubProject	No (only in 5-level hierarchy)
L4	Blue	Task	Yes (all hierarchies)
L5	Gold	Allocation	Yes (all hierarchies)

Scheduling Engine Summary

Task Position	Start Date	Finish Date	Duration	Work & Cost	% Complete
Future (all dates > Report Date)	Forecast	Forecast	Forecast	Forecast	N/A

Task Position	Start Date	Finish Date	Duration	Work & Cost	% Complete
In Progress (straddles Report Date)	Actual	Forecast	Remaining	Actual + Forecast	Calculated
Completed (all dates < Report Date)	Actual	Actual	0 (complete)	Actual	100%

Resource Allocator Actions

Action	Method
Open Resource Allocator	Main Menu > Resources
Create Allocation	Drag resource from allocator, drop on Task (Blue L4 bar)
Check Availability	Double-click resource row (weekly view only) – shows yellow bars
View Usage Details	Click Usage icon on Resource Allocator
Search Resources	Type in search box (searches all fields)
Filter by Type	Check Human/Generic/All Resources boxes
Sort Resources	Click column header to sort
Show More Metadata	Click Width Icon

Supply vs. Demand Grid Summary

View Mode	When to Use	Granularity
FTE, Monthly	Strategic planning, executive reporting	Monthly FTE capacity
Hours, Monthly	Detailed tactical planning	Monthly work hours
FTE, Weekly	Short-term resource leveling	Weekly FTE capacity
Hours, Weekly	Granular task scheduling	Weekly work hours
GroupBy	Shows	
Project	Resource demand by project	
Resource Name	Individual resource utilization	
Role	Capacity and demand by job function/skill	

Work Hours Calculation

Hours = Percent Allocated × Work Day Hours × Remaining Duration

Cost = Calculated Hours × Pay Rate

Getting Help

For additional assistance:

- **Common Help Page** – Comprehensive guide for shared functionality across all products
- **Timebars Presentation** – [Comprehensive benefits and features presentation](#)
- **Support Contact** – Contact your Timebars administrator or support team

Optimize Your Resource Scheduling: Visual planning. Accurate forecasting. Balanced capacity.

Part 2 — The User Interface

Product Help - Common Across Products

What You Should Know First

The following sub-section describes important Need to Know information First, before you can even get started. The following help topics describe how to use any Timebars Ltd. products. For help related to an individual product see other tabs in the knowledge base for each product (Timebars, Agilebars, Costbars)

How to Use for Free

You can try our products for free without registering. The free version limits how many projects, tasks and Pubsets you can hold, and there is no time limit — use it for as long as you like. The current limits are on the sales site.

There is no software to install. We apply data limits rather than a trial period so that users with small teams and portfolios can keep the price down.

Important Note: We don't install cookies on your device. Our management software products are secure web pages. Your data is stored in the browser cache, not on our servers (unless you choose to use the Pubset feature).

How to Get a License

After registering and purchasing a subscription you can begin using our Apps as a licensed user. See video: how to purchase a subscription. In summary, your new license will automatically install when you log into one of our Apps.

If you wish to purchase a subscription, use the Quick Buy button found throughout the site. Subscriptions are single person monthly purchases. Register with an email and password, purchase your desired subscription, then use the email/password combination as your license to log in and use our products with the desired data limits.

Our monthly subscription model has three pricing tiers. Each tier increases the data limits to suit your needs. After you purchase a subscription and log in, it automatically alters the data limits imposed by the free version to suit your new subscription, your project and resource data remains unchanged. You can buy one month at a time or up to 12 months in one credit card transaction. Read the Pricing Table & Sales FAQ at the Sales Site!

How to Login/Logout

Click the Yellow Icon in the top most menu, right side, enter user name and password, same ones used when registering. That is it you are now logged in as a licensed user.

Switching Products

There are scenarios where you may be running agile bars product and you drag and drop a Time bars based spreadsheet onto the canvas and you will find you may get an error about the license not being valid or correct. this may be normal And it requires you to log in again with the username and password from the desired product subscription.

About Security and Product Limits

Data limits are enforced by product line and subscription tier. It calculates the number of days remaining until the license expires by comparing the current date with the expiration date store with the license. If the license is valid, it checks to determine the type of license and performs additional UI trimming based on the license type. For each product and license type, it checks if the bar count exceeds certain limits. If the limits are exceeded, it displays the appropriate user information message and re-establishes a demo license status so the user can continue with nag screens until a valid license is established.

About Demo Data

When the App is loaded for the first time, the demo data is populated as part of the page loading process. This is a one-time thing and it's done only only the very first time you use the product you can and start new with fresh new demo data.

It is important to always start off with demo data because the mandatory seed data is installed. Seed data for the default pick lists (tags/metadadata), timescale settings etc. The demo data consists of a demo license for free access to the product for life with data limits as stated in the sales site at www.Timebars.com.

There are two demo data sets, a small demo data set with a few Timebars and there is a large data set with over 350 bars. To load switch demo data, click on the hamburger icon on the top left of the screen to load the admin menu, click Data Actions, click clear demo data. Click either load demo data Small or load core demo data large defined by S or L in Brackets.

The population of demo data includes not only project data but all the seed data. If you make changes to your time Bars by dragging them around the screen and if you create custom fields and other operations you will lose them all when you clear demo data and restore demo data from scratch so you have to be careful.

Top Menu and the Main Menu

The top menu shows the product icon and name, import and export arrows, bar refresh icon, shopping cart icon and login icon. In the middle of the top menu is the currently logged in user, if the license is invalid or not downloaded yet it will show license status instead.

The Main Menu is a row of icons for launching core functionality of the application, an explanation of each follows within the Core Product Functionality section.

The Hamburger Icon

Click the hamburger icon in the Main Menu to access links to various features, including the Filter Menu, Getting Started Page, AI Search, FAQ, WBS Generator, and Data Actions (such as Import, Export, Backup, and more).

The Canvas

Our application comprises two key components: the Kanban and Timescale canvases. The Timescale canvas, featuring Timebars, offers distinct advantages over traditional Gantt charts. Unlike Gantt charts, Timebars on the Timescale canvas empower users to effortlessly drag and drop bars to any position they desire. In contrast to the rigid one-row-per-bar format of Gantt charts, Timebars allow for greater flexibility in bar placement.

This flexibility not only accommodates a larger number of bars on the screen but also enhances resource allocation and project scheduling, ultimately boosting the productivity of project management personnel.

About the Timescale Canvas

The first thing you encounter when logging in with fresh demo data is the Getting Started Page. Once closed, the Timescale Canvas becomes visible by default. The Switch Modes icon on the Main Menu offers a choice between the Timescale Canvas and the Kanban Canvas.

The Timescale Canvas comprises a series of vertical lines that can be separated by either weeks or months. You have the flexibility to select the timescale (weeks or months) from the canvas settings window. Click Tools > Set Canvas to load the window. A flyout menu will appear, with the Set Canvas icon at the top. Click on it to adjust your time scale preferences such as either a weekly or monthly view. Furthermore, you can zoom in and out by selecting a zoom factor. The default zoom factor is 1.5, but alternatives include 0.5, 1.5, 2, and so forth. Be sure to save and close to apply your chosen settings. The Canvas will automatically reload.

Numerous other Canvas settings await exploration, such as toggling between light and dark modes, displaying baseline bars or ghost bars, and adjusting levels. We will delve into these topics later, as they are somewhat more intricate and warrant further documentation immediately following the Kanban discussion.

About the Kanban Canvas

Kanban is a potent visual management tool that empowers teams to plan, track, and manage their work with exceptional efficiency. Its visual representation of project workflows fosters collaboration and transparency.

To switch to Kanban mode, simply locate the "Switch Modes" icon on the main menu at the top center of the screen. Clicking it again will return you to the Timescale mode.

The Kanban board is available in all our products, but each functions differently to cater to either Agile or Classic management processes. For more specific information, please refer to the dedicated help topics for the product you are interested in.

The Timebar Hierarchy

Timebars conforms to 3 different hierarchies. The minimum is a 3 level hierarchy, and maximum is a 5 level hierarchy with a 4 level hierarchy in the middle. The rules are built into the tool to not allow violation of these hierarchies as you drag and drop Timebars to create schedules. Each level is color coded to make it easy to remember the levels in the hierarchy and to allow choice as to what levels are needed.

L1 - Dark Grey L2 - Green (Always a Project for all 3 hierarchies) L3 - Orange L4 - Blue (Always a Task for all 3 hierarchies) L5 - Gold (Always an Allocation for all 3 hierarchies)

The maximum 5 Levels:

You can optionally look at the 5 level hierarchy as follows:

L1-Portfolio L2--Project/Program L3---SubProject L4----Task (incl milestones and gates) L5-----Resource Allocation

The medium is a L4 level hierarchy:

For most small and medium sized projects 4 levels is sufficient. This one is the same as the 5 level except it allows a user to not use the Orange bars.

L1-Portfolio L2--Project/Program L3----Task (incl milestones and gates) L4-----Resource Allocation

The minimum is a L3 level hierarchy:

For most small and medium sized projects 4 levels is sufficient. This one is the same as the 5 level and 4 level except it allows a user to not use the Dark Grey L1 bars.

L1--Project/Program L2---Task (incl milestones and gates) L3---Resource Allocation

Canvas Settings

Adjusting the Weekly/Monthly Settings:

- To switch to a weekly timescale:
 1. Click on the wrench icon on the Main Menu.
 2. Click on "Canvas Settings".
 3. Change the radio button to "weekly".
 4. Click "save and close".
- For optimal viewing, it's recommended to set the Zoom Factor to 1. Click and drag the slider icon and drop it to see the zoom factor.

Adjusting the Time Scale Start, Report Date The timescale must always start on a Monday so that the scheduling engine has a work week reference point. The Report Date is one of the most important settings for the automatic cost, hours and date calculations to be accurate. It must be set to a work week day, not a weekend to improve accuracy. When you change the Time Scale Start, Report Date, the settings stored in IndexedDB and reused when the page loads.

The general rule is that if a bar is visually behind the Report Date line on the Timescale canvas, it will automatically set actual dates, costs and hours based on the configurations. You will learn about how to configure this later.

Show Baseline, Ghost and Visible Levels Show Baseline Bars: Yes/No - If the user has set a baseline and this value is set to Yes, it will show Baseline Bars on the Canvas as thin bars below each timebar

Show Ghost Bars: Yes/No - If the user has this value is set to Yes, the dit will show Ghost Bars on the Canvas. This is useful to see the original position of the bar as a light ghostly bar.

Visible Levels, L1, L2, L3, L4 or L5 - The user can check off which levels in the hierarchy to hide on the Canvas. For example if L2 is checked, bars at levels L1 and L2 are shown, hiding the levels below. This is useful to reduce clutter on the screen for large data sets.

Hide Completed Bar True/False - If set to true, bars in the past (before the Report Date) are hidden.

L3 / L4 / L5 Level Tick Boxes on the Main Menu

Three tick boxes sit in the Main Menu, beside the timescale changer, and decide how far down the hierarchy the canvas draws:

Tick box	Hides when cleared
L3	Sub-Projects
L4	Tasks, Milestones and Gates
L5	Allocations

L1 Portfolio and L2 Project bars are always shown — only the three lower levels are optional. Clearing L5 on a large plan is usually the quickest way to turn a wall of allocations back into a readable portfolio.

They apply to the **Timebar View, Modern View, Filters View, and the Vertical and Horizontal Timelines**. They do not appear on the Gantt Schedule View, which has its own tick boxes in its filter panel, and not in Agilebars, which has no L1, L3 or L5 hierarchy. They are also hidden while the Filter Menu is loaded — that menu already decides which rows are on screen, so the tick boxes would do nothing there.

Your choice is saved with your settings, so it survives a page refresh and rides along with a backup and restore.

Bar Spacing and the "Fit" Tick Box

Next to the level tick boxes are a **spacing picklist** (Small, Medium, Large) and a **Fit** tick box. This is the one place in the application where vertical bar spacing is chosen.

Spacing means different things depending on how a view places its bars:

View	What the spacing setting does
Modern View, Filters View	Sets the row height. Bars always sit on even rows
Timebar View	Nothing <i>until you tick Fit</i> — see below
Vertical and Horizontal Timelines	Sets the spacing between points
Gantt Schedule View, Kanban	Ignored — these lay themselves out

Fit is the trade-off tick box, and it only appears on the Timebar View.

The Timebar View's defining feature is that you place bars wherever you like vertically — that freedom is what lets you fit far more work on one screen than a conventional Gantt chart. The cost is that a plan built that way can end up looking ragged.

Tick **Fit** and the Timebar View lays every bar out on even rows at the spacing you picked, cleaning the view up instantly. You keep horizontal drag and resize, so you can still reschedule; what you give up is **vertical reorder** —

bars are on their rows and cannot be dragged up or down. Clear the tick box and your hand-placed positions come back, because they were never discarded.

Use it to tidy a plan for a screenshot or a review, and clear it when you go back to laying bars out by hand.

Both the spacing and the Fit setting are saved with your settings and survive a refresh, backup and restore.

The picklist hides in the two cases where there is no spacing to pick — while the Filter Menu is in use (it draws bars at the exact position stored on each row), and in Agilebars.

Refresh the Canvas

Quickly refresh the Canvas and bars without a full browser page reload.

- Click the Refresh icon on the top menu (black)
- Will invoke a full cost schedule rollup

Reload the Canvas (Full Browser Page Reload)

Sometimes you may want a full browser page reload if the page seems to be rendering something unexpected.

- Right click on Canvas > Reload Page or
- Use the standard browser page reload that is available on all browsers, typically located to the left of the URL (e.g., tb.timebars.com)

Recalculate All System Generated Data

The System can perform a complete recalculation: hours/cost rollup, updates L1 to L5 hierarchy names, hierarchy numbers (WBS), and re-populates the tbMDJoined data store if you notice data anomalies.

- Right click on Canvas > Recalculate All
- There are other Recalculate All buttons, such as on the Reports Page, to ensure that the system data has complete and updated information

How to Manage your Data

Download Open Office Spreadsheet

This OpenOffice Calc spreadsheet is specially designed to sync with all Timebars products with the ability to optionally synchronize your data with OpenOffice Calc to move data in and out of our products in a fast, clean and simple manner. Click on the Hamburger icon to launch the Admin menu, near the bottom is the link to download the spreadsheet. See the Spreadsheet syncing topic below for further details on how to use it.

Download Excel Spreadsheet

This Excel spreadsheet is specially designed to sync with all Timebars products with the ability to optionally synchronize your data with Excel to move data in and out of our products in a fast, clean and simple manner. See the Spreadsheet syncing topic below for further details on how to use it.

Spreadsheet Syncing

tbClient Pre-configured Spreadsheet Syncing: How It Works

Licensed users will have access to our spreadsheet, which contains custom code that facilitates the synchronization of data between exported CSV files and the spreadsheet. To achieve this synchronization, the code inside the spreadsheet imports CSV files stored in a user-configurable location on the hard drive.

From the app, users should first click the export button to save CSV files to this designated location. Once saved, the data in these CSV files can be imported into the spreadsheet by clicking the "import all" button located on the setup page.

Below is a description of the setup page inside the spreadsheet (a screenshot would typically be here). Notably, both the import and export locations can be adjusted to fit any directory on your hard drive. On the setup page, you can utilize the "import all" button to pull data from all stores that have a "yes" marked in the "import yes/no" column. However, if you prefer to selectively import specific data, such as time bars and metadata, you can navigate to the respective tabs and use the individual import buttons.

Within the spreadsheet, users have the flexibility to manually edit data. This includes adding new tasks, modifying resources, changing tag data, and using common Excel functionalities like drag-and-drop, copy-paste, and fill-down.

When you've finished making your edits, ensure you save your changes. After saving, simply drag and drop the spreadsheet onto the application canvas, and the app will update to reflect the modifications. It's crucial to remember not to make edits in the app while the data is in the spreadsheet, as doing so could lead to overwriting and loss of your changes.

Export as CSV (multi-file)

Export all data stores to csv for import into spreadsheet

Database Backup and Restore

Drag and drop the spread sheets or the JSON backup file for full restore of all data

Editing Data Inside the Application

Sometimes it is quicker to edit data in the application than to go out to the spreadsheet. Every store now has its own **tabular report** on the Report Menu, rather than one combined grid:

To edit	Go to
Bars, tasks, metadata	Report Menu > Project > General Tabular View , or the Gantt Report
The resource pool	Report Menu > Resource > Shared Resource Pool
Picklist values	Report Menu > Other > Picklist Values
Form and layout configuration	Report Menu > Other > Fields Values, Schema Values, CoreReport Values

The schedule and metadata grids are edited **in place** — click a cell and type. The four configuration reports are read-only in the grid and edited on their form; see the *Forms, Reports and Graphs Guide* for why.

The old **Data Management Grids** screen, which put every store behind one set of tabs, has been removed. Each of its tabs is now a proper report in the list above.

Data Actions Menu

The Admin menu — the **Hamburger icon** at the top left of the main menu — carries the data actions. Each one is covered in full in the *Synchronization and Data Control User Guide*; this is the summary.

Menu item	What it does
Full Backup	writes every store to one JSON file — your restore point
Export to CSV (SpreadSheet sync.)	writes 15 CSV files, one per store, for the workbook round trip
Export to JSON	writes 15 JSON files, one per store, for integrations
Import SpreadSheet, CSV or JSON	opens the drop panel — drop a file, or browse for one
Bulk Manage Bars	duplicate a bar and its children, or transfer bars to another product
Migrate Rich Text	converts legacy rich text to the current editor format; safe to run any time
Clear Timebars & Metadata	a clean canvas, keeping your resources, picklists and configuration
Load Demo Data	replaces the data with a demo set for this product
Delete Database	destroys everything; refresh afterwards to rebuild with demo data
Download Excel SS / Libre Office Calc SS	the workbook template

Two things are worth knowing before you use any of them:

- **A Full Backup is taken automatically** before an import, a demo data load, or Clear Timebars & Metadata. **Delete Database is the exception** — it does not back up for you.
- **An import replaces each store rather than merging into it.** If you have edits in the application that are not yet in the spreadsheet, the application **blocks the import** and shows a warning icon on the top menu. Double-click that icon for the way out. This is covered properly in the *Data Synchronization Backup Recovery And Retention User Guide*.

Clearing data by hand

You can clear individual stores from the browser's own developer tools — F12, Application tab, IndexedDB, right-click a store, Clear. Deleting the whole database there works too, but **no backup is taken first**. Refresh afterwards and a new database is built with demo data; drop one of your backup files on the import panel to get back to where you were.

Prefer **Clear Timebars & Metadata** on the menu for the everyday case — it backs up first and keeps your configuration.

Core Product Functionality

This section provides details about the user interface that enhances your productivity and serves as the core functionality across all our products.

Automatic Calculations

Scheduling Engine

The scheduling engine is at the heart of the application and it is proprietary. It is not something that you can see or hear but it is listening and running in the background. With its advanced algorithms, it can recalculate your schedule in a matter of seconds when you make any changes to your project such as when you drag and drop a bar, adding resources, changing the percent allocated or changing the time scale.

For example if you have a set of bars on the timescale and one bar is moved, this bar and the child bars will have new dates, costs and hours calculated by the scheduling Engine. Tasks with relationships to other tasks also calculate new dates for the successors in addition to the child bars. If a bar has a constraint on it it states will not be recalculated by the scheduling engine.

Automatic Cost and Schedule Rollup

In order for the reports graphs and charts to be accurate we have to continually roll up or sum up the data as it's being created or modified. The process of automatically rolling up data is to aggregate cost and hours data from lower levels of a hierarchy into higher levels. All products enforce a 5 tier hierarchy also known as levels which are abbreviated like this: L1, L2, L3, L4 and L5.

Switch Modes (Timescale/Kanban)

Switch to Kanban mode from the Main Menu by selecting the "Switch Modes" option.

Imagine a project represented as a green bar. In Kanban mode, this project is managed using swim lanes. The blue bars signify individual work items or tasks, representing the backlog items chosen for the current Sprint.

There's also a filter option which allows users to transition between different projects. Using this filter, one can view various work items in the project's backlog. The project team then decides which tasks they'll undertake. Once these are selected, a burn down chart can be plotted to visualize progress. As tasks shift between stages like 'doing' and 'done', progress is tracked automatically. The burn down chart can be revisited at any time to compare the initial plan with the ongoing progress.

Additionally, users have the option to toggle between viewing tasks as bars or text boxes in the Kanban view.

Filter Menu

The Filter Menu is derived from the underlying hierarchical data. Each row is a hyperlink that allows drill-down or filtering at specific levels of the bar hierarchy to reduce clutter. Double-click the pink FM tab to launch the Filter Menu. Click the Filter Icon to remove the filter completely and hide it.

The menu renders three levels: L1 (brown bars), L2 (green bars), and L3 (orange bars). Agilebars renders one level: Project. When a level is clicked, a filter is applied. The selected level is saved so that when the user returns it will show the filter menu and the filtered project by default.

Main Filter Panel

The Main Filter Panel helps users refine and focus on specific projects and tasks based on freeform search or metadata picklists.

- Choose Main Filters on the Main Menu
- Click button to Reset filters to get back to an unfiltered state

- Click the X to close the Filter Panel and go back to the Canvas with all bars loaded
- Notice that the bars are all shown on individual lines, not beside each other as shown when the Filter Panel is closed

How to Create Bars

Launch the **Creator Bars** popup to create new bars. Drag a bar from the popup and drop it onto an existing bar to establish a hierarchical structure.

- Choose **Main Menu > New Bar**

Steps to Create a New Bar

1. If there are no existing bars, a special step is required to create the first bar at the top of the hierarchy
2. Begin dragging a **Creator Bar**, and follow the instructions displayed in the **dashed box** before dropping it as instructed

Create new L1 Portfolios, L2 Projects, L3 Sub-projects, L4 Tasks and Milestones and L5 Allocations. Also create Gate Notes to link and show notes on the Canvas.

Bars are created by drag and drop that forces the hierarchy of L1 to L5. When you drag the L2 Project Creator Bar, instructions pop up on the right. If an L1 exists, drop it on top to create and link the L2 bar. Or you can decide not to link it to the L1, this could be considered as a standalone project, not belonging to a Portfolio L1 bar if you choose.

The Filter Menu will show it as such. We recommend always using an L1 bar as a scheduling best practice. L3 orange bars can only be created by dropping the Creator Bar on an L3 bar, dropping it anywhere else is not allowed and the user is notified. Task and Milestone Creator Bars can only be dropped on L2 or L3. Gate Notes can be dropped at any level, L1 to L5.

Example in Agilebars

- **Task bars** can only be created by **dropping the Task Creator Bar (blue) onto a Project Bar (green)**
- Dropping the Task Creator Bar elsewhere will trigger a **user notification** and prevent the action

Filter Menu

The filter allows the user to drill down into specific levels in the hierarchy. All products enforce a 5 tier hierarchy known as levels abbreviated like this: L1, L2, L3, L4 and L5. In the graphic above the pink FM tab, when its double clicked it opens the Filter Menu and the user can click to drill and filter. When they click, the "isFilteredFrom" value is stored in indexedDB, so the next page reload it will hit drawBarsFromP3Menu(isFilteredFrom, tbID) function more on this later. The graphic below shows the Filter Menu, it can only render L1, L2 and L3 items. In this case the brown bar indicates it is an L1, Green is L2 and Orange is an L3 (See the "L" at bottom left of all bars). For example If the user clicks on an orange bar, the filtering will look like this:

Shortcut Menu

The Shortcut Menu is standard "Right Click" functionality like in any app. Right click on the canvas and the shortcut menu will pop up. There is a different shortcut menu for the kanban View.

Right-click anywhere on the Canvas to bring up the Shortcut Menu for quick access to: Canvas Settings, Bulk Move, Refresh Bars, Filter Menu and Toggle Bars vs Time Boxes on the Kanban Board. Bulk Move allows moving

many bars at once to a new position on the Canvas.

Toggle Light/Dark Mode

In software development, "dark mode" and "light mode" refer to two different color schemes or visual styles that can be applied to user interfaces.

1. **Dark Mode:** Dark mode, also known as night mode or dark theme, is a design option where the user interface predominantly uses dark or black backgrounds with lighter text and elements. This color scheme is intended to reduce the amount of light emitted by the screen, making it more comfortable to view in low-light environments. Dark mode is often preferred by users who find it easier on the eyes or who want to reduce eye strain, particularly in dimly lit conditions.
2. **Light Mode:** Light mode, on the other hand, is the default or traditional design option in many software applications. In light mode, the user interface typically uses light or white backgrounds with darker text and elements. This color scheme is more reminiscent of the classic paper-like appearance and is often the default choice for readability in well-lit environments.

The choice between dark mode and light mode is largely a matter of personal preference and user experience. Some individuals may find dark mode more visually appealing or less intrusive, while others may prefer the traditional light mode. Many software applications and operating systems now provide the option to switch between these modes to accommodate different user preferences and environments.

Creating Relationships between Tasks

To create relationships between Tasks, this tells the scheduling engine to reschedule the successor bar as well, by the same amount of time. To create a relationship drag a task by grabbing the beginning of it and dragging it over the ending of the target bar and dropping it as shown below.

Show/Hide Hierarchy Lines

To show the hierarchy lines on the Canvas:

- Right click on the Canvas to show the Shortcut Menu and click "Hierarchy Lines" or
- Choose Tools > "Hierarchy Lines"

For example to see the hierarchy through the lines: The Gold Bar (L5) is linked to the Blue Bar (L4), the Blue Bars are linked to the Orange Bar (L3), Green Bar (L2) is linked to the Brown Bar at L1. The Brown bar is the top of the hierarchy.

How to Turn On/Off Bar Relationship Lines

To toggle relationship lines between tasks:

- Right click on the Canvas to show the Shortcut Menu and click "Bar Relationships" or
- Choose Tools > Bar Relationships

Task Bar Lines (Relationships)

To show the Task Lines on screen, turn on Task Bar lines as described above. Relationship lines visually connect predecessor and successor tasks, showing task dependencies in your schedule.

Drop to Delete Bar Lines

To remove a relationship, drag and drop the successor bar on the "Drop to Delete Task Lines" Pad.

Manage Constraints

Add constraints to tasks so they are not rescheduled ever. Click the Edit Menu button on the Main Menu then click on the Manage Constraints link. The pin now shows and the Canvas has been darkened. The pin can be dragged and dropped on a task to constrain it. Remove the constraint by clicking on the pin on the bar and dragging it off the bar and dropping it on the Canvas.

Popup Cost Schedule Form Off of Bar

To load the popup the user clicks on the lower portion of the bar e.g. T:3773... it will load the form below the bar associated with the bar, works at all levels.

How to Make Edits to Bar and Metadata

There are multiple ways to edit bars and metadata throughout the application:

Method 1: Flyout Form

- Click on the **bar Name** to open the right side flyout form
- Click on Name Field, make changes
- Data is saved automatically when you move off the field
- Close the form when done

Method 2: Freeze Bars for Quick Edits

- Click the **pink FZ Tab** on the left side of the Canvas to freeze the Bars so they cannot be moved
- Simply change the bar name directly on the Canvas
- Click on another bar to save your changes
- Click the **Refresh Icon** on top black menu to remove the freeze

Method 3: Core Report

Launch the Core Report from multiple locations:

- Click on a **bar ID** (bottom left of any bar) to launch the Cost Schedule popup, click "Core Report"
- Launch the Core Report from rows in the **Risk and Issues page**
- Launch the Core Report from rows in **Tabular and Card View reports** available from the Reports Menu

Method 4: Spreadsheet Sync Feature

- Use the Spreadsheet Sync Feature to change metadata in bulk
- Export data to spreadsheet, make changes, and load it back into the application
- See Data Management section above for details

Note: Edits are saved automatically when you move off a field throughout all editable fields in the applications.

Metadata Coding

Metadata coding is a flexible way to organize and break down a project into groups of activities. By establishing a coding structure and assigning codes to Timebars it will allow virtually any view or report of your project information.

The metadata codes are maintained in the custom spreadsheet template and are updated in the Timebars Web App by drag and drop gestures on the Canvas. There is no complex code tables and look up tables to maintain,

just spread sheet maintenance and Timebars does the rest saving you time.

Metadata View

Available from the Cost/Schedule Popup form.

How to Create a Baseline

When planning is complete and before work begins, create a baseline to generate your Planned data. Use the Planned data to compare with current data and actuals for variance reporting.

Easy Baseline Management: When Sprint planning is done and before the work starts, set a baseline or snapshot with one click. Compare current forecast to the baseline (original plan) with the burndown chart and other reports with one click. Unlimited baselines supported.

Steps to Create a Baseline

1. Click on a **Bar ID** (bottom left of any bar) to open the Cost Schedule popup
2. From the popup, click the **"Set Baseline"** button to create a baseline for this bar and all its child bars
3. The checkbox is enabled by default, so it will **overwrite an existing baseline**
4. You can set a baseline at the **L2 Project Level (green bar)** or any level below it, but **not at the L1 (brown bars) level**

When the Set Baseline button is clicked and the check mark is on, the previous baseline is lost. The baseline will be created at the current level and all child bars below it in the hierarchy.

How to Update a Baseline for Added Scope

If the checkbox is unchecked, this Timebar and any child Timebars without an existing baseline will not delete existing baseline data. This is necessary for scenarios where the baseline is initially set, and later a new bar and/or its children need to be added to the baseline.

Use Case: The Baseline is initially set at L1, then later, an individual bar and its children need to be re-baselined without affecting existing baselines.

How to Delete Bars

To delete a bar, **drag and drop it onto the Trash Can** located on the **left side of the Canvas**. This will remove the bar along with all of its child bars.

When a bar is dragged, the Edit Menu shows automatically, allowing you to delete one bar at a time or the dropped bar and all child bars, one level below. For example if you delete a Task, the Allocations will be deleted also, if you wish to delete just the Task and keep the Allocations, drop the bar on the Delete One pad.

Important Notes

- If you want to delete a bar and keep the child bars, first move the children to another bar in the hierarchy using **drag and drop**
- **Deleting cannot be undone**, so if needed, **make a backup first:**
 - Select **Hamburger Icon > Full Backup**
 - To restore a backup, simply **drag and drop the backup file onto the Canvas**

- **Undo Bar Move does not cover deletion.** It puts *moved* bars back where they were; it cannot bring a deleted bar back. A backup is the only way back from a delete. See *Undo Bar Move* in the **Data Synchronization, Backup, Recovery and Retention Guide**

Drop to Duplicate Bars

Below is the result of duplicating a project. After dragging and dropping the duplicate onto the L1 bar (brown), it shows in the Filter Menu as a new project. The child bars were not duplicated. There are other ways to do this such as the Spreadsheet. More on this later.

Bulk Move Tasks

To Bulk Move Tasks or to drag and drop many tasks at one time, click the Edit Menu button on the main menu then click on the Bulk Move Tasks link. Now click on each task that is to be moved, then click again and drag and drop to the right place on the timescale. Notice that selected bar are highlighted with blue dotted lines. When done, screen refreshes and bars have been rescheduled including child bars or Allocations.

- Tip: When you get to the last bar that you plan to move, double-click it then drag.

Bulk Move Projects (Costbars Only)

To bulk move Projects (green bars) in Costbars:

- Choose Tools Menu > Bulk Move Projects link to reschedule Projects (green Bars) in bulk
- Click on two or more projects to highlight them, then drag and drop to reposition them

Notice that selected bars are highlighted with green dotted lines. When done, refresh the screen and confirm they have been rescheduled including the child bars.

- Tip: When you get to the last bar that you plan to move, double-click it then drag.

Change Creator Bar Names

The names and descriptions of the six Creator bars are configurable, and they live in the Tags store.

Go to **Report Menu > Other > Picklist Values**, filter the **Picklist** column to the relevant list, then click the edit icon on the value you want. Change its **Short Name** — that is the text shown on the Creator bar — and its **Purpose**, which is the instruction text in the Creator Bar popup. Changes save as you type.

Leave the other columns alone. A value that carries a **padlock** cannot be renamed at all: the application compares against that exact text, and renaming it would break a calculation silently. Adding your own values is always safe.

You can also change these in the Tags worksheet and sync back to the app.

Metadata View

The metadata view is available from several locations within the application, such as the General Tabular View and from each bar on the canvas. This view is a large collapsible report or form that allows for editing metadata based on business driven headings such as business case cost and schedule information etc.

Get Started Page

From the main menu click on the hamburger icon to launch the admin menu. The "Getting Started" menu graphic is shown below. From there you can fire up the getting started page. Below is a view of the getting started page which can be configured to fire up by default every time the application loads for the first time.

FAQ

The FAQ can be launched by the link on the Admin Menu similar to how the Getting Started page is set up and called. The graphic below shows the frequently asked questions page each question is in a drop-down when you click on another next question the current one collapses.

Intro

The intro can be launched from the right side tools menu. Each of the three applications has been configured so that the intro is launched whenever a new database is created in other words when a user first uses the app. Most users will read the intro once and click close don't show the intro again or if you click skip intro it will close and fire up the next time you relaunch the page.

Tour

The tour can be launched from the tools menu slide out right tools menu by clicking on the tour link when the tour is fired it starts at the hamburger icon and you just keep clicking next and the tour stops off at each of the visible UI portions of the application it's a nice little tool to inform the users quickly who are new about the usability and features.

Risk & Issues Features

Overview of Risks and Issues

The risk system has the same look and feel as the risks and issues system but data storage is different. Click on the Risks and Issues button on the main menu to launch the risk list. From the filter menu, find the project you want and click on it to view linked Risks and Issues.

Create New Risks and Issues

Risks and Issues are created by adding a task with the Bar Creator and launching the Slide Out Right Form. The SubType can be set as Risk or Issue, and additional metadata fields can be configured for filtering and grouping.

Filter Risks and Issues

Use the Risks/Issues page to track and manage lists of risks and issues. Filter by project and view the status of each item. Edit details of a Risk or Issue by launching the FOCD Form.

Edit Risks and Issues

Edit the status, title, and content of a Risk or Issue by clicking on the edit icon and entering data in markdown format.

Add fields to the Form

Use FOCD to add or remove fields in the form to customize it according to specific needs or methodologies.

Local Reports Menu

The Reports Menu gives access to tabular reports, charts and the configuration tables. It is grouped into five menus: **Portfolio**, **Project**, **Task**, **Resource** and **Other**.

Portfolio, Project and Task Reports

Projects by Portfolio, Drilldown from Portfolio and Portfolio Detail at portfolio level; the General Tabular View and Baseline Variance at project level; the Task Tabular Report and the Gantt Report at task level.

Resource Reports

Resource Usage — the supply and demand grids and charts — and the Shared Resource Pool.

Other

Print WBS, and the five configuration reports: Picklist Values, Fields Values, Schema Values, CoreReport Values and Config Integrity.

See the *Forms, Reports and Graphs Guide* for what each one does.

Joining tbTimebars and tbMetaData Stores

The function in scripts/tbdatabase.js joins the tbTimebars store and the tbMetaData store, creating a new store called tbMDJoined. This combined data store is useful for reporting, dashboard code, and integration with other tools.

Burndown Chart

What is a Burndown Chart

Easily generate Burndown Charts with one click that are based on Agilebars data in the browser that was generated while using the Kanban Board canvas and/or the Timebars Timescale canvas. Switching lanes in Kanban mode updates the progress of sprint work based on internal rules. For example dropping an Agilebar in the Done lane, sets the work item to 100% complete which feeds the burndown chart.

To launch the burndown chart click on the Project ID, in this case Pj:40 and launch the Burndown Chart. Click the "Refresh Button" to calculate the graph, then click Run to launch it.

How to Calculate the Burndown Chart

The graphic below shows the core function that will timephase the data for the burndown chart. It calls many other functions, not shown. There are 2500 lines in kanvanv2.js file.

Local Dashboard

What is Local Dashboard

The Local Dashboard, (not to be confused with the Cloud Dashboard), is the place where your data is transformed into information. It has four tabs for the Agilebars and Timebars Apps (Summary, Resource Demand Charts, Project Status and Finance), and three additional tabs for the Costbars PPM functionality (Programs, Priorities and Strategy). The filter menu drives the dashboard at the L1 level. Each page of the

Dashboard is a FOCD form so the user can add and remove fields to suit, there are up to 150 different fields available.

Timebars Ltd. Products provides two dashboards because many customers will not be allowed to store their data in the cloud due to security reasons. Corporate data is often proprietary and if exposed to unwanted eyes it could pose serious threats to the organization. Therefore there is a local dashboard that is available on the main menu that pulls data from the index DB inside the browser.

In the example below the L1 title “Digital Works Space - Transformation” is available because there is one L1 in the data set. Each page of the Dashboard is a FOCD form so the user can add and remove fields to suit, there are up to 150 different fields available.

Summary Tab

Currently shows the following fields as part of the Demo Data. This form renders data from L1 rows only. User can add and remove fields as needed.

Resources Tab

Shows the same set of buttons as on the Resource Allocator to show Weekly Demand by various metadata. This form renders data from ResCalcs2 store.

Project Tab

Shows the projects which are children of the L1 (Digital Workspace - Transformation). Multiple L1 bars would be listed in the Filter Menu. This form renders data from L2 rows only in the tbTimebars store.

Financial Tab

This form renders data from L1 rows only. User can add and remove fields as needed. The Cost Analysis field can be edited using the rich text editor.

Cloud Publishing

Overview of Publishing

The Cloud Publishing feature is opt-in functionality with secure Cloud login and authentication. The Cloud Dashboard provides customizable and dynamic reports and graphs, for insightful visualizations. Also rehydrate the Cloud data on another device. e.g. work from your PC one minute and on your Ipad the next.

Cloud Login

Code location and details about the cloud login process.

Show License

Clicking on the Show License Button displays the license details, including the quantity of PubSets available and the expiry date.

Create Your Pubsets

Users need to activate PubSets themselves by clicking the activate PubSets button.

Publish to Update Cloud Dashboard

Publishing PubSets allows updating the cloud dashboard, rendering charts and reports.

Part 3 — Data Model and Scheduling Engine

Data Model and Scheduling Engine Guide

Table of Contents

- 1. Introduction
- 2. Data Hierarchy
- 3. Parent–Child Relationships & WBS Ordering
- 4. Field Rules — What Every Row Must Carry
- 5. Rollup Rules & Field Aggregation
- 6. Scheduling Engine & Scenarios
- 7. Predecessor & Constraint Logic
- 8. Where the Full Field List Lives
- 9. What Recalculate All Does
- 10. Values the Code Depends On
- 11. Status and Health Automation

1. Introduction

The platform is a multi-product scheduling application available in three variants — **Agilebars (AB)**, **Timebars (TB)**, and **Costbars (CB)**. All three products share the same underlying data model and a single client-side IndexedDB database. The differences between products are driven by different scheduling engines and configuration, not by separate codebases.

At the heart of the platform is a **five-level parent child data hierarchy**, where every item — from a top-level portfolio down to an individual resource allocation — is a record in the **tbTimebars** store with a **tbType** field that identifies its level:

Level	tbType	Bar colour	Typical purpose	What its tbSelfKey2 must point at
L1	Portfolio or Program	dark brown	Top-level grouping of related projects	nothing — a root node has no parent
L2	Project	green	A discrete deliverable or initiative	the parent L1, when a Portfolio exists
L3	Sub-Project or Work-Package	orange	A work package or component within a project	the parent L2
L4	Task, Milestone or Gate	blue	Individual work items or schedule control points	its immediate parent — L3, or L2 when there is no Sub-Project

Level	tbType	Bar colour	Typical purpose	What its tbSelfKey2 must point at
L5	Allocation	gold	A resource's assignment to a task (TB/CB only)	the parent L4 Task — and it must also carry a tbResID that exists in tbResources

A row may never skip a level or point at a parent that does not exist. An unresolved tbSelfKey2 orphans the row: it disappears from the hierarchy and contributes nothing to any rollup.

Important Note: Agilebars is a Two Tier Hierarchy, Projects and Tasks, it has its own scheduling engine that skips the other levels.

Parent-child relationships are established via the tbSelfKey2 field: every child record stores the tbID of its parent. This single field is what the scheduling engine, rollup functions, and hierarchy rendering all rely on to traverse the tree — upward to aggregate totals and downward to render the timescale and Kanban views.

Work and cost values are entered at the lowest level in use — normally L5 Allocations, but any childless row (a Task with no allocations, or even a Sub-Project, Project, or Portfolio with nothing beneath it) acts as the leaf for its branch and may carry directly entered values — and they are **rolled up automatically** through the hierarchy. The tbWork, tbAWork, and tbWorkRemaining fields represent forecast hours, actual hours, and remaining hours respectively; tbCost, tbACost, and tbCostRemaining follow the same pattern for costs. A parent **with children** always holds derived totals: the rollup overwrites any manually set values on it, ensuring consistency from allocation to portfolio.

The three products use that model differently:

Product	Hierarchy	Scheduling
Agilebars	two levels — Project (L2) → Task or Story (L4) . L1, L3 and L5 are not used	lane-based progress steps on the Kanban board, plus a sprint burndown
Timebars	the full five levels on a gantt-style timescale	dragging an L5 Allocation recalculates actual start, actual and remaining hours and cost, and percent complete, then rolls the totals up the chain
Costbars	the full five levels, same cost and hours model	as Timebars, extended with portfolio pipeline features — strategic value scoring and demand vs. capacity analysis

This guide documents how data flows through that hierarchy — from the scheduling engine calculations at L5 up through each rollup stage to the portfolio total — and how the two stores (tbTimebars for core fields, tbMetaData for extended fields) relate to each other across all levels.

2. Data Hierarchy

The tbTimebars store in IndexedDB holds rows that represent different levels of work items. Each row is keyed by tbID (unique ID) and references its parent with tbSelfKey2. The tbType field determines its role in the hierarchy.

Timebars and Costbars

The full hierarchy is:

Portfolio → Project → Sub-Project → Task → Allocation

Example:

- **Portfolio:** "Digital Workplace - Transformation"
- **Project:** "Synchronisation Client"
- **Sub-Project:** "Private Cloud Phase 1"
- **Task:** "Consume requirements"
- **Allocations:** "Alloc 1" and "Alloc 2"

Each row stores cost, work, dates, durations, and scheduling metadata. Higher-level rows (Sub-Projects, Projects, Portfolios) that **have children** always hold rolled-up totals — anything entered directly on them is overwritten by the rollup. A higher-level row **without children** is the leaf of its branch and may hold directly entered cost and work (see *Manual Cost, Work & Percent Complete* below).

At the task level, the rollup engine checks whether a Task has child Allocations. If Allocations exist, the Task's cost and work totals are summed from them. If no Allocations exist, the Task itself is treated as the leaf node and its own stored values are used directly in the rollup. This means allocations are optional — tasks without them still participate correctly in the hierarchy rollup.

Agilebars

Agilebars uses a simplified two-level hierarchy:

Project → Task (or Story)

L1 (Portfolio/Program), L3 (Sub-Project), and L5 (Allocation) are not supported. Example:

- **Project:** "Sprint 4 - Mobile App"
- **Tasks / Stories:** "Design login screen", "Write unit tests", "Fix auth bug"

Tasks are always the leaf nodes in Agilebars. Cost and work values are entered directly on each Task, and the Project rolls them up. There is no allocation layer — resource assignment in Agilebars is handled through the Kanban board lanes and sprint membership rather than discrete allocation records.

3. Parent–Child Relationships & WBS Ordering

Every row in **tbTimebars** links to its parent through a single field: **tbSelfKey2**, which stores the **tbID** of the parent row. Root nodes (Portfolios with no parent) have **tbSelfKey2** set to null or empty. This one field is what every traversal — rollup aggregation, hierarchy rendering, connector drawing — depends on.

Breadcrumb labels (tbL1 through tbL5) are derived display fields, not structural ones. They are cleared and regenerated each time **updateL1ToL5()** runs. Each row gets its ancestor names written into these fields by walking up the **tbSelfKey2** chain, so that any row can show its full path without a live tree traversal at render time:

Field	Contains
tbL1	Portfolio or Program name

Field	Contains
tbL2	Project name
tbL3	Sub-Project name
tbL4	Task name
tbL5	Allocation name

WBS ordering (tbHierarchyOrder) is a dot-separated position string, zero-padded to two digits per level, always five segments deep regardless of the actual depth of the row. Examples:

- Portfolio: 01.00.00.00.00
- Project: 01.01.00.00.00
- Sub-Project: 01.01.02.00.00
- Task: 01.01.02.03.00
- Allocation: 01.01.02.03.01

This field is used for display ordering and WBS-style report sorting. Like tbL1–tbL5, it is a computed field regenerated on demand by updateTbHierarchyOrder () — it is not a source of truth for structure, only for ordering. This pre-processed data make managing project data in the spreadsheets and the reports easier for the users.

4. Field Rules — What Every Row Must Carry

The rules in this section apply to **every** route data takes into the product: typed on a form, generated by AI, or synced from the workbook. Data that breaks them does not throw an error — it renders wrongly, or calculates wrongly, or quietly does nothing.

Mandatory fields

Every tbTimebars row, at every level:

Field	What it must hold
tbID	a unique alphanumeric identifier — the key everything else joins to
tbName	the display label shown on the bar, in grids and in reports
tbStart, tbFinish	planned dates, DD-MMM-YYYY (for example 15-Jan-2026)
tbType	one of the reserved values — Portfolio, Project, Sub-Project, Task, Milestone, Gate, Allocation
tbSelfKey2	the parent's tbID, per the table in section 1. Empty only on a root Portfolio
canvasNo	always 1

Every tbMetaData row: tbMDID, which must **exactly** match its bar's tbID. That single field is the entire join between the two stores — one metadata row per bar, no more and no less.

Allocations (L5) additionally require the fields the scheduling engine calculates from: `tbPayRate`, `tbPercentTimeOn`, `tbCalendar`, `tbWork`, `tbWorkRemaining`, `tbDuration`, `tbRemainingDuration` and `tbPercentComplete`. Section 6 shows how they interact.

Field typing

Three rules, and they are absolute.

Numeric fields must be numbers, and default to 0 — never `null`, never `"na"`, never blank:

`tbWork`, `tbAWork`, `tbWorkRemaining`, `tbCost`, `tbACost`, `tbCostRemaining`, `tbBudgetCost`, `tbDuration`, `tbRemainingDuration`, `tbPercentComplete`, `tbPayRate`, `tbPercentTimeOn`, `tbCalendar`, `tbCoordTop`, `kbCoordTop`

A `null` or `"na"` in any of these propagates through every sum it touches: one bad value at allocation level can empty a portfolio total. This is the single most common defect in data arriving from a spreadsheet.

Date fields must be `DD-MMM-YYYY` strings — `07-Oct-2026`:

`tbStart`, `tbFinish`, `tbAStart`, `tbAFinish`

Text fields hold `" "` when empty, never `null` or `undefined`. Every null and undefined is blanked on the way in.

Where a bar sits on the canvas

`tbCoordTop` is the vertical pixel position of a bar. On the Timebar View you place bars where you like, and that freedom is what lets far more work fit on one screen than a conventional gantt chart — so the position has to be stored, not derived.

Data created **inside** the application gets this for free. Data you **generate or import** must set it, or every bar stacks at the same height:

- Give the first bar in the hierarchy a starting value — **100 pixels** is the default if you have no reason to choose otherwise.
- Step each following bar down from it. The Bar Creator uses **50 pixels**; AI-generated rows use **35**, and place an Allocation **40** below its parent Task.

Any consistent step works. The numbers above are what the application itself uses.

Metadata defaults

When a metadata row is created without them, three fields are filled in so the row is never left in an unreportable state:

Field	Default
<code>tbMDStatus</code>	<code>New</code>
<code>tbMDHealth</code>	<code>Not Assessed</code>
<code>tbMDPriority</code>	<code>Not Assessed</code>

Risks, Issues and Change Requests are Tasks

There is no **tbType** of **Risk**. A risk, an issue and a change request are all **Tasks** carrying a **tbSubType**:

To create a	Set tbType	Set tbSubType
Risk	Task	Risk
Issue	Task	Issue
Change Request	Task	CR

tbType is forced to **Task** for all three. The **tbSubType** value is what routes the item to the Risks and Issues views, draws its flag icon on the canvas, and keeps it out of resource calculations — a risk is tracked work, not scheduled effort.

5. Rollup Rules & Field Aggregation

Costs and work flow upward through the hierarchy via the rollup engine (**tbschedulingengineRollups.js**). It builds an in-memory map of the entire hierarchy keyed by **tbID**, links children to parents via **tbSelfKey2**, then walks the tree bottom-up with **aggregateCostsAndWork()**.

The engine applies one rule at every node, regardless of type or level:

- **If a node has children:** its cost and work fields are overwritten with the sum of its children's values.
- **If a node has no children** (leaf node): its own stored values (**originalValues**) are written back as-is and passed upward unchanged.

The leaf-node rule is **not type-aware**. It does not distinguish between an Allocation, a Task, a Sub-Project, or a Project. Any node without children becomes the leaf at that branch and its own stored cost and work values participate in the rollup. This means it is valid to mix approaches within a single portfolio — some projects fully broken down to allocations, others with a cost and work figure entered directly at the project or sub-project level — and the rollup will aggregate all of them correctly into the portfolio total.

The general convention is that cost and work are entered at the Allocation level and that higher-level rows hold only derived totals. But this is a convention, not an enforcement. The engine simply sums whatever exists beneath each node and falls back to the node's own values when nothing is beneath it.

Running the rollup multiple times produces the same result — it is fully idempotent.

Fields that roll up at every level:

Field	Meaning
tbWork	Forecast hours
tbAWork	Actual hours to date
tbWorkRemaining	Remaining hours
tbCost	Forecast cost
tbACost	Actual cost to date
tbCostRemaining	Remaining cost

The full bottom-up flow:

Allocation (or any childless node)
→ summed into Task (or nearest parent with children)
→ summed into Sub-Project
→ summed into Project
→ summed into Portfolio

The Portfolio row is always the root node (`tbSelfKey2` = null). Its totals equal the sum of every leaf node beneath it, wherever in the tree those leaves sit.

Percent complete is derived, not summed. Percentages cannot be meaningfully added, so `tbPercentComplete` is excluded from the summed fields. Instead, immediately after a parent's children are summed, its percent complete is derived from the fresh totals using the same rule hierarchy as manual entry: 100 when the parent itself is complete (it has an actual finish date), otherwise from work when the parent has work (`tbAWork` ÷ `tbWork`), from cost when it has cost but no work (`tbACost` ÷ `tbCost`), and left unchanged when it has neither.

Manual Cost, Work & Percent Complete (rows without Allocations)

When a user chooses not to create Allocations, cost and work are entered by hand on the lowest row in use — typically the L4 Task, but the same applies to a childless L3, L2, or L1. These rows are edited in the **Cost & Schedule popup form** (launched from the bar's status icon), where the Forecast row's three cost cells, three work cells, and % complete render as editable dotted-border inputs on Timebars and Costbars. Rows **with** children stay read-only there — their values come from the rollup, and any manual entry on them would be overwritten (children always win).

Every manual entry is auto-reconciled so the seven fields stay mathematically aligned:

You enter	The system derives
Work (first entry)	Work Remaining = Work, A Work = 0, % Complete = 0
A Work later	Work Remaining = Work – A Work, % Complete = A Work ÷ Work × 100
% Complete later (work present)	A Work = Work × %, Work Remaining = Work – A Work
Work Remaining directly	A Work is kept; Work is re-forecast to A Work + Work Remaining (same convention as the Allocation hours-to-go override)
Cost fields , no work present	identical rules on the cost triple; % Complete derived from cost
Cost fields , work present	Cost Remaining = Cost – A Cost kept consistent, but % Complete stays work-driven
% Complete with neither work nor cost	the typed percent is simply held for reporting

The rule hierarchy for `tbPercentComplete` is therefore: **complete first, then work, then cost, user-entered as last resort**. If Allocations are later created beneath the row, the next rollup replaces the manual values with the children's totals — no cleanup needed.

Completion always wins (all bar types). A bar that carries an actual finish date (**tbAFinish**) is finished — that single field is what makes the Schedule Status read *Completed* and what the *Hide Completed Bars* filter looks for. The scheduling engine sets it on **any** bar type whose finish falls behind the report date, not just Allocations, so **tbPercentComplete** is forced to **100** on those rows too: Task, Sub-Project, Project and Portfolio as well as Milestone, Gate and Release. Previously only Allocations got the matching 100, which left completed rows sitting at whatever percent they happened to hold — complete by date, part-done by number — and Hide Completed Bars (which needs both the actual finish *and* 100%) never hid them.

Actual work and actual cost are **not** rewritten to match. The hours and money a row really consumed stay exactly as entered, so a completed row can legitimately read 100% with A Work below Work. Dates decide *whether* a row is done; the numbers stay a record of what it took. To show less than 100% on such a row, clear its actual finish date or move the bar out past the report date — percent then falls back to the work/cost rules on the next recalculation.

Because data can also arrive via AI generation and Spreadsheet Sync, these seven fields can drift out of alignment. The **Recalculate All** button runs a consistency checker before the rollup: on every childless L4–L1 row it treats forecast and actual work (and cost) as the source of truth, corrects the remaining and percent complete fields to match, and assumes actuals of 0 when only a forecast value exists. Allocation (L5) rows are never touched by the checker — they belong to the scheduling engine.

The scheduling engine (**tbschedulingengine2.js**) is triggered when a user drags or resizes an allocation bar on the timescale. It only recalculates **allocation-level fields** — tasks, sub-projects, projects, and portfolios are updated afterwards by the rollup engine, not by the scheduling engine directly.

Important — Manual Cost & Work at L1–L4: The scheduling engine only recalculates cost and work fields for **Allocation (L5)** rows. If cost and work have been entered manually on a Task, Sub-Project, Project, or Portfolio row (i.e. the row has no children and is acting as a leaf node), those values will **not** be recalculated when the bar is dragged or resized on the timescale. The engine will still update the bar's dates and duration, but **tbWork**, **tbCost**, and all related actual and remaining fields will remain unchanged. To have cost and work recalculate automatically on drag, the values must be held at the Allocation level.

The one progress field the engine does write on these rows is **tbPercentComplete**, and only on completion: when the drag, resize, or report-date change puts the bar entirely behind the report date, the engine stamps the actual finish date and sets percent complete to 100. Work and cost are still left alone.

Manually entered values are maintained through the **Cost & Schedule popup form** (click the bar's status icon): on a childless L4–L1 row the Forecast row's cost, work, and % complete cells render as editable (dotted-border) inputs, and every entry is auto-reconciled by the manual calculation rules described in the *Manual Cost, Work & Percent Complete* section below.

6. Scheduling Engine & Scenarios

The scheduling engine runs when a user drags or resizes an **Allocation** bar on the timescale. It recalculates that allocation's dates, duration, hours, cost and percent complete; everything above it in the hierarchy is then updated by the rollup engine, not by the scheduling engine directly.

Input: Pixels to Days

The entry point is `prepForDateHoursCostCalculator()`, which converts the pixel offset of the drag (`leftDiff` for a move, `resizeDelta` for a resize) into days using the current timescale pixel factor (`tsPxFactor`, set per the weekly or monthly view). This produces `daysdiffS` (start shift) and `daysdiffF` (finish shift), which are passed to `dateHoursCostCalculator()`.

Scenario Detection

The engine reads the allocation's `tbStart` and `tbFinish` dates, converts them to JavaScript Date objects, then compares against `apStatusDate` (the user's status/report date) to determine which of three scenarios applies:

Condition	Scenario
<code>dStart < statusDate</code> AND <code>dFinish > statusDate</code>	In Progress
<code>dStart < statusDate</code> AND <code>dFinish < statusDate</code>	Completed
<code>dStart >= statusDate</code>	Not Started

Scenario 1 — In Progress

The allocation has started but not yet finished. The actual start is locked to the bar's start date; actual finish remains empty.

Standard (no override):

```

tbDuration          = working days (dStart → dFinish)
tbADuration         = working days (dStart → statusDate)
tbRemainingDuration = tbDuration - tbADuration
tbWork              = tbCalendar × (tbPercentTimeOn / 100) × tbDuration
tbAWork             = tbCalendar × (tbPercentTimeOn / 100) × tbADuration
tbWorkRemaining     = tbCalendar × (tbPercentTimeOn / 100) ×
tbRemainingDuration
tbCost              = tbWork × tbPayRate
tbACost             = tbAWork × tbPayRate
tbCostRemaining     = tbWorkRemaining × tbPayRate
tbPercentComplete   = (tbWork - tbWorkRemaining) / tbWork × 100
tbAStart            = dStart (formatted DD-MMM-YYYY)
tbAFinish           = "" (not yet finished)

```

Override path (`tbSchEngOverride = "Overridden"`): The user has manually fixed `tbWorkRemaining`. The engine respects that stored value instead of recalculating it from duration:

```

tbAWork              = tbCalendar × (tbPercentTimeOn / 100) × tbADuration
tbWorkRemaining     = stored value (not recalculated)
tbWork              = tbWorkRemaining + tbAWork
tbCost/tbACost/tbCostRemaining recalculated from the above
tbPercentComplete   = tbAWork / tbWork × 100

```

Scenario 2 — Completed

The entire bar sits before the status date. The allocation is 100% done.

```
tbAStart      = dStart (formatted DD-MMM-YYYY)
tbAFinish     = dFinish (formatted DD-MMM-YYYY)
tbDuration    = working days (dStart → dFinish)
tbADuration   = tbDuration
tbRemainingDuration = 0
tbWork        = tbCalendar × (tbPercentTimeOn / 100) × tbDuration
tbAWork       = tbWork
tbWorkRemaining = 0
tbCost        = tbWork × tbPayRate
tbACost       = tbCost
tbCostRemaining = 0
tbPercentComplete = 100
```

Scenario 3 — Not Started

The bar's start date is on or after the status date. No actual progress exists.

```
tbAStart      = "" (not started)
tbAFinish     = "" (not started)
tbDuration    = working days (dStart → dFinish)
tbRemainingDuration = tbDuration
tbWork        = tbCalendar × (tbPercentTimeOn / 100) × tbDuration
tbAWork       = 0
tbWorkRemaining = tbWork
tbCost        = tbWork × tbPayRate
tbACost       = 0
tbCostRemaining = tbCost
tbPercentComplete = 0
```

Key Allocation Fields Used by the Engine

Field	Role
tbCalendar	Work hours per day for this resource
tbPercentTimeOn	Percentage of time allocated to this task (0–100)
tbPayRate	Hourly cost rate — used to derive all cost fields from work hours
tbSchEngOverride	"Overridden" locks tbWorkRemaining against recalculation

After the scenario calculation completes, results are written back to IndexedDB via `UpdateProgressHoursCalculator()` (called from the drag/resize path in `tbschedulingengine.js`; the allocation popup form path fills the form via `initSchEngForm()`), and the rollup engine then propagates the updated totals up the hierarchy chain.

Important — Progress is Calculated Against the Status Date, Not Today's Date: The engine always compares bar positions against **apStatusDate** — the status or report date set by the user in the admin panel — not against the current calendar date. This is a deliberate project management discipline, not a technical limitation.

In formal project management, schedules are updated on a defined reporting cycle: weekly, fortnightly, or monthly. The status date is the agreed "as of" date for that cycle — the point in time at which the team has collectively assessed progress and recorded actuals. Using today's date instead would mean that the same schedule opened on a Monday shows different percent complete, actual hours, and remaining work than it does on a Friday, even if no update has been made and no real work has been recorded. Reports sent to a steering committee or PMO would not match the numbers the PM sees in the tool, and trend analysis across reporting periods would become unreliable because the reference point keeps drifting.

By anchoring all calculations to a fixed status date, the platform ensures that every report, chart, and rollup total reflects a consistent snapshot: what the project looked like as of that date. The PM advances the status date at the start of each new reporting period, updates the schedule, and the next snapshot is produced cleanly.

Note: The Agilebars Kanban board offers an optional setting (**apABuseStatusDate**) that allows teams to use today's date instead of the status date, for teams that prefer to update progress continuously rather than on a fixed reporting cycle.

7. Predecessor & Constraint Logic

Predecessor (**tbPredecessor**)

A row can reference one other row as its predecessor by storing that row's **tbID** in the **tbPredecessor** field. This establishes a **Finish-to-Start** relationship in intent: the successor is expected to start when its predecessor finishes. The field stores the predecessor's **tbID** as a plain string and defaults to "" when no predecessor exists.

Predecessors are supported on **Projects (L2)** and **Tasks, Milestones, and Gates (L4)**. Sub-Projects (L3), Allocations (L5), and Portfolios (L1) do not carry a predecessor reference in their rendered bar HTML and will not display connector lines even if a value exists in **tbPredecessor** on the underlying record.

Constraint Type (**tbConstraintType**)

The **tbConstraintType** field controls whether a bar's position is locked against movement caused by a predecessor cascade. The meaningful value is:

Value	Effect
"Pinned"	The bar is locked. A predecessor cascade will not move it. A pin icon is rendered on the bar.
"" (empty)	No constraint. The bar moves freely and participates in predecessor cascades normally.

Pinning is useful when a task has a hard deadline or a fixed contractual date that must not drift when upstream bars are rescheduled.

Connector Rendering

When the timescale draws bars, each bar element receives `data-pred` and `data-tbid` HTML attributes. After rendering, `tbCanvas.js` iterates the bar elements, reads `data-pred` on each successor, looks up the corresponding predecessor element by `id="key{tbID}"`, and draws a connector line between the two. If the predecessor element is not found in the current view (filtered out or off-screen), the connector is silently skipped.

Important — Simplified Dependency Relationships (Scheduling Light)

Traditional project scheduling tools support four dependency relationship types: **Finish-to-Start (FS)**, **Start-to-Start (SS)**, **Finish-to-Finish (FF)**, and **Start-to-Finish (SF)**, each optionally combined with a positive or negative lag value. This platform intentionally implements a simplified subset of that model.

Only **Finish-to-Start** relationships are supported, with the option to apply a **positive lag** (a gap between the predecessor finishing and the successor starting) or a **negative lag** (which effectively approximates a Start-to-Start with overlap, allowing the successor to begin before its predecessor finishes). Finish-to-Finish and Start-to-Finish relationships are not supported, as Finish-to-Finish is rarely practical in day-to-day scheduling and Start-to-Finish is almost never used in practice.

When a **Task** is dragged on the timescale, the engine automatically cascades the move to any successor Tasks linked to it via `tbPredecessor`, recursively cascading through the full chain. A successor Task that has been **Pinned** will not move regardless of what its predecessor does — the pin takes priority over the cascade. This cascade is **limited to Tasks only** — dragging a Project bar does not trigger a cascade to linked successor Projects. The predecessor connector line will display on Project bars as a visual reference, but Project-level predecessors must be manually rescheduled.

This approach is intentionally described as **Scheduling Light** — enough dependency structure to communicate sequence and intent clearly on the timescale and in reports, without requiring teams to manage the complexity of a fully automated predecessor network across all levels of the hierarchy.

8. Where the Full Field List Lives

Every field in the system is a row in the `tbSchema` store, and you read it in the application rather than here: **Report Menu → Other → Schema Values**.

For each field it gives you the internal name, its label, **what it is for**, the picklist it draws from and whether that picklist is a tag list or the live resource pool, its data type, whether it is mandatory, whether it is validated, and which spreadsheet sheet it belongs to. It is filterable by sheet and by picklist, and its purpose column is full-text searchable.

That report is the field reference. A field list printed into a guide would start drifting from the product the day it was written; the report cannot.

The related stores, in one line each:

Store	What it holds
<code>tbTimebars</code>	every bar — the scheduling backbone
<code>tbMetaData</code>	the extended fields for each bar, joined on <code>tbMDID = tbID</code>

Store	What it holds
tbBaseline	a point-in-time snapshot that mirrors the tbTimebars structure exactly — the same fields populate it
tbResources	the resource pool, one row per person or generic role. L5 Allocations link to it through tbResID
tbTags	every picklist value, grouped by tbTagGroup , each group pointed at a field by tbTagTbInternalName

The picklists and their values are documented in the *Forms, Reports and Graphs Guide*; the stores and how they move in and out of the product are in the *Data Synchronization, Backup, Recovery and Retention Guide*.

9. What Recalculate All Does

Recalculate All is the single button that puts every derived value back in step. It matters because data arrives from three directions — typed in the app, generated by AI, synced from a workbook — and only the first of those keeps the derived fields current as it goes.

It runs in this order, and the order is the point: each step feeds the next.

#	Step	What it fixes
1	Consistency check	on every childless L4–L1 row, treats forecast and actual work and cost as the truth and corrects the remaining and percent-complete fields to match. Runs first , so corrected leaf values feed the rollup in the same pass. Allocations are left alone — they belong to the scheduling engine
2	Cost and work rollup	sums the six cost and work fields bottom-up in a single pass, and derives each parent's percent complete from its fresh totals
3	Hierarchy names	clears and rewrites the tbL1–tbL5 breadcrumb fields by walking tbSelfKey2 up the tree
4	WBS ordering	regenerates tbHierarchyOrder , the five-segment sort path
5	Grouping fill-down	empty grouping fields inherit the nearest tagged ancestor's value, so reports do not group everything as untagged. Runs after the ordering is fresh and before the rebuild below, so the joined rows carry the filled values
6	Joined table rebuild	rebuilds tbMdJoined , the cached join of bars and metadata that every report reads

Run it after any spreadsheet import, any AI generation, and any bulk edit — and before publishing, exporting, or trusting a portfolio total. Running it twice changes nothing the second time; every step is idempotent.

Resource calculations are not part of this. The weekly resource demand figures behind the charts and grids are rebuilt separately, so a resource chart that looks stale after a Recalculate All needs its own refresh rather than another run of this.

10. Values the Code Depends On

This section replaces the old *Picklist Values Locked By Code* document and the old *System-Reserved Metadata Fields and Values* section, which said similar things in different words and had both drifted from the product.

144 values across 46 picklists are locked. Twelve picklists are locked end to end. Every locked value is shown in **bold** in the A–Z reference in the picklist reference in the *Forms, Reports and Graphs Guide*, so there is one list to read rather than two to reconcile.

The picklists whose meaning is not obvious

Most locked values explain themselves. These do not, and getting them wrong is expensive:

Picklist	Field	Why the exact text matters
tbType	tbType	Tells the scheduling engine whether a row is a Project, Task, Milestone or Allocation. Everything downstream branches on it.
SubType	tbSubType	Turns a Task into a Risk, Issue or CR. The Risks & Issues module, and the health rules in section 7, find their rows by this value.
Timebar Status	tbMDStatus	Drives the whole PPM pipeline — what counts as a proposal, in-flight work, or closed. Rejected , On Hold and Blocked also make a row immune to the status automation.
Schedule Status	tbMDHealth	The word behind the schedule colour. Section 6 maps each value

Picklist	Field	Why the exact text matters
		to a colour by exact text.
Escalation Level	tbMDEscalationLevel	Executives and Directors push a project's scope, risk or issue health straight to Red, and drive Enterprise Dashboard notifications.
Budget Estimate	tbMDROMEstimate	Band names are parsed as money ranges, used as the fallback when Budget Cost is zero in the PPM select/kill process.
Risk Probability, Risk Impact	tbMDProbability, tbMDImpact	Each value carries a hidden 1–5 weight held in code, not in sort order . Rename one and risks stop scoring — see the warning below.
Labour Type	tbResLabourType	Human is what makes a resource appear in every people-picker; Generic and Human split the supply-and-demand analysis.
Health Cost/Hours/Issues/Overall/Risk/Schedule/Scope	tbMDHealth*	Green, Yellow, Red, Not Assessed are matched exactly to select an indicator colour. There is no Grey

Picklist	Field	Why the exact text matters
		— grey is how Not Assessed renders.
Executive Commitment	tbMDSeniorLevelCommitment	Feeds PPM scoring and notification thresholds.
Show In Reports	tbMDShowIn	Each value is a report filter. A renamed value silently removes rows from that report.

Risk Probability and Risk Impact deserve a specific warning. The 1–5 weights behind them are an explicit map in the code, deliberately *not* taken from the tag sort order — because **Not Assessed** shares a sort order with the worst value in both lists, and taking sort order would score an unassessed risk as the worst possible one. The consequence for you: **renaming a probability or impact value stops it scoring altogether**, and the risk silently reads 0 / Not Assessed. Add values freely; do not rename the five in each list.

Why this list can drift, and what to do about it

The lock list is maintained alongside the code, and the picklist values are maintained in the spreadsheet. Nothing automatically checks one against the other, and they have gone out of step before — the **Part-time Full-time** rename detached two values the portfolio report still branches on, and left the resource form's picker opening empty for months.

Two things protect you:

- **Config Integrity** after every import.
- The A–Z in the picklist reference in the *Forms, Reports and Graphs Guide* is **generated from the live data**, so a value that has been renamed away simply stops appearing in bold rather than sitting in the documentation as a value that no longer exists.

11. Status and Health Automation

The product can set status, stage, phase, schedule status and all seven health indicators for you, across the whole database, from the dates, baselines, budgets and risk register you already hold. This section documents what each button does.

Where: the buttons sit on the **Core Form** and on the **CS status form**.

Button	Sets	On
Update Status	Status, PPM Stage, PMI Phase, Schedule Status	Projects and Sub-Projects (all four); Tasks, Milestones, Allocations (Status only)
Sch Status	Schedule Status	Projects, Sub-Projects, Tasks, Milestones, Allocations
	...and the Health Schedule colour	Projects and Sub-Projects only — it is the only level the colour is reported on
Cost Hours	Health Cost, Health Hours	Projects and Sub-Projects
RIC	Health Scope, Health Risk, Health Issues	Projects and Sub-Projects
Overall	Portfolio rollup, then Health Overall	Portfolios, Projects, Sub-Projects
Run All	all five, in the correct order, in one pass	everything

Run All before sending reports to management. The buttons can be pressed individually and in any order, but Overall reads what the others wrote, so running them out of order gives an Overall based on stale inputs.

What automation never touches

- **A row whose Status is Rejected, On Hold or Blocked is skipped entirely.** Somebody put it there deliberately; nothing automatic moves it out, and skipping the whole row means it can never end up half updated.
- **Actual work and actual cost are never rewritten.** The automation reads them.
- Every indicator is recomputed from current state on every run, so a withdrawn escalation or a corrected date moves a project back *down* as readily as up. Nothing gets stuck on Red.

Update Status — lifecycle from the actual dates

A row's state is read from its actual dates alone:

The row has	State	Status	Stage	Phase	Schedule Status
no actual dates	Not started	New	1 Proposed	Initiating	Not Assessed
an actual start	Started	In progress	6 Delivering	Executing	On schedule
an actual finish	Finished	Closed	8 Closed	Closing	Completed

On Projects and Sub-Projects all four fields move together, so the row always reads consistently. On Tasks, Milestones and Allocations **only Status** is set — stage, phase and schedule status are left alone.

Sch Status — schedule health

Two methods, tried in order:

With a baseline — compares forecast finish against baseline finish. The tolerances scale with the length of the item, because five days late on a three-week task is not the same event as five days late on a two-year

programme:

Default	
Slip tolerance	the greater of 3 days or 5% of baseline duration
Late tolerance	the greater of 10 days or 10% of baseline duration
Result	When
Early	earlier than baseline by more than the slip tolerance
On schedule	within the slip tolerance either way
Slipping	late, but within the late tolerance
Late	later than the late tolerance
Completed	the row has an actual finish — this wins over everything

Without a baseline — it can only see what is already overdue against the Report Date, which is a genuinely different question. So it returns **Late** or **Not Assessed** and never **Early** or **Slipping**. Work still ahead of the Report Date stays **Not Assessed** rather than being flattered with an "On schedule" there is no evidence for.

The **Health Schedule** colour is a pure projection of the word, so the two can never disagree: Early and On schedule and Completed are Green, Slipping is Yellow, Late is Red, Not Assessed is grey.

Each run tells you which method it used. If it reports that no baselines were found, every schedule status on that run came from the Report Date alone — useful, but not schedule variance. **Set a baseline to get the real answer.**

Cost Hours — cost and hours health

Two questions, combined, with the **worse of the two** winning:

- **Variance** — is the forecast bigger than the plan? An estimating question. The plan is the baseline if there is one, otherwise the entered Budget Cost or Budget Hours. Green within 5%, Yellow within 15%, Red beyond — with an absolute floor (**5,000** currency units, **40** hours) so a small project's rounding noise does not read as an overrun.
- **CPI** — are we burning faster than we are earning? An execution question, meaningful only once something has actually been spent. Green at 0.95 and above, Yellow at 0.85, Red below.

Two safeguards worth knowing, because they are what stop a dashboard lying:

- **With neither a baseline nor a budget the answer is Not Assessed, never Green.** A brand-new project reporting green cost health is the most common way a portfolio dashboard misleads.
- **Below 15% complete, a variance-driven Red is reported as Yellow.** Early on, a forecast above plan is an estimate revision rather than an overspend — attention, not alarm. A genuinely bad CPI still shows Red at any percent complete, so nothing is hidden.

RIC — scope, risk and issue health

Each of the three indicators counts the Change Requests, Risks or Issues beneath the project. A row stops counting once its Status is **Closed** or **Rejected**.

Result	When
Not Assessed	the register is empty — no items of that kind at all
Green	items exist and none are open
Yellow	an open item is escalated to the Project Office , or 3 or more remain open
Red	an open item is escalated to Executives or Directors , or a risk is marked Escalated

Overall — the summary indicator

First, **Portfolios take their six indicators from their Projects** — the worst reading among them. Sub-Projects are deliberately excluded, because their costs already roll into the Project totals and their risks are already counted at Project level; reading them again would count the same problem twice. A Portfolio with no Project in progress stays **Not Assessed** rather than claiming health it has not earned.

Then Health Overall is set from the other six indicators on the same row:

Result	When
Red	any indicator is Red, or 4 or more are Yellow
Yellow	1 to 3 indicators are Yellow
Green	all assessed indicators are Green
Not Assessed	no indicator has been assessed

Indicators reading **Not Assessed** are excluded from the count entirely, so an unassessed dimension neither dilutes nor inflates the result. Zero assessed dimensions gives **Not Assessed**, never a confident green built on no evidence.

Tuning the thresholds

Every threshold above is a default, and every one can be overridden per install through an **apMetadataRules** JSON value on the admin panel row. Absent or unparseable, the defaults apply — so a malformed entry degrades to standard behaviour rather than breaking anything.

Key	Default	Controls
slipDaysMin / slipPct	3 / 0.05	Slipping tolerance
lateDaysMin / latePct	10 / 0.10	Late tolerance
costGreen / costYellow	0.05 / 0.15	cost variance bands
costMateriality	5000	absolute floor below which cost variance is ignored
hoursGreen / hoursYellow	0.05 / 0.15	hours variance bands
hoursMateriality	40	absolute floor, in hours
cpiGreen / cpiYellow	0.95 / 0.85	earned-value bands
evEarlyPct	15	percent complete below which variance cannot force Red

Key	Default	Controls
ricYellowCount	3	open items before an unescalated register goes Yellow
overallYellowToRed	4	Yellow count that forces Overall to Red

The dependency you must not break

Every value in the tables above — **In progress**, **On schedule**, **Executives**, **Green** — is matched by **exact text** against the picklists. The automation writes the value, and the picker has to be able to offer the same value back to the user afterwards.

Rename one of these values and the automation stops working on that field, silently. They are locked for exactly this reason (section 4). Note the lower-case second word in **In progress** and **On schedule**; they are not **In Progress** or **On Schedule**.

End of guide.

Part 4 — Data Synchronization, Backup, Recovery and Retention

Data Synchronization, Backup, Recovery and Retention User Guide

This guide covers everything about getting data **into** and **out of** Agilebars, Timebars and Costbars, and about keeping it safe while it moves: where your data lives, how to export it, how to bulk-edit it in a spreadsheet, how to bring it back, what protects you when the two get out of step, and how to recover when something goes wrong.

One thing to read before anything else. A spreadsheet import is **not a merge**. It clears each store and replaces it. Anything you changed in the app since your last export is gone. The application now watches for exactly this and **stops the import** rather than letting it happen — see [The import gate](#). It is the most important section in this guide.

Contents

- 1. [Where your data lives](#)
- 2. [The data stores](#)
- 3. [Which copy is the source of truth](#)
- 4. [Moving data out](#)
- 5. [Moving data in](#)
- 6. [The import gate and the sync alert](#)
- 7. [The spreadsheet workflow](#)
- 8. [Other data actions](#)
- 9. [Undo Bar Move](#)
- 10. [Backup, recovery and retention](#)
- 11. [Cloud publishing](#)
- 12. [Validation and integrity](#)

13. [Troubleshooting](#)

14. [Best practices](#)

1. Where your data lives

The products follow a **client-first** architecture:

- **Your data lives in your browser**, in IndexedDB
- **No internet is required** — the application works completely offline
- **Nothing moves unless you move it** — every export, import and publish is something you initiate
- **Cloud publishing is optional**

Three storage layers

Layer	Where	What it is for
IndexedDB	your browser	live work; every edit lands here immediately
Files	your computer	bulk editing, backups, offline work
Timebars Cloud	Timebars servers	team sharing and cross-device access, opt-in



About IndexedDB

Database name	TB
Version	1
Typical size	~5–20 MB for 100–500 bars; up to ~100 MB for large portfolios
Browsers	Chrome, Firefox, Edge, Safari

What this means in practice:

- Data survives closing the browser and restarting the machine.
- Data is **isolated per browser** — your Chrome data is not your Firefox data.
- Data is **per domain** — the Timebars site and the Costbars site hold separate databases.
- **Clearing browser data deletes it.** Take a Full Backup before you ever clear site data, change machines, or let IT reimaging your laptop.

The hierarchy itself — the five levels, what may parent what, and how rollups work — is covered in the *Data Model and Scheduling Engine Guide*, which owns that subject. This guide is about moving the data, not modelling it.

2. The data stores

Fifteen stores. Ten round-trip through the workbook; the rest are calculated.

Stores you control

Store	Holds	Edited through
tbTimebars	every bar on the canvas — portfolios, projects, tasks, milestones, allocations	canvas, Bar Creator, flyout forms, tabular grids
tbMetaData	100+ extended fields per bar — status, health, priority, business case	Core Form, FOCD forms, tabular grids
tbResources	the resource pool — people and generic resources	Shared Resource Pool report, or the workbook
tbAdminPanel	settings — report date, timescale, zoom, product, view state	Canvas Settings and the toolbars
tbTags	every picklist value	Picklist Values report, or the workbook
tbSchema	the master field list — one row per field in the system	Schema Values report, or the workbook
tbFields	which fields sit on which FOCD form	Fields Values report, or the workbook
tbCoreReport	the Core Form layout — field, section, level	CoreReport Values report, or the workbook
tbBaseline	baseline snapshots of tbTimebars	Set Baseline on the CS form
tbCharts	saved chart configurations	saving an Agilebars burndown

tbSchema is the newest of these and the most important to know about. It is the master field list every other configuration table joins to, and it replaced the old **BulkOperations** configuration table in the workbook. If your workbook predates it, see [The workbook sheets](#).

Stores the application calculates

These are rebuilt from the stores above. Do not edit them, and do not carry them back into the workbook.

Store	Holds	Rebuilt when
tbMdJoined	tbTimebars joined to tbMetaData, for reporting	canvas load, Recalculate All
tbResCalcs	time-phased hours and costs for burndown charts	burndown calculation
tbResCalcs2	weekly resource demand by role, skill, location, department	canvas load, Recalculate All
tbResCalcsUsage	weekly resource availability	resource data changes
tbDocuments	<i>deprecated</i> — retained for compatibility, nothing writes to it	—

3. Which copy is the source of truth

This is the most important idea in the guide, and the one that explains every alert, block and bulk-update tool that follows.

Your data can exist in four places at once — the browser, a workbook, a backup file, and a published pubset. **Only one of them is current at any moment**, and the product's job is to stop you overwriting the current one with a stale one.

The chain, in order

1. Editing in the app makes the browser the source of truth — immediately.

Change a picklist value on Picklist Values, a field label on Schema Values, a placement on CoreReport Values, or any bar or metadata field in a grid, and it is written to IndexedDB on the spot. There is no save step. From that instant, **the browser holds the only current copy**.

2. The workbook is now behind.

It still holds the configuration as it stood at your last export. Nothing about it looks wrong — it simply describes an earlier version of your product.

3. The application blocks the import rather than let you lose the change.

This is what the sync alert is. Rather than allow a stale workbook to overwrite a newer browser, the product **stops the import** and tells you which stores are affected. Export the changed stores, take them into the workbook, and the workbook becomes the source of truth again — at which point importing it is safe.

The full mechanism is in section 6. What matters here is *why* it exists: an import replaces stores wholesale, so without the gate the newer copy always loses.

4. Anything already published or backed up is behind too — and nothing warns you.

The gate protects the browser. It cannot protect copies that already left.

A pubset published last month, or a Full Backup taken last month, carries **last month's Tags, CoreReport and Fields**. Re-hydrate from it and your configuration silently reverts — renamed picklist values come back with their old names, fields you moved go back where they were.

This is not a defect; it is what a snapshot is. But it means **a configuration change makes every existing pubset and backup progressively more out of date**, and the more configuration you tune, the faster that happens.

5. Bulk Update is how published data catches up.

Rather than republish every pubset by hand after a configuration change, the **Publishing and Syncing Timebars** page has a **Bulk Update** tab that pushes one store into many existing pubsets at once.

Bulk Update mode	Use it for
Complete Replacement (JSON file)	replacing a whole store — the usual choice after a picklist or layout change

Bulk Update mode	Use it for
Filter & Set (no file)	changing one field's value wherever it currently equals something else, across the selected pubsets
Row-Level Patch (CSV file)	updating named rows only. First column is the store key, the rest are the fields to change

Select the target store, optionally filter the subset list by publish date, tick the subsets to update, preview, and apply. The stores it can push are **tbTimebars**, **tbMetaData**, **tbResources**, **tbTags**, **tbFields**, **tbCoreReport** and **tbSchema**.

tbSchema is available in **Complete Replacement** and **Row-Level Patch** only. Filter & Set builds its field list from the picklist table, which does not describe Schema's own columns, so Schema is greyed out in that mode.

What to do after a configuration change

- 1. **Export** the changed stores — the sync alert panel offers exactly those
- 2. **Import them into the workbook**, so it is the source of truth again
- 3. **Bulk Update** your live subsets with the changed configuration store
- 4. **Take a fresh Full Backup**, so your newest restore point matches your newest configuration

Steps 3 and 4 are the ones people skip, and they are the ones that hurt months later.

4. Moving data out

Three exports, for three different jobs. All of them write to your browser's download folder.

Menu item	Produces	Use it for
Full Backup	one file, tbFullBackup_YYYY-MM-DD_HH-MM-SS.js	restore points — this is your safety net
Export to CSV (SpreadSheet sync.)	15 CSV files , one per store	the spreadsheet round trip
Export to JSON	15 JSON files , one per store	integrations, AI agents, external tools

All three are on the **Hamburger Icon** menu.

Full Backup is the one that matters. It contains every store, so restoring it returns the application to exactly the state it was in. The other two are working exports.

A fourth route exists and is often the best one: the **sync alert panel** exports **only the stores you have actually changed** — see section 6.

Automatic backups

The application takes a Full Backup for you, without being asked, before:

- importing a spreadsheet, CSV or JSON file
- loading Demo Data
- Clear Timebars & Metadata

They land in your downloads folder with the same `tbFullBackup_...` naming. This is a safety net, not a backup strategy — the files accumulate unsorted among your downloads.

5. Moving data in

The drop panel

Hamburger Icon → **Import SpreadSheet, CSV or JSON** opens the **Drop To Import Data** panel. Drop a file onto it, or use the browse button.

Earlier documentation said to drag files onto the canvas. Use the import panel.

Accepted: `.xlsx`, `.xlsm`, `.ods`, `.csv`, `.js`, `.json`.

The filename rule

The application decides what a file *is* from the **start of its name**. A file whose name does not begin with one of these is rejected before anything is read:

Name starts with	The file is treated as
<code>tbClient...</code>	the workbook — all its sheets
<code>tbFullBackup...</code>	a full restore
<code>transferBars...</code>	bars transferred from another product
a store name — <code>tbTags...</code> , <code>tbMetaData...</code> , <code>tbTimebars...</code>	that single store

The last row is why the per-store exports round-trip cleanly: a file exported as `tbTags.json` can be dropped straight back in. It is also why **renaming a file breaks the import**. Keep the name the application gave it.

What an import actually does

This is the single most misunderstood behaviour in the product:

An import clears the store and replaces it. It does not merge.

Every row in the store is removed and the file's rows are written in their place. A row you added in the app and did not export is not preserved, not conflicted, not reported — it is simply gone.

The application does two things to protect you:

1. It takes a **Full Backup first**, automatically.
2. It **refuses to import at all** when it knows you have unexported local changes. That is section 6.

6. The import gate and the sync alert

Why this exists

You export to the workbook on Monday. On Tuesday you edit a few bars in the app, add a picklist value, set a baseline. On Wednesday you import the workbook you edited. Tuesday's work is destroyed, silently, with nothing on screen to suggest it was ever at risk.

The application now tracks this.

The alert icon

When any workbook-backed store holds edits that are **not yet in the spreadsheet**, a **warning icon appears on the top menu**.

Action	What happens
Single click	a message: <i>"Local changes pending — double-click for instructions."</i>
Double click	the sync alert panel opens

Double-click rather than single, so a stray click cannot throw a panel over whatever you were doing.

While the icon is showing, imports stop

Attempting an import gives you:

Import stopped. You have local changes that are not yet in the spreadsheet. Double-click the alert icon on the top menu for instructions.

The panel opens automatically, listing the stores concerned.

The check happens **after** the filename is validated and **before** the pre-import backup runs — so a blocked import has touched nothing at all. No backup file, no partial write, nothing to clean up.

The two ways out

The panel gives you two buttons, and they are not equivalent:

Button	What it does
Export to Remove Block	writes one JSON file per changed store to your downloads folder, ready to import into the workbook, then clears the block
Remove Block	clears the block and keeps your local changes — the next import will overwrite them

Export to Remove Block is the safe path. Remove Block is there because without an escape hatch a stale flag would leave the application unable to import at all — use it when you genuinely do not want the local changes.

Exporting a store through the normal Export to CSV or Export to JSON also clears that store's own mark automatically.

What gets watched

Nine stores can be changed from inside the app, and each is watched for the work that changes it:

Store	Marked dirty by
Timebars	moving, resizing or rescheduling a bar; changing the Report Date; any field edit on a form or grid — Owner included
MetaData	field edits on a FOCD form, the Core Form or a tabular grid
Resources	edits in the Shared Resource Pool view or on the Resource form
Tags	editing, adding or deleting a value in Picklist Values
Baseline	setting or clearing a baseline from the CS form, at any level
Charts	saving a burndown chart in Agilebars
Fields	edits in Fields Values
Schema	edits in Schema Values
CoreReport	edits in CoreReport Values

Five stores are deliberately **not** watched. Four of them — ResCalcs, ResCalcs2, ResCalcsUsage and MdJoined — are calculated, rebuilt by Recalculate All, and never worth carrying back. The fifth, **AdminPanel**, holds your settings and view state and is written on almost every click; watching it would leave the warning showing permanently and telling you nothing.

Two routes into the workbook

They are alternatives, not steps:

- **Export to CSV** on the main menu gives you **everything, every time**, and you decide which sheets to import.
- **The sync alert panel** gives you **only the changed stores** — less to import, and less chance of overwriting a sheet you did not mean to touch.

Setting it up

The tracking needs one column, **apDirtyStores**, on the **AdminPanel** worksheet. It needs no database change — if your workbook does not have it yet, add it. Without the column the app still works; it simply cannot remember the warning across a reload.

7. The spreadsheet workflow

The template

Excel	Hamburger Icon → Download Excel SS — tbClientMaster...xlsm
LibreOffice Calc	Hamburger Icon → Download Libre Office Calc SS — tbClientMaster...ods

Both sync with all three products. **The filename must still begin with [tbClient](#)** — that prefix is how the application recognises it.

The cycle

1. Export to CSV from the app (15 files)
2. Import All on the workbook's Setup tab
3. Edit in Excel or Calc
4. Save the workbook
5. Import the workbook back through the drop panel

Step 1 — export

Hamburger Icon → **Export to CSV (SpreadSheet sync.)**. Wait for all files to finish downloading before moving on.

Step 2 — tell the workbook where they are

On the **Setup** tab, set the **Location** column to your browser's download folder. The **Import Yes/No** column controls which stores are pulled in by *Import All*, so you can leave the calculated stores out.

Step 3 — import into the workbook

Import All on the Setup tab pulls every sheet marked Yes. Individual sheets have their own Import button if you only want one.

Step 4 — edit

This is the point of the exercise: copy-paste, fill-down, formulas, filters, sorting, conditional formatting, bulk status changes, adding and deleting rows.

Sheet structure:

Row	Contents
Row 1	database field names — tbID , tbName , tbStart . Never change these.
Row 3	friendly labels, for your reference
Row 4+	your data

Changing Row 1 breaks the import. It is how each column is matched to a field.

Step 5 — save, and step 6 — import back

Save the workbook (macros enabled writes the updated CSVs out), then drop it on the import panel. The app takes its automatic backup, replaces each store from its sheet, and redraws.

The workbook sheets

Fifteen sheets map to the fifteen stores:

Sheet	Store	Edit it?
Timebars	tbTimebars	yes
MetaData	tbMetaData	yes
Resources	tbResources	yes
AdminPanel	tbAdminPanel	yes — settings and holiday exceptions
Tags	tbTags	yes — picklist values
Schema	tbSchema	yes — the master field list
Fields	tbFields	yes — FOCD form layout
CoreReport	tbCoreReport	yes — Core Form layout
Charts	tbCharts	yes
Baseline	tbBaseline	no — snapshot
MdJoined	tbMdJoined	no — calculated
ResCalcs	tbResCalcs	no — calculated
ResCalcs2	tbResCalcs2	no — calculated
ResCalcsUsage	tbResCalcsUsage	no — calculated
Documents	tbDocuments	no — deprecated

How a sheet is laid out

Every data sheet in the workbook follows the same three-row anchor, and the macros, the exporter and the importer all depend on it:

Row	Holds
Row 1	the exact internal field names — tbID , tbName , tbStart — as column headers
Row 3	friendly column titles, for reading. Nothing reads these
Row 4	the first data row

Do not insert or delete rows above row 4, and do not edit row 1. A field name typed into row 1 that does not exist in the Schema sheet is not imported and is not reported — the column is silently ignored. Change a name in Schema first, then in row 1.

The two sheets that carry your plan are **Timebars** (which maps to **tbTimebars**) and **MetaData** (which maps to **tbMetaData**), joined by **tbMDID = tbID**. Where a MetaData column mirrors a Timebars one — **tbMDName**, **tbMDTypes**, **tbMDSubtype**, **tbMDCustomerID** — take the value from the matching Timebars row rather than typing it again.

Validation, and where it comes from

The workbook's macros check data quality as you edit — mandatory columns, data types, valid picklist values — and highlight problems in red.

The validation rules now come from the *Schema* sheet. They used to come from a table on the *BulkOperations* sheet. If you are maintaining the macros, *FormatAndValidateV1.bas* reads *CFG_SHEET = "Schema"*. The *BulkOperations* sheet still exists for its import and export control ranges, but its field-configuration table is gone — one row per field now lives in *Schema*, so no field can be configured twice with two different answers.

Two Excel-side tools worth knowing about:

- **The Column Managers** — show and hide groups of columns on the Timebars and MetaData sheets, so you can work on one section at a time instead of scrolling across hundreds of columns.
- **PickListsV1** — one picklist engine driving the dropdowns for both Tags and Resources values.

Before you import: the six checks

The workbook's validators catch most of this as you type, but these are the six that break an import or produce a broken hierarchy, and they are worth a deliberate pass:

1. Every Timebars row has a **unique *tbID***, and every MetaData row has a ***tbMDID* that matches one**
2. Every non-root row has a ***tbSelfKey2* pointing at a parent ID that exists**
3. ***tbType*** holds only the reserved values — Portfolio, Project, Sub-Project, Task, Milestone, Gate, Allocation
4. All dates are ***DD-MMM-YYYY***
5. ***canvasNo* is 1** on every row
6. Headers are in **row 1** and data starts at **row 4**

Numeric columns must hold numbers — **0**, never blank, never **na**. The full field rules are in the *Data Model and Scheduling Engine Guide*.

After every import: run Config Integrity

An import can leave the four configuration tables pointing at each other incorrectly — a field renamed on one sheet and not another, a row deleted while others still reference it, a picklist emptied. **None of it throws an error.**

Report Menu → Other → Config Integrity measures it. Make it the last step of every import. See the *Forms, Reports and Graphs Guide* for what it checks.

The rule about editing in two places

Do not edit in the application while the same data is out in a spreadsheet. The import gate now catches the dangerous half of this automatically, but the discipline is still worth keeping:

1. Export
2. Edit in the workbook
3. Import
4. Resume working in the app

8. Other data actions

All on the **Hamburger Icon** menu.

Bulk Manage Bars

Duplicate bars and children — enter the Bar ID (bottom-left of the bar on the canvas) and click Duplicate. Copies appear near the original, names prefixed **duplicate**, IDs suffixed **-d**. Good for template projects and what-if scenarios.

Transfer bars between products — enter the Bar ID and an optional ID prefix to avoid collisions, click Transfer, and a **transferBars...json** file lands in your downloads. Open the target product and drop that file on the import panel. Use it when a waterfall project becomes agile, or an agile work package becomes a portfolio item.

Migrate Rich Text

Converts legacy rich text from the old Quill format to the current editor's format. **Safe to run at any time** — already-converted and plain-text fields are left alone.

Run it once after upgrading, and again after importing an old backup file. If descriptions or business-case sections look like raw markup, this is the fix.

Clear Timebars & Metadata

Gives you a clean canvas without losing your configuration. A Full Backup is written automatically first, and the action asks for confirmation.

Cleared	Kept
Timebars, MetaData, Baseline, Charts , and all four calculated stores	AdminPanel, Schema, Tags, Resources, Fields, CoreReport

Note that **baselines and saved charts go too** — they describe bars that will no longer exist. The undo history is cleared for the same reason. Everything that makes the application *yours* — your picklists, your form layouts, your resource pool, your settings — survives.

Load Demo Data

Replaces current data with a demo dataset for the product you are running. A Full Backup is written automatically first.

Delete Database

Destroys everything, permanently. Refresh the browser afterwards and a fresh database is built with demo data. **Back up first** — this one does not do it for you.

9. Undo Bar Move

You moved a few bars, and the plan now looks wrong. **Undo Bar Move** puts them back.

Where it is: right-click anywhere on the canvas to open the **Shortcut Menu**. Undo Bar Move is the first item.

Undo remembers your **last 5 moves**, most recent first. Each click steps back one move.

Undo is not a backup. It covers bar moves only, only since your last data load, and only in this browser. For anything larger — a spreadsheet sync, a bulk import, a version upgrade — take a **Full Backup** first.

What Undo puts back

A drag or resize does not change one bar. It changes everything the scheduling engine recalculates as a result, and Undo reverses **all** of it:

- **Bar positions** — where each bar sat on the timescale, and its row
- **Dates** — Start, Finish, Actual Start, Actual Finish
- **Durations** — Duration and Remaining Duration
- **Work** — Work, A Work, Work Remaining
- **Cost** — Cost, A Cost, Cost Remaining
- **Progress** — % Complete
- **Allocation values** — Calendar hours, % Allocated, Pay Rate, Override flag
- **Baseline bars** — the matching baseline bar returns to its row

Because a move cascades, Undo restores more than the bar you touched. Move a Project and you moved its sub-projects, tasks, allocations, any linked successor tasks, its baseline bar, and the rolled-up totals on the Portfolio above it. Undo puts that whole branch back — which is what makes it trustworthy, but does mean undoing a Portfolio drag is a big operation.

What Undo never touches

Metadata is completely outside Undo. One rule with two halves, and both are worth learning:

	Undo restores it?	Does editing it cost an undo step?
Schedule fields (dates, durations, work, cost, % complete, allocation values, bar position)	Yes	Yes — see below
Everything else (Name, Owner, tags, descriptions, Status, Priority, all Metadata, Budget Cost, Budget Hours, WBS Description)	No	No

So you can rename bars, change owners, tap picklists, write status updates and edit descriptions **whenever you like** — before, between or after moves. Undo will not put any of it back, and none of it uses up your undo steps.

This works field by field, not bar by bar. Drag a bar, rename that same bar, then press Undo: the bar's **dates go back** and its **new name stays**.

Your typed schedule values are protected

Some values you type by hand are the *same* values Undo restores — hours to go, % Allocated, and the manual Cost / Work / % Complete cells on the status form. Without a safeguard, undoing an older move would quietly wipe out what you typed.

It cannot. **Typing one of those values clears every restore point older than it.**

1. You move two bars → 2 moves available to undo
2. You type a new **Hours to go** on an allocation → **undo list clears**, the button greys out
3. You move two more bars → 2 moves available again

4. You undo both → the bars go back, and **your typed value is still there**

You can undo moves made *after* your edit, but never back *past* it. Hover the menu item and it tells you exactly what happened, and when.

Which edits clear the list:

- **Hours to go, % Allocated** or **Available hours per day** on the Allocation Hours Calculator
- **Remove Override** on the Allocation form
- **Manual Cost, A Cost, Cost Remaining, Work, A Work, Work Remaining** or **% Complete** on the CS status form

Editing a bar's Name, Owner, tags, description, Status, Priority, Budget Cost or Budget Hours does **not** clear the list — Undo would never have touched those anyway.

Reading the menu item

What you see	What it means
Undo Bar Move	one restore point available
Undo Bar Move (3)	three moves can be undone, most recent first
greyed out	nothing to undo — hover it and the text says why

When Undo clears itself

Undo **survives a page refresh**. It is **cleared** whenever the underlying data is replaced:

- Loading Demo Data
- Clear Timebars & Metadata
- Restoring a Full Backup
- Importing a spreadsheet, CSV or JSON that replaces bars
- Typing a schedule value by hand (above)

This is deliberate. A restore point describes the bars that existed when it was taken; applying it to freshly loaded data would put back bars that no longer belong.

Undo is per browser, per machine

Restore points live in your browser's local database alongside your project data. They are **not** published to the cloud, **not** shared with colleagues, and do **not** travel to another computer or browser profile. Two people working on the same plan each have their own Undo.

Undo history is **not included in a Full Backup** — backup files are unchanged in size and content by this feature, and restoring one correctly clears the undo list.

Undo troubleshooting

Symptom	Cause and fix
Undo is greyed out	No restore point. Hover it — either nothing has moved since the last data load, or a typed schedule value cleared the list

Symptom	Cause and fix
My undo list disappeared	You typed a schedule value. That protects your edit. Moves from now on are undoable again
Undo put back more than my last drag	Two things do this, both intended: a move cascades to children, siblings and linked tasks; and several drags within half a second are grouped into one restore point
Undo put back a bar I did not move	Same cause — it was part of the cascade
Undo did not restore my Name / tag / description change	Correct. Metadata is outside Undo entirely
Undo is slow on a large plan	It re-runs the full recalculation and redraw, the same cost as Recalculate
My colleague cannot undo my move	By design — undo history is local to one browser

10. Backup, recovery and retention

What each kind of file contains

File	Contains	Restores to
Full Backup — <code>tbFullBackup_...</code>	every store	exactly the state the application was in when it was taken
Per-store JSON / CSV — <code>tbTags.json</code> , <code>tbTimebars.csv</code> ...	one store	that store only, replacing what is there
The workbook — <code>tbClient...</code>	the ten editable stores	those ten stores
A pubset in the cloud	fifteen stores	the application, on any device you log in from

Only the Full Backup is a restore point. The others are working files: each replaces part of your data and leaves the rest as it is, which is exactly what you want for a bulk edit and exactly what you do not want after a mistake.

Automatic backups

The application takes a Full Backup for you, unasked, immediately before it does anything destructive:

- importing a spreadsheet, CSV or JSON file
- loading Demo Data
- Clear Timebars & Metadata

They land in your downloads folder with the same `tbFullBackup_...` name. **This is a safety net, not a backup strategy.** The files accumulate unsorted among everything else you have downloaded, and after a fortnight you will not be able to tell which one is the good one.

How to restore

Drop the file onto **Hamburger Icon** → **Import SpreadSheet, CSV or JSON**.

A Full Backup restore **replaces every store**. Anything done since that backup was taken is gone — including configuration changes, which is the case people forget. Take a fresh backup of the current state first if there is any chance you will want to come back to it.

Recovery scenarios

What happened	What to do
Browser data was cleared — you open the app and your work is gone	Restore your most recent Full Backup. Then stop the browser clearing site data on exit, or this repeats
A different browser or profile — the data looks missing but nothing was lost	IndexedDB is per browser, per profile, per machine. Go back to the browser you were working in, or restore a backup into this one
You imported the wrong file	Restore the automatic Full Backup the application took immediately before that import. It is the newest tbFullBackup_... in your downloads folder
An import overwrote work you had done in the app	Same — the automatic pre-import backup holds it. This is what the import gate exists to prevent; see section 6
Moving to a new machine	Take a Full Backup on the old machine and import it on the new one. Publishing to the cloud and re-hydrating also works and carries thirteen of the stores
The numbers look wrong rather than missing	Not a restore problem. Run Recalculate All , then Config Integrity . See section 12
A picklist opens empty, or a field lost its list	A configuration reference broke, usually from a partial import. Run Config Integrity — it names what no longer resolves

What a backup does not contain

- **Undo history.** Undo lives outside the backup, is cleared whenever data is loaded or imported, and is per browser. It is not a recovery tool — see section 9.
- **Your browser's view state** beyond what the AdminPanel store holds.

Retention

A backup you cannot find is not a backup.

Practice	Why
Keep a dedicated backups folder — not Downloads	the automatic pre-import backups land in Downloads and bury the ones you meant to keep
Keep three to five recent Full Backups	enough to step back past a bad day; few enough to tell apart

Practice	Why
Keep one monthly backup for as long as the plan matters	the daily set will have rolled over long before anyone asks what the plan looked like at approval
Keep the filename the application gave it	the timestamp is in the name, and the import decides what a file is from the start of its name. Rename it and it will not import
Take one before, not after	any spreadsheet import, bulk operation, demo-data load, configuration change, or browser change
Take a fresh one after a configuration change	so your newest restore point matches your newest configuration — see section 3

How often? Daily while you are actively working a critical plan; weekly for stable work. The honest test is how much re-typing you are willing to do, because that is exactly what a restore costs you.

11. Cloud publishing

Optional, user-initiated, and covered fully in the *Cloud Publishing Guide*. In outline:

Publish — Main Menu → Publish, log in, then **Re-Publish!** on the PubSet. Your data uploads and the Cloud Dashboard updates.

Re-Hydrate — on another device, Main Menu → Publish, log in, **Re-Hydrate!**. Data downloads to that device.

Control	user-initiated only, never automatic
PubSets	up to 10 separate datasets, subscription dependent
Dashboard PubSet	the one with the blue background
Security	HTTPS in transit, token authentication, data isolated per account

Re-Hydrate replaces all local data. Same warning as an import: back up first, and check the PubSet's Last Published date. If your local data is newer, publish before you re-hydrate.

12. Validation and integrity

Four layers, at different moments:

As you type — required fields enforced, date pickers, picklists, numeric fields reject text.

In the workbook — macro validators check mandatory columns, data types, references and duplicate IDs, and highlight problems in red before you import. Driven by the **Schema** sheet.

During import — field names checked against the schema, types checked, unique constraints enforced. Multi-store operations are all-or-nothing: an error rolls the whole thing back.

After import — **Config Integrity** for the configuration tables (section 7), and **Recalculate All** to rebuild every calculated store.

Referential behaviour worth knowing: deleting a parent bar deletes its children.

13. Troubleshooting

My data disappeared after restarting the browser

Browser data was cleared, or you are in a different browser or profile. Restore your most recent Full Backup through the import panel. Then check your browser is not set to clear site data on exit.

The import is blocked and I do not know why

You have local changes that are not yet in the spreadsheet. That is the gate working. Double-click the warning icon on the top menu, then use **Export to Remove Block** to keep those changes, or **Remove Block** to discard them. See section 6.

A pubset or backup brought back old picklists or an old form layout

It was published or taken before you changed the configuration, and a snapshot carries the configuration of its moment. Re-apply the change, then use **Bulk Update** on the Publishing and Syncing page to push the current store into your existing pubsets, and take a fresh Full Backup. See section 3.

The file will not import at all

The filename does not start with a recognised prefix. It must begin with **tbClient**, **tbFullBackup**, **transferBars**, or a store name. Do not rename exported files, and do not drop folders.

The import ran, but my changes are not there

Two likely causes. Either the sheet you edited was not marked Yes in the Setup tab's Import column, or you edited the sheet but did not save the workbook before dropping it. Re-export, re-import, and check the Setup tab.

An import overwrote work I had done in the app

An import replaces stores; it does not merge. The gate exists to prevent exactly this — if it did not fire, the changes were made before the tracking was set up, or **apDirtyStores** is missing from your AdminPanel sheet. Restore the automatic backup taken immediately before the import; it is in your downloads folder.

A picklist opens empty, or a field lost its list

A configuration reference has broken — usually a value or picklist renamed on one sheet and not another. Run **Config Integrity**.

Rich text fields show raw markup

Run **Migrate Rich Text** from the Hamburger menu. Safe to run repeatedly.

Storage quota exceeded

Rare. Export your data, delete the database, and import back only what you need. Consider splitting very large portfolios.

The data looks inconsistent — totals do not add up

Run **Recalculate All** (right-click the canvas). It rebuilds every calculated store. If it persists, restore a backup.

Macros are not working in the workbook

Excel blocks macros in files downloaded from the internet. Right-click the file → Properties → **Unblock**, or enable macros when prompted. Without them the Import and Export buttons do nothing.

14. Best practices

Backups

Covered in full in section 10. The short version: keep them in a dedicated folder, keep three to five recent ones plus a monthly, take one before anything destructive, and take a fresh one after a configuration change.

Backups vs. Undo. Undo handles the small everyday mistake — a bar dragged to the wrong place. It does not replace a backup: bar moves only, five of them, and cleared whenever data is loaded or imported.

The sync loop

1. **Export** — everything, or just the changed stores from the alert panel
2. **Edit** in the workbook; do not edit the same data in the app meanwhile
3. **Save** with the **tbClient** name intact
4. **Import** through the drop panel
5. **Run Config Integrity**
6. **Run Recalculate All**
7. Keep the workbook as a record of what changed

Test a major restructuring against demo data before running it on real data.

Multiple devices

1. Nominate one device as primary
2. Publish from it after changes
3. Re-hydrate on secondary devices *before* working, never after
4. Do not edit on two devices at once
5. Agree a team convention: last person to edit publishes

Data hygiene

- Delete test and demo bars before production use
 - Archive completed projects
 - Keep naming conventions consistent
 - Review unused resources and picklist values periodically
 - Run **Config Integrity** after every import, not just when something looks wrong
-

Related documents

For	See
Picklists, forms, and the configuration reports	<i>Forms, Reports and Graphs Guide</i>
The hierarchy, rollups and the scheduling engine	<i>Data Model and Scheduling Engine Guide</i>
Field-by-field detail	Schema Values report
Publishing and the Cloud Dashboard	<i>Cloud Publishing Guide</i>
Moving from a legacy system	<i>Getting Started</i> — its data-migration chapter

Supersedes: Spreadsheet Sync User Guide, Data Management User Guide, and Data Structure User Guide.

Part 5 — Forms, Reports and Graphs

Forms, Reports and Graphs Guide

This guide covers everything on the **Report Menu**, and it comes in two parts.

Part A is the reports about your work — tabular views, cards, charts, timelines and the Gantt Report, with the filters, grouping and inline editing that go with them.

Part B is the four **Editable Reports** that configure the product itself: what the picklists offer, which fields sit on which form, and how the Core Form is laid out. They sit on the same menu because they are reports; they are documented here because they are also the only place that configuration is done.

Contents

Part A — Reports and graphs

1. [How to access reports](#)
2. [Report categories](#)
3. [Tagging rows into reports — the Show In field](#)
4. [Special reports](#)
5. [Common report features](#)
6. [Where report data comes from](#)
7. [Publishing to the Cloud Dashboard](#)
8. [Tips](#)
9. [Troubleshooting](#)

Part B — Configuring the product through the Editable Reports

10. [The two rules that decide everything](#)
11. [Where configuration lives](#)
12. [Picklists and tagging](#)
13. [Dynamic forms \(FOCD\)](#)
14. [The Core Form](#)
15. [What is configurable, by product](#)

[Appendix A — Picklist reference A–Z](#)

Part A — Reports and Graphs

1. How to Access Reports

Choose **Main Menu > Reports** to open the Reports menu. It is organised into five menus:

Menu	Contains
Portfolio	Projects by Portfolio, PPM Cards Report, Drilldown from Portfolio, Portfolio Detail Report
Project	General Tabular View, Detail Cards, Baseline Variance
Task	Task Tabular Report, Gantt Report
Resource	Resource Usage, Shared Resource Pool
Other	Print WBS, Picklist Values, Fields Values, Schema Values, CoreReport Values, Config Integrity

Most reports offer search, filtering, sorting and export. Reports can be published to the Timebars Cloud Dashboard with one click.

If you are looking for a report you remember by name, several that used to be separate menu items are now reached from inside another report:

Looking for	It is now
Items Overdue, Milestone Horizon	tick-box filters inside the Task Tabular Report , and on the Gantt Schedule View header (reached from the Main Menu, not the Report Menu)
Project Charter, Project Assessment	embedded on Project cards in Detail Cards
Portfolio Status and Balancing	embedded per portfolio in Portfolio Detail Report
Editable Grids / Data Management Grids	removed — each store now has its own report, see <i>Configuration reports</i> below

2. Report Categories

General Reports

General Tabular View

- Comprehensive data table displaying all projects, tasks, milestones, and allocations
- Powerful search, filter, and sort capabilities across all fields
- Multiple column presets available:
 - Default Fields: Essential project information
 - Common Fields: Frequently accessed metadata
 - Notes & Descriptions: Detailed project narratives
 - Allocation Fields: Resource assignment details
 - KPI Fields: Key performance indicators

- Variance Fields: Baseline comparison metrics
- Investment Fields: Strategic alignment (Costbars)
- Project Charter Fields: Comprehensive investment analysis
- Export to Excel/CSV with one click
- Color-coded rows by type (Portfolio, Project, Task, Milestone)
- Click any row to open the edit form

General Card View

- Visual card layout showing projects and tasks
- Each card displays health indicators, status, priority, and key metadata
- Expandable sections for descriptions and notes
- Filter and search across all visible cards
- Color-coded by hierarchy level (Portfolio, Project, Sub-Project, Task)

Print WBS (Work Breakdown Structure)

- Professional WBS reports organized by hierarchy
- Shows parent-child relationships with indentation
- Includes WBS number, name, and description
- Copy and paste into markdown editors for sharing
- Export as documentation for project charters or scope baselines

All Drilldown (Portfolio Hierarchy)

- Interactive 5-tier hierarchy visualization
- Shows complete structure: Portfolio → Program → Project → Task/Milestone → Allocation
- Collapse and expand any level to focus on specific areas
- Color-coded levels for instant recognition
- Validates parent-child relationships and identifies orphaned items
- Export hierarchy as diagrams (Mermaid or GraphViz formats)

Portfolio Reports

PPM Tabular Report

- Strategic portfolio analysis with dual scoring dimensions
- **Strategic Value (SV) Score:** Alignment with organizational objectives
- **Ability to Execute (AE) Score:** Resource availability and capability
- **Final Decision Score:** Weighted combination for prioritization
- Weekly resource demand grid (32 weeks visibility)
- Investment category, objective, strategy, and initiative fields
- System-generated decision notes
- Search and filter by project selection criteria
- Export for portfolio review meetings

PPM Cards Report

- Card-based view of portfolio projects with visual indicators
- Expandable tabs for Benefits, Impact Analysis, and Notes
- Health dashboard with 7 dimensions

- Filter by investment category, status, priority

Drilldown from Portfolio

- Portfolio report with color hierarchy showing all levels
- Visual representation of portfolio structure
- Click to drill down into specific portfolios or programs
- Shows relationships between portfolios, programs, and projects

PPM Stop Light Report

- Visual health dashboard with color-coded status indicators
- Red/Yellow/Green indicators across all portfolio dimensions
- Quickly identify projects requiring attention
- Exception-based management view

Project Reports

Project Status

- High-level summary of all projects
- Green, yellow, and red health indicators
- Key metrics: schedule variance, cost variance, budget utilization
- Resource allocation status
- Risk and issue counts
- Escalation flags
- Sortable by any column
- Export for weekly status reporting

Project Core Report

- Command center for project data in customizable card layout
- **7-Dimension Health Dashboard:** Overall, Scope, Schedule, Cost, Hours, Risk, Issues
- Expandable rich text sections: Executive summaries, descriptions, notes
- Inline editing: Update project fields without leaving the report
- Print-ready layouts for stakeholder distribution
- Real-time health monitoring
- Color-coded cards by project status

All Tabular

- One large table showing all columns and all rows available
- Complete dataset with every field in the system
- Advanced filtering across 100+ fields
- Export entire dataset to Excel or CSV
- Useful for comprehensive data analysis

Project Stop Light Report

- Individual project health metrics with color indicators
- Drill-down into specific performance areas
- Track trends over time

- Early warning system for project issues

Baseline Variance Report

- Compare actual performance against approved baselines
- Variance calculations for:
 - Start Date Variance (days early/late)
 - Finish Date Variance (days early/late)
 - Duration Variance (work days)
 - Work Variance (hours)
 - Cost Variance (budget units)
- Advanced filtering with operators (>, <, =, >=, <=)
- Filter by hierarchy level (L1-L5)
- Type, status, priority, owner filters
- Earned value analysis support

Project Charter Document

- Comprehensive investment analysis report
- Strategic alignment (category, objective, strategy, initiative)
- Risk and complexity scoring with automated calculations
- Financial metrics (cost, hours, ROI indicators)
- Baseline comparison metrics
- Health indicators (7 dimensions)
- Resource demand by role and project
- Timeline visualization
- Export to PDF for board presentations

Task Reports

The Task menu holds two reports over the same data. The **Task Tabular Report** is a table; the **Gantt Report** is the same table with a timescale beside it.

Task Tabular Report

- Every Task, Milestone and Gate across the whole hierarchy in one table
- Free text name search, plus picklist filters for Status, PM, Owner, State, Portfolio (L1) and Project (L2)
- **L3 / L4 / L5 tick boxes** decide how deep the table goes. L1 Portfolio and L2 Project rows are always shown; the three lower levels are opt-out
- Column-set picker — Default, Schedule Standard, Schedule Advanced, Notes, Variance, Risks & Issues, All Columns
- **Inline editing** — click a cell and type; picklist cells open a picker and save on the tap (see *Direct Editing* below)
- Two override tick boxes replace the retired *Items Overdue* and *Milestone Horizon* reports (see below)

Overdue and Milestone Horizon (the two override tick boxes)

These sit above the table on the Task Tabular Report, and on the header of the **Gantt Schedule View** (Main Menu). They are *overrides*: while one is ticked it replaces the L3/L4/L5 level filtering with its own result set. They are mutually exclusive — ticking one clears the other — and touching any level tick box clears both and returns you to the normal view.

Tick box	What it shows
Overdue	Tasks and Milestones that are late — planned start <i>or</i> planned finish is in the past — and are still missing an actual start or an actual finish. A bar that is genuinely finished on time never appears
Milestone Horizon	Milestones. On the Task Tabular Report this is <i>every</i> Milestone. On the Gantt Schedule View it is limited to a recent horizon — nothing whose start or finish is more than 14 days in the past — so the list stays focused on what is coming up or has only just passed

Ticking **Overdue** on the Gantt Schedule View also switches the columns to the variance set, because the question you are about to ask is "by how much?". Un-ticking it puts the default columns back.

While either override is on, rows are banded by real hierarchy — Portfolio, Project and, where one genuinely exists, Sub-Project — so a long overdue list still reads as "whose is this?". A Task or Milestone with no Sub-Project sits directly under its Project band, and empty levels produce no placeholder band.

Gantt Report

A two-pane report: the filterable, editable table on the left, and the bars on a timescale on the right. Both panes always render the same filtered set, so what you see in the table is what you see on the timescale.

- **Filter panel** (collapsible): name search, Reset Filters, and picklists for Status, PM, Owner, State, Portfolio (L1) and Project (L2)
- **L3 / L4 / L5 tick boxes** at the foot of the panel — same rule as elsewhere, L1 and L2 always show
- **Show In** picklist along the top — narrows to rows tagged for one report (see *Tagging Rows Into Reports* below). It combines with everything else
- **Columns** picker and **row spacing** picker along the top
- **Grouping By** picklists — see the next section
- Inline editing throughout, including **Predecessor** on Task and Project rows
- L2 Project rows are emphasised: darker green band, larger bold name, and the project name drawn on the timescale itself
- Sort by clicking any column header; while grouping is on, the clicked column sorts the tasks *inside* each band rather than the whole list

The Overdue and Milestone Horizon tick boxes are deliberately **not** on the Gantt Report — they live on the Task Tabular Report and the Gantt Schedule View. The Gantt Report's job is the grouped structural view.

Grouping the Gantt Report

Set one or both **Grouping By** picklists and the Gantt Report stops being a flat hierarchy and reorganises the tasks under bands. The structure is:

L1 Portfolio	(real row)	
L2 Project	(real row)	← grouping starts under L2
▶ Sub-Project	(light orange band, showing the Sub-Project name)	
▶ Owner: Jane	(group band)	
task, task...		
▶ Owner: Sam	(group band)	
task...		

► Owner: Not Tagged (muted band)
task...
(if the project has no Sub-Projects, the group bands sit directly under L2)

Points worth knowing:

- **The whole data set is grouped, not one project.** Every Portfolio, and every Project under it, is drawn with its own bands. Filter to a single Project on the L2 picklist if you want to look at just one
- **Two levels of grouping.** Set the first picklist to group, set the second to sub-divide inside each of those bands. Leave either on *No grouping* to use fewer levels
- **Nothing is ever hidden.** Tasks with no value for the grouping field are collected under a **Not Tagged** band rather than dropped. Tasks whose parent was removed by a filter are still shown at the end
- **Bands are labels, not bars.** A band has no dates, so the right-hand pane draws a band across the row instead of a bar
- **Allocations (L5) are not grouped** — grouping works on Tasks, Milestones and Gates
- Inside a band, tasks are ordered by planned start until you click a column header

The dimensions on offer are Sched Status, Owner, WBS, Priority, Phase, Category, Department, Product, Weighting and Risk/Issue/CR. The list is a single configurable set — adding a dimension to it makes it appear in **both** grouping picklists at once, so the report can be extended without redesigning the panel.

Blank group bands? Grouping fields are usually tagged at Project level and left empty on the tasks below, which would put everything into *Not Tagged*. **Recalculate All** fixes this — see *Grouping Fields Are Filled Down the Hierarchy* below.

Resource Reports

Resource Usage

- Comprehensive resource demand visualization
- Multiple rollup options available
- Stacked bar charts showing allocation over time
- 32-week visibility window
- Identify over/under allocation patterns
- Forecast hiring needs
- Balance workloads across teams

Resource Usage Stacked Charts By:

- **Resource Name:** Individual allocation patterns
- **Primary Role:** Developer, Architect, PM, Tester, etc.
- **Primary Skill:** Technical skills distribution
- **Location/Office:** Geographic resource distribution
- **Department:** Organizational unit allocation
- **Project:** Project-based resource demand
- **Initials:** Condensed view for large teams

Resource Usage Detail Table

- Tabular view with 32-week visibility

- Weekly hours breakdown
- Resource metadata (location, role, skill, department)
- Sortable and exportable
- Supports capacity planning analysis

Shared Resource Pool

- Complete resource pool management view
- All resources in the tbResources store
- Shows all resources added by Administrator
- Typically managed via Spreadsheet integration
- Includes:
 - Skills, roles, and competencies
 - Department, location, organizational structure
 - Pay rates, cost codes, financial attributes
 - Availability calendars and capacity
 - Manager, team leader, reporting relationships
- Click any resource to open full resource form for updates
- Skills gap analysis support
- Organization chart maintenance

Which Generic resources count as supply

Generic resources are role placeholders — "Senior Developer", "Tester", "Project Manager" — and not all of them represent capacity you can plan work against. The rule the application uses everywhere is simple:

A Generic resource counts as supply only when its **Quantity is greater than zero**.

That one number is the switch, and it controls three things at once:

Where	What Quantity > 0 does
Supply and Demand grids	Only these Generic roles form the supply target. Capacity is Quantity × calendar hours × working days
AI Resource Plan	Only these roles are offered to the AI, so only these can be added as Allocations
Resource charts and rollups	Only these contribute to organisational capacity totals

This makes the resource pool manager the gatekeeper. Roles that exist for reporting or org structure rather than delivery — governance boards, steering committees, management overhead — should be set to **Quantity 0**. They stay in the pool, stay reportable, and stop turning up in plans and capacity totals.

The practical consequence: if an AI-generated resource plan keeps proposing management roles you did not want, do not fight the prompt — set those Generic roles to Quantity 0 in the Shared Resource Pool and generate again. Equally, a delivery role sitting at Quantity 0 is invisible to planning, which is the usual reason a role you expected never appears.

See the *How to Use Ask AI* guide for the resource plan itself, and the *Supply and Demand Grids User Guide* for capacity analysis.

3. Tagging Rows Into Reports — the Show In Field

Several report sections are not built from a query at all. They show **only the rows you have tagged for them**, using the **Show In** field on the metadata form (picklist: *Show In Reports*). This is deliberate: an executive-facing section should show the five risks that matter, not all forty.

A row can carry more than one Show In value, so one Milestone can appear in a Project Charter and on a portfolio pack at the same time.

Which tag feeds which section

Report and section	Shows	Tag its Show In field with
Project Charter → 5. High-Level Risks	Risks belonging to that project	Key Risks
Project Charter → 6. Summary Milestone Schedule	Tasks and Milestones	Key Deliverables
Project Charter → Resource Requirements	Allocations	Project Charter
Portfolio Status and Balancing → Notable Risks by Project	Risks	Key Risks
Portfolio Status and Balancing → Key Deliverables	Tasks and Milestones	Key Deliverables

The Charter's risk and deliverable sections use the **same two tags** as the Portfolio Status and Balancing Report. Tag a risk **Key Risks** once and it appears in both — you do not tag the same row twice for two reports. Only the Charter's **Resource Requirements** section still uses the **Project Charter** tag, because allocations are charter-specific.

Hover the heading of any of these sections in the app and the tool-tip tells you which tag it reads.

If a section is empty

An empty section is almost always an untagged one, and each says so in place of its content — for example *"No Key Risks for this project. Tag a Risk's Show In field with 'Key Risks' to display it here."* To fix it:

1. Open the Risk, Task, Milestone or Allocation
2. Click its **Show In** field
3. Pick the value from the list

The change saves on the tap, and the report shows it on the next open.

Bulk tagging by spreadsheet

Tagging a whole portfolio one row at a time is slow. Export to the spreadsheet, fill the **Show In** column down, and sync back. Make sure your Tags sheet carries these three values in the **Show In Reports** group, or the picklist will not offer them:

- **Project Charter**
- **Key Deliverables**
- **Key Risks**

See the *Forms, Reports and Graphs Guide* for adding picklist values, and the *Data Synchronization, Backup, Recovery and Retention Guide* for the spreadsheet round trip.

4. Special Reports

Risk, Issue & Change Management Reports

Risks and Issues Report (Card View)

Risk Cards (Blue):

- Probability × Impact scoring matrix
- Automated risk score calculation
- Mitigation status and plans
- Escalation tracking
- Assign owners and track mitigation activities

Issue Cards (Red):

- Priority and severity classification
- Assignment and ownership
- Resolution tracking
- Impact assessment

Change Request Cards (Gradient Blue):

- Requestor and department
- Impact assessment (cost, hours, schedule)
- Benefits and justification
- Options analysis
- Version tracking

All cards support:

- Rich text editor for detailed descriptions
- Expandable sections for plans and strategies
- Filter by status, priority, owner
- Search across all text fields
- Export for audit trails

Risks and Issues Report (Tabular View)

- Same data as card view in table format
- Sortable columns
- Multi-row selection
- Bulk export capabilities

Timeline Visualizations

Horizontal Timeline Report

- Interactive timeline with draggable event cards
- Events positioned along horizontal date axis

- Persistent positioning (saved in browser)
- Filter and search by name, type, status
- Connector lines from timeline to event details
- Reset positions for fresh layouts

Vertical Timeline Report

- Similar to horizontal but oriented vertically
- Project chronologies and milestone sequences flow downward
- Milestone planning visualization
- Dependency relationships

Configuration Reports (Other menu)

Five reports that show the tables the application is configured from, rather than your project data. They are covered fully in the *Forms, Reports and Graphs Guide*; in outline:

Picklist Values

- Every value behind every picklist — 97 lists in the shipped data
- Filter to one picklist, then open a value on its form to edit it
- Add and delete values; locked values carry a padlock and cannot be renamed
- A picklist can never be emptied — the last value is renamed rather than deleted

Fields Values

- The fields placed on the dynamic FOCD forms, one row per field per form
- Add a field to a form, remove one, change its label, type and picklist
- Label, Top, Left and Width are edited directly in the grid — the only configuration columns that are

Schema Values

- The master field list every other configuration table joins to
- Read and update only — rows cannot be added or deleted, because a row here corresponds to a database index and a spreadsheet column

CoreReport Values

- The Core Form layout — which field sits in which section, at which level
- Read and update only

Config Integrity

- Checks the four configuration tables against each other and reports what does not resolve: orphaned field names, duplicate placements, picklists that are empty or referenced by nothing
- It never blocks and never repairs — it measures
- **Run it after every spreadsheet import.** Broken configuration references do not throw errors; a picker whose list has gone simply opens empty

5. Common Report Features

All reports in the system share these capabilities:

Search & Filter

- Free text search across names and descriptions
- Filter by hierarchy level (L1-L5)
- Filter by type (Portfolio, Project, Sub-Project, Task, Milestone, Allocation)
- Filter by status (Active, On Hold, Cancelled, Complete)
- Filter by priority (High, Medium, Low)
- Filter by owner (Resource names)
- Multiple filters can be combined

Data Export

- **Excel:** Full data export with formatting
- **CSV:** Raw data for external analysis
- **Print:** Print-ready layouts with proper page breaks
- **Copy:** Copy to clipboard for quick sharing

Visual Indicators

- **Color Coding:** By type, status, priority, health
- **Health Indicators:** Red/Yellow/Green status
- **Icons:** Visual markers for risks, issues, milestones
- **Progress Bars:** Completion percentage visualization

Direct Editing

Not every grid is editable, and the difference is deliberate.

Report	Editing
Schedule and metadata reports — General Tabular View, Task Tabular, Gantt Report, Detail Cards	inline — click a cell and type; picklist cells open a picker
Shared Resource Pool	inline, plus a full edit form
Picklist Values, Fields Values, Schema Values, CoreReport Values	read-only in the grid; the grid finds the row, the form edits it

The configuration reports work that way because every one of their columns carries a rule — a name that must exist elsewhere, a list that must not be emptied — and enforcing those rules in two places would let the two drift apart. The exception is the four layout columns on Fields Values, which are plain numbers and edit in the grid.

Edits save as you go. There is no separate Save step, and no need to switch screens.

Editing Predecessors in a Grid

Predecessor is editable in the grids, but not on every row, and the list you are offered depends on what you clicked:

Row type	Predecessor cell
Task	Editable. The list holds the other Tasks in the same project — a task cannot depend on work in a different project
Project	Editable. The list holds all Projects , with no hierarchy restriction
Portfolio, Sub-Project, Milestone, Gate, Allocation	Read-only, like the Type column

The picker shows each candidate's **ID and name**, with long names trimmed to fit — hover a row to read the full name. Picking a row stores the **ID** as the predecessor while showing you the name, so you never have to look an ID up.

The current row is always left out of its own list, so a bar can never be made its own predecessor.

Unlike the other picklists, this one is **live**: it is built from your actual bars at the moment you click, not from a stored list of values. Create a task and it is immediately available as a predecessor.

Grouping Fields Are Filled Down the Hierarchy

Most people tag a project once — one Department, one Product, one PM — and leave the tasks below it blank. Grouping and filtering would then be nearly useless: every task would land in the *Not Tagged* band.

Recalculate All solves this. As part of the recalculation, empty grouping fields inherit the value of their **nearest tagged ancestor**, all the way down the hierarchy: Portfolio → Project → Sub-Project → Task → Allocation. Tag a project as *Finance* and every task beneath it groups and filters as *Finance* without you touching one of them.

The fields filled down are:

Field	Set at
Owner	any level
Department	any level
Product	any level
Project Number	any level
Project Manager	any level

Three rules make this safe to run as often as you like:

- 1. **A value you typed is never overwritten.** Only empty fields are filled. A task tagged to a different department than its project keeps its own value
- 2. **Blank means blank in every form** — an empty cell, spaces, or a stray "null" left behind by a spreadsheet import are all treated as empty and get filled
- 3. **Gaps in the hierarchy are skipped.** A task sitting directly under a project with no sub-project between them inherits from that project

Run it from **Right click on Canvas > Recalculate All**, or any *Recalculate All* button, including the one on the Reports page. Do it after a spreadsheet import and before you rely on grouped or filtered output.

Not every field that you can group by is filled down. A field is filled down only where inheriting the parent's value is genuinely meaningful — Status and Priority, for instance, belong to the individual task, so they are left alone and blanks show as *Not Tagged*.

Report Persistence

- Last viewed report is remembered
- Filter settings saved per report
- Return to your exact view next session

6. Where Report Data Comes From

Reports read the browser database. Two stores do most of the work:

- **tbMdJoined** — the cached join of bars and metadata that nearly every tabular and card report reads. It is rebuilt by **Recalculate All**, which is why a report showing stale numbers is usually a report that needs a recalculation, not a bug.
- **tbResCals2** — the weekly resource demand figures behind the resource charts and grids. Rebuilt separately from Recalculate All.

Baseline variance reads **tbBaseline**, and the resource reports read **tbResources**. The stores themselves are documented in the *Data Synchronization, Backup, Recovery and Retention Guide*.

7. Publishing to the Cloud Dashboard

One-Click Publishing: All reports can be published to the Timebars Cloud Dashboard:

- No configuration required
- Customize graph types and filter data sets
- Share with stakeholders via secure links
- Access from any device with internet connection
- Real-time updates when you publish new data

Learn more in the [Cloud Publishing Guide](#).

8. Tips

1. **Recalculate before grouping.** Run Recalculate All after any spreadsheet import, so grouping fields are filled down before you group or filter on them
2. **Tag once, report twice.** **Key Risks** and **Key Deliverables** feed both the Project Charter and the Portfolio Status and Balancing Report — tag a row once and it appears in both
3. **Keep Generic quantities honest.** Quantity 0 on management-only roles keeps them out of resource plans and capacity totals
4. **Check the level tick boxes first** when a report looks empty. L4 off means no tasks
5. **Run Config Integrity after every import**, not only when something looks wrong

9. Troubleshooting

Report is Empty or Shows No Data:

- Check that you have data loaded in the application

- Verify filters are not excluding all results
- Click "Reset Filters" to clear all filter settings
- Check the **L3 / L4 / L5** tick boxes — if L4 is off, no tasks are shown
- Check the **Show In** picklist is not narrowed to a tag nothing carries

A Report Section Says "No Key Risks" / "No Key Deliverables":

- The section is tag-driven, not a query — nothing is tagged for it yet
- Open the row, click its **Show In** field, and pick the value the message names. See *Tagging Rows Into Reports* above

Everything Lands in the "Not Tagged" Band:

- The grouping field is set on the project but empty on the tasks below
- Run **Recalculate All** to fill grouping fields down the hierarchy, then re-open the report. See *Grouping Fields Are Filled Down the Hierarchy*

The Overdue / Milestone Horizon Tick Box Keeps Switching Off:

- That is by design. They are overrides — ticking one clears the other, and touching any L3/L4/L5 tick box clears both and returns the normal view

A Generic Role Never Appears in a Resource Plan or Supply Grid:

- Its **Quantity** is 0. Only Generic resources with Quantity greater than zero count as supply — see *Which Generic resources count as supply* above

A cell will not accept an edit:

- Not every column is editable, and the difference is deliberate — see *Direct Editing*
- On the configuration reports the grid is read-only by design; the form does the editing
- System fields such as Type and Hierarchy Order are read-only everywhere

Part B — Configuring the Product Through the Editable Reports

Four reports on **Report Menu** → **Other** are not reports about your projects at all: they are where the product itself is configured — what the dropdowns say, which fields sit on which form, and how the Core Form is laid out. They are called the **Editable Reports**, and a fifth report checks that they still agree with each other.

Everything in this part used to live in a separate Configurable Data guide. It is here because the reports *are* the configuration interface — there is nothing to describe separately from them.

10. The Two Rules That Decide Everything

Almost every question in this guide — can I change this? where do I change it? — is answered by two rules.

Rule 1: the app edits what is purely data; it does not edit what is also code

An internal field name such as **tbMDStatus** is not just a label. The same string is a column heading in the workbook, an index in the browser database, and a literal in the application source. The app cannot rewrite the source, so it will not let you rewrite the name.

That single idea decides every "can I edit this?" in the product:

Where it changes	
The values in a picklist	in the app, or the spreadsheet
Which picklist a field opens	in the app, or the spreadsheet
A field's label , position, width	in the app, or the spreadsheet
Which fields sit on a form	in the app, or the spreadsheet
An internal field name	the spreadsheet, alongside a code change
A primary key	never

Rule 2: the grid finds the row, the form edits it

On all four Editable Reports the **grid is read-only**. You filter and sort to find the row you want, then open it on its **form** to change it.

There is one deliberate exception: four layout columns on Fields Values — **Label, Top, Left, Width** — stay editable in the grid, because positioning a field means comparing its numbers against its neighbours', and that is exactly what a grid is good at.

A companion rule follows from Rule 1: **names that reach code are picked from a list, never typed**. Every cross-table reference is a picklist sourced from the table that owns the name. The one place free typing is allowed is a **new picklist name** on the Picklist Values form — because a new picklist has to start somewhere.

11. Where Configuration Lives

The four Editable Reports, and the validator

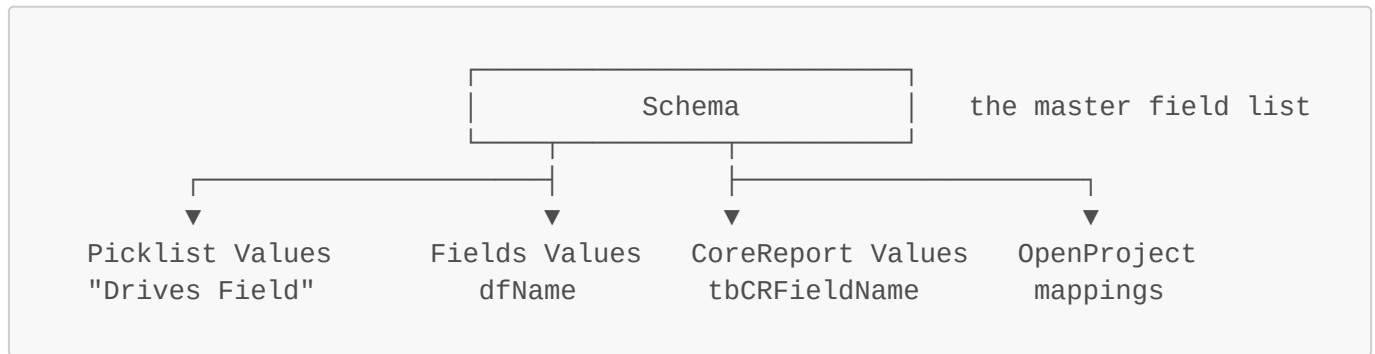
All five are on **Report Menu → Other**. The first four are the **Editable Reports** — the places you change how the product is configured. The fifth checks their work.

Report	What a row is	You can
Picklist Values	one value inside one picklist	edit, add, delete
Fields Values	one field placed on one FOCD form	edit, add, delete
Schema Values	one field that exists in the system	edit only
CoreReport Values	one field placed on one Core Form section	edit only
Config Integrity	(a validation report)	run it

Schema Values and CoreReport Values have **no Add and no Delete**. A Schema row corresponds to a database index and a spreadsheet column, neither of which the app can create; a CoreReport row places a field that must already exist.

Schema is the spine

The four configuration tables are not independent lists. **Schema is the master field list, and everything else names a field in it.**



A name that does not resolve does not throw an error. A write addressed to a row that is not there is a silent no-op, and a picker whose picklist has gone simply opens with no rows in it. Nothing turns red. That is why the fifth report exists.

Config Integrity — run it after every spreadsheet import

An import replaces whole tables. Rename a field on one sheet and not another, delete a row three tables still point at, empty a picklist — none of it announces itself. **Config Integrity** measures it: it checks that every field name resolves, that placements are unique, that **dfType** and **dfPicklist** agree, that no picklist is down to its last value or referenced by nothing.

It never blocks and never repairs. Bad data must not stop the tool opening, and a fix belongs where the value is owned — which is usually the spreadsheet.

Make it a step in your import routine, immediately after the import.

Every write is stamped and trimmed

Two housekeeping behaviours worth knowing:

- Every change made in the app stamps that store's **Last Modified** column with the date, in **DD-MMM-YYYY**.
- Values are **trimmed on write**, and every comparison trims both sides. A trailing space typed into a name cannot create an invisible mismatch.

12. Picklists and Tagging

The two kinds of picklist

This is the single most important thing in this section, and the most common source of wasted effort.

1. A tag-backed picklist. The values are rows in the Tags table sharing one picklist name. You maintain them — add, edit, remove, reorder. Most picklists in the product are this kind.

2. A live people-picker. The field names a *person*, so its list is not stored anywhere — it is read live from the **Resource Pool** (**Human** rows in Resources) every time the picker opens. There are no tag values to maintain, and adding some would achieve nothing.

If a field names a person, maintain the Resource Pool, not the Tags table.

The fifteen people-picker fields:

Field	Shown as
tbOwner	Owner
tbMDPM	Project Manager
tbMDExSponsor	Executive Sponsor
tbMDBusinessAdvisor	Business Advisor
tbMDDeliveryManager	Delivery Manager
tbMDResponsibility	Responsible Person
tbMDBusinessOwner	Business Owner
tbMDPrimaryContact	Primary Contact
tbMDContact	CR Requestor
tbMDOrgManager	Org Manager
tbMDStageApprover	Stage Approver
tbMDEscalationOwner	Escalation Owner
tbMDCharterApprovedBy	Charter Approved By
tbMDProductOwner	Product Owner
tbMDScrumMaster	Scrum Master

This list is growing. Fields move from tag-backed lists to the live pool over time, because a person's name kept in two places drifts. **tbMDResponsibleTeam** is deliberately *not* here — it names a team, and stays a tag-backed picklist.

Changing what a picklist offers

Report Menu → Other → Picklist Values.

1. Filter the **Picklist** column to the list you want.
2. Click the **edit icon** on a row to open it on the form. Change the name, short name, sort order, purpose or owner. It saves as you go — there is no separate Save step.
3. **Add** puts a new value into whichever picklist is currently filtered — so filter first; the button asks you to if you have not. Then open the new row on the form to name it.
4. **Delete** removes the value from the list. Any record already holding that value keeps it; the value simply stops being offered from now on.

Remember Rule 2: the grid itself is read-only here. It is how you *find* the value; the form is where you change it.

The same values live on the **Tags** worksheet, and a spreadsheet import replaces them wholesale. Both routes are valid. Use the app for a handful of values, the workbook for a wholesale revision.

Giving a field a picklist

A field opens a picklist because a Tags row points at it — not because of anything in the code.

Set a value's **Drives Field** to the field you want, and that value's **Picklist** becomes that field's list.

Two constraints:

- **A picklist name must not contain a hyphen.** Forms encode the name as the last segment of a hyphen-separated key, so a hyphen splits it in the wrong place and the picker opens empty. This has bitten before: **Part-time Full-time** had to become **Parttime Fulltime**.
- **A picklist cannot be empty.** See below.

The never-empty rule

A picklist exists only as the set of rows carrying its name. Delete the last one and the picklist itself is gone — leaving a field whose picker opens with nothing in it and no way to put anything back.

So the app refuses. Deleting the **last** value in a picklist renames it to **Not Assessed** instead. The picklist survives with one neutral value in it, and the value you wanted rid of stops being offered. If you would rather remove it outright, add a replacement first.

Locked values and the padlock

Some values are read by the application itself. A schedule status of **Late** is not just a word on a bar — it selects a colour. A budget band of **301K to 1000K** is what a portfolio score is calculated from. Rename one and nothing warns you: the comparison simply stops matching and the result quietly becomes wrong.

Those values carry a **padlock** in Picklist Values. On a locked row:

Locked	the value's name, and its picklist name
Still yours	sort order, short name, purpose, owner
Delete icon	replaced by the padlock

The picklist name is locked as well as the value, because renaming a picklist detaches every value in it from its field — that breaks a whole list, not one entry.

You may still add your own values to any locked picklist. Only the specific values the code names are fixed. Section 4 lists them, and they are marked in bold in the A–Z reference.

13. Dynamic Forms (FOCD)

FOCD — Forms on Cached Data — lets you add and remove fields on the product's built-in forms without writing code. The number of forms is fixed; what goes on them is yours.

The seven forms

Form name	The form it draws
PF1	L1 Portfolio flyout

Form name	The form it draws
PJ1	L2 Project flyout
SPJ1	L3 Sub-Project flyout
TASK1	L4 Task flyout
ALLOC1	L5 Allocation flyout
dfRisk	Risk form
dfIssue	Issue form

What a Fields row holds

Report Menu → **Other** → **Fields Values**. One row is one field on one form.

Column	What it does
Field	the field this row places — picked from Schema , never typed
Form	which of the seven forms it appears on
Type	how it renders — see below
Label	the caption shown to the user on this form ; free text
Picklist	which list the field opens, when Type is WithPicklist
Tooltip	the hover text on the field
Top / Left / Width	position and size, in pixels
Show	whether it is drawn at all
Other 1	which table the value is saved to — tbTimebars or tbMetaData
Other 2	disabled makes the field read-only on the form; blank leaves it editable

Other 1 and Other 2 are load-bearing despite their names. Other 1 decides where the field's value is stored and Other 2 whether the user can type into it. Other 3 and Other 5 are unused and are left blank on a new row.

The five field types:

Type	Renders as
NoPicklist	a plain input
WithPicklist	an input that opens a picklist
TextArea	a multi-line box
Editor	a rich text editor
Band	a section heading bar, not a field

The label is deliberately per-form

A field's **official label** lives once, in Schema, and is what the field is called everywhere by default. The **Label on a Fields row overrides it for that one form** — and that is not an inconsistency to be tidied away, it is the point.

The person configuring a form decides what the field should be called *in that context*, in whatever language the team using the form actually speaks. The same underlying field can read differently on each form it appears on:

Form	Field	Official label (Schema)	Label on this form
ALLOC1	<code>tbName</code>	Name	Allocation Name
ALLOC1	<code>tbPercentTimeOn</code>	Percent Time Allocated	% Allocated
ALLOC1	<code>tbCalendar</code>	Calendar	Availability/Calendar
dfIssue	<code>tbStart</code>	Start	Created On
dfIssue	<code>tbMDHealth</code>	Schedule Status	Timebar Health

Roughly forty of the shipped Fields rows carry a label that differs from the official one for exactly this reason. Leave it blank and the official label is used; fill it in whenever the form's context calls for different wording. It is a decision for the administrator and the users of that form to make together, and nothing else in the product is affected by it.

The Core Form works the other way. Its captions are *derived* from the official Schema label and cannot be overridden per placement — see [The Core Form](#). The per-form override is a FOCD feature only.

WithPicklist is one word. The old guidance said "With Picklist" with a space; a field configured that way does not open a picklist. You no longer have to remember this — the type is picked from a list.

Adding, changing and removing fields

- **Add** is scoped to the form you are filtered to, because "which form" cannot be guessed. The new row lands below everything already on that form and inherits its neighbours' left, width and target store. **Name is left empty** — an invented field name is exactly the broken reference these rules exist to prevent — and the form opens straight onto it for you to pick one.
- **Type and Picklist move together.** Choosing a picklist sets the type to `WithPicklist`; choosing any other type clears the picklist. They cannot disagree.
- **Delete** warns you in the two cases whose consequence is not visible in the row: deleting a **Band** row removes the form's heading, and deleting the **last** row leaves the form empty.

Positioning fields

Two ways, and they suit different jobs.

In the grid — Label, Coord Top, Coord Left and Width are inline-editable, the only four columns on any configuration report that are. Use this when laying out a whole form, where the point is comparing one field's numbers against its neighbours'.

On the form itself — the blue icons at the top of a FOCD form move fields directly. The second icon moves one field at a time, the third moves several at once; dotted outlines show what is movable. Drag, then click the blue square to save. Use this for fine-tuning.

Rich text

Rich text fields use **Tiptap**. (Earlier documentation referred to Quill, which was replaced.) Whether a field is a rich text editor is set by its type here, and for the Core Form by a list in the application — see [The Core Form](#).

14. The Core Form

The Core Form is the large collapsible form that shows a bar's full metadata, grouped under business headings — Essential Facts, Progress & Health, Charter, Investment Alignment and so on. It is the form a PM spends most time in.

It is configured from a different table than the FOCD forms. FOCD forms are built from the **Fields** table (section 5); the Core Form is built from the **CoreReport** table. They share nothing but the Schema field list underneath them, so a change to one never affects the other.

	FOCD forms	The Core Form
Configured from	tbFields — <i>Fields Values</i> report	tbCoreReport — <i>CoreReport Values</i> report
A row is	one field on one of the seven FOCD forms	one field in one Core Form section
Add and delete rows in the app	yes	no — spreadsheet only
Field caption	free text, per form (section 5)	derived from the field's Schema label
Positioning	pixel coordinates you set	column widths inside a four-column section
Sections	one Band row per heading	a fixed set of named sections

Everything below is about the Core Form and the CoreReport table.

There are two ways to change the Core Form: **in the application**, one row at a time, and **in the spreadsheet**, in bulk. They edit exactly the same data — the CoreReport table — so use whichever suits the job.

In the application

Report Menu > Other > CoreReport Values. One row is one field placed on one section of the form. Filter to the section or field you want, then click the edit icon to open the row on its form.

To change	Edit
Which section a field is in	Section Name — pick from the list of sections the form actually draws
Order within a section	ID — fields are drawn in ascending ID within their section
Field width	Columns — 1, 2 or 3
Shown at a level	the five level flags — Portfolio through Allocation

Two things you will notice are read-only, and hovering them explains why:

- **Label** is **derived**. It always holds the field's own label from the Schema table, and changing it there updates every placement at once. Change a caption in **Schema Values**, not here.
- **Section Key** is written for you when you choose a Section Name. It has to match a section the form actually draws, so it is never typed.

Rows cannot be **added or deleted** in this report — a CoreReport row places a field that must already exist, and adding placements is a spreadsheet job.

tbCRForm is now vestigial. Every row is v2; version 1 of the Core Form is dead at runtime.

In the spreadsheet

Almost everything about how the Core Form looks is held in the **CoreReport sheet**, not in the code, and takes effect the next time you import the sheet and reopen the form. Each row is one field on the form.

tbCRSection says which section it appears in, so moving a field from Essential Facts to Other Metadata is a matter of changing that one cell — nothing in the code names a field's section. tbCRID decides the running order: fields are drawn in ascending tbCRID within their section, so to move a field you give it a number that falls where you want it to sit. tbCRNoOfColumns is how many columns wide the field is (1 is a normal field, 2 is half a four-column row, 4 is the full row), and it applies to rich text editors exactly as it does to plain fields — an editor set to 2 sits beside its neighbour instead of taking a row of its own. A field behaves the same wherever you put it: width is the field's own, from the sheet, while how many columns a section is divided into belongs to the section, so moving a field into a different section does not drag a column count in with it. Every section is four columns wide, and a width larger than that is capped at the full row, so a stray value in the sheet costs you a wide field rather than a broken one. tbCRLabel is the caption shown above the field — but it is **derived from the field's label in Schema** and rewritten whenever that changes, so set the caption there rather than here. tbCRSectionName is the heading printed above the section. The five tbCRPortfolio ... tbCRAllocation columns decide, per hierarchy level, whether the field appears at all: show means show, anything else means hide. Set tbCRForm to v2 on every row, or the form will not see it. (Every row is v2 now — v1 is dead at runtime — so this column is vestigial, but a blank still hides the row.)

A few things are not in the sheet, because it has no column for them. Whether a field is a **rich text editor** comes from RICH_TEXT_FORM_FIELDS in scripts/focd/dfmain.js — that one list also drives how the field renders in the tabular reports and the Gantt grid, so adding a field there changes it everywhere at once. Whether a field is a **date picker** comes from its Type being Date on the master field list, and whether it is **read-only** or forced onto a **new row** comes from scripts/reports/coreFormV2Fields.js. Which **picklist** a field opens is data of a different kind: add or point a tbTags row's tbTagTbInternalName at the field and its tbTagGroup becomes that field's list, with no code change — except for people-pickers such as Owner or Sponsor, which always read the live resource pool and are listed in PEOPLE_PICKER_FIELDS in scripts/reports/editableCells.js. Finally, the Investment Alignment and Benefits & Financial Value sections are hidden outside Costbars regardless of what the sheet says; that rule is HIDDEN_BY_PRODUCT in scripts/reports/coreFormV2.js, and when it fires it says so in the browser console.

Where each thing lives

To change...	Edit	Notes
Which section a field is in	tbCRSection	Must match a section id in CFV2_SECTIONS

To change...	Edit	Notes
Order within a section	<code>tbCRID</code>	Ascending; see the warning below
Field width	<code>tbCRNoOfColumns</code>	1–5; editors included
Field caption	<code>tbSchLabel</code> in Schema	<code>tbCRLabel</code> follows it automatically; it is read-only in the app
Section heading	<code>tbCRSectionName</code>	Taken from the section's first row that has one
Shown at a level	<code>tbCRPortfolio ...</code> <code>tbCRAllocation</code>	<code>show</code> , or anything else to hide
Which picklist a field opens	<code>tbTags</code> row	<code>tbTagTbInternalName</code> → <code>tbTagGroup</code>
Rich text editor on/off	<code>RICH_TEXT_FORM_FIELDS</code> , <code>dfmain.js</code>	Also changes the grids
Date picker	Type <code>Date</code> on the master field list	Uses the app's <code>DD-MMM-YYYY</code> format
Read-only, forced new row	<code>coreFormV2Fields.js</code>	Generated file; see its header
Columns a section is divided into	<code>columns</code> on the section in <code>CFV2_SECTIONS</code>	Every section is four across; set this only to break that
Sections and their order	<code>CFV2_SECTIONS</code> , <code>coreFormV2Fields.js</code>	
Costbars-only sections	<code>HIDDEN_BY_PRODUCT</code> , <code>coreFormV2.js</code>	Overrides the sheet

Two things that will bite you

`tbCRID` is unique. The store enforces it, and an import that leaves two rows sharing a number is rejected part-way through. The safe way to reorder a section is to reuse the numbers that section already has, dealt out in the new order — then nothing outside the section moves and no number can clash with another section.

A field must be known in two other places or it will look broken rather than missing. If it has no entry in `CFV2_FIELDS` the form skips it and logs a warning to the console. If it is absent from the join's field list in `scripts/tbdatabaseasync.js` it draws as an empty box that accepts what you type and then shows blank again when you reopen the form — the field saved correctly, but the form cannot read it back. If a field you added to the sheet behaves that way, that list is the first place to look.

15. What Is Configurable, by Product

A summary of what can be tailored. All three products share one build, so anything marked as common applies everywhere.

Common to all three

Scheduling and calendar

- Report Date — the "as-of" date every progress calculation is measured against
- Timescale start and finish, weekly vs. monthly view, zoom level
- Working calendar exceptions and holidays (AdminPanel sheet, rows 6–100)

Canvas display

- Dark / light theme; bar row spacing; hierarchy level filter (L1–L5)
- Show or hide: completed bars, baseline bars, ghost bars, dependency lines, hierarchy connector lines
- Show or hide Risk, Issue and Change Request bars

Metadata and tagging

- The full metadata field set, and the picklist behind any dropdown (section 3)
- Rich-text business case sections
- Custom sort order per item

Forms

- Which fields appear on which FOCD form, their labels, tooltips, picklists, position, width and read-only state (section 5)
- The Core Form's sections, field order, widths and per-level visibility (section 6)

Status and health

- All automation thresholds, via `apMetadataRules` (section 7)

Data

- Full JSON backup and restore; per-store export; baseline snapshots

Licence *(display only, not user-editable after install)*

- Licensed organisation, expiry date, bar and project limits

Agilebars

- Five-lane Kanban — Backlog, Will Do, Doing, Finalizing, Done — with automatic percent-complete mapping per lane
- Card content as text or as a bar chart
- Progress measured against the Report Date or against today (`apMPO`)
- Fixed-sizing mode, where hours never change when a bar is dragged
- Sprint name per task; multiple boards; burndown chart
- Hierarchy is L2 and L4 only — L1, L3 and L5 are hidden by product

Timebars

- Full five-level hierarchy, drag-to-schedule with working-day calculations
- Resource pool: role, skill, department, manager, location, team leader
- Work calendar and general availability percentage per resource

- Resources per line, and pixels-per-person display scaling
- Weekly and monthly demand views by name, role, skill, location and department
- Baseline snapshots and overlay

Costbars

- PPM tabular, card and scorecard views; demand grids
- Prioritisation and scoring configuration
- Investment category, objective, strategy and initiative classification
- ROM estimate, estimation class, size and weighting
- Optional financial scoring: NPV, IRR, payback period, economic value added, benefit-cost ratio
- Risk vs. Size and Complexity assessment

Appendix A — Picklist Reference A–Z

97 picklists, 608 values. Every picklist in the product, with the field it drives, its purpose, and its values in sort order.

Values in bold are locked - the code compares against that exact text (144 of them). See *Values the Code Depends On* in the *Data Model and Scheduling Engine Guide*.

This reference is **generated from the live data** by `tools/buildPicklistReference.py`. Run that rather than editing entries here; hand-editing is how the previous version came to list twelve picklists that no longer existed.

Picklists named after a person are absent by design - those fields read the Resource Pool live and have no values to list. See *The two kinds of picklist* in section 12.

Approval State - `tbMDState`

Workflow Life-cycle tag

Values: **Spawned**, **Awaiting Input**, **Requested Approval**, **Awaiting Approval**, **Approved**, **Awaiting Review**

Every value in this picklist is locked. Add your own freely; do not rename these.

Benefit Cost Ratio - `tbMDBenefitCostRatio`

Financial scoring value used at the project level.

Values: > **5:1**, **3:1** - **5:1**, **2:1** - **3:1**, **1.5:1** - **2:1**, **1:1** - **1.5:1**, **Not Assessed**

Every value in this picklist is locked. Add your own freely; do not rename these.

Benefit Owner - `tbMDBenefitOwner`

Values for the `tbMDBenefitOwner` field

Values: HR, Finance, Engineering, Shareholders

Benefit Type - `tbMDBenefitType`

Values for the `tbMDBenefitType` field

Values: Financial, Cost Avoidance, Non-Financial

Billing Type - `tbMDBillingType`

Mainly Used in CR's and Tasks

Values: Billable, Non Billable, Not Assessed

Budget Estimate - `tbMDROMEstimate`

Fixed ranges for sizing a project budget

Values: **0 to 100K, 101K to 300K, 301K to 1000K, 1001K to 5000K, 5000K Plus**

Every value in this picklist is locked. Add your own freely; do not rename these.

Budget Fiscal Year - `tbMDBudgetFiscalYear`

Values for the `tbMDBudgetFiscalYear` field

Values: FY2026, FY2027, FY2028

Calendar - `tbResResourceCalendar`

Workday hours, used cor calculating wok on an Allocation

Values: 4, 7.5, 8, 24

Closure Reason - `tbMDClosureReason`

Values for the `tbMDClosureReason` field

Values: **Completed, Cancelled**, Deferred

Cost Center - `tbMDCostCenter`

Values for the `tbMDCostCenter` field

Values: CC-1000 Operations, CC-2000 Technology, CC-3000 Corporate

Cost Code - `tbResCostCode`

Custom Cost Code Metadata for reporting costs, mainly used at Task or Allocation level

Values: A26, B44, C99, D11, E11, E112

Cost Type - `tbCostType`

Cost Type Metadata to separate costs e.g. Software, Equipment . Mainly used at Task or Allocation level

Values: Equipment, Hardware, Software

CR Category - `tbMDCategoryCR`

Metadata to categorize Change Requests. Used at Task level if Sub-Type is CR

Values: Field, Engineerng, Equipment, Product, Defect

Cross Team Collaboration - `tbPASCrossTeamCollaboration`

Helps determine the Project Assessment Score.

Values: Single team, 2-3 teams, 4-6 teams, 7-10 teams, Many teams/departments

Currency - `tbMDCurrencyCode`

Values for the `tbMDCurrencyCode` field

Values: CAD, USD, EUR

Department - `tbMDDepartment`

Task or Allocation Level tagging identifying what department is responsible

Values: Sales and Marketing, Operations, Finance, Procurement, Business Development, Research and Development, Engineering

Domain Experience - `tbPASDomainExperience`

Helps determine the Project Assessment Score.

Values: Extensive experience, Good experience, Some experience, Limited experience, New domain

Escalation Level - `tbMDEscalationLevel`

Metadata to support the Escalation process, mainly for Projects but can be used at any level.

Values: Executives, Directors, **Project Office, Not Assessed**

Estimation Class - `tbMDEstimationClass`

Project Tagging for the quality of the Estimate

Values: **SWAG, Based on History, Resource Driven**

Every value in this picklist is locked. Add your own freely; do not rename these.

Executive Commitment - `tbMDSeniorLevelCommitment`

Project Tagging for degree of management commitment.

Values: **Full, Strong, Moderate, Limited, Not Clear, Not Assessed**

Every value in this picklist is locked. Add your own freely; do not rename these.

External Integrations - `tbPASExternalIntegrations`

Helps determine the Project Assessment Score.

Values: None, 1-2 simple integrations, 3-5 moderate integrations, 5+ complex integrations, Many critical integrations

ExtSystemResID - `tbResExtSystemResID`

Use to tie into or integrate other systems in the organization

Values: 456M23, 456M24

Funding Source - `tbMDFundingSource`

Values for the `tbMDFundingSource` field

Values: Capex, Opex, Client Funded

Gate - `tbMDGate`

Project Level tagging identifying a project gating process

Values: 1 Approved to Design, 2 Approved to Develop, 3 Approved to Implement, Not Applicable

Health Cost - `tbMDHealthCost`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Health Hours - `tbMDHealthHours`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Health Issues - `tbMDHealthIssues`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Health Overall - `tbMDHealthOverall`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Health Risk - `tbMDHealthRisk`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Health Schedule - `tbMDHealthSchedule`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Health Scope - `tbMDHealthScope`

Core Health Indicators mainly used at Portfolio and Project and Sub-Project/Work Pckage Level

Values: Green, Yellow, **Not Assessed**, Red

Internal Rate of Return - `tbMDInternalRateOfReturn`

Tagging financial values at project level.

Values: **> 50%**, **25% - 50%**, **15% - 25%**, **5% - 15%**, **< 5%**, **Not Assessed**

Every value in this picklist is locked. Add your own freely; do not rename these.

Investment Category - `tbMDInvestmentCategory`

Project Portfolio Balancing â€œ Product/System Focus

Values: **Not Assessed**, Business Systems, Our Core Products, Productivity Tools, **Innovation**, Digital Transformation, Infrastructure and Security, Data and Analytics

Investment Initiative - `tbMDInvestmentInitiative`

Project Portfolio Balancing â€œ Capability Building

Values: **Not Assessed**, Strengthen Cybersecurity Infrastructure, Build AI/ML Capabilities, Faster response to new business opportunities, Improve employee skills, Improve product reliability, Improve productivity with Agile Teams, Source Cloud services for our Core products, Support modern digital workplace

Investment Objective - `tbMDInvestmentObjective`

Project Portfolio Balancing â€œ Business Impact

Values: **Not Assessed**, **Grow the Business**, **Run the Business**, Modernize the Business, **Transform the Business**

Investment Strategy - `tbMDInvestmentStrategy`

Project Portfolio Balancing â€œ Specific Business Goals

Values: **Not Assessed**, Attract and retain the best people, Improve our core products, Ensure compliance, Accelerate Time-to-Market, Improve customer service, Build Recurring Revenue Streams, Increase share of market, Reduce waste, Drive Contract Renewals, Server more customers

Issue Category - `tbMDIssueCategory`

Metadata to categorize Issues. Used at Task level if Sub-Type is Issue

Values: Product, Procurement, Construction, **Technical**, Schedule

Labour Type - tbResLabourType

Type of Resource Generic, Human or other

Values: **Generic**, **Human**, Other

Location - tbResLocation

Where is resource located

Values: Edmonton, Halifax, Ottawa

Manager - tbResManager

The Resources Manager

Values: George Bush, John Cougar, Sally Res, Rich Delment, Mitsy Meta, Manny Fact, Marg Ting, Ron Hay, Ron Scope, Al League, Sally Sail, Joe Purch

Market Timing - tbPASMarketTiming

Helps determine the Project Assessment Score.

Values: Perfect timing, Good timing, Acceptable timing, Challenging timing, Poor timing

Mitigation Status - tbMDMitigationStatus

Identify the status if ongoing mitigation for a Risk

Values: Identified, Assessed, Mitigation Planned, **In Progress**, Mitigated, Under Review, Accepted, Transferred, **Escalated**, Closed, Deferred

Net Present Value - tbMDNetPresentValue

Tagging financial values at project level.

Values: > \$5M, \$1M - \$5M, \$250K - \$1M, \$50K - \$250K, < \$50K, **Not Assessed**

Every value in this picklist is locked. Add your own freely; do not rename these.

Parttime Fulltime - tbResPartTimeFullTime

Is the Resource Full time employee or part time

Values: **Full Time**, **Part Time**

Every value in this picklist is locked. Add your own freely; do not rename these.

Payback Period - tbMDPaybackPeriod

Set financial values at the Project Level

Values: < 1 year, 1 - 2 years, 2 - 3 years, 3 - 5 years, > 5 years, Not Assessed

Every value in this picklist is locked. Add your own freely; do not rename these.

Percent Available - **tbResPercentGeneralAvailability**

Default Percent Available for the allocation.

Values: 10, 25, 50, 75, 100

PercentTimeAllocated - **tbPercentTimeOn**

Can be used to override the percent allocated on the Allocation

Values: 10, 25, 50, 75, 100

Phase - **tbMDPhase**

High-level Phase of Project or Program

Values: **Closing**, Executing, **Initiating**, **Planning**

Popular - **tbTagPopular**

Groups tags so commonly used ones surface first

Values: General, NewBars, NewBarsNo, OpenProject Integration, PPM, Project Management, Resource Management, Risks and Issues Management, StdTasks

Portfolios - **tbMDPortfolio**

Optional field to tag a project with another Portfolio Scheme

Values: Building Services, Business Support Systems, Core Product Delivery, IT Platform, R&D New Technology

Primary LOB - **tbMDPrimaryLineOfBusiness**

Optional field to tag a project with a Business Owner

Values: Business Development, Business Technology Solutions, Corporate Affairs, Engineering and Product Delivery, Human Resources, Research and Development

Primary Role - **tbResPrimaryRole**

Set role for resource demand calculations

Values: **Executive Sponsor**, Director, Manager, Department Head, **Project Manager**, Vendor Manager, R&D, Innovation Manager, Engineer, Solution Architecture, Developer, Software Engineer, DevOps Engineer, Database Administrator, UX/UI Designer, Graphic Designer, Technical Writer, Test Automation, Tester, Risk Manager, Network Administrator, Financial Analyst, HR SME, Finance SME, Marketing Manager, Sales Manager, RF Engineer, Field Service Technician, NOC Technician

Primary Skill - **tbResPrimarySkill**

Tag resources for resource for skill visibility

Values: IT Architecture, Testing, Testing Software, Design, Design - Mechanical, Project Management, Procurement, Catch All Skill, na, Leadership, Management, Strategic Planning, Business Analysis, Stakeholder Management, Client Relationship Management, Vendor Management, Contract Management, Product Management, **Innovation**, Change Management, Solution Architecture, Siebel CRM Development, Cloud Computing, DevOps, Database Administration, Data Analytics, Artificial Intelligence, Robotics, UX/UI Design, Technical Writing, Document Management, Quality Assurance, Network Security, Risk Management, IT Operations, Incident Management, Network Administration, Network Operations (NOC), Customer Support, Training & Development, Human Resources, Financial Analysis, Facilities Management, Communication, Marketing, Sales, RF Engineering, Field Services, Site Acquisition, Design - UI, Testing Hardware, Java Development, JavaScript / Web Development, Python Development

Product - tbMDProduct

If this timebar is for creating a product, use this tag for reporting by Product

Values: Access Point, Analyzers, Antenna, Billing System, CEP, Customer Solution, DM System, Hot spot, Hot Spot, HR System, NA, Network, SAP System, Secure Cloud System, Security System, Ticketing system, Tools, VNS System, WIFI System

Program Activity Alignment (PAA) - tbMDProgActivityAlignment

GOC Field for tracking programs.

Values: 1 1 1 1 Sub-Sub-Program: Banks support of One-Benefit distribution, 1 1 1 2 Sub-Sub-Program: Tax System changes to suit new One-Benefit., 1 1 1 Sub-Program: One Canada wide monthly payment to each Canadian, 1 1 Program: Common One-Benefit service for all Canadians, Taxes, Food

Programs - tbMDProgram

Optional field to tag a project to a Program for Program Reporting

Values: Building and Office Modernization, Business Continuity Management Program, Centralized Billing and Receivables, Continuous Improvement Transformation, Continuous Internal Delivery Enablement, Core Operations Program, Digital Workplace - Transformation

Project Assessment Status - tbPASStatus

Helps determine the Project Assessment Score.

Values: Draft, **Completed**, **Approved**, **Rejected**, **On Hold**

Project Assessment Type - tbPASProjectType

Helps determine the Project Assessment Score.

Values: Incremental Improvement, **Enhancement**, New Feature, New Product, Breakthrough Innovation

Project Category - tbMDCategory

Metadata to categorize bars at all leveles, Projects, Tasks etc.

Values: SW Upgrade, Business System Configuration, HW Upgrade, Cloud Service New or Upgrade, HW, SW System Upgrade, New Ground to Cloud Service, Not Assessed, New HW, SW System, Upgrade Ground to Cloud Service, Not Applicable

Project Similarity - `tbPASProjectSimilarity`

Helps determine the Project Assessment Score.

Values: Very Similar, Somewhat Similar, Moderately Different, Quite Different, Completely New

RandIStage - `tbMDStageRIC`

Controlling Risk and Issue life-cycle workflow

Values: 1 Proposed, 2 Triaged, 3 Prioritized, 4 Assigned, 5 Solving, 6 Mitigating, 7 Closed

RandIState - `tbMDStateRIC`

Workflow Life-cycle tag

Values: Awaiting Input, Awaiting Approval, Approved, Awaiting Review

RandIStatus - `tbMDStatusRIC`

Timebar Task, Issue, Risk or CR Status.

Values: Managing, **Concerned**, **Critical**, **Completed**

Regulatory Approvals - `tbPASRegulatoryApprovals`

Helps determine the Project Assessment Score.

Values: None required, Internal approvals, Industry standards, Government approvals, Multiple regulatory bodies

Resource Class - `tbResResourceClass`

Resource Class External, Seasonal etc

Values: External, Permanent, Seasonal

Resource Department - `tbResDepartment`

Resource belongs to what department

Values: Finance, Field Operations, Information Technology, Legal Compliance, Engineering, Management

Resource Type - `tbResResourceType`

Resource type equipment, labour, material etc.

Values: Equipment, Labour, Material

Responsible Team - **tbMDResponsibleTeam**

Project Tagging

Values: Product Engineering, Business Analysis, Information Technology, IT Operations, Accounting, Corporate Communications, Finance, Procurement, Procurement & Vendor Mgmt, Design, Documentation, Executive Leadership, Facilities, Field Operations, Human Resources, Marketing, Product & Innovation, Product Management, Project Delivery, Quality Assurance, Sales, Security & Risk, Software Engineering, Solutions Engineering, Strategy & Innovation

Risk Category - **tbMDRiskCategory**

Metadata to categorize Issues. Used at Task level if Sub-Type is Risk

Values: Schedule, Financial, **Operational**, **Technical**, Organizational

Risk Impact - **tbMDImpact**

Tagging risks for the risk score result

Values: **Very Low**, Low, Medium, High, **Very High**, **Not Assessed**

Risk Probability - **tbMDProbability**

Tagging risks for the risk score result

Values: **Very Unlikely**, **Unlikely**, **Likely**, **Very Likely**, **Certain**, **Not Assessed**

Every value in this picklist is locked. Add your own freely; do not rename these.

Risk Vs Size - **tbMDRiskVsSizeAndComplexity**

Supports bubble charts

Values: **Not Assessed**, **41-50 Complex but Manageable**, **0-10 Very Small and Simple**, **51-60 Significant Complexity**, **11-20 Small and Straightforward**, **61-70 Large and Complex**, **21-30 Medium with Some Complexity**, **71-80 Very Large and Complex**, **31-40 Large with Moderate Complexity**, **81-90 Highly Complex and Risky**, **91-100 Extremely Complex and Risky**

Every value in this picklist is locked. Add your own freely; do not rename these.

Schedule Status - **tbMDHealth**

For Reporting Color indicators

Values: **Completed**, **Early**, **Late**, **On schedule**, **Slipping**, **Not Assessed**, **Blocked**

Every value in this picklist is locked. Add your own freely; do not rename these.

Severity - **tbMDSeverity**

Indicates how severe a priority item is, in terms of impact

Values: **Critical**, Important, **Normal**

Show In Reports - tbMDShowIn

Mainly used at Task, Issue, Risk, CR levels to ensure the bar shows in specific system report

Values: Facility Schedule, **Key Deliverables**, Assembly Shop, **Key Milestones**, **Key Risks**, **Project Charter**

Size - tbMDSize

Project or Task Tagging

Values: Small, Medium, Large, Huge

Sponsoring Department - tbMDSponsoringDepartment

Project Tagging

Values: Finance, IT Department, Presidents Office, Communications Public Relations, Legal Compliance, Operations

Stage - tbMDStage

Controlling project life-cycle workflow

Values: 1 Proposed, 2 Triaged, 3 Prioritized, 4 Selected, 5 Assigned, 6 Delivering, 7 Delivered, 8 Closed

Stakeholder Count - tbPASStakeholderCount

Helps determine the Project Assessment Score.

Values: 1-5 stakeholders, 6-10 stakeholders, 11-20 stakeholders, 21-50 stakeholders, Over 50 stakeholders

Strategic Priority - tbMDPriorityStrategic

Project Tagging

Values: High, Low, Medium

SubType - tbSubType

Task, Risk, Issue or Other. Determines Blue bar sub type

Values: **Issue**, Other, **Risk**, Task, CR

Supervisor - tbResSupervisor

Resources supervisor, not Manager

Values: John Hammond, Nora Antenna, Jim Key, Sam Backhaul, Helen Hunt, Lena Spectrum

Task Category - tbMDCategoryTask

Categorises a Task, kept separate from the Project-level Category

Values: Engineering, Purchasing, Assembly, Construction, Operate

Task Status - `tbMDStatusTask`

Status of a Task, kept separate from the Timebar Status picklist

Values: **Not Started**, Started, **Late**, **Not Assessed**

tbType - `tbMDTypes`

System Internal element that defines the bar type. E.g. Task, Project etc.

Values: **Portfolio**, Project, **Sub-Project**, **Milestone**, Task, **Allocation**

tbTypeInternal - `tbType`

System Internal element that defines the bar type. E.g. Task, Project etc.

Values: **Portfolio**, Project, **Sub-Project**, Task, **Milestone**, **Allocation**

Team Leader - `tbResTeamLeader`

Resource or person tagging

Values: France Finney, Mack Engineer, Chris Java, Kai Pipeline, Mike Side, Jack Top, App Team, CRM Team, Desktop Team

Team Size - `tbPASTeamSize`

Helps determine the Project Assessment Score.

Values: 1-3 people, 4-8 people, 9-15 people, 16-30 people, Over 30 people

Team Tech Experience - `tbPASTeamTechExperience`

Helps determine the Project Assessment Score.

Values: Expert level, Advanced, Intermediate, Basic, No experience

Technology Novelty - `tbPASTechnologyNovelty`

Helps determine the Project Assessment Score.

Values: Proven/Mature, Established, Emerging, Cutting-edge, Experimental

Test Status - `tbMDTestStatus`

Values for the `tbMDTestStatus` field

Values: **Not Started**, **In Progress**, Passed

Timebar Priority - `tbMDPriority`

Set timebar priority

Values: High, **Immediate**, Low, **Normal**, **Not Assessed**

Timebar Status - `tbMDStatus`

Mandatory values for enabling reporting.

Values: **In progress**, New, **Rejected**, Closed, **On Hold**, **Blocked**

Type Of Initiative - `tbMDProjectType`

Project Type Tagging for Portfolio Balancing

Values: **Mandatory Compliance**, **Maintenance Operational**, **Growth Enhancement**, External Contract Delivery, Internal Delivery

Value Stream - `tbMDValueStream`

Values for the `tbMDValueStream` field

Values: Customer Experience, Operations, Technology Platform

Vendor Dependencies - `tbPASVendorDependencies`

Helps determine the Project Assessment Score.

Values: No vendors, 1 reliable vendor, 2-3 known vendors, Multiple/new vendors, Critical unknown vendors

YesNo - `tbMDYesNo`

for tagging a time bar for true or false on unknown

Values: No, Yes, NA

Related Help Topics

- [Data Synchronization, Backup, Recovery and Retention Guide](#) - Import/export data for bulk editing
- [Cloud Publishing Guide](#) - Publish reports to the cloud
- [User Interface Guide](#) - Navigate the application
- [Data Synchronization, Backup, Recovery and Retention Guide](#) - Understand the data model

Part 6 — Risks, Issues and Change Requests

Risks, Issues & Change Requests User Guide

This guide explains how to track and manage Risks, Issues, and Change Requests (RIC) in Agilebars, Timebars, and Costbars applications. The RIC system helps you identify potential problems, log current issues, and manage proposed changes—all directly on your project timeline.

Overview: What is the RIC System?

The **RIC (Risks, Issues, Change Requests) system** allows you to:

- **Track Risks:** Potential problems that might occur in the future
- **Manage Issues:** Problems that are happening right now
- **Handle Change Requests:** Proposed changes to project scope, schedule, or budget

Key Concepts:

- RIC items are created as regular tasks using the Bar Creator
 - Each item is classified by changing the **SubType** field (Risk, Issue, or CR)
 - Visual indicators appear automatically on the Canvas (R, I, or CR prefix)
 - Dedicated views provide filtering, searching, and detailed tracking
 - Customize forms to match your organization's methodology
-

Quick Start: Creating Your First RIC Item

Step-by-Step Creation

1. Create a Task Bar:

- Click **Main Menu > New Bar**
- Drag the Task Creator Bar (blue) onto a Project Bar (green)
- Drop to create the task

2. Open the Form:

- Click on the **bar name** to open the Slide Out Right Form

3. Change Name:

- Update the task name to describe your Risk, Issue, or CR
- Examples:
 - "R-Database Migration Failure Risk"
 - "I-Server Outage on Production"
 - "CR-Add Mobile App Support"

4. Set SubType:

- Find the **SubType** field in the form
- Change from "Task" to "**Risk**", "**Issue**", or "**CR**"

5. Visual Indicator Appears:

- **Risk:** Bold **R** prefix appears before bar name
- **Issue:** Bold **I** prefix appears before bar name
- **Change Request:** Bold **CR** prefix appears before bar name

6. Save and Close:

- Data saves automatically
- Close the form

That's it! You've created a RIC item. Now let's explore how to track and manage it.

Visual Indicators on the Canvas

RIC items display with automatic visual indicators to help you quickly identify them:

Prefix Letters

- **R** = Risk
- **I** = Issue
- **CR** = Change Request

Color Coding by Status

When you set the **Status** field to specific values, the visual indicator changes:

Status	Visual Effect	When to Use
Normal	Black bold letter	Standard priority
Concerned	Red letter with thin red border	Needs attention
Critical	RED letter with thick red border	Urgent action required

Example:

- Normal Risk: **R** Database Migration Failure (black)
 - Concerned Issue: **I** Server Outage on Production (red letter, thin border)
 - Critical Risk: **R** Security Vulnerability Discovered (red letter, thick border)
-

Accessing the RIC Management Page

To view and manage all your Risks, Issues, and Change Requests in one place:

1. Choose **Main Menu > RIC** (or **Main Menu > Risks/Issues**)
2. The RIC page opens with **Card View** as the default

Two View Options:

- **Card View:** Visual cards with expandable details (recommended for viewing)
 - **Tabular View:** Sortable table with all fields (recommended for bulk analysis)
-

Card View: Visual Management

Card View displays your RIC items as color-coded, expandable cards in a two-column responsive grid.

Card Colors by Type

- **Risk Cards:** Blue gradient background
- **Issue Cards:** Red gradient background
- **Change Request Cards:** Pink/Purple gradient background

Card Structure

Each card shows:

1. Header Section:

- **Breadcrumb:** Project hierarchy (Portfolio > Program > Project)
- **SubType Chip:** Colored badge (Risk/Issue/CR)
- **ID and Name:** Item identifier and title
- **Action Buttons:**
 - **Edit** (pencil icon) - Opens full FOCD form
 - **Print** (printer icon) - Prints this card

2. Metadata Section (Always visible):

- **Parent Project:** Which project this belongs to
- **Dates:** Start/Finish (planned) and Actual dates (if started/completed)
- **Schedule Health:** Color-coded indicator (Green/Yellow/Red)
- **Risk-Specific:**
 - **Probability:** Likelihood rating
 - **Impact:** Severity rating
 - **Score:** Auto-calculated (Probability × Impact) with color coding
 - **Mitigation Status:** Current mitigation progress
- **Escalation Level:** Color-coded (e.g., red for Executives)
- **Owner:** Person responsible
- **Status:** Current status (New, In Progress, Closed, etc.)
- **State:** Workflow state (Awaiting Approval, Approved, etc.)
- **Category:** Classification type
- **Priority:** Color-coded (Immediate, High, Normal, Low)

3. Details Section (Expandable - Click "Show Details"):

For Risks:

- Description
- Executive Summary
- Notes
- Risk Response Strategy (rich text)
- Mitigation Plan (rich text)
- Contingency Plan (rich text)

For Issues:

- Description
- Executive Summary
- Notes

For Change Requests:

- **Tabbed Interface** with 4 tabs:
 - **Overview:** Description, Executive Summary, Problem/Opportunity
 - **Impact Assessment:** Schedule, Budget, Scope, Resource Commentary

- **Benefits & Justification:** Expected Benefits, Cost-Benefit Analysis, Options Analysis
- **Notes:** Implementation Approach, Next Steps, Dependencies

Inline Editing in Cards

You can edit text fields directly in the cards:

1. Click "Show Details" to expand the card
2. Click on any text field
3. Edit the content
4. Click outside the field to save automatically
5. Rich text editors available for Risk strategy/plan fields

Note: Some fields require opening the full FOCD form to edit.

Tabular View: Data Analysis

Tabular View displays all RIC items in a sortable, scrollable table—ideal for analysis and reporting.

Features

- **Sortable Columns:** Click any column header to sort ascending/descending
- **Fixed Actions Column:** Edit button always visible on the left
- **Horizontal Scrolling:** Scroll right to see all columns
- **Compact View:** See many items at once
- **Quick Access:** Click Edit button to open FOCD form

Available Columns

The table includes columns for:

- Actions (Edit button)
- Hierarchy (L1, L2, L3)
- ID, Name, SubType
- Start Date, Finish Date, Actual Start, Actual Finish
- Owner, Status, State, Priority
- **Risk Fields:** Probability, Impact, Score, Category, Mitigation Status, Escalation Level
- **Health Indicators:** Schedule Health (color-coded)
- Description, Executive Summary, Notes

Sorting Data

1. Click any column header (e.g., "Priority")
 2. Data sorts ascending
 3. Click again to sort descending
 4. Default sort: Hierarchy Order (top to bottom as on Canvas)
-

Search and Filter

Both Card View and Tabular View include powerful search and filter tools at the top of the page.

Free-Text Search

Search Box (top of page):

- Type any text to search across all fields
- Results filter in real-time as you type
- Searches: Names, Descriptions, Notes, all text content

Example:

- Type "database" to find all items mentioning database
- Type "migration" to find migration-related items

Picklist Filters

Filter Dropdowns:

- Multiple dropdown filters available
- Filter by any picklist field:
 - **SubType:** Show only Risks, Issues, or CRs
 - **Status:** Active, Closed, On Hold, etc.
 - **Priority:** Immediate, High, Normal, Low
 - **Owner:** Filter by responsible person
 - **Category:** Filter by classification
 - **Escalation Level:** Show only escalated items
 - **State:** Filter by workflow state

Combining Filters:

- Select multiple filters to narrow results
- Example: Show only "High Priority Risks" with "Critical Status"

Recalculate All Button

Click "**Recalculate All**" to:

- Refresh all calculated fields
 - Recalculate every Risk Score from its current Probability and Impact (use this after any change to those picklists, and to clear scores left over from the old 0-100 scale)
 - Ensure data consistency across views
 - Rebuild system-generated data
-

Editing with FOCD Forms

FOCD (Field-Oriented Control Design) forms provide detailed editing of all RIC metadata.

Opening a FOCD Form

Three Ways to Open:

1. Click **Edit button** (pencil icon) on any card
2. Click **Edit button** in Tabular View row
3. Click on a bar on the Canvas, then click **Edit** in the popup

Two Form Types:

- **Risk Edit Form (dfRisk)**: For Risks and Change Requests
- **Issue Edit Form (dfIssue)**: For Issues

Form Features

Auto-Save:

- Changes save automatically to the database when you move off a field
- No "Save" button needed for most edits

Rich Text Editors (Risks Only):

 Three fields use rich text editors:

- **Risk Response Strategy**: Formatted text with bold, italic, lists
- **Mitigation Plan**: Detailed action plans with formatting
- **Contingency Plan**: Backup plans with formatting

Risk Score Calculator:

- When you change **Probability** or **Impact**, the **Score** field updates automatically
- Formula: $\text{Score} = \text{Probability (1-5)} \times \text{Impact (1-5)} = 1 \text{ to } 25$, higher is worse

Field Types:

- **Text Inputs**: Standard single-line text
- **Dropdowns**: Picklist selections
- **Text Areas**: Multi-line text with auto-resize
- **Rich Editors**: Formatted text (Risks only)
- **Date Pickers**: Calendar date selection
- **Section Headers**: Organize fields into groups

Form Actions

Save Button:

- Click to explicitly save all changes
- Returns to Card or Tabular View

Move/Resize Fields Button:

- Click to enter customization mode
- Drag fields to reposition
- Resize using edge handles
- Click "Stop" to save layout

Close Button (X):

- Saves changes and closes form

- Reloads application with updated data

Managing Risks

Risks are potential problems that might occur in the future. Effective risk management helps you prevent issues before they happen.

Core Risk Fields

When managing a Risk, you'll work with these key fields:

1. Probability (Likelihood)

How likely is this risk to occur? Each value carries a weight of 1 to 5, used to calculate the Score:

Value	Weight
Very Unlikely	1
Unlikely	2
Likely	3
Very Likely	4
Certain	5
Not Assessed	<i>no weight — the risk scores 0</i>

Do not rename these five values. The weights are held in the application code and matched by exact text, so a renamed value stops scoring altogether and the risk silently reads Not Assessed. You may add values, but only these five produce a score. See *Customizing picklists* below.

2. Impact (Severity)

How severe would the consequences be if this risk occurs? Consider schedule, budget, scope and quality. Each value carries a weight of 1 to 5:

Value	Weight
Very Low	1
Low	2
Medium	3
High	4
Very High	5
Not Assessed	<i>no weight — the risk scores 0</i>

The same warning applies: these five values are matched by exact text and must not be renamed.

3. Score (Auto-Calculated)

The Score follows the standard **5x5 probability-impact matrix** used by PMBOK, ISO 31000, PRINCE2 and M_o_R:

Score = Probability (1-5) x Impact (1-5) → 1 to 25, where higher is worse.

It is calculated for you the moment you set both axes, on every surface — the risk form, the cards, the tabular view and the inline grid.

Set only one axis and the score stays 0. So does setting either to *Not Assessed*. A score of 0 means *not assessed*, and is deliberately outside the valid 1-25 range so it can never be mistaken for a real, low assessment.

The four bands:

Band	Score	Reading
Low	1-4	green
Moderate	5-9	amber
High	10-14	orange
Extreme	15-25	red
<i>Not Assessed</i>	0	grey

The score displays as the number and its band together — **12 - High** — so the reading never depends on remembering where the boundaries fall.

Worked examples:

Probability	Impact	Score	Band
Certain	Very High	25	Extreme
Likely	High	12	High
Unlikely	Medium	6	Moderate
Very Unlikely	Very Low	1	Low
Likely	<i>(not set)</i>	0	Not Assessed

The risk matrix panel

The risk form carries a **5x5 heat map** showing all 25 combinations, with your assessed cell ringed. Probability runs down the rows worst-first and Impact across the columns, so it reads like any published risk heat map. Hovering a cell tells you its combination, score and band, and a legend underneath shows the four band ranges.

The panel is draggable — move it aside if it covers a field you are working on.

A note if you have used earlier versions. The score used to run the other way: it was normalised to 0-100 and inverted, so 100 meant *safest*. Most of the product read it as higher-is-worse, which meant the safest risk on a card rendered in the loudest red and the most dangerous one in plain small text. The scale, the colours and the matrix now all run the same direction. **Scores recorded under the old scale**

are meaningless on the new one — reassess, or run **Recalculate All**, which recalculates every score from its Probability and Impact.

4. Category

Classify the type of risk:

Default Values for Risks:

- Strategic: Alignment with business goals
 - Financial: Budget and funding risks
 - Operational: Day-to-day operations
 - Technical: Technology and system risks
 - Organizational: People and structure risks
-

5. Risk Response Strategy

Your written approach to handling the risk.

Common Strategies:

- **Avoid:** Change plans to eliminate the risk
- **Mitigate:** Reduce likelihood or impact
- **Transfer:** Shift risk to third party (insurance, vendor)
- **Accept:** Acknowledge risk and prepare to deal with it

Field Type: Rich text editor (supports formatting)

6. Mitigation Plan

Detailed actions to reduce the risk.

What to Include:

- Specific steps to take
- Who is responsible for each step
- Timeline for completion
- Resources needed
- Success metrics

Field Type: Rich text editor (supports formatting)

7. Mitigation Status

Track progress of your mitigation efforts.

Default Values:

- Identified: Risk recognized but not yet assessed

- Assessed: Analysis complete
 - Mitigation Planned: Plan created
 - In Progress: Executing mitigation
 - Mitigated: Risk reduced to acceptable level
 - Under Review: Evaluating effectiveness
 - Accepted: Acknowledged without mitigation
 - Transferred: Shifted to third party
 - Escalated: Raised to higher management
 - Closed: Risk no longer relevant
 - Deferred: Postponed to later
-

8. Contingency Plan

Backup plan if the risk occurs despite mitigation.

What to Include:

- Fallback actions
- Resources reserved for response
- Decision triggers
- Communication plan

Field Type: Rich text editor (supports formatting)

9. Escalation Level

Who needs to be informed if this risk materializes?

Default Values:

- Executives: C-level notification
- Directors: Department heads
- Project Office: PMO oversight
- Not Assessed

Display: Color-coded based on level

10. Trigger Event

What event or condition signals this risk is occurring?

Examples:

- "Server CPU usage exceeds 80% for 24 hours"
- "Project spend reaches 90% of budget with 40% work remaining"
- "Key vendor misses two consecutive delivery dates"

Field Type: Plain text area

11. Early Warning Indicators

Metrics or signals to detect the risk early.

Examples:

- Weekly defect trend increasing
- Resource availability dropping below threshold
- Velocity declining for two sprints
- Stakeholder engagement score decreasing

Field Type: Plain text area

Risk Management Workflow

Step 1: Identify

1. Create Risk item on Canvas (SubType = Risk)
2. Give it a clear, descriptive name
3. Set Category for classification

Step 2: Assess

1. Open RIC page (Main Menu > RIC)
2. Find your risk and click Edit
3. Set **Probability** and **Impact**
4. Review calculated **Score**
5. Set **Escalation Level** if needed

Step 3: Plan Response

1. Write **Risk Response Strategy** (Avoid/Mitigate/Transfer/Accept)
2. Detail **Mitigation Plan** with specific actions
3. Write **Contingency Plan** as backup
4. Set **Trigger Event** and **Early Warning Indicators**
5. Assign **Owner**
6. Set **Mitigation Status** to "Mitigation Planned"

Step 4: Execute Mitigation

1. Update **Mitigation Status** to "In Progress"
2. Add progress **Notes** regularly
3. Monitor early warning indicators
4. Update **Status** to "Concerned" or "Critical" if risk increases

Step 5: Monitor and Close

1. Review risk regularly (weekly/monthly)
2. Update Probability/Impact if conditions change
3. When risk is addressed:
 - Set **Mitigation Status** to "Mitigated" or "Closed"
 - Set **Status** to "Closed"

- Document outcome in **Notes**
-

Managing Issues

Issues are problems that are happening right now and require immediate attention.

Core Issue Fields

Issues use simpler fields than Risks since the problem has already occurred:

Key Fields:

- **Description:** What is the problem?
- **Executive Summary:** High-level summary for management
- **Notes:** Updates on resolution progress
- **Status:** Current state (New, In Progress, Closed, etc.)
- **State:** Workflow tracking (Awaiting Input, Under Review, etc.)
- **Priority:** Urgency level (Immediate, High, Normal, Low)
- **Owner:** Person responsible for resolving
- **Escalation Level:** Who needs to know (Executives, Directors, Project Office)
- **Category:** Type of issue

Category Values for Issues:

- Product: Product defects or problems
- Financial: Budget or cost issues
- Operational: Process or operational problems
- Technical: Technical failures or bugs
- Schedule: Timeline delays

Issue Management Workflow

Step 1: Log the Issue

1. Create Issue item on Canvas (SubType = Issue)
2. Give it a clear name describing the problem
3. Set **Priority** based on urgency
4. Set **Category** for classification

Step 2: Assess

1. Open RIC page and click Edit
2. Write detailed **Description** of the problem
3. Write **Executive Summary** for stakeholders
4. Set **Escalation Level** if critical
5. Assign **Owner** to resolve

Step 3: Resolve

1. Update **Status** to "In Progress"
2. If urgent: Set Status to "Concerned" or "Critical" (triggers red visual indicator)

3. Add **Notes** with resolution activities
4. Update **State** as it moves through workflow (e.g., "Awaiting Approval")

Step 4: Close

1. When resolved:
 - Set **Status** to "Closed"
 - Document resolution in **Notes**
 - Set **Actual Finish** date
 2. Review **Notes** to capture lessons learned
-

Managing Change Requests

Change Requests (CRs) track proposed changes to project scope, schedule, budget, or resources.

Core CR Fields

CRs have extensive fields to justify and analyze the proposed change:

Overview Tab:

- **Description:** What is the change?
- **Executive Summary:** High-level summary
- **Problem/Opportunity:** What drives this change?

Impact Assessment Tab:

- **Schedule Commentary:** How does this affect timeline?
- **Budget Commentary:** Cost impact
- **Scope Commentary:** How does scope change?
- **Resource Commentary:** Resource needs

Benefits & Justification Tab:

- **Expected Benefits:** What do we gain? (Rich text editor)
- **Cost-Benefit Analysis:** Is it worth the investment?
- **Options Analysis:** What alternatives were considered?

Notes Tab:

- **Implementation Approach:** How will we execute?
- **Next Steps:** Immediate actions
- **Key Dependencies:** What's needed to proceed?

Common Fields:

- **Owner:** CR sponsor/requestor
- **Status:** Approval status
- **State:** Workflow state (Awaiting Approval, Approved, etc.)
- **Priority:** Importance level

Change Request Workflow

Step 1: Submit CR

1. Create CR item (SubType = CR)
2. Name it clearly (e.g., "CR-Add Mobile App Support")
3. Open RIC page, find CR, click Edit

Step 2: Document Justification

1. Overview Tab:

- Write detailed **Description**
- Write **Executive Summary**
- Explain **Problem/Opportunity**

2. Benefits Tab:

- Document **Expected Benefits** (use rich text formatting)
- Complete **Cost-Benefit Analysis**
- Describe **Options Analysis** (alternatives considered)

Step 3: Assess Impact

1. Impact Assessment Tab:

- **Schedule Commentary**: Add X weeks? Delay milestone?
- **Budget Commentary**: Cost increase? Source of funds?
- **Scope Commentary**: What's added/removed?
- **Resource Commentary**: Need additional people/skills?

Step 4: Plan Implementation

1. Notes Tab:

- **Implementation Approach**: How will we execute?
- **Key Dependencies**: What must happen first?
- **Next Steps**: Immediate actions

2. Set **Owner** (requestor)

3. Set **Priority**

Step 5: Approval Process

1. Set **State** to "Requested Approval"
2. Update **Status** and **State** as it moves through workflow:
 - Awaiting Approval
 - Approved / Rejected
 - Awaiting Implementation
3. Add **Notes** with decision rationale

Step 6: Implementation or Closure

- If Approved: Create tasks to implement, link to CR
- If Rejected: Set **Status** to "Rejected", document reason
- If Complete: Set **Status** to "Closed"

Customizing Forms and Picklists

The Risk and Issue forms, and every picklist behind them, are configured the same way as everything else in the product. The **Forms, Reports and Graphs Guide** is the full reference; this section covers only what is specific to Risks, Issues and CRs.

Earlier versions of this guide routed you through *Reports > Data Management Grids*. **That screen no longer exists.** Configuration now lives on five reports under **Report Menu > Other**.

Show, hide, move and add fields

Report Menu > Other > Fields Values. Filter the **Form** column to **dfRisk** or **dfIssue** and you have every field on that form, one per row.

To do this	Do this
Hide a field	open it on the form and set Show to No
Show it again	set Show back to Yes
Reposition or resize	edit Top , Left and Width directly in the grid
Add a field	filter to the form first, click Add , then pick the field name
Remove a field	delete its row

The layout columns are the only cells in that report you edit in the grid rather than on the form — because laying out a form means comparing one field's numbers against its neighbours'.

Or move fields by hand. Open a Risk or Issue form, click the **Move/Resize Fields** button, drag the fields where you want them, and click **Stop** to save. Easier than typing coordinates for small adjustments.

Which picklists drive Risks, Issues and CRs

Report Menu > Other > Picklist Values. Filter to the picklist, open a value, edit it.

Picklist	Field	Notes
Risk Probability	tbMDProbability	Do not rename the five scoring values — see below
Risk Impact	tbMDImpact	Do not rename the five scoring values
Risk Category	tbMDCategory	free to customise; add your own
Issue Category	tbMDCategory	free to customise
CR Category	tbMDCategory	free to customise
Mitigation Status	tbMDMitigationStatus	free to customise, with one exception: Escalated is read by the health rules
Escalation Level	tbMDEscalationLevel	Executives, Directors, Project Office are read by the health rules

Picklist	Field	Notes
Timebar Status	tbMDStatus	Closed and Rejected are what make an item stop counting
Approval State	tbMDState	the workflow states
Timebar Priority	tbMDPriority	free to customise
RandIStage, RandIState, RandIStatus	R&I workflow	free to customise

Values in bold are locked. They carry a padlock in Picklist Values and cannot be renamed there, because the application compares against that exact text. You may always **add** your own values to any of these lists.

Why the locked values matter here specifically

Risks, Issues and Change Requests do not just describe themselves — they drive their parent project's **Scope, Risk and Issue health indicators**. Those rules read your items by exact value:

- An item stops counting toward health once its Status is **Closed** or **Rejected**.
- An open item escalated to **Executives** or **Directors**, or a risk whose Mitigation Status is **Escalated**, turns the parent's indicator **Red**.
- An open item escalated to **Project Office**, or three or more open items, turns it **Yellow**.

Rename one of those values and the health rules stop seeing your escalations — without any error. The *Forms, Reports and Graphs Guide* covers the full rule set.

People fields come from the Resource Pool

Contact / Requestor, and the other person-named fields on these forms, no longer read a picklist of typed names. They read the **live Resource Pool** — the **Human** rows in Resources. To change who can be picked, maintain the resource pool, not the Tags table.

Tips and Best Practices

Risk Management

1. **Regular Reviews:** Review risks weekly or bi-weekly
2. **Update Probability/Impact:** Reassess as conditions change
3. **Monitor Early Warnings:** Track indicators proactively
4. **Document Triggers:** Define clear conditions that signal risk is occurring
5. **Keep Plans Current:** Update mitigation plans as you learn more
6. **Use Visual Indicators:** Set Status to "Concerned" or "Critical" for urgent risks

Issue Management

1. **Log Immediately:** Create issues as soon as problems occur
2. **Escalate Appropriately:** Set Escalation Level for critical issues
3. **Update Progress:** Add Notes regularly with resolution activities

4. **Set Clear Ownership:** Always assign an Owner
5. **Close Properly:** Document resolution and lessons learned
6. **Use Red Indicators:** Set Status to "Critical" for urgent issues to trigger visual alerts

Change Request Management

1. **Complete Justification:** Fill all Impact Assessment fields
2. **Quantify Benefits:** Use specific metrics in Expected Benefits
3. **Consider Alternatives:** Document Options Analysis thoroughly
4. **Track Through Workflow:** Update State as approval progresses
5. **Link to Implementation:** Create implementation tasks and link to CR

General Best Practices

1. **Consistent Naming:** Use prefixes (R-, I-, CR-) in names
 2. **Use Categories:** Classify items for better filtering
 3. **Set Priorities:** Helps sort and filter critical items
 4. **Leverage Search:** Use search box to find items quickly
 5. **Print Cards:** Print individual cards for meetings or reviews
 6. **Regular Recalculation:** Click "Recalculate All" periodically to ensure data accuracy
 7. **Customize Forms:** Remove unused fields to simplify forms
 8. **Standardize Picklists:** Ensure team uses consistent terminology
-

Troubleshooting

Visual Indicator Not Appearing

Problem: R, I, or CR prefix doesn't show on Canvas

Solution:

1. Verify **SubType** field is set to "Risk", "Issue", or "CR" (exact spelling)
 2. Refresh Canvas (click Refresh icon on top menu)
 3. If still missing, Right-click Canvas > Recalculate All
-

Status Not Turning Red

Problem: Set Status to "Concerned" but letter doesn't turn red

Solution:

1. Ensure exact spelling: "Concerned" or "Critical"
 2. Check tbTags store has these values defined
 3. Refresh Canvas after changing Status
-

Risk Score Not Calculating

Cause: One axis is missing, or a value has been renamed.

Solution:

1. Set **both** Probability **and** Impact — one alone always scores 0.
2. Check neither is set to **Not Assessed**, which also scores 0.
3. If both are set and the score is still 0, the picklist value has been renamed. The five scoring values in each list are matched by exact text: **Very Unlikely**, **Unlikely**, **Likely**, **Very Likely**, **Certain** and **Very Low**, **Low**, **Medium**, **High**, **Very High**. Restore the original spelling in **Picklist Values**.
4. Run **Recalculate All** to recalculate every score in the database.

Cause: The score looks wrong rather than absent — a trivial risk showing red, or a severe one showing green.

Solution: The row is holding a score from the old 0-100 scale, where 100 meant safest. Run **Recalculate All**; every score is recalculated from its Probability and Impact on the current 1-25 scale.

Fields Missing in Form

Problem: Expected fields don't appear in Risk or Issue form

Solution:

1. Go to **Report Menu > Other > Fields Values**
 2. Filter **Form** to **dfRisk** or **dfIssue** and find the field
 3. Open it on the form and check **Show** is **Yes**
 4. If the field has no row on that form at all, **Add** one and pick the field name
 5. Refresh the application after changes
-

Cannot Edit in Card View

Problem: Click on field in card but can't edit

Solution:

1. Click **"Show Details"** to expand card first
 2. Some fields require full FOCD form—click **Edit button**
 3. Ensure you're not in read-only mode
-

Filters Not Working

Problem: Select filter but no results appear

Solution:

1. Check that items exist with selected filter value
 2. Clear other filters—they may be combining to exclude all results
 3. Try Search box to verify items exist
 4. Click "Recalculate All" to refresh data
-

Data Storage

Understanding where RIC data is stored helps with backups, spreadsheet sync, and troubleshooting.

IndexedDB Stores

tbTimebars Store:

- Contains all RIC items as task records
- Key field: **tbSubType** = "Risk", "Issue", or "CR"
- Includes: ID, Name, Type, Start/Finish dates, Hierarchy (L1-L5), Owner

tbMetaData Store:

- Contains all metadata fields (100+ fields)
- Links to tbTimebars via **tbMDID = tbID**
- Includes: Probability, Impact, Score, Status, Priority, Description, Plans, Commentary

tbFields Store:

- Defines FOCD form layouts
- Contains field positions (dfCoordLeft, dfCoordTop)
- Controls field visibility (dfShowYN)
- Filter by **dfFormName** = "dfRisk" or "dfIssue"

Spreadsheet Sync

RIC items can be exported/imported via spreadsheet:

Export:

1. Click **Hamburger Icon > Export to CSV**
2. Files created: **Timebars.csv** and **MetaData.csv**
3. Open tbClientMaster.xlsm spreadsheet
4. Click "Import All" to load data

Edit in Spreadsheet:

- Timebars worksheet: ID, Name, SubType, Dates, Hierarchy
- MetaData worksheet: All 100+ metadata fields
- Edit, add rows, copy-paste as needed

Import:

- Save spreadsheet
- Drag and drop onto Canvas
- RIC items update from spreadsheet data

Related Help Topics

- [User Interface Guide](#) - Creating and editing bars
 - [Data Synchronization, Backup, Recovery and Retention Guide](#) - Data stores, backup, import and export
 - [Local Reports Guide](#) - Risk and Issue reports
 - [Configurable Data Guide](#) - Forms, picklists and the status/health rules
-

Summary

Key Takeaways:

1. **Three Item Types:** Risks (potential problems), Issues (current problems), Change Requests (proposed changes)
2. **Create as Tasks:** Use Bar Creator, then change SubType field
3. **Visual Indicators:** R, I, CR prefixes appear automatically; turn red for Concerned/Critical status
4. **Two Views:** Card View (visual) and Tabular View (analytical)
5. **Powerful Search:** Free-text search plus multiple picklist filters
6. **FOCD Forms:** Detailed editing with auto-save, rich text editors and the risk matrix panel
7. **Risk Management:** Assess Probability and Impact; the Score is the 5x5 matrix product, 1-25, higher is worse
8. **Customizable:** Show/hide fields, move/resize layout, customise picklists - all from Report Menu > Other
9. **Track Workflows:** Use Status and State fields to track approval and resolution
10. **Regular Reviews:** Update items frequently; use visual indicators for urgency

Effective RIC management helps you prevent problems, resolve issues quickly, and manage changes systematically.

Part 7 — How to Use Ask AI



How to Use Ask AI

What "Ask AI" Is

Ask AI is the assistant built into your product. It does two things for you:

1. **Creates your project data for you.** Instead of building Projects, Work Packages, Tasks, Milestones and resource Allocations by hand, you describe what you want in plain English and the AI builds the bars on your timeline — at whatever level of the hierarchy you are working in.
2. **Answers your questions.** The **Ask AI** button on the main menu opens a help chat that answers "how do I..." questions about using the app, based on the official documentation.

You do not need to know any technical field names or rules. You write the way a project manager, portfolio manager or engineer would talk, and the AI fills in the rest.

Ask AI is powered by Google Gemini. After the AI finishes creating data, **refresh the page (press F5)** to see the new bars appear on the canvas.



Getting Started: the "Get Started" L1 bar

When you open a brand-new, empty canvas in **Timebars** or **Costbars**, the app automatically places one starter bar near the top for you:

L1 Get Started here using AI, click the L1 ID below

This is your starting point — a top-level **Portfolio (L1)** that runs from today for about six months. You don't have to create anything first; just use it to launch the AI.

Agilebars note: Agilebars uses a simpler two-level structure and does **not** create this starter bar. Agilebars users begin at the Project level (see the Agilebars guide).

The two things you click

Everything in Ask AI starts from a bar on the canvas:

1. **Click the bar's ID** (the little ID label/circle on the bar). This opens that bar's **status pop-up form**.
2. On that form, click the ✨ **sparkle icon** (the "Ask AI" button). This opens the **AI Create** dialog for that bar.

The AI Create dialog always knows **which bar you launched it from** — it shows the ID, name and level at the top — so anything it creates is placed correctly underneath that bar.

Some levels also have **named buttons** on the status form (for example **Resource Plan** or **Create Allocations**) that run a specific action directly. These are described below.

What AI Creates at Each Level

Think of your plan as a set of nested levels:

Portfolio (L1) → Project (L2) → Work Package / Sub-Project (L3) → Task (L4) → Allocation (L5 — a person assigned to a task)

You move down this ladder one step at a time. At each step you click the bar's ID, click the ✨ sparkle icon, and tell the AI what you want.

1 From a Portfolio (L1): create a Project

This is where most people begin, using the **Get Started** bar.

1. Click the **ID** of the Portfolio bar, then click the ✨ **sparkle icon**.
2. In the big text box, describe the project. You can:
 - paste in a **Business Case** — formal (with headings) or rough — and the AI maps it into the Project Charter fields, or
 - simply type a **project name and a short description**, plus any background you have.
3. Optional tick-boxes:
 - **Fill in Project Charter fields with AI** — after the project is created, the AI also enriches its Project Charter fields (value proposition, benefits, success criteria, and so on).
 - **Create a Resource Plan** — the AI adds one **"Resource Plan"** task with **generic role placeholders** for the kinds of people the project will need (see *Resource Plan* below).
4. Click **Create**.

Result: one new **Project (L2)** appears under your Portfolio as a **New** initiative, with its Project Charter fields filled from your Business Case. Anything the AI cannot map to a specific field is preserved under headings in **Management Notes** (tbMDNotes) so nothing is lost.

 The more you tell it — scope, goals, timeline, budget, who is involved — the more specific and useful the result.

2 From a Project (L2): create Work Packages, Tasks and Milestones

1. Click the **ID** of the Project bar, then the ✨ **sparkle icon**.
2. Describe how the project should be broken down (for example: *"Break this into a design phase, a build phase, and a testing phase"*). If you are not specific, the AI uses sensible project-management judgment.
3. Optional tick-box:
 - **Assign people to the Tasks (Human resources)** — after the Tasks are created, the AI also assigns the best-fit **real person** from your Resource Pool to each Task (see *Assigning people* below).
4. Click **Create**.

Result: new **Work Packages (Sub-Projects, L3)** are created under the Project, each containing the **Tasks (L4)** and **Milestones** needed.

Also on the Project status form: a **Resource Plan** button. Click it any time to generate the generic-role Resource Plan on demand (handy if you didn't tick the box when the project was first created).

3 From a Work Package / Sub-Project (L3): create Tasks and Milestones

1. Click the **ID** of the Work Package bar, then the ✨ **sparkle icon**.
2. Describe the work in that package.
3. Optional tick-box:
 - **Create Allocations for the Tasks (assign people)** — the AI assigns a real person to each new Task.
4. Click **Create**.

Result: **Tasks (L4)** and **Milestones** are created under the Work Package (up to six tasks per package), optionally staffed with people.

Also on the Work Package status form: a **Create Allocations** button. Click it to staff the Work Package's **existing** Tasks with people — useful when you created the tasks earlier without ticking the "assign people" box. (If the Work Package has no tasks yet, the assistant will tell you to create tasks first.)

4 From a Task (L4): assign people to it

1. Click the **ID** of the Task bar, then the ✨ **sparkle icon**.
2. Describe the roles you need in plain words, for example:
 - *"Add a developer and a tester"*
 - *"I need a business analyst and a project manager"*
3. Click **Create**.

Result: the AI finds the matching people in your Resource Pool and adds them as **Allocations (L5)** under the Task. If you ask for "a developer," it includes every matching role (for example *Senior Developer* and *Web Developer*), one allocation each. New allocations always appear neatly **below the task**, so they are easy to see.

🧩 Two kinds of staffing: Resource Plan vs. Assigning People

It helps to know the difference between the two ways AI adds resources:

	Resource Plan	Assign People / Create Allocations
What it adds	Generic role placeholders (e.g. "Project Manager", "R&D")	Real, named people from your Resource Pool
Answers	<i>What kinds of people will this project need?</i>	<i>Who specifically will do this work?</i>
Where	Portfolio "Create a Resource Plan" tick-box, or the Resource Plan button on a Project	"Assign people" tick-box at Project/Work Package level, the Create Allocations button, or the ✨ button on a Task
Used when	Early planning, before you know exact names	Detailed planning, staffing the actual work

Resource Plan is for early estimating — it judges which roles the project needs and, if it is unsure, always includes a Project Manager and an R&D role. It never assigns specific named people.

Assigning People picks the best **named person** for each task by matching the task to each person's **role and skill**, and it checks that the person is not already over-committed. It keeps people sensibly loaded and will not pile work onto someone who is already very busy. If no suitable person is free, it drops in a generic role placeholder instead, so the task is never left empty without telling you.

 You control which roles a Resource Plan can use

A Resource Plan does not see your whole Generic resource list. It only sees Generic resources whose **Quantity is greater than zero**.

That gives you a simple lever. Roles that exist for reporting or org structure rather than for doing the work — governance boards, steering committees, management overhead — should be set to **Quantity 0** in the Shared Resource Pool. They stay in the pool and stay reportable, but the AI will never propose them as allocations.

Quantity	Effect
Greater than 0	The role is real capacity — the AI can plan with it, and it counts towards supply in the Supply and Demand grids
0	The role is invisible to planning — never proposed, never counted as capacity

Two things follow from this, and they are the usual surprises:

- **The AI keeps proposing managers you did not want.** Do not rewrite the prompt — set those roles to Quantity 0 and generate again
- **A role you expected never appears.** It is almost certainly sitting at Quantity 0. Give it a quantity and it becomes available

Keeping the quantities honest is the resource pool manager's job, and it is the single highest-value thing you can do for the quality of AI resource plans.

 While the AI Works

When you click **Create**, the dialog shows a **running progress log** — gathering context, calling the AI, creating each row — and finishes with a short **summary** of what was created (and a note of anything it adjusted or skipped). If a step has a problem, it is clearly flagged, and any work that did succeed is kept.

When it is done, **refresh the page (F5)** to see the new bars on your timeline.

Tips for Great Results

Do:

- Give the AI real context — paste a Business Case, scope notes, or a solid description.
- Mention timelines, budgets, key phases, and who is involved when you know them.
- Work **top-down**: create the Project first, then break it into Work Packages and Tasks, then assign people.
- Keep a healthy **Resource Pool** (named people with roles, skills, rates and availability) so staffing has good people to choose from.

× **Avoid:**

- One-word requests like *"make a project"* — you'll get a generic result.
 - Expecting people to be assigned if your Resource Pool is empty — add resources first.
-

The Ask AI Help Assistant

The **Ask AI** button on the main menu (and on the canvas toolbar) opens the **Help Assistant** — a chat window where you can ask questions about how to use the app, such as *"How do I create a baseline?"* or *"How do resource charts work?"*. It answers from the official product documentation for your edition (Agilebars, Timebars or Costbars).

- Type your question and press **Ask** (or press Enter).
 - Use **Clear** to start a fresh conversation.
 - Click the ✕ to close it.
-

Troubleshooting

"I clicked Create but nothing appears on the canvas." Refresh the page (F5). New bars show after a reload.

"No people were assigned to my tasks." Your Resource Pool may have no suitable **named (Human) resources**. Add people with roles and skills, then use the **Create Allocations** button on the Work Package, or the ✨ button on a Task.

"The tasks don't quite match what I wanted." Give more detail — name the phases, deliverables or roles you expect — and try again on the same bar.

"I don't see the Get Started bar." It only appears on an **empty** Timebars or Costbars canvas (when there are no Portfolios yet). Agilebars does not use it. If you already have bars, just use the ✨ button on any existing bar.

Ready to build your plan? Click an ID, click the ✨ sparkle, and tell the AI what you need. 

Part 8 — Supply and Demand Grids

Timebars/Costbars Supply and Demand Grids User Guide

Table of Contents

- [Overview](#)
 - [Purpose and Use Cases](#)
 - [The Five Summary Rows](#)
 - [View Modes and Configurations](#)
 - [Primary Roles: Named and Generic Resources](#)
-

Overview

The Supply vs Demand grid is a strategic resource management tool that provides visibility into resource capacity planning and allocation across projects. It enables resource managers and project managers to identify capacity gaps, prevent over-allocation, and make informed decisions about resource assignments and hiring needs.  Supply vs. Demand Grid

Purpose and Use Cases

What Are Supply and Demand Grids?

Supply represents the available capacity of resources in your organization - the total FTE (Full-Time Equivalent) or hours that resources can work based on their employment contracts, calendars, and availability.

Demand represents the actual resource requirements from project allocations - how much capacity is being consumed by tasks and projects that resources are assigned to.

Why Do We Need These Grids?

The gap between supply and demand reveals critical resource management issues:

- **Over-allocation:** When demand exceeds supply, resources are over-committed and projects are at risk
- **Under-utilization:** When supply exceeds demand, you have unused capacity that could be deployed
- **Future capacity planning:** Identifying when you'll need to hire, train, or contract additional resources
- **Portfolio decisions:** Understanding if you have capacity to take on new projects

Who Uses These Grids?

Resource Managers use these grids to:

- Balance workloads across the organization
- Identify hiring needs 3-6 months in advance
- Resolve resource conflicts and over-allocations
- Optimize resource utilization rates

Project Managers use these grids to:

- Understand if sufficient resources are available for their projects

- Plan project schedules based on resource availability
- Request additional resources when gaps are identified
- Adjust project timelines when resources are over-committed

Portfolio Managers use these grids to:

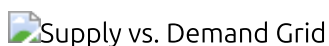
- Assess organizational capacity for new work
- Make go/no-go decisions on project proposals
- Prioritize projects based on resource constraints
- Plan strategic initiatives around resource availability

Executive Leadership uses these grids to:

- Make hiring and budgeting decisions
- Understand organizational capacity at a high level
- Evaluate departmental utilization and productivity
- Align business strategy with resource capacity

The Five Summary Rows

At the top of the Resource Demand Grids, five critical rows provide aggregated metrics:



1. Supply Target (FTE) (First Row)

What it shows: Total available resource capacity across all Generic Resources

Data source:

- Sourced from the **tbResources** store (resource pool)
- Fields:??
- Each Role has monthly FTE availability defined automatically calculated using tbResQuantity

Example: ??

Timing considerations:

- This allows forward planning for known hiring or departures

2. Supply Pool (FTE) (Second Row)

What it shows: Total available resource capacity across all resources in the pool

Data source:

- Sourced from the **tbResources** store (resource pool)
- Fields: **tbResMonth1** through **tbResMonth24**
- Each resource has monthly FTE availability defined (typically 1.0 for full-time)
- Sums ALL resources regardless of the groupBy selection

Example: If you have 6 full-time resources (1.0 FTE each), the supply shows 6.0 FTE per month

Timing considerations:

- Resource start dates (**tbResStart**) are factored in - new hires only contribute from their start month
- Resource finish dates (**tbResFinish**) are factored in - departing resources stop contributing
- This allows forward planning for known hiring or departures

3. Variance (ST - SP) (Third Row)

What it shows: ???

4. Demand (FTE) Row - Green Background

What it shows: Total resource requirements from all project allocations

Data source:

- Calculated from the **tbResCalcsUsage** store
- Generated by aggregating task allocations across all projects
- Fields: **tbResCalcMonth1** through **tbResCalcMonth32**
- Respects the **groupBy** filter - only sums demand for the filtered resources/projects/roles

Calculation process:

1. Task allocations are split into monthly buckets (via **splitAllocHoursMonthly**)
2. Each task's total work is distributed across its duration
3. Work hours per day are calculated: **totalWork / numberOfWorkDays**
4. Monthly hours are summed for each resource
5. Converted to FTE: **monthlyHours / (8 hours/day × 20 work days/month)**

5. Variance (S-D) Row - Yellow Background

What it shows: The difference between Supply and Demand (Supply minus Demand)

Color coding:

- **Green text** (positive variance): You have excess capacity - resources are under-utilized
- **Red text** (negative variance): You have a capacity deficit - resources are over-allocated

Strategic interpretation:

- Small negative variance (< 0.2 FTE): Minor over-allocation, manageable with overtime
- Large negative variance (> 0.5 FTE): Serious capacity problem, need to hire or delay projects
- Consistent positive variance: Opportunity to take on more work or reduce headcount

6. Month Header Row - Gray Background

What it shows: The calendar months for each column

Format: "MMM YYYY" (e.g., "Sep 2025", "Oct 2025")

Start date: Begins with the **apStatusDate** from the Admin Panel configuration

- This is typically set to the current month or the project baseline date
- Allows historical analysis by setting status date in the past
- Enables future planning by projecting 24+ months forward

7. Resource/Project/Role Header Row - White Background

What it shows: The first column header, which changes based on groupBy:

- "Resource ID/Owner" when grouping by ResourceName
- "Project Name" when grouping by Project
- "Role" when grouping by Role

View Modes and Configurations

FTE vs Hours Toggle

FTE (Full-Time Equivalent):

- Standard view for strategic planning
- 1.0 FTE = one full-time resource for one month
- Easier to understand capacity at a glance
- Recommended for executive reporting and high-level planning

Hours:

- Detailed view for tactical planning
- Shows actual work hours (e.g., 160 hours/month for 1.0 FTE)
- Useful for detailed project scheduling
- Helps with billing and time-tracking analysis

Conversion formula:

$$\text{FTE} = \text{Hours} / (\text{Calendar hours per day} \times \text{Working days per period})$$

For monthly:

$$\text{FTE} = \text{Hours} / (8 \text{ hours/day} \times 20 \text{ days/month}) = \text{Hours} / 160$$

For weekly:

$$\text{FTE} = \text{Hours} / (8 \text{ hours/day} \times 5 \text{ days/week}) = \text{Hours} / 40$$

Note: Supply totals are always calculated in FTE from the resource pool, then converted to hours if needed. The conversion uses: $\text{Hours} = \text{FTE} \times 8 \times 21$ (assuming 21 working days per month).

Weekly vs Monthly Toggle

Monthly View:

- Displays resource capacity and demand by calendar month
- Better for long-term strategic planning (6-24 months out)
- Smooths out weekly variations
- Recommended for portfolio planning and hiring decisions
- Supply and demand totals are shown

Weekly View:

- Displays resource capacity and demand by work week (Monday-Friday)
- Better for short-term tactical planning (1-3 months out)
- Shows week-to-week fluctuations
- Useful for identifying specific bottleneck weeks
- Currently does NOT show supply/demand totals (demand data only)

Why monthly is preferred for supply/demand analysis:

- Resource supply data is typically managed monthly (hiring, contracts)
- Weekly granularity creates noise in capacity planning
- Monthly aligns with financial planning and reporting cycles

Group By Options

The groupBy selection controls how demand rows are organized, but does NOT affect supply totals.

Group By: Project

Use case: Understanding capacity allocated to each project

What you see:

- One row per project showing demand
- Supply row shows total capacity across ALL resources
- Variance shows if organization has capacity for all projects combined

Strategic questions answered:

- Which projects are consuming the most resources?
- Do we have enough capacity for all active projects?
- Can we take on a new project without hiring?

Example scenario: You have 3 projects (CRM Migration, ERP Upgrade, Website Redesign). The grid shows how much FTE each project demands per month, and whether your total resource pool can support all three simultaneously.

Group By: Resource Name

Use case: Monitoring individual resource utilization

What you see:

- One row per named resource (e.g., "Joe Invent", "Ally HR")
- Each row shows that resource's project allocations
- Supply row still shows organizational total (all resources)

Strategic questions answered:

- Which individuals are over-allocated?
- Who has available capacity for new work?
- Are workloads balanced across the team?

Example scenario: Joe Invent shows 1.2 FTE demand in March (over-allocated by 20%), while Julia Finance shows 0.6 FTE (has 40% available capacity). You can rebalance by moving some of Joe's work to Julia.

Group By: Role

Use case: Understanding capacity by job function or skillset

What you see:

- One row per role (e.g., "Developer", "Engineer", "Finance SME")
- Shows aggregated demand for all resources in that role
- Critical for skills-based capacity planning

Strategic questions answered:

- Do we have enough developers vs. testers?
- Which role is the bottleneck in our organization?
- Should we hire for a specific skillset?

Example scenario: Your "Developer" role shows 3.5 FTE demand but you only have 3.0 FTE supply of developers. You need to hire an additional 0.5 FTE developer (potentially a contractor) or delay some development work.

Primary Roles: Named and Generic Resources

The system supports two types of resources that work together for flexible capacity planning:

Named Resources (Humans)

What they are: Individual people with specific identities

Characteristics:

- **tbResLabourType:** "Human"
- Unique identifier: **tbResID** (e.g., "700", "701", "702")
- Named individuals: "Joe Invent", "Ally HR", "Julia Finance"
- Assigned a Primary Role: "R&D", "HR SME", "Finance SME"
- Quantity is typically 1

Why we need them:

- Track specific people's workloads and allocations
- Manage vacation, training, and individual availability
- Support performance management and utilization reporting
- Enable skills-based assignment to tasks

Primary Role assignment: Each named resource is assigned one Primary Role (e.g., Joe Invent has Primary Role = "R&D"). This:

- Categorizes the resource by function for reporting
- Enables role-based filtering and grouping
- Supports career development and succession planning
- Allows comparison of individuals within the same role

Generic Resources

What they are: Placeholder resources representing a pool of similar workers

Characteristics:

- **tbResLabourType:** "Generic"
- Role-based identifier: **tbResID** (e.g., "800", "818", "830")
- Generic names matching the role: "R&D", "HR SME", "Finance SME"
- Assigned a Primary Role that matches the resource name
- Quantity can be > 1 (e.g., "2" means 2 FTE of this role)

Why we need them:

- Early project planning before specific people are assigned
- Representing contractor pools or offshore teams
- Capacity planning when you don't know who specifically will do the work
- Future hiring - you know you need "3 developers" but haven't hired them yet

How quantity works: If Generic Resource "Developer" has **tbResQuantity:** "2" and **tbResMonth1:** "2.00", this means:

- You have 2.0 FTE of developer capacity available
- This could represent 2 full-time developers
- Or 4 half-time developers
- Or any combination totaling 2.0 FTE

Why Have Both Named and Generic?

This dual approach supports the full project lifecycle:

Early Planning Phase:

- Use generic resources ("we need 2 developers and 1 tester")
- Estimate capacity needs without knowing specific people
- Justify hiring or budget requests

Assignment Phase:

- Replace generic allocations with named resources
- Assign Joe, Chris, and John to specific tasks
- Track actual people doing the work

Capacity Planning:

- Generic resources fill the gap between demand and available named resources
- If named resources show 4.5 FTE demand but only 4.0 FTE supply, the 0.5 FTE gap might be filled by generic "Developer" resource (representing a contractor)

Reporting Flexibility:

- Group by Resource Name: See named resources only
- Group by Role: See combined named + generic resources by function

- Supply totals: Includes both named and generic for total capacity

Primary Role in Generic Resources

Generic resources have their Primary Role set to match their name:

- Resource Name: "Developer" → Primary Role: "Developer"
- Resource Name: "R&D" → Primary Role: "R&D"

This ensures that when grouping by Role, both named resources with that role AND the generic resource pool for that role are aggregated together, giving a complete picture of capacity for that function.

Part 9 — Text Notifications

Text Notifications User Guide

Timebars Ltd. products — Agilebars, Timebars, and Costbars — include an automated notification system that alerts portfolio and project managers when projects require attention. Notifications are delivered to your phone via the **Pushover** app. An optional SMS escalation channel (Costbars only) is available via Twilio for executive-level alerts.

The system evaluates your in-progress projects automatically every four hours and sends a push notification when your configured thresholds are met. You can also trigger a notification at any time from the admin panel — for example, immediately after publishing a new dataset.

Which products support notifications? All three products — Agilebars (AB), Timebars (TB), and Costbars (CB) — support push notifications. The number of conditions evaluated and the available threshold settings differ by product. Costbars includes the full five-scenario evaluation and optional SMS escalation. See [How the System Evaluates Projects](#) for details.

This document has two parts. Part 1 is the user guide — setting up Pushover, creating a notification configuration, choosing thresholds and testing them. **Part 2 — Technical Reference** covers the server side: environment variables, authentication, API security, cron scheduling and Strapi integration. Part 2 is for system administrators and developers; a portfolio manager never needs it.

Table of Contents

Part 1 — Using Notifications

1. [Overview](#)
2. [Preparing Your Projects for Notifications](#)
3. [Setting Up Pushover on Your Phone](#)
4. [Creating a Notification Configuration](#)
5. [How the System Evaluates Projects](#)
 - [Agilebars — 2-Condition Check](#)
 - [Timebars — 3-Condition Check](#)
 - [Costbars — 5-Scenario Check](#)
 - [The Chain-of-Values Rule](#)

6. [When Notifications Are Sent](#)
 - [Automated Schedule](#)
 - [On-Publish Notifications](#)
 - [Timing Gates](#)
7. [Escalation — SMS for Costbars](#)
8. [Testing Your Configuration](#)
9. [Managing Notifications](#)
10. [Troubleshooting](#)
11. [Quick Reference — Fields by Product](#)

Part 2 — Technical Reference

12. [Environment Variables](#)
13. [Data Sources — How Projects Are Fetched](#)
14. [Authentication Flow](#)
15. [Route Protection](#)
16. [API Route Security](#)
17. [Strapi Collections](#)
18. [Strapi User Fields Required](#)
19. [Automated Cron Job](#)
20. [On-Publish Endpoint](#)
21. [Manual Test Button — What It Actually Does](#)
22. [Low-Level Connectivity Test](#)

Part 1 — Using Notifications

Overview

The notification system works in three stages:

1. **Evaluate** — The system reads your in-progress projects and checks them against the thresholds you have configured.
2. **Gate** — Before sending, the system checks timing rules: business hours, quiet hours, cooldown period, and daily send limits.
3. **Send** — If at least one project triggers an alert and all timing gates pass, a push notification is sent to the manager's phone via Pushover.

You manage all of this from the **Notifications** page in the admin panel at </admin/notifications>.

Preparing Your Projects for Notifications

Before notifications can fire, two things must be true about each project record, and the relevant metadata fields must be populated. The notification engine silently skips projects that do not meet the prerequisites.

Prerequisites — every product

Requirement	Field	Value
Must be a project-type bar	<code>tbType</code>	<code>Project</code>

Requirement	Field	Value
Must be actively in progress	tbMDStatus	In progress (exact match, case-sensitive)

Tasks, milestones, allocations, and projects with any other status (e.g., Complete, On Hold) are not evaluated.

Fields the engine reads — by product

The engine reads these metadata fields from your project records. Empty fields are not neutral — each has a default behaviour that may trigger an alert.

Agilebars — 2 fields:

Field	Form label	Pick-list values	Empty treated as
tbMDHealthOverall	Health Overall	Red, Yellow, Green	Red (triggers)
tbMDSeniorLevelCommitment	Senior Level Commitment	Full, Strong, Moderate, Limited, Not Clear, Not Assessed	Moderate (triggers)

Timebars — 3 fields (includes the 2 above, plus):

Field	Form label	Pick-list values	Empty treated as
tbMDRiskVsSizeAndComplexity	Risk Vs Size And Complexity	0–10 Very Small and Simple ... 91–100 Extremely Complex and Risky	Score 51+ (triggers)

Costbars — 5 scenarios (includes the 3 above, plus):

Field	Form label	Notes	Used in scenario
tbMDPriorityStrategic	Strategic Priority	Numeric score — higher = more critical	1, 2, 4, 5
tbCost	Actual Cost	Current actual cost spend	4 (Budget Overrun)
tbBudgetCost	Budget Cost	Approved budget	4 (Budget Overrun)
tbWork	Actual Work (hours)	Current actual hours	5 (Work Overrun)
blWork	Baseline Work (hours)	Baseline hours snapshot	5 (Work Overrun)
tbMDEscalationLevel	Escalation Level	Set to Executives to trigger SMS escalation	SMS only

Tip — force a test result: Set Health Overall = Green, Senior Level Commitment = Not Assessed, and (for TB/CB) Risk = 0–10 Very Small and Simple on a test project. These are the most permissive values and will match against almost any notification threshold, making it easy to verify the system end-to-end.

Setting Up Pushover on Your Phone

Pushover is a third-party push notification service used to deliver alerts to your phone. It has a one-time cost of approximately \$5 USD and includes a free 30-day trial.

Step 1 — Install and create an account

1. Install **Pushover** from the App Store (iOS) or Google Play (Android).
2. Create an account at pushover.net or directly within the app.

Step 2 — Get your User Key

After logging in to pushover.net, your **User Key** is displayed on the home dashboard in the top section. Copy it — this is your **PUSHOVER_USER_KEY**.

Step 3 — Create an App Token

1. In pushover.net, go to **Your Applications** → **Create an Application / API Token**.
2. Give it a name such as **Timebars PPM**.
3. Copy the **API Token** — this is your **PUSHOVER_APP_TOKEN**.

Step 4 — Confirm the keys are in your environment

Your system administrator will have added these two values to the server's environment configuration file:

```
PUSHOVER_APP_TOKEN=    ← From pushover.net → Your Applications → API Token
PUSHOVER_USER_KEY=    ← From pushover.net → Home dashboard, top section
```

Once these are in place, notifications can be sent. You can verify the connection is working from the [Testing](#) section before creating your first configuration.

Creating a Notification Configuration

Navigate to [/admin/notifications](#) and click **Add Notification** (or the **+** button). The form has four tabs.

Tab 1 — Basic

Field	Description
Manager Name	The name of the person who will receive this notification
Phone	Manager's phone number (used for SMS escalation if enabled)
Email	Manager's email address
Timezone	The manager's local timezone — used for business hours and quiet hours calculations

Field	Description
Order	Select the product licence order this config belongs to
Active	Toggle to ON to enable this notification config

Tab 2 — Preferences

Field	Description
Primary Channel	Set to push to use Pushover. (SMS is the escalation channel, not the primary channel.)
Business Hours Only	If enabled, notifications are only sent between 09:00 and 17:00 in the manager's timezone
Send on Weekends	If disabled, notifications are skipped on Saturday and Sunday
Quiet Hours Start / End	Times during which notifications are suppressed (e.g., 22:00–07:00)
Cooldown (hours)	Minimum hours between notifications for this manager. Set to a low value (e.g., 1 hour) during initial testing.
Daily Limit	Maximum number of notifications per day for this manager

Tab 3 — Thresholds

The fields shown on this tab depend on your product licence. See [How the System Evaluates Projects](#) for a full explanation of each threshold.

Agilebars (AB) — 2 fields:

Field	Options	Default
Overall Health Threshold	Green, Yellow, Red	Red
Senior Level Commitment Threshold	Full, Strong, Moderate, Limited, Not Clear, Not Assessed	Moderate

Timebars (TB) — 3 fields:

Field	Options	Default
Overall Health Threshold	Green, Yellow, Red	Red
Senior Level Commitment Threshold	Full, Strong, Moderate, Limited, Not Clear, Not Assessed	Moderate
Risk / Size / Complexity Threshold	0–10 Very Small and Simple ... 91–100 Extremely Complex and Risky	51–60 Significant Complexity

Costbars (CB) — 5 fields:

Field	Options	Default
Overall Health Threshold	Green, Yellow, Red	Red

Field	Options	Default
Senior Level Commitment Threshold	Full, Strong, Moderate, Limited, Not Clear, Not Assessed	Moderate
Risk / Size / Complexity Threshold	0–10 Very Small and Simple ... 91–100 Extremely Complex and Risky	51–60 Significant Complexity
Strategic Value Threshold	Numeric (e.g., 50)	50
Budget Overrun % Threshold	Numeric percentage (e.g., 10)	10
Work Hours Overrun % Threshold	Numeric percentage (e.g., 20)	20

Choosing your threshold values

Each threshold answers "how bad does this have to get before I want to hear about it?" The defaults are a reasonable starting point; these are the questions each one is really asking:

Threshold	What it watches	A worked example
Overall Health	the project's rolled-up health indicator	set to Red to hear only about projects already in trouble; Yellow to catch them earlier
Senior Level Commitment	the recorded level of executive support	set to Moderate to be told when sponsorship is drifting below solid backing
Risk / Size / Complexity	the project's risk and complexity score	set to 51–60 to ignore small, simple projects and hear only about substantial ones
Strategic Value	the strategic score from portfolio analysis	set to 50 so a high-value project in difficulty reaches you ahead of a low-value one
Budget Overrun %	forecast cost against budget	set to 10 to be told when a project passes 10% over
Work Hours Overrun %	forecast hours against budgeted hours	set to 20 to allow more tolerance on effort than on money

Remember the [chain-of-values rule](#): a threshold is not an exact match. Setting Overall Health to **Yellow** catches Yellow **and** Red, because the check climbs toward the most critical end of the scale.

Start loose, then tighten. Set generous thresholds first, watch what actually arrives over a reporting cycle or two, and tighten from there. Thresholds that are too tight produce silence, which reads exactly like a system that is not working; thresholds that are too loose produce noise, and a manager who stops reading the alerts. Both failures are quiet, which is why the dry run below matters.

Tab 4 — Escalation

Escalation is a Costbars-only feature that sends an SMS via Twilio in addition to the Pushover notification. See [Escalation — SMS for Costbars](#). For Agilebars and Timebars, leave escalation off.

How the System Evaluates Projects

The system reads all projects in your published dataset where **Status = In progress** and evaluates each one against your notification configuration. The logic applied depends on your product licence.

The Chain-of-Values Rule

All categorical threshold comparisons use a **chain-of-values** model. Rather than checking for an exact match, the system checks whether a project's value is **at or beyond the threshold you set**, climbing toward the most critical end of the scale.

Overall Health — climbs toward Red (worst):

Threshold you set	Fires for these project health values
Green	Green, Yellow, Red
Yellow	Yellow, Red
Red	Red only

An empty health field on a project is treated as **Red**.

Senior Level Commitment — climbs toward Full (strongest):

Threshold you set	Fires for these project commitment values
Not Assessed	Not Assessed, Not Clear, Limited, Moderate, Strong, Full
Not Clear	Not Clear, Limited, Moderate, Strong, Full
Limited	Limited, Moderate, Strong, Full
Moderate	Moderate, Strong, Full
Strong	Strong, Full
Full	Full only

An empty commitment field on a project is treated as **Moderate**.

Risk / Size / Complexity — climbs toward higher complexity:

Each label maps to a numeric score. The system fires when a project's score is **greater than or equal to** the score of your configured threshold.

Label	Score
Not Assessed	0
0–10 Very Small and Simple	10
11–20 Small and Straightforward	20
21–30 Medium with Some Complexity	30
31–40 Large with Moderate Complexity	40

Label	Score
41–50 Complex but Manageable	50
51–60 Significant Complexity	60
61–70 Large and Complex	70
71–80 Very Large and Complex	80
81–90 Highly Complex and Risky	90
91–100 Extremely Complex and Risky	100

An empty risk field on a project is treated as triggering (score 51+).

Agilebars — 2-Condition Check

A notification fires when **both** conditions are true for the same in-progress project:

1. The project's **Overall Health** is at or beyond your configured health threshold (climbing toward Red).
2. The project's **Senior Level Commitment** is at or above your configured commitment threshold (climbing toward Full).

Example: You configure Health = Yellow and Commitment = Moderate. The system will send a notification for any in-progress project where health is Yellow **or** Red, **and** commitment is Moderate, Strong, **or** Full — both must be true simultaneously.

Timebars — 3-Condition Check

A notification fires when **all three** conditions are true for the same in-progress project:

1. The project's **Overall Health** is at or beyond your configured health threshold.
2. The project's **Senior Level Commitment** is at or above your configured commitment threshold.
3. The project's **Risk / Size / Complexity** score is greater than or equal to the score of your configured risk threshold.

Example: You configure Health = Yellow, Commitment = Moderate, Risk = 51–60 Significant Complexity (score 60). A notification fires for any in-progress project where health is Yellow or Red, **and** commitment is Moderate or higher, **and** risk score is ≥ 60 .

Costbars — 5-Scenario Check

Costbars evaluates five independent scenarios against every in-progress project. A notification fires if **any one** scenario triggers.

#	Scenario	Fires when
1	Critical High-Value Project	Health is at or beyond your health threshold AND strategic value exceeds your strategic value threshold

#	Scenario	Fires when
2	Missing Strategic Data	Health is at or beyond threshold AND strategic value field is blank AND risk score $\geq$ your risk threshold
3	Missing Health Status	Health field is blank AND commitment is at or above your commitment threshold AND risk score $\geq$ your risk threshold
4	Budget Overrun	Actual cost exceeds budget by more than your budget overrun % AND the project is high-value (strategic value $\geq$ threshold)
5	Work Hours Overrun	Actual hours exceed baseline hours by more than your work overrun % AND the project is high-value

The notification message you receive identifies which scenario(s) triggered and which project(s) were flagged.

SMS escalation note: If escalation is enabled on your configuration and a flagged project has its **Escalation Level** field set to **Executives**, an SMS is sent in addition to the Pushover notification. See [Escalation — SMS for Costbars](#).

When Notifications Are Sent

Automated Schedule

The system runs an automated check every **four hours**, around the clock. Each run evaluates all active notification configurations against your published dataset. If a configuration's timing gates pass and projects are flagged, a notification is sent.

On-Publish Notifications

When you publish a dataset from any of the three client apps, the system can trigger an **immediate** notification check — bypassing the four-hour schedule. This is useful for ensuring managers are alerted as soon as new data is available.

On-publish notifications bypass business hours and quiet hours checks. Cooldown and daily limit rules still apply to prevent duplicate sends on rapid re-publishes.

Note: On-publish notifications require no additional configuration. They are triggered automatically when you publish, provided you have an active notification configuration set up.

Timing Gates

Before any notification is sent during an automated run, the system checks each manager's configuration against the following gates in order:

Gate	What it checks
Active	The notification config must have Active set to ON
Business hours	If enabled, the current time must be between 09:00 and 17:00 in the manager's timezone

Gate	What it checks
Weekend block	If weekends are disabled, the current day must be Monday–Friday
Quiet hours	The current time must be outside the configured quiet hours window
Cooldown	Sufficient hours must have passed since the last notification was sent to this manager
Daily limit	The manager must not have reached their configured maximum notifications for the current day

If any gate fails, the notification is skipped for that run. Manual sends from the admin panel bypass all timing gates and send immediately.

Escalation — SMS for Costbars

Costbars configurations include an optional SMS escalation channel. When enabled, any flagged project that has its **Escalation Level** metadata field set to **Executives** will trigger an SMS message in addition to the standard Pushover notification.

To configure escalation:

1. Open the notification configuration form and go to **Tab 4 — Escalation**.
2. Enable the escalation toggle.
3. Enter the **escalation phone number** to receive the SMS.

Important: SMS escalation only fires when the flagged project's `tbMDEscalationLevel` field is set to **Executives**. Projects with other escalation level values receive the Pushover notification only. Agilebars and Timebars projects always receive Pushover only — the escalation level field is not used in those products.

Testing Your Configuration

Manual Dry Run

On the `/admin/notifications` page, scroll to the **Notification Testing & Message Generation** section and click **Run Manual Test**. This runs the full evaluation logic in your browser using your current project data and notification configurations. It shows:

- How many in-progress projects were found
- Which projects were flagged and which scenario or condition triggered
- The full notification message that would be sent
- Debug logs for each condition checked (click **Show Debug Logs**)

No notification is sent during a dry run. Data is not written to the server.

Sending a Test Notification

If projects are flagged during the dry run, an **AdminNotificationSender** section appears below the results. Click **Send to [Manager Name]** to fire the actual Pushover notification immediately. This bypasses all timing gates

and is intended for verifying that your Pushover keys and phone setup are working correctly.

Tip: If you want to force a flagged result during testing, temporarily set permissive threshold values — for example, Health = Green (fires for all health values), Commitment = Not Assessed (fires for all), and a low risk or strategic value threshold. Remember to restore your intended thresholds after testing.

Low-Level API Test

To confirm that Pushover itself is reachable (before configuring anything in the admin UI), your system administrator can run a direct test from the server:

```
curl -s \
  --form-string "token=YOUR_PUSHOVER_APP_TOKEN" \
  --form-string "user=YOUR_PUSHOVER_USER_KEY" \
  --form-string "message=Test from Timebars!" \
  --form-string "title=PPM Test" \
  https://api.pushover.net/1/messages.json
```

A successful response looks like: `{"status":1,"request":"..."}` and a push notification arrives on the phone within seconds.

Best Practice

- **Run a dry run every time you change a threshold.** It costs nothing and tells you immediately whether the change did what you expected.
- **Start with looser thresholds** and tighten once you have seen real behaviour over a reporting cycle.
- **Keep a short cooldown while testing** (an hour), then raise it for live use so a manager is not alerted repeatedly about the same project.
- **Reserve executive escalation for genuinely high-impact projects.** An escalation channel that fires often stops being treated as an escalation.
- **Write down the thresholds you chose and why.** The numbers are easy to change and easy to forget, and the next person to look at them will want the reasoning, not just the values.

Managing Notifications

All notification configurations for your licences are listed on [/admin/notifications](#). From this page you can:

- **Add** a new notification configuration with the **+** button
- **Edit** an existing configuration by clicking on it
- **Toggle Active** on or off directly from the list to temporarily pause a configuration without deleting it
- **View the threshold summary** for each configuration at a glance — the list shows the key configured values by product

Multiple configurations can exist for the same order (for example, one per portfolio manager). Each fires independently based on its own thresholds and timing settings.

Troubleshooting

Symptom	What to check
No push notification received	Run the curl test in Low-Level API Test to verify your Pushover token and user key are valid
Pushover returns <code>status:0</code>	Your <code>PUSHOVER_APP_TOKEN</code> or <code>PUSHOVER_USER_KEY</code> is incorrect — check for typos or placeholder values
Dry run shows "No flagged projects"	Lower your thresholds temporarily (e.g., Health = Green) to verify the evaluator runs at all. Also confirm your projects have <code>Status = In progress</code> .
Notification was expected but not sent during automated run	Check Active = ON, verify the manager's timezone is correct, and review business hours / quiet hours / cooldown / daily limit settings
SMS not received (Costbars)	Confirm escalation is enabled on the config and that the flagged project's <code>Escalation Level</code> field is set to <code>Executives</code>
On-publish notification not received	Confirm cooldown has elapsed since the last send — on-publish sends still respect the cooldown setting
Admin panel shows a 500 error on the notifications page	Contact your system administrator — this typically indicates a missing server-side environment variable

Quick Reference — Fields by Product

Field	Agilebars (AB)	Timebars (TB)	Costbars (CB)
Overall Health Threshold	✓	✓	✓
Senior Level Commitment Threshold	✓	✓	✓
Risk / Size / Complexity Threshold	—	✓	✓
Strategic Value Threshold	—	—	✓
Budget Overrun % Threshold	—	—	✓
Work Hours Overrun % Threshold	—	—	✓
SMS Escalation	—	—	✓
Conditions evaluated	2	3	5 scenarios
Automated check frequency	Every 4 hours	Every 4 hours	Every 4 hours
On-publish notification	✓	✓	✓

Part 2 — Technical Reference

This part covers server-side configuration, authentication, API security, cron scheduling and Strapi integration. It is intended for **system administrators and developers**. Nothing in Part 1 requires it.

Environment Variables

The following env vars must be set on the server:

```
PUSHOVER_APP_TOKEN=      # From pushover.net → Your Applications → API Token
PUSHOVER_USER_KEY=       # From pushover.net → Home dashboard, top section
NOTIFICATION_STRAPI_KEY # Used by the automated endpoint to fetch configs and
                        # project data from Strapi (no user session needed – designed for cron use)
SYSTEM_ADMIN_EMAIL=      # The email whose active pubset is used to fetch
                        # project data for automated runs
```

Note: `PUSHOVER_API_URL` and `ORTHSURL` env vars are not used. The code hardcodes `https://api.pushover.net/1/messages.json` directly in `pushover.js` and `pushover/route.js`.

Data Sources — How Projects Are Fetched

The automated notification route always fetches from the **timebars Strapi collection** (the pubset), filtering to the `SYSTEM_ADMIN_EMAIL`'s active pubset. It reads `tbmdjoined` rows and filters down to:

- `tbType === "Project"`
- `tbMDStatus === "In progress"`

Only those rows enter the evaluation engine. Projects that are not of type Project, or whose status is not exactly `"In progress"`, are silently skipped.

The notification configurations are fetched from the **notifications** Strapi collection, each linked to an **order** in the **orders** collection. The order carries the product licence (e.g., Agilebars Tier 2, Costbars Tier 1), and the system reads the `product_code` from that order to decide which evaluation logic to apply.

Product code prefix convention

Prefix	Product	Scenarios evaluated
CB*	Costbars	5 scenarios
TB*	Timebars	3-condition check
AB*	Agilebars	2-condition check

Authentication Flow

When a user signs in, NextAuth runs a JWT callback (`app/auth/auth.js`) that performs two things beyond standard login:

1. Stores the Strapi JWT in the session (`session.jwt`) — used for all subsequent Strapi API calls.
2. Makes an extra call to Strapi to fetch two RBAC fields from the user record:
 - `primary_role` — determines admin access
 - `customer_id` — organisation identifier for data scoping

These are then available on `session.user` throughout the app.

Route Protection

Route	Guard
<code>/admin</code>	Must be authenticated AND <code>primary_role === 'admin'</code> or email is the designated admin email
<code>/admin/notifications</code>	Must be authenticated only — no role check

`/admin/notifications` is reachable by any logged-in user. The main `/admin` page is the harder gate.

API Route Security

Route	Protection
<code>POST /api/notifications/automated</code>	System-only — uses <code>NOTIFICATION_STRAPI_KEY</code> env var. No user session required. Designed for cron.
<code>GET/POST /api/notifications/pushover</code>	Requires <code>Authorization: Bearer <token></code> header
<code>GET/POST /api/notifications/twilio</code>	No auth check — open if reachable
<code>POST /api/notifications/on-publish</code>	Validated via Strapi JWT token

Note: The Twilio route validates input but does not verify the caller's identity. As long as it is only called server-side from `AdminNotificationSender` this is acceptable in practice, but it could be called directly by anyone who knows the URL. Consider adding a bearer token check if the route becomes externally reachable.

Strapi Collections

orders — stores purchases, each linked to a product and a user. Filtered by the `owner` field (email address) to scope to each user's licences.

When `/admin/notifications` loads, it calls `getMyOrders(email, jwt)` which hits:

```
GET /api/orders?populate[0]=product&populate[1]=user
&filters[owner][$eq]={your-email}
```

Fields displayed from each order:

UI label	Strapi field
Order name	<code>order.name</code>
Product name / code	<code>order.product.name</code> / <code>order.product.product_code</code>
Cost	<code>order.total</code>

UI label	Strapi field
Expiration	<code>order.expires_on</code>
Active status	<code>order.active_status</code>

notifications — stores per-manager notification configurations, each linked back to an order by `order.id`. Multiple notification configs can exist per order (one per manager). Fields cover all thresholds, channels, quiet hours, cooldown, and daily limit settings.

Strapi User Fields Required

Each user record in the Strapi `users` collection needs two custom fields:

Field	Type	Purpose
<code>primary_role</code>	String	Set to 'admin' to grant /admin access
<code>customer_id</code>	String / Number	Organisation identifier for data scoping

If `primary_role` is null or missing, the user passes the `/admin/notifications` check (authenticated only) but is redirected away from `/admin` itself.

Automated Cron Job

The automated notification chain runs as a bash cron job on the production server.

Chain of execution

```

cron job (server)
├─ runs automate_pushover_twillio.sh
│   └─ internal guard: exits if weekend or outside 09:00-18:00 server time
│       └─ POST https://www.timebars.com/api/notifications/automated
│           └─ fetches all active notification configs (via
NOTIFICATION_STRAPI_KEY)
│               └─ fetches all in-progress projects (via
fetchProjectDataFromPubset)
│                   └─ calls evaluateInProgressProjects() ← scenarios run here
│                       └─ for each flagged manager:
│                           checks timezone / quiet hours / cooldown / daily limit
│                           sends Pushover (+ SMS if escalation conditions met)
│                           updates last_notification_sent + notification_count_today in
Strapi

```

Script location and schedule

The script lives at `~/docker/tbwwwp/automate_pushover_twillio.sh` and is registered in root's crontab via `sudo crontab -e`.

Recommended schedule — every 4 hours, any day:

```
0 */4 * * * /home/jcox/docker/tbwwwp/automate_pushover_twillio.sh >>
/var/log/ppm-notifications.log 2>&1
```

Previous schedule — hourly, weekdays only:

```
0 6-23 * * 1-5 /home/jcox/docker/tbwwwp/automate_pushover_twillio.sh
```

Absolute path required: `sudo crontab -e` registers in root's crontab, so `~/` expands to `/root/` not `/home/jcox/`. Always use the full path.

Setup checklist

```
# Verify script exists and is executable
ls -la /home/jcox/docker/tbwwwp/automate_pushover_twillio.sh
chmod +x /home/jcox/docker/tbwwwp/automate_pushover_twillio.sh

# Create log file writable by root
sudo touch /var/log/ppm-notifications.log
sudo chmod 644 /var/log/ppm-notifications.log

# Verify cron registration
sudo crontab -l
```

On-Publish Endpoint

`POST /api/notifications/on-publish` triggers an immediate notification check when users publish a dataset from any of the three client apps.

Behaviour differences from automated runs

Behaviour	Automated cron	On-publish
Business hours check	Applied	Bypassed
Quiet hours check	Applied	Bypassed
Cooldown	Applied	Applied (prevents duplicate sends on rapid re-publish)
Daily limit	Applied	Applied
Authentication	<code>NOTIFICATION_STRAPI_KEY</code>	Strapi JWT token

Deduplication

Two-layer deduplication prevents duplicate sends:

1. **In-memory cache** (per user per day) — fast, survives normal runtime restarts
2. **Strapi `last_notification_sent` field** — persistent, survives full server restarts

Client integration

A reference integration snippet is at `app/admin/notifications/tbrunp-integration-snippet.js`. The design is fire-and-forget: notification failures never block or break the publish operation. Missing notification configs or no in-progress projects result in silent skips.

Manual Test Button — What It Actually Does

The **Run Manual Test** button on `/admin/notifications` calls `evaluateInProgressProjects()` **directly in the browser** using project data and notification configs that were loaded server-side when the page rendered. It does not make any HTTP request and does not write to the server.

- Results and debug logs appear in the UI (Show Debug Logs button) and in the browser console (F12 → Console).
 - The server log at `/var/log/ppm-notifications.log` is only written when the cron job fires the real API call.
 - The **Send to [Manager]** buttons in `AdminNotificationSender` call the Pushover/Twilio routes directly and bypass the cron, timing gates, and cooldown/daily-limit counters entirely. They are intended for verifying that keys and phone setup work.
-

Low-Level Connectivity Test

Before configuring anything in the admin UI, confirm Pushover is reachable from the server:

```
curl -s \  
  --form-string "token=YOUR_PUSHOVER_APP_TOKEN" \  
  --form-string "user=YOUR_PUSHOVER_USER_KEY" \  
  --form-string "message=Test from Timebars!" \  
  --form-string "title=PPM Test" \  
  https://api.pushover.net/1/messages.json
```

A successful response: `{"status":1,"request":"..."}` — push arrives on the phone within seconds.

You can also test the app's own API route once the dev server is running:

```
curl -X POST http://localhost:3001/api/notifications/pushover \  
  -H "Content-Type: application/json" \  
  -d '{"message":"Hello from Timebars","title":"PPM Test","priority":1}'
```

Part 10 — Cloud Publishing

Cloud Publishing and Bulk Operations Guide

Applies to: Agilebars, Timebars, Costbars **Version:** 2.2 **Last Updated:** 2026-09-10

Table of Contents

1. [Overview](#)
 2. [Account Registration & Subscription](#)
 3. [Logging In to Timebars Cloud](#)
 4. [Publishing Page Interface](#)
 5. [Viewing Pubsets](#)
 6. [Publishing to the Cloud](#)
 7. [Downloading from Cloud to Local \(Re-Hydrate\)](#)
 8. [Clearing a Published Dataset](#)
 9. [License and Activation](#)
 10. [Cross-Device Synchronization](#)
 11. [OpenProject Sync](#)
 12. [Bulk Update](#)
 13. [Re-Publish with Scheduling Engine](#)
 14. [Support Resources](#)
 15. [Quick Reference](#)
-

1. Overview

Cloud Publishing is an **optional feature** that allows you to securely upload your project data from the client applications (Agilebars, Timebars, or Costbars) to the Timebars Cloud. The Cloud Publishing page is the central hub for managing data between your local browser (IndexedDB) and the Timebars Cloud. From here you can:

- **Publish** your local project data to the cloud for sharing and dashboards
- **Download** a published dataset back into your local browser
- **Clear** published data from the cloud
- **Bulk Update** specific fields across multiple published datasets at once
- **Re-Publish** datasets with recalculated scheduling engine data
- **Sync** with OpenProject (external PM system)

Published datasets are called **Pubsets**. Each pubset is a complete snapshot of your project data stored in the cloud. The active pubset feeds your Personal Cloud Dashboard.

Key Benefits:

- **Cloud Dashboard Access:** View interactive reports and charts from any device
- **Cross-Device Sync:** Work on one device, continue on another
- **Data Backup:** Secure cloud storage for your project data
- **Team Collaboration:** Share data with stakeholders (requires individual licenses)
- **Real-Time Updates:** Publish changes instantly to the cloud

Important Notes:

- Publishing is entirely optional—client apps work fully offline
 - Your local data remains in the browser unless you choose to publish
 - Cloud Publishing requires an active subscription
 - Data is encrypted during transmission (HTTPS/TLS)
 - You must log in each session — credentials are not saved or persisted
-

2. Account Registration & Subscription

Create an Account

1. **Register:** Visit www.timebars.com/register, provide your email address and create a secure password, then click "Register"
2. **Verify Email:** Check your email for the verification link and click it to confirm your account

Choose a Subscription Plan

1. Navigate to www.timebars.com/pricing and review available subscription tiers
2. Select a plan based on your needs — you can purchase up to 10 PubSets depending on tier
3. Complete the credit card transaction; your license activates automatically
4. Log in at www.timebars.com/login to view subscription details and manage PubSets

Subscription Includes:

- Client application license (Agilebars, Timebars, or Costbars)
 - Cloud Publishing capabilities
 - Cloud Dashboard access
 - Number of PubSets based on tier
 - Technical support
-

3. Logging In to Timebars Cloud

Steps

1. Navigate to the publishing page from the main menu (**Main Menu > Publish**)
2. You will land on the **Login** tab
3. Enter your **Email** and **Password**
4. Click **Cloud Login!**

What Happens

- Your credentials are validated against the Timebars Cloud authentication service
- On success, a green message appears: *"Login successful! Your pubsets are loaded."*
- After 1 second the page switches to the **Pubsets** tab showing your datasets
- The Pubsets (Cards) and Pubsets (Table) tabs become visible

Note: The cloud login is a **secondary login** separate from your local app login, providing an additional security layer. Your data requires both local app login AND cloud login to access.

Important Notes

- You must log in each session — credentials are not saved or persisted
- No cookies or tokens are stored between sessions
- If you try to access other tabs before logging in, you will be redirected back to the Login tab

4. Publishing Page Interface

Tab Navigation

The publishing page has six tabs:

Tab	Visible	Purpose
Login	Always	Cloud authentication
Pubsets (Cards)	After login	Card-style view of your pubsets
Pubsets	After login	Table view with sorting and filtering
OpenProject	Always	Sync projects, bars and people with OpenProject (see section 11)
Bulk Update	Always	Bulk data operations across pubsets
Re-Publish	Always	Download, recalculate, and republish

Click any tab button to switch. The active tab is highlighted with a blue underline.

Top Navigation Bar

The blue navigation bar at the top of the publishing page provides quick actions:

Button	Action
Show License	Displays your current license tier, product code, and expiration date
Recalculate All	Runs all system calculations (hierarchy names, rollups) to ensure data accuracy before publishing
Dashboard	Opens your Personal Cloud Dashboard in a new browser tab
Purchase	Opens the Timebars pricing page to buy or upgrade a license
Activate	One-time setup to create your pubset containers (Tier-2 and Tier-3 only)
Sync Users	Admin-only: syncs resource pool data to portal user accounts
Close (X icon)	Closes the publishing page and returns to the main application

5. Viewing Pubsets

After logging in, your pubsets are displayed in two views:

Card View

- Shows each pubset as a card with its name and action buttons
- The **active pubset** (the one feeding your Cloud Dashboard) has a **blue background**
- Use this view for a quick visual overview

Table View

- Shows all pubsets in a sortable, searchable table
- Columns: Active status, Pubset Name, ID, Published Date, Published By, Customer ID, Actions
- Click column headers to sort; use the search box to filter by name, owner, or customer ID
- The active pubset is highlighted in blue
- Action buttons for each pubset: **Download, Publish, Clear**

6. Publishing to the Cloud

Publishing takes all your local project data and uploads it to a pubset in the cloud.

Prerequisites

- Active Timebars Ltd. subscription
- Client application installed and running
- Project data created in the client application
- Internet connection
- Logged into the client app locally (gold icon, top right) AND the cloud (see Section 3)

Steps

1. Go to the **Pubsets** tab after logging in
2. Click the **Publish** button next to the target pubset
3. Enter a name for the pubset when prompted (or keep the existing name)
4. Wait 3–5 seconds for publishing to complete

What Gets Published

All six data stores are uploaded:

Store	Content
tbTimebars	Tasks, projects, milestones, allocations (scheduling data)
tbMetaData	Extended metadata: descriptions, status, priority, assessments
tbResources	Resource definitions (people, roles, teams)
tbTags	Picklist tag definitions with colours and groupings
tbFields	Dynamic form field configurations
tbCoreReport	Report configurations and settings

After the stores are published, the system also updates RBAC fields (role-based access control) and enterprise metadata (customer ID, access grants).

Dashboard PubSet vs Storage PubSets

- **Blue Background PubSet** — your **Dashboard PubSet**: directly connected to the Cloud Dashboard. Publishing here updates all dashboard reports and charts immediately.
- **White Background PubSets** — **Storage PubSets**: used for backup and cross-device synchronization. Publishing here does **not** update the dashboard.

Best Practice: Use the blue Dashboard PubSet for day-to-day work. Use white PubSets for backups and snapshots.

After Publishing

- A green toast notification confirms: *"Publishing complete, go to dashboard to see results!"*
 - Your data is immediately available in the Personal Cloud Dashboard
-

7. Downloading from Cloud to Local (Re-Hydrate)

Downloading restores a published dataset from the cloud into your local browser.

Steps

1. Go to the **Pubsets** tab
2. Click the **Download** button next to the pubset you want to restore
3. Confirm by entering **y** when prompted: *"Over-write local data?"*
4. Wait for the download to complete

Important

- **This overwrites all your local data** with the cloud pubset's data
 - If you cancel (enter **n**), your local data is not affected
 - All six stores are replaced with the downloaded data
 - The tbMdJoined table is automatically rebuilt after download
 - No page reload is required — views update automatically
 - Ensure you have published any unsaved local work before re-hydrating
-

8. Clearing a Published Dataset

Clearing removes all data content from a pubset in the cloud while keeping the pubset container.

Steps

1. Go to the **Pubsets** tab
2. Click the **Clear/Delete** button next to the pubset
3. Confirm the action when prompted

What Happens

- All six store fields are set to empty arrays
- The pubset container (name, owner, dates) is preserved

- You can publish new data to the same pubset later
- Your local IndexedDB data is **not affected**
- If the cleared pubset was the active pubset, the Cloud Dashboard will show empty data

9. License and Activation

License Tiers

Tier	Products	Pubsets	Features
Tier-1	ABT01, TBT01, CBT01	0	Local only, no cloud publishing
Tier-2	ABT02, TBT02, CBT02	2	Cloud publishing with 2 pubsets
Tier-3	ABT03, TBT03, CBT03	5–8	Full cloud features with multiple pubsets

Activation (One-Time Setup)

Tier-2 and Tier-3 users must activate pubsets before first use:

1. Click **Activate** in the top navigation bar
2. The system creates your pubset containers based on your tier
3. One pubset is marked as active (feeds the Cloud Dashboard)
4. Remaining pubsets are inactive (for backup or testing)
5. Tier-1 users will see an alert that activation is not available

You only need to activate once. After activation, your PubSets are permanent until you manually clear them.

Viewing Your License

Click **Show License** to see your product code, expiration date, and tier information.

License Type: TBT03

Expires: December 31, 2025

PubSets Available: 5

PubSets Used: 3

10. Cross-Device Synchronization

The Timebars Cloud enables seamless work across multiple devices.

Scenario: Work on PC, Continue on iPad

On Your PC (Primary Device):

1. Work on your project data normally
2. Choose **Main Menu > Publish** and log in to cloud service
3. Click **Publish** on your Dashboard PubSet (blue background)

On Your iPad (Secondary Device):

1. Open the same client app and log in locally (gold icon, top right)
2. Choose **Main Menu > Publish** and log in to cloud service
3. Click **Download** on the Dashboard PubSet — your project data loads
4. Continue working where you left off

Syncing Changes Back:

- After making changes on the iPad, click **Publish** to sync back to the cloud
- Back on the PC, click **Download** to get the latest changes

Key Rules:

- Always publish after making changes
- Always download before starting work on a different device
- Downloading will overwrite local data — save any local work first

Real-Time Dashboard Access

After publishing, click the **Dashboard** button in the top nav bar, or visit www.timebars.com/dashboard from any browser. The dashboard is touch-optimized and works on tablets and phones.

11. OpenProject Sync

The **OpenProject** tab on the Publishing page connects Timebars to an OpenProject instance and moves data in either direction, one project at a time.

Before anything syncs, three things have to be true: Timebars knows which instance to talk to, the two systems agree on what each field is called, and the project has a declared owner. The tab is laid out in that order — the scenario checkboxes at the top for day-to-day running, the **OpenProject Integration** buttons below them for the set-up and repair work you do less often.

A note on wording: projects, and everything under them

OpenProject calls **every** scheduled row beneath a project a *work package*. Timebars does not — it has levels, and the level is what decides where a row goes:

Timebars	Becomes, in OpenProject
L2 Project (green bar)	an OpenProject project
L3 Sub-Project (orange bar)	a row of type L3 SP/WP
L4 Task	L4 Task
L4 Task with subtype Risk / Issue / CR	L4 Risk / L4 Issue / L4 CR
L4 Milestone	L4 Milestone
L5 Allocation (gold bar)	L5 Allocation , carrying the assignee

An OpenProject user who says "work package" almost always means what Timebars calls a **Sub-Project** — the L3 row, which is why its OpenProject type is named **L3 SP/WP**. The rest of this section says *Sub-Project*, *Task* and *Allocation* rather than *work package*, because those are the words on your bars.

L1 Portfolio does not travel. OpenProject has no type for it. A sync run always starts at a Project and covers everything under it.

11.1 Point Timebars at your instance

Control	What it does
OpenProject URL	The instance every sync talks to. Type op.yourcompany.com — the scheme and any trailing slash are tidied for you. Saved to your settings as the admin panel row apOpURL , so it survives a reload and a change of instance no longer needs a rebuild. Leave it empty and the build-time default is used.
Test Connection	Reads your identity and the instance's type list. It confirms the API key works and prints the work package type ids this instance actually has. Run it first, every time; almost every "nothing happened" report starts here.

Types are matched **by name at run time, never by id** — a customer's **L5 Allocation** is not necessarily id 13 — so a type renamed in OpenProject admin stops resolving until the name matches again.

11.2 The four scenarios

Each project has a **sync master**: the side that owns it, held in **tbMDSyncMaster**. The create steps set it; the update steps refuse to run against a project the other way round, and name the button to press instead. Changing master is a deliberate edit on the project, not something a run does behind you.

	Timebars is master	OpenProject is master
Create	A1 — create the project and every row under it in OpenProject	B1 — create the project and every bar under it in Timebars
Update	A2 — update OpenProject, creating anything missing	B2 — update Timebars, creating anything missing

Tick the box and you are asked for the project id — a Timebars **tbID** for A1 and A2, an OpenProject project id for B1 and B2. Each run previews what it will do and asks once before writing.

There is **no separate bulk-create step**. A2 and B2 create what is missing as part of the update, so a project that has grown since the last run does not need a different button. A row that has been deleted on the master side is **reported, never deleted** on the other — removals stay a human decision.

A2 recalculates the hierarchy and rollups before it sends, so the numbers that reach OpenProject are the ones the app would show you.

Every run writes its progress into the feedback list under its own checkbox, and every row it could not carry is named with the reason. Read those lines before assuming the run worked.

11.3 The Integration buttons

Button	Reads	Writes	When to use it
Test Connection	OpenProject	nothing	first, and whenever a run behaves oddly
Discover OP Configuration	OpenProject	nothing — four CSVs to your Downloads folder	after any field or picklist change in OpenProject admin
Reconcile with OpenProject	OpenProject	tbSchema, tbTags and tbResources in Timebars, plus five CSVs	straight after Discover, and after every OpenProject configuration change
Create OP Users & Groups	tbResources	OpenProject users and groups	when setting up people, once the resource pool has emails
Create "All Users" Group	OpenProject	an <i>All Users</i> group holding everybody	when a permission has to be granted to everyone at once

Discover walks the Project, Work Package and User schemas and writes nothing at all — it is safe to run any time.

Reconcile is the one that closes the loop. It matches every Timebars field and every picklist value against OpenProject **by name**, and records the ids it finds. Those ids are per-instance — your `customField107` is not another customer's — so they are never typed by hand, and reconciling them from the live instance is the entire point of the button. It previews, then asks before writing, and it changes IndexedDB only. Take a backup first if you want a way back.

Two things Reconcile deliberately does **not** do:

- **It never writes to OpenProject.** OpenProject's API cannot create a custom field or a picklist value, so those are added in its admin screens — Reconcile tells you which ones are missing and downloads the list.
- **It never sets Sync Yes/No.** Whether a field travels is your decision, held in the workbook; an id is a fact about the instance, and Reconcile now records one for every field that declares a mapping whether or not that field is switched on.

11.4 Where the field mapping lives

The mapping is **data, not code**. Two workbook sheets own it, and two reports in the app show it.

The Schema sheet (`tbSchema`) is the master field list — one row per field in the system. Alongside each field's label, type and purpose it carries the OpenProject columns:

Column	Meaning
<code>tbSchOpSyncYN</code>	Sync Yes/No — whether this field travels at all. Yours to set; nothing else writes it
<code>tbSchOpTableName</code>	which level it maps at — Project, WP, or Users
<code>tbSchOpFormat</code>	OpenProject's own type for the field, which decides the payload shape. <code>Formattable</code> is what routes a rich text field onto the

Column	Meaning
	markdown path
<code>tbSchOpFieldName</code>	the field's label in OpenProject — the name its admin UI shows
<code>tbSchOpInternalName /</code> <code>tbSchOpInternalNameWP</code>	the <code>customFieldNN</code> ids, at project and work package level. Written by Reconcile, never by hand

The **Tags sheet** (`tbTags`) does the same for every picklist value: `tbTagOpFieldName` and `tbTagOpInternalName` say which OpenProject field the value belongs to, and `tbTagOpTagName /` `tbTagOpCustomOptionID` hold the value's own name and id in OpenProject — again filled in by Reconcile.

In the app, both are visible and partly editable:

Report	Where	What you can do
Schema Values	Report Menu → Other	Search and filter the whole field list; OP Field and OP Internal are shown and locked, because the picklist values join to them and a rename here would orphan them. Label, purpose, width, alignment, Validate and Mandatory are editable
Picklist Values	Report Menu → Other	Filter to one picklist and open a value. Its OpenProject Mapping section lets you pick OP Field and OP Internal from the names the Schema table owns; OP Value and OP Custom Option ID are read-only, because OpenProject owns those
Config Integrity	Report Menu → Other	The standing check across Schema, Tags, Fields and CoreReport. Run it after every spreadsheet import — the drift it finds never throws an error on its own

`Sync Yes/No`, the level and the format are **workbook columns and are not editable in the app**, on purpose: turning a field on or off is a configuration decision that has to survive the next spreadsheet import, and the workbook is what both stores are rebuilt from.

The loop to remember. Discover and Reconcile write to the browser's database and to your Downloads folder. The workbook is what Timebars is rebuilt from on the next import. Until the reconciled ids are pasted back into the Schema and Tags sheets, the next import throws the reconciliation away. That is what `tbSchemaForWorkbook.csv` and `tbTagsForWorkbook.csv` are for.

11.5 Two things that travel outside the mapping

Almost everything the sync sends is a **custom field** in OpenProject, and the Schema sheet says where each one goes. Two project attributes are not custom fields, and so cannot be described in the Schema sheet at all:

In OpenProject	What it is
Description	the block of text at the top of the project overview page
Status	the coloured chip beside the project name — On track, At risk, Off track and three more

These are OpenProject **built-ins**. The Schema sheet's internal-name columns hold `customFieldNN` ids, Reconcile regenerates them from what OpenProject reports about its *custom* fields, and a built-in appears

nowhere in that walk — so a built-in name typed into the workbook by hand would be wiped by the next Reconcile run. Both are therefore written by the sync itself, and **the workbook needs no change for either**.

This is in addition to the mapping, not instead of it. `tbMDDescription` still travels to the *TB Description* custom field and `tbMDHealth` still travels to *Schedule Status*, exactly as before. What changed is that the same two values now also fill the OpenProject built-ins, so a project reads correctly to somebody looking at the stock OpenProject page who has never heard of Timebars.

Description. The project's `tbMDDescription` is converted to markdown and written to the OpenProject description. A blank description is **not** sent — an update will not wipe text somebody typed in OpenProject just because the Timebars field has not been filled in yet.

Status comes from the project's **Health** (`tbMDHealth`, the *Schedule Status* picklist), not from Timebar Status. OpenProject's project status answers "*is this project in trouble*" rather than "*what stage is it at*", which is the question Health answers too:

Timebars Health	OpenProject project status
On schedule	On track
Early	On track
Slipping	At risk
Late	Off track
Blocked	Off track
Completed	Finished
Not Assessed	Not started

The seven values above are the whole table, and it is **fixed in code**. OpenProject's six project statuses are a closed set it does not let an administrator add to, so unlike a work package status there is no id to reconcile and nothing that can drift.

A Health value outside that list — one added to the Tags sheet but not to the table above — is reported in the run's feedback list and in its anomaly CSV. What happens then depends on the scenario: **A1** creates the project as *On track*, and **A2** leaves the project's status in OpenProject exactly as it is rather than resetting it. The same applies to a project whose Health is blank.

Work packages are not affected. Their own description and status fields are untouched by this and continue to travel only as the custom fields the Schema sheet names. A work package status in OpenProject is an administrator-defined, per-instance list with numeric ids — a different problem from the fixed six above, and not one this table speaks for.

11.6 The CSV files, and what each one is for

Every button and every run downloads its findings, so you are never reading a mapping problem off a screen. All of them land in the browser's Downloads folder.

From Discover — what OpenProject actually holds:

File	What it answers
opFields.csv	every custom field on this instance, at every level
opCustomOptions.csv	every picklist value on this instance
opFieldComparison.csv	only the fields where OpenProject and tbSchema disagree
opTagComparison.csv	only the picklist values where OpenProject and tbTags disagree

The two comparison files are the ones to open first: they are short by design, and a field that is not in them is a field that is already fine.

From Reconcile — what to do next:

File	What it answers
tbSchemaForWorkbook.csv	the whole Schema sheet in sheet column order, ids filled in. Paste it over the sheet, below the header rows
tbTagsForWorkbook.csv	the same for the Tags sheet
opFieldsToCreate.csv	the fields a Timebars row names that this instance does not have. Admin-UI work — the API cannot create them
opRemainingWork.csv	the picklist values still to add in OpenProject. Same — admin UI
opUnmatchedPeople.csv	resources with no matching OpenProject user, so an assignee cannot be set
currentOPFormFieldConfigStatus.csv	every field each work package type must have on its OpenProject form - where the requirement comes from, whether tbSchema maps it, and whether the form exposes it today. The list to configure a type's form from

The two workbook files are written **whether or not** you confirm the write into Timebars, because they describe what would change either way.

From a run — what did not carry:

opSyncAnomalies_A1.csv, _A2.csv, _B1.csv, _B2.csv — one row per thing worth knowing about that run, at three severities:

Severity	Meaning
blocked	did not sync. Something has to change before it will
guessed	synced, but on a fallback rather than an exact match. Worth a look
note	nothing was lost; worth knowing

A run whose anomalies are all notes is a clean run. The same lines are printed on screen as the run goes, so the CSV is the copy you can sort, filter and send to whoever owns the fix.

11.7 When a field will not sync

Almost every "the field is empty in OpenProject" report is one of five things, and the files above name which:

1. **Sync Yes/No is not Yes** on that **tbSchema** row. Nothing else is wrong; the workbook has decided the field does not travel.
2. **The field does not exist in OpenProject yet** — it is in **opFieldsToCreate.csv**. Add it in OpenProject admin, then Reconcile.
3. **The labels disagree**. Matching is by label, so **Cost Centre** against **Cost Center** resolves at neither level. **opFieldComparison.csv** lists exactly these.
4. **The field exists but is not on that type's form**. OpenProject refuses a value for a field the type does not expose, so the run drops it and says so. Fixing it is **Administration** → **Work packages** → **Types** → **type** → **Form configuration**, by hand — OpenProject has no API for form configuration. A run names each such field **once**, at the top, and only for fields that belong on that type; **currentOPFormFieldConfigStatus.csv** from Reconcile is the full list, type by type.
5. **A picklist value has no option id** — it is in **opRemainingWork.csv**. Add the value in OpenProject admin, then Reconcile.

Order of operations, whenever anything changes on either side: **change it** → **Discover** → **Reconcile** → **paste the two workbook CSVs back into the sheets** → **run A2 or B2**.

11.8 Formatted text fields

43 metadata fields hold formatted text — descriptions, executive summaries, options analyses and the rest of the Charter and Assessment content. They are stored in Timebars in the editor's own format and **converted at the sync boundary**: Timebars sends markdown, and markdown coming back is converted for display. Nothing reaches OpenProject as raw markup, and no formatting has to be re-typed.

Two things follow from that:

- **Set those fields to *Long text* in OpenProject**. A field left as single-line *Text* has its formatting flattened on purpose, because markdown syntax in a one-line box is noise. Change the field's type in OpenProject admin and re-run Reconcile — it upgrades itself, with no code change.
 - **A round trip loses underline and colour**. Headings, lists, bold, italic, links, quotes and code survive in both directions; markdown cannot carry the other two. Everything else comes back as it went.
-

12. Bulk Update

The Bulk Update tab lets you modify data across multiple pubsets simultaneously without downloading and re-publishing each one individually.

Getting Started

1. Go to the **Bulk Update** tab
2. Choose an **Update Mode** (see modes below)
3. Select the **Target Store** you want to update
4. Optionally set a **Published Since** date to filter which pubsets are shown
5. Click **Load Pubsets** to fetch your available pubsets from the cloud
6. Check the boxes next to the pubsets you want to update (or use Select All)

7. Configure the mode-specific options (see below)
 8. Click **Apply Bulk Update**
 9. Confirm the operation in the dialog
-

Mode 1: Complete Replacement (JSON File)

Use case: Replace the entire contents of a store with data from a prepared JSON file.

Steps:

1. Select **Complete Replacement (JSON file)** mode
2. Choose the target store from the dropdown
3. Click **Load Pubsets** and select which pubsets to update
4. Upload a JSON file containing the replacement data
5. The preview shows: file name, size, and record count
6. Click **Apply Bulk Update** and confirm

JSON file format:

```
[
  { "tbID": "A001", "tbName": "Task One", "tbWork": 40 },
  { "tbID": "A002", "tbName": "Task Two", "tbWork": 80 }
]
```

The file must contain a JSON array of objects. The entire store content on each selected pubset is replaced with this array.

Mode 2: Filter and Set

Use case: Find records matching a specific value in a field and change that field to a new value. Ideal for bulk metadata updates like changing status, priority, or category across many pubsets at once.

Available stores: tbMetaData and tbResources only.

Steps:

1. Select **Filter & Set (no file needed)** mode
2. The **Target Store** defaults to tbMetaData
3. The **Published Since** date defaults to one month ago
4. Click **Load Pubsets** and select which pubsets to update
5. If tbMetaData is selected, a **Set Values by Type** filter appears — choose All Types or a specific type (Portfolio, Project, Sub-Project, Task, Milestone, or Allocation)
6. Select a **Field Name** from the dropdown (populated from your Tags definitions in the local database)
7. Select the current value in **Where Value Equals**
8. Select the replacement value in **Set To New Value**
9. Click **Apply Bulk Update** and confirm

Example: Change the priority of all Tasks from "Low" to "Medium" across 5 pubsets:

- Store: tbMetaData | Type filter: Task | Field Name: tbMDPriority | Where Value Equals: Low | Set To New Value: Medium

How the dropdowns are populated:

Dropdown	Data Source
Field Name	Local tbTags store, filtered by prefix (tbMD for tbMetaData, tbRes for tbResources)
Where Value Equals	Unique values found in the actual pubset data for the selected field. Includes (Blank) for empty/null values. Respects the type filter.
Set To New Value	Tag values from the local tbTags store where tbTagTbInternalName matches the selected field

Mode 3: Row-Level Patch (CSV File)

Use case: Update specific rows identified by their unique ID, changing one or more fields per row.

Steps:

1. Select **Row-Level Patch (CSV file)** mode
2. Choose the target store from the dropdown
3. Click **Load Pubsets** and select which pubsets to update
4. Upload a CSV file with your updates
5. The preview shows: file name, size, row count, column names, and a sample row
6. Click **Apply Bulk Update** and confirm

CSV file format:

```
tbMDID,tbMDStatus,tbMDPriority
A001,Complete,High
A002,In Progress,Medium
A003,Not Started,Low
```

Rules:

- The first column must be the store's unique key (tbID for tbTimebars, tbMDID for tbMetaData, tbResID for tbResources, etc.)
- Only the fields listed in the CSV are changed; all other fields on the record are preserved
- Rows in the pubset that don't match any CSV key are left unchanged

Bulk Update Results

After execution, a results table appears:

Column	Content
Pubset Name	Name of the pubset

Column	Content
ID	Strapi record ID
Status	success or failed
Detail	"Replaced" (Mode 1), "X rows matched" (Mode 2), "X rows patched" (Mode 3), or error message
• Green background: all pubsets succeeded • Red background: one or more pubsets failed (check Detail column for error)	

13. Re-Publish with Scheduling Engine

The Re-Publish feature downloads pubsets from the cloud, runs the PM scheduling engine to recalculate all derived data (dates, hours, costs, percent complete, rollups, resource demand), then publishes the refreshed data back to the cloud.

When to Use

- After a bulk update that changed fields affecting calculations (e.g. status, work hours, dates)
- To refresh derived data across many pubsets without manually opening each one
- When the scheduling engine or rollup logic has been updated and you want all pubsets recalculated

Steps

1. Go to the **Re-Publish** tab
2. Optionally set a **Published Since** date filter
3. Click **Load Pubsets**
4. Select the pubsets you want to re-publish (checkboxes or Select All)
5. Choose which calculation steps to run (all checked by default):
 - **Scheduling Engine** — recalculates dates, hours, costs, and percent complete for allocations
 - **Rollups & Hierarchy** — rolls up costs and hours from children to parents, rebuilds L1–L5 hierarchy names, regenerates the tbMdJoined table
 - **Resource Calculations** — regenerates weekly and monthly resource demand data (tbResCalcs2, tbResCalcsUsage)
6. Click **Execute Re-Publish**
7. Confirm the operation in the dialog
8. Watch the progress log as each pubset is processed

The 5-Step Pipeline (per pubset)

Step	Operation	Description
1	Download to IDB	Fetches the pubset from cloud and loads all stores into local IndexedDB
2	Scheduling Engine	Runs <code>calcCostWorkTBCB()</code> on allocations to recalculate dates, hours, costs, and percent complete
3	Rollups & Hierarchy	Runs <code>performFullSystemUpdate()</code> to roll up values through the hierarchy and rebuild joined tables

Step	Operation	Description
4	Resource Calculations	Runs <code>createUnifiedWeeklyUsageData()</code> to regenerate weekly/monthly resource demand splits
5	Publish Back	Runs <code>publishToTimebarsCloud()</code> to upload all stores, then updates RBAC and enterprise metadata

Steps 2, 3, and 4 can be individually enabled or disabled via the checkboxes.

Progress Log

A dark terminal-style log panel shows timestamped entries as each step executes:

```
[14:23:45] --- Starting pubset: Project Q4 (ID: 123) ---
[14:23:46] Step 1: Downloading pubset data to IndexedDB...
[14:23:48] Step 1 complete: Data loaded into IDB
[14:23:50] Step 2: Running scheduling engine on allocations...
[14:23:53] Step 2 complete: Scheduling engine finished
[14:23:54] Step 3: Running rollups, hierarchy, and joined table rebuild...
[14:23:56] Step 3 complete: Rollups and hierarchy rebuilt
[14:23:57] Step 4: Regenerating resource calculations...
[14:23:59] Step 4 complete: Resource calculations regenerated
[14:24:00] Step 5: Publishing all stores back to cloud...
[14:24:04] Step 5b: Updating RBAC and enterprise metadata...
[14:24:05] Pubset "Project Q4" re-published successfully!
```

- **Blue text:** Start/end markers | **Green text:** Successful step | **Orange text:** Warnings | **Gray text:** Skipped steps | **Red text:** Fatal errors

Processing Notes

- Pubsets are processed **one at a time** (sequential) to avoid database conflicts
- Typical time per pubset: 20–25 seconds with all calculation options enabled
- Your local IndexedDB will contain the data from the last processed pubset when complete
- Toast notifications confirm each pubset as it completes

14. Support Resources

Knowledge Base

- Browse articles at www.timebars.com/knowledgebase
- Search for specific topics or error messages
- View step-by-step tutorials and videos

Customer Support

- **Email:** jcox@tbcox.com
- **Support Tickets:** www.timebars.com/support

- **Response Time:** 24–48 hours (varies by subscription level)
- **Phone:** (613) 255-5374

Feature Requests

- Email: jcox@tbcx.com with a detailed description and use case
- All requests are reviewed for the product roadmap

15. Quick Reference

Publishing Workflow

1. Log in to client app (gold icon, top right)
2. Main Menu > Publish
3. Cloud Login (email + password)
4. Activate PubSets (first time only)
5. Click Publish next to blue PubSet (Dashboard PubSet)
6. View Dashboard button or www.timebars.com/dashboard

Cross-Device Workflow

- Device A:
1. Make changes
 2. Main Menu > Publish → Publish (to save to cloud)
- Device B:
1. Main Menu > Publish → Download (to get latest from cloud)
 2. Make changes
 3. Main Menu > Publish → Publish (to save changes back)
- Device A:
1. Main Menu > Publish → Download (to get Device B's changes)

Troubleshooting Quick Reference

Problem	Solution
Cannot log in to cloud	Check email/password; verify subscription at www.timebars.com/login
Publishing hangs	Wait 10+ seconds; check browser console for errors
Data not in dashboard	Ensure you published to the active pubset (blue highlight)
Re-Hydrate fails	Check "Last Published" date on pubset — if empty, pubset has no data
"No pubsets found" (Bulk Update)	Adjust or clear the Published Since date filter

Problem	Solution
Field Name dropdown empty (Bulk Update)	Ensure local tbTags store has definitions with matching tbTagTbInternalName prefix
Pubset is empty (Re-Publish)	Pubset has been cleared — publish data to it first
OpenProject run does nothing	Run Test Connection first; check the OpenProject URL field (apOpUrl)
A field is empty in OpenProject	Check Sync Yes/No on its Schema row, then opFieldComparison.csv for a label mismatch
A picklist value will not sync	It has no option id — see opRemainingWork.csv , add it in OpenProject admin, then Reconcile
A2 or B2 refuses to run	The project's sync master is the other side. Use the other button, or change the master on the project
Reconciled ids disappear after an import	The two workbook CSVs were not pasted back into the Schema and Tags sheets
Project status in OpenProject looks wrong	It comes from the project's Health, not Timebar Status — see the table in section 11.5

Cloud Publishing and Bulk Operations Guide v2.2 — Timebars Ltd.

Part 11 — The Personal Dashboard

Personal Dashboard Guide

The Timebars Personal Dashboard delivers a focused suite of interactive reports, visualizations, and data management tools for individual project managers and team leads. It connects directly and automatically to your **single active pubset** — the pubset you have marked as Active (or "connected") in your Timebars, Agilebars, or Costbars client application. No additional configuration is needed; publish your data and the Personal Dashboard reflects it immediately.

Personal Dashboard vs. Enterprise Dashboard The Personal Dashboard ([/personaldashboard](#)) is designed for a single-pubset workflow: one user, one active data source, instant access. If your organisation needs to combine data from multiple pubsets across projects or teams into a shared portfolio view, see the [Enterprise Dashboard Guide](#) ([/dashboard](#)).

Table of Contents

- [Personal Dashboard Guide](#)
 - [Table of Contents](#)
 - [What is the Personal Dashboard?](#)
 - [How to Access](#)
 - [How Publishing Connects to Your Dashboard](#)
 - [Dashboard Navigation](#)
 - [Report Categories](#)

- [Project Status Report \(Dashboard Home\)](#)
- [Project Variance Report](#)
- [Interactive Card-Based Drilldown](#)
 - [Project Status Cards — Hierarchical Navigation](#)
 - [Four-Level Hierarchy](#)
 - [Interactive Features](#)
- [Agilebars Burndown Chart](#)
- [Resource Cost & Usage — Pie Charts](#)
- [Resource Usage — Bar Charts](#)
- [Common Features Across All Reports](#)
 - [Search & Filter](#)
 - [Sorting](#)
 - [Inline Editing](#)
 - [Health Indicators](#)
 - [Responsive Design](#)
 - [Performance](#)
- [Data Export & Integration](#)
 - [Export Options](#)
 - [Cloud Integration](#)
- [License Tiers & Report Availability](#)
- [Getting Started with the Personal Dashboard](#)
- [Tips for Effective Use](#)
- [Troubleshooting](#)
 - [Dashboard Not Loading](#)
 - [Reports Show No Data](#)
 - [Visualization Not Updating](#)
 - [Cannot Edit Data](#)
- [Related Help Topics](#)
- [Support](#)

What is the Personal Dashboard?

The Personal Dashboard is a web-based platform that transforms data from your Agilebars, Timebars, or Costbars application into dynamic visualizations — graphs, line charts, pie charts, bar charts, and tabular reports. It is powered by the **one pubset you have set as Active** in your client application. Unlike the client apps, which work offline, the Personal Dashboard requires an internet connection and an active subscription.

Key Benefits:

- **Zero Configuration:** No dashboard sources to create or manage — your Active pubset is the data source
- **Real-Time Updates:** Changes sync automatically when you re-publish from your client app
- **Anywhere Access:** View reports from any device with a modern web browser
- **Interactive Intelligence:** Filter, sort, search, and drill down through hierarchical data
- **Seven-Dimension Health Tracking:** Monitor project health across Overall, Scope, Schedule, Cost, Hours, Risk, and Issues

How to Access

Prerequisites:

- Active Timebars Ltd. Cloud subscription
- Published data from Agilebars, Timebars, or Costbars client application
- An **Active pubset** set in the client app (the one flagged as connected)
- Internet connection

Two Ways to Access:

1. From Client Applications:

- After publishing your data, click the **Dashboard** button in the client app
- You will be automatically logged in and directed to the Personal Dashboard

2. Direct Web Access:

- Visit www.timebars.com/personaldashboard
- Log in with your credentials

How Publishing Connects to Your Dashboard

The Personal Dashboard is powered exclusively by your **Active pubset** — the pubset you have designated as active (connected) in the Publish panel of your client application. This is the key difference from the Enterprise Dashboard, which lets you combine multiple pubsets.

Workflow:

1. Open your Timebars, Agilebars, or Costbars client
2. Navigate to **Main Menu** → **Publish**
3. Log in to the cloud service
4. Identify the pubset with a blue background (your Dashboard/Active pubset)
5. Click **Re-Publish!** to send updated data to the cloud
6. Your Personal Dashboard reflects the new data automatically

For full details on the publishing process, see the [Cloud Publishing Guide](#) - Publish reports to the cloud

Dashboard Navigation

The Personal Dashboard uses a **left sidebar navigation** to browse available reports.

Report Categories

1. **Executive Portfolio Reports:** Portfolio-wide visibility and variance tracking
2. **Card-Based Drilldown:** Interactive hierarchical navigation with health indicators
3. **Charts and Visualizations:** Burndown charts, pie charts, and bar charts
4. **Resource Reports:** Resource allocation, utilization, and cost analysis

To View a Report:

- Click any report name in the sidebar
- The report displays in the main viewing area

- Use filters and search to focus on specific data
-

Project Status Report (Dashboard Home)

Your command center for portfolio-wide visibility. This comprehensive tabular report displays all portfolios, projects, work packages, tasks, and resource allocations from your active subset in a unified view.

Key Features:

- **Multi-Level Search:** Search by project name or unique ID with instant filtering
- **Advanced Filtering:** Filter by Project (L2), Type, Owner, and Status simultaneously
- **Smart Sorting:** Click any column header to sort ascending/descending
- **Color-Coded Rows:** Visual distinction between:
 - Portfolios (light green background)
 - Projects (light orange background)
 - Tasks (default background)
- **Hierarchical Indentation:** Clear visual hierarchy showing Portfolio → Project → Work Package → Task relationships
- **One-Click Details:** Click any row to open a detailed item dialog with full editing capabilities
- **License-Aware Columns:** Display adapts to show Agilebars, Timebars, or Costbars-specific fields based on your license
- **Real-Time Updates:** Data refreshes automatically using SWR caching

When to Use:

- Daily portfolio status reviews
- Executive briefings
- Identifying project bottlenecks
- Tracking ownership and accountability
- Quick access to any project detail

Export Options:

- Export to Excel with all filtered/sorted data
 - Export to CSV for external analysis
 - Maintains column order and formatting
-

Project Variance Report

Track deviations from baseline plans with variance-specific metrics showing the delta between planned and actual performance across cost, schedule, and scope dimensions.

Key Features:

- **Variance-Focused Columns:** Baseline vs. actual comparisons for:
 - Start Date Variance (days early/late)
 - Finish Date Variance (days early/late)
 - Duration Variance (work days)
 - Work Variance (hours)
 - Cost Variance (budget units)

- **Deviation Highlighting:** Quickly identify projects off-track from original plans
- **Same Powerful Filtering:** All search, filter, and sort capabilities from the main report
- **Historical Tracking:** Compare current state against original baseline commitments
- **Performance Indicators:** Visual cues for schedule variance, cost variance, and scope creep

When to Use:

- Earned Value Management (EVM) analysis
- Monthly project performance reviews
- Identifying projects requiring corrective action
- Forecasting final project outcomes

Note: Baselines must be set in the client application before publishing for this report to show variance data. The [Enterprise Dashboard](#) offers an equivalent Variance Report across combined pubset sources.

Interactive Card-Based Drilldown

Project Status Cards — Hierarchical Navigation

Navigate project data in a visually engaging, click-to-drill format with collapsible cards representing each level of your portfolio hierarchy. Navigate from portfolio to project to work package to individual tasks with interactive health indicators on every card.

Four-Level Hierarchy

Level 1 — Portfolio Cards (Light Brown Background)

- High-level portfolio overview cards
- Aggregated health indicators for all projects within
- Click any portfolio card to reveal its projects

Level 2 — Project Cards (Light Green Background)

- All projects within the selected portfolio
- Displays project-level health, dates, owner, status
- Click any project card to reveal work packages and associated Tasks/Risks/Issues

Level 3 — Work Package Cards (Light Orange Background)

- Work packages within the selected project
- Click any work package to reveal its Level 4 items

Level 4 — Tasks, Risks, Issues (Tabbed Interface)

- Detailed tabular data for individual work items
- Separate tabs for:
 - **Tasks Tab:** All tasks with status, dates, owner, progress
 - **Risks Tab:** Risk probability, impact, mitigation status
 - **Issues Tab:** Issue priority, severity, resolution tracking
- Full editing capabilities on each tab

Interactive Features

Breadcrumb Navigation Path:

- Shows current location: L1 > L2 > L3
- Click breadcrumb levels to navigate back up the hierarchy

Hide/Show Controls:

- Toggle visibility of card levels as needed
- Collapse levels to focus on specific areas

Health Indicator Lights: Every card displays 7 color-coded health indicators:

-  **Green:** On track
-  **Yellow:** Needs attention
-  **Red:** Critical issue
-  **Orange:** Not yet assessed

Seven Health Dimensions:

1. Overall Health
2. Cost Health
3. Hours Health
4. Risk Health
5. Schedule Health
6. Scope Health
7. Issues Health

Planned vs. Current Comparison:

- See baseline dates and current dates side-by-side
- Variance indicators on each card

When to Use:

- Executive presentations requiring visual engagement
- Project health assessment at-a-glance
- Understanding risk/issue context within project structure
- Mobile access (card interface is touch-friendly)

Agilebars Burndown Chart

Track sprint velocity and work completion rates with the classic agile burndown chart showing ideal vs. actual progress over time.

Chart Components:

- **Ideal Line** (Blue): Planned work burn rate based on sprint timeline
- **Actual Line** (Red/Green): Actual work completion to date
- **X-Axis:** Sprint timeline (days or iterations)
- **Y-Axis:** Remaining work (hours or story points)

Key Features:

- Visual comparison of planned burn rate against actual progress
- Sprint performance evaluation and team velocity assessment
- Forecast completion dates

Prerequisites:

- Must publish the Active Pubset from the Agilebars client application
- Must run the Burndown calculation in the client before publishing
- Data must include sprint-based work items

When to Use:

- Sprint retrospectives and planning
 - Daily standup reference
 - Velocity trend analysis across sprints
-

Resource Cost & Usage — Pie Charts

Visualize cost distribution across your active pubset with interactive pie charts breaking down resource costs and hours by multiple dimensions.

Four Pie Charts Displayed (2×2 grid):

1. **By Project (L2):** Which projects consume the most resources
2. **By Resource Role:** Cost distribution across Developer, Manager, QA, Architect, etc.
3. **By Resource Location:** Geographic cost analysis (Office locations, Remote, Offshore)
4. **By Department:** Departmental resource allocation and spend

Indicator Cards (Top of Page):

- **Total Cost Cards:** Prominent cards showing total costs for each dimension
- **Currency Formatting:** Professional \$USD formatting with thousands separators

Interactive Features:

- **Hover Details:** Hover over pie slices for exact values and percentages
- **Legend:** Color-coded legend with labels
- **Percentage Labels:** Show percentage on larger slices

When to Use:

- Monthly financial reviews
 - Budget allocation decisions
 - Chargeback reporting to business units
 - Resource planning and forecasting
-

Resource Usage — Bar Charts

Detailed bar chart analysis of resource utilization across six critical dimensions showing both hours and cost consumption patterns.

Six Resource Dimensions:

1. **By Resource Name:** Individual resource utilization and costs
2. **By Project (L2):** Project-level resource consumption
3. **By Resource Role:** Role-based allocation analysis (PM, Dev, QA, Architect, BA, etc.)
4. **By Primary Skill:** Skill-based resource distribution (Java, Python, Cloud, Database, etc.)
5. **By Resource Location:** Geographic resource allocation
6. **By Department:** Departmental resource usage

Chart Features:

- **Stacked Bar Charts:** Show both hours (blue) and cost (green) in single view
- **Dual Y-Axis:** Left axis for hours, right axis for costs
- **Legend Toggle:** Show/hide hours or cost bars independently
- **Configurable Metadata:** Works with your custom metadata fields from the Timebars client

When to Use:

- Resource capacity planning meetings
- Skill gap identification for hiring
- Utilization rate analysis
- Workforce planning decisions

Common Features Across All Reports

Search & Filter

Text Search:

- Search by project name or description, or by unique ID
- Instant filtering as you type (case-insensitive)

Multi-Select Filters:

- **By Project:** Filter to specific project (L2 level)
- **By Type:** Portfolio, Project, Sub-Project, Task, Milestone, Allocation
- **By Owner:** Filter by assigned resource
- **By Status:** Active, On Hold, Cancelled, Complete
- **"All" Option:** Clear a specific filter dimension instantly

Filter Combinations:

- Combine multiple filters for precise data views
- Filters persist during navigation
- Clear all filters with one click

Sorting

- Click any column header to sort ascending/descending
- Arrow indicators show current sort direction
- Maintains filter state during sort operations

Inline Editing

- Click the row action button (pencil icon) to open the edit dialog
- Full CRUD operations directly from reports
- Built-in field validation prevents data errors
- Changes sync immediately to the backend

Health Indicators

Seven-Dimensional Health Tracking — all cards and detail views display:

Dimension	Green	Red
Overall Health	On track	Critical
Cost Health	Under budget	Over budget
Hours Health	On track	Over hours
Risk Health	Low risk	High risk
Schedule Health	On schedule	Late
Scope Health	Stable	Scope creep
Issues Health	No critical issues	Critical issues

Color Coding:

-  Green: On track / healthy
-  Yellow: Needs attention / at risk
-  Red: Critical issue / off track
-  Orange: Not yet assessed / no baseline

Responsive Design

- Touch-optimized controls for tablets and phones
- Adaptive column display on smaller screens
- Fixed headers during scroll on desktop

Performance

- **SWR Caching:** Client-side data caching for instant navigation
- **Automatic Revalidation:** Data stays fresh with background updates
- **Loading States:** Professional loading indicators during data fetch
- **Error Handling:** Graceful error messages with retry options

Data Export & Integration

Export Options

Available on all tabular reports:

- **Excel Export:** Full data export with formatting
- **CSV Export:** Raw data for external analysis tools

- **Maintains Filters:** Export only filtered/sorted data
- **One-Click Download:** Simple download button

Cloud Integration

Seamless Data Flow:

1. **Publish from Client:** One-click publish from the Timebars/Agilebars/Costbars desktop client selects the Active pubset and sends data to the cloud
2. **Real-Time Sync:** Changes reflect in the dashboard within seconds of publishing
3. **Secure Authentication:** Enterprise-grade JWT security, encrypted data transmission (HTTPS/TLS)
4. **Multi-User Viewing:** Multiple users can view the dashboard simultaneously

See the [Cloud Publishing Guide](#) - Publish reports to the cloud for full publishing instructions.

License Tiers & Report Availability

Report	Basic (T01)	Standard (T02)	Professional (T03)
Project Status Report	✓	✓	✓
Export to Excel/CSV	✓	✓	✓
Project Variance Report	—	✓	✓
Resource Pie Charts	—	✓	✓
Resource Bar Charts	—	✓	✓
Card-Based Drilldown	—	✓	✓
Burndown Charts (Agilebars)	—	—	✓
Advanced analytics	—	—	✓

Check Your License:

- View license details in the client app: **Main Menu** → **Publish** → **Show License**
- Upgrade at www.timebars.com/pricing

Getting Started with the Personal Dashboard

Step-by-Step:

1. **Publish Your Data:**
 - Open your Timebars, Agilebars, or Costbars client
 - Navigate to **Main Menu** → **Publish**
 - Log in to the cloud service
 - Click **Activate** if this is your first time (one-time setup)
 - Identify the pubset with a blue background (your Active/Dashboard pubset)
 - Click **Re-Publish!** to send data to the cloud

2. Access the Personal Dashboard:

- Visit www.timebars.com/personaldashboard
- Sign in with your credentials

3. Explore Reports:

- Use the sidebar navigation to browse reports
- Start with the Dashboard Home for a portfolio overview
- Try Card-Based Drilldown for visual navigation

4. Filter & Search:

- Use search boxes to find specific projects
- Apply filters to focus on project types, owners, or status
- Combine filters for precise views

5. Analyze with Charts:

- View Resource charts for capacity planning
- Use Pie charts for cost distribution analysis
- Track sprint progress with Burndown charts (Agilebars)

Tips for Effective Use

1. **Regular Publishing:** Publish data daily or weekly to keep the dashboard current
2. **Consistent Metadata:** Use consistent tagging and naming for better filtering
3. **Baseline First:** Set baselines in the client before using the Variance Report
4. **Health Tracking:** Update health indicators regularly for accurate status
5. **Use Cards for Presentations:** Card view is excellent for executive briefings
6. **Export for Offline:** Create Excel exports for offline analysis or board meetings
7. **Check License:** Ensure your license is current for uninterrupted access
8. **Need Multi-Pubset Views?:** If you need to combine multiple pubsets (e.g., across teams or products), use the [Enterprise Dashboard](#) instead

Troubleshooting

Dashboard Not Loading

Possible Causes: Internet connection issue, subscription expired, or no data published yet.

Solutions:

- Check your internet connection
- Verify subscription status at www.timebars.com/login
- Publish data from the client application

Reports Show No Data

Possible Causes: Data not published from client, filters excluding all data, or wrong pubset published.

Solutions:

- Publish from the client using the Active pubset (blue background)
 - Click **All** to clear all filters
 - Verify data exists in the client before publishing
-

Visualization Not Updating

Possible Causes: Browser cache, old SWR cache, or publishing to the wrong pubset.

Solutions:

- Refresh the browser page (F5 or Ctrl+R)
 - Clear the browser cache
 - Re-publish from the client to the Active pubset
-

Cannot Edit Data

Possible Causes: Network connectivity issue or session expired.

Solutions:

- Check your internet connection
 - Log out and log back in
 - Verify your subscription is active
-

Related Help Topics

- [Enterprise Dashboard Guide](#) — Combine multiple pubsets into shared portfolio views with RBAC
 - [Cloud Publishing Guide](#) — How to publish data to the cloud and manage pubsets
 - [Local Reports Guide](#) — Reports available within client applications
 - [Data Synchronization, Backup, Recovery and Retention Guide](#) — Understanding the data model
 - [User Interface Guide](#) — Navigate the client apps
-

Support

Need Help?

- Knowledge Base: www.timebars.com/knowledgebase
 - Email Support: jcox@tbcox.com
 - Feature Requests: jcox@tbcox.com
 - Phone: (613) 255-5374
-

The Personal Dashboard transforms your active project data into strategic intelligence — accessible anytime, anywhere, on any device.

Part 12 — The Enterprise Dashboard

Enterprise Dashboard Guide

The Timebars Enterprise Dashboard provides a consolidated, organisation-wide view of your project portfolio. It draws data from one or more published datasets (pubsets) — from Agilebars, Timebars, or Costbars — that are combined into a **Dashboard Source**: a single, unified snapshot that powers all reports, charts, and visualizations in the enterprise view.

Enterprise Dashboard vs. Personal Dashboard The Enterprise Dashboard (</dashboard>) is designed for teams and organisations that need to combine data across multiple pubsets — for example, rolling up projects from different teams, products, or time periods into one portfolio view. Role-based access control (RBAC) governs who can see which pubsets and sources.

If you work with a single active pubset and want an instant, zero-configuration view of your own data, see the [Personal Dashboard Guide \(/personaldashboard\)](/personaldashboard).

Table of Contents

1. [Getting Started](#)
2. [Understanding Dashboard Sources](#)
3. [Role-Based Access Control \(RBAC\)](#)
4. [Creating and Managing Dashboard Sources](#)
5. [Managing Pubsets](#)
6. [The Executive Dashboard Home](#)
7. [Reports](#)
8. [Visualizations](#)
9. [Drilldown](#)
10. [Analytics](#)
11. [Shop Schedule \(Gantt\)](#)
12. [Preprocessing Resource Data](#)
13. [Quick Reference](#)

Getting Started

To access the Enterprise Dashboard, sign in and navigate to **Dashboard** in the main navigation menu. You will land on the Executive Dashboard home page at </dashboard>.

If you see a message saying you have no access, contact your **Administrator** to have your role assigned or a pubset shared with you (see [RBAC](#) below).

What you need before using the Enterprise Dashboard

1. A user account with a role of **Administrator**, **Project Manager**, or **Team Member**
2. At least one **pubset** published from Timebars, Agilebars, or Costbars (see the [Cloud Publishing Guide - Publish reports to the cloud](#))
3. At least one **Dashboard Source** created from your pubsets (see [Creating and Managing Dashboard Sources](#))

Coming from the Personal Dashboard? The Enterprise Dashboard requires an extra step — you must create a Dashboard Source before reports will display data. This is because the Enterprise Dashboard lets you combine multiple pubsets, and you choose which combination to activate. The [Personal Dashboard](#) skips this step by automatically connecting to your single active pubset.

Understanding Dashboard Sources

A **Dashboard Source** is the data backbone of the Enterprise Dashboard. It is a consolidated snapshot built from one or more pubsets and stored in the system. All reports, charts, and visualizations draw from the currently active source.

Key facts

- Each source has a **name**, creation date, and an **isActive** flag
- Only **one source can be active at a time** for your organisation
- Sources are created by selecting pubsets at [/dashboard/pubsets](#)
- You can switch the active source at [/dashboard/sources](#)
- Sources are **snapshots** — they do not update automatically; create a new source each time your data changes

How this differs from the Personal Dashboard

	Personal Dashboard	Enterprise Dashboard
Data source	Your single Active pubset (auto-connected)	One or more pubsets combined into a Dashboard Source
Source creation	None required	Must create a Dashboard Source
Pubset selection	Automatic (flagged as Active in client)	Manual — choose which pubsets to combine
Multi-pubset support	No	Yes
Access control	Per-user	RBAC (Admin, PM, Team Member)

Source naming convention

When a source is generated, the system proposes a name in this format:

{Product} – {PubsetNames} [{IDs}] – MMM DD, YYYY HH:MM

Example:

Timebars – Q1 Portfolio, Infrastructure Refresh [12-14] – Apr 06, 2026 09:30

Edit this name before saving to choose something meaningful that lets you identify snapshots later.

Role-Based Access Control (RBAC)

The Enterprise Dashboard uses RBAC to control which pubsets and dashboard sources each user can see.

Roles

Role	What they can see
Administrator	All pubsets and sources in the organisation
Project Manager	Their own pubsets + any pubsets where they have been granted PM access
Team Member	Their own pubsets + any pubsets where they have been granted TM access
<i>(No role)</i>	No access to the Enterprise Dashboard

How access is granted

- **Ownership:** If you created (own) a pubset, you always have access to it
- **Admin grant:** An Administrator can see all org pubsets by default
- **Explicit grant:** In Timebars/Agilebars/Costbars, the pubset owner can add a user's email to the `grant_pm_access_to` or `grant_tm_access_to` fields when publishing

Changing a user's role

Roles are managed in **Strapi** (the backend CMS) by an Administrator:

1. Log in to the Strapi admin panel
2. Navigate to **Content Manager** → **Users**
3. Find the user and edit their `primary_role` field
4. Set to **Administrator**, **Project Manager**, or **Team Member**
5. Save — the change takes effect on the user's next page load

Creating and Managing Dashboard Sources

Step 1 — Publish your data

Before creating a dashboard source, project data must be **published as a pubset** from within Timebars, Agilebars, or Costbars. See the [Cloud Publishing Guide](#) for full publishing instructions. Each publish creates a new pubset record.

Step 2 — Browse and select pubsets

Navigate to **Dashboard** → **Manage Pubsets** (`/dashboard/pubsets`). You will see all pubsets your RBAC role grants you access to. Use the checkboxes to select one or more pubsets to consolidate.

Step 3 — Generate a dashboard source

With pubsets selected, click **Generate Dashboard Source**. You will be taken to the consolidation page (`/dashboard/pubsets/consolidated`) where you can:

- Review which pubsets are included
- Edit the auto-generated source name
- Click **Save Dashboard Source** to create the source

Step 4 — Set the active source

Navigate to **Dashboard** → **Manage Sources** (</dashboard/sources>). You will see a list of all sources accessible to you. Click **Set Active** on the source you want to use. All reports will now draw from this source.

Managing existing sources

From the Sources page you can:

- **View** source details (which pubsets are included, creation date)
 - **Set active** to switch which source powers the dashboard
 - **Delete** sources you no longer need
-

Managing Pubsets

Navigate to **Dashboard** → **Manage Pubsets** (</dashboard/pubsets>) to manage your enterprise pubsets.

What you can do here

- Browse all pubsets you have access to (RBAC-filtered)
- See pubset status, product, owner, and creation date
- Select multiple pubsets and generate a consolidated dashboard source
- View an individual pubset report

Pubset report

Click **View Report** on any pubset to open a read-only report for that single dataset at </dashboard/pubsets/report/{id}>.

The Executive Dashboard Home

Route: </dashboard>

The home page provides an executive-level summary of your active dashboard source. It shows:

- Key portfolio metrics and health indicators
- Summary cards with quick-access links to reports
- Status of the currently active dashboard source

If no source is active, you will be guided to create one via the Manage Pubsets page.

Reports

Navigate to **Dashboard** → **Reports** (</dashboard/reports>) to access the full report catalogue.

Project Status Report

Route: </dashboard/reports/projects>

Detailed project and portfolio status drawn from the active dashboard source. Shows timelines, budgets, and health indicators for every project in the portfolio — spanning all pubsets included in the active source. Use this report for regular project review meetings.

Portfolio Status Report**Route:** </dashboard/reports/portfolio>

Portfolio-level insights and strategic analysis across all projects in the active source. Provides cross-project comparisons and helps identify which projects are on track, at risk, or in trouble. Ideal for portfolio review sessions.

Variance Report**Route:** </dashboard/reports/variance>

Compares current project data against baseline plans to identify schedule and cost deviations across all pubsets in the active source. Quickly highlights where projects are running ahead or behind plan, and by how much.

This report is unique to the Enterprise Dashboard in that it covers all pubsets in your Dashboard Source.

Risk Register**Route:** </dashboard/reports/risks>

A consolidated view of all identified risks across the entire portfolio (all pubsets in the active source). Each risk is scored for probability and impact. Includes mitigation plans and escalation tracking so risk owners can take action before issues materialise.

Issues Log**Route:** </dashboard/reports/issues>

Tracked issues across the entire portfolio, including impact assessment, mitigation status, and escalation levels. Use this log to ensure open issues are being actively managed and resolved.

Change Requests**Route:** </dashboard/reports/change-requests>

Formal change requests affecting project scope, schedule, or budget. Tracks approval status and the financial or schedule impact of each change. Useful for governance and audit purposes.

Resource Pool Report

Route: </dashboard/reports/resources>

Comprehensive resource capacity and allocation data drawn from all pubsets in the active source. Shows availability by role, location, and department alongside current assignments. Use this report to understand where your resource capacity sits across the organisation.

Capacity vs Demand**Route:** </dashboard/reports/capacity-demand>

Side-by-side view of resource supply and demand by week and role, consolidated across all included pubsets. Highlights overloaded or underutilised resource pools so you can rebalance allocations or plan hiring decisions proactively.

Executive Portfolio Summary**Route:** </dashboard/reports/executive-summary>

An at-a-glance summary built for executive audiences. Presents portfolio health across seven key dimensions and includes a financial summary. Designed to be shared in board-level or steering committee meetings with minimal preparation.

Project Prioritization Matrix**Route:** </dashboard/reports/prioritization>

Composite scoring of every project based on health, cost performance, schedule performance, and risk level. Helps leadership decide which projects to continue as-is, restructure, or terminate. Output is a ranked list with a recommended action for each project.

Project Selection Bubble Chart**Route:** </dashboard/reports/bubble-chart>

A strategic scatter plot of **Strategic Value** vs **Ability to Execute** with bubble size representing project cost. Includes a selection algorithm and quadrant analysis to guide portfolio optimisation decisions. Adjust weights and thresholds interactively to simulate different selection scenarios.

Balanced Scorecard**Route:** </dashboard/reports/balanced-scorecard>

Assesses portfolio balance across eight KPI dimensions including investment strategy alignment, objectives coverage, risk distribution, resource mix, and project duration spread. Surfaces imbalances that may not be visible from individual project views.

Portfolio Financial Summary

Route: </dashboard/reports/financial-summary>

Aggregates financial data across the entire portfolio including NPV, IRR, Payback Period, and ROM Estimates. Provides a single source of truth for portfolio cost analysis and investment return tracking across all included pubsets.

Strategic Alignment Heatmap**Route:** </dashboard/reports/strategic-alignment>

A heatmap of **Investment Objective vs Category** showing where portfolio spend is concentrated. Quickly reveals whether the portfolio is aligned to your organisation's strategic priorities or skewed towards certain objectives.

What-If Scenario Dashboard**Route:** </dashboard/reports/what-if>

An interactive simulation tool. Adjust Strategic Value and Ability to Execute weights and scoring thresholds to model different portfolio decisions and see the financial and health impact of each scenario. Useful for pre-planning workshops and scenario analysis.

Visualizations

Navigate to **Dashboard** → **Visualizations** (</dashboard/visualizations>) to access interactive chart views. All visualization data is drawn from the active Dashboard Source.

Resource Cost Charts**Route:** </dashboard/visualizations/cost-charts>

Interactive pie charts showing resource cost distribution across four dimensions: by project, by role, by location, and by department. Drill into any segment to see the contributing detail. Covers all pubsets in the active source.

Prerequisite: Resource data must be preprocessed before these charts will display. See [Preprocessing Resource Data](#).

Resource Usage Charts**Route:** </dashboard/visualizations/usage-charts>

Bar charts showing resource utilization across six dimensions with weekly breakdowns. Use these alongside the Resource Pool Report and Capacity vs Demand report to understand allocation patterns across the organisation.

Prerequisite: Resource data must be preprocessed. See [Preprocessing Resource Data](#).

Burndown Charts

Route: </dashboard/visualizations/burndown>

Sprint burndown charts comparing planned vs forecast progress over time. Includes both computed project burndown from Timebars data and sprint-level charts from Agilebars. Covers all included pubsets in the active source.

Drilldown

Navigate to **Dashboard** → **Drilldown** (</dashboard/drilldown>) to explore the portfolio hierarchy interactively.

Project Cards Drilldown

Route: </dashboard/drilldown/cards>

Navigate through your combined portfolio in four levels: **Portfolio** → **Project** → **Work Package** → **Task**. Each card displays health indicators, status, and key metrics. At the task level, tabbed sections show related risks and issues. This view covers all pubsets consolidated in the active Dashboard Source.

The [Personal Dashboard](#) offers an equivalent drilldown scoped to your single active pubset.

Analytics

Route: </dashboard/analytics>

The Analytics section is planned for a future phase. It will provide performance and cost analytics for data-driven project decisions. Check this section for updates as new capabilities are released.

Shop Schedule (Gantt)

Route: </dashboard/facilities>

The Shop Schedule presents facility and shop tasks as a Gantt chart. Filter by schedule type and task status to focus on specific workstreams. Use this view for operational planning when projects involve physical facilities, workshops, or lab schedules.

Preprocessing Resource Data

Route: </dashboard/settings/preprocess>

Before the Resource Cost Charts and Resource Usage Charts can display data, the allocation rows in your active dashboard source must be extracted and reshaped into the resource calculations format ([tbrescalcs2](#)).

When to preprocess

- After setting a new active dashboard source
- If resource charts show no data or stale data

How to preprocess

1. Navigate to **Dashboard** → **Settings** → **Preprocess Resource Data**
2. Confirm the active dashboard source shown is the correct one
3. Click **Run Preprocess**
4. Wait for the operation to complete (a confirmation message will appear)
5. Return to the visualization pages — charts should now display updated data

Note: Preprocessing overwrites the previous resource calculations data for your organisation. If you switch the active source, run preprocess again before viewing resource charts.

Quick Reference

Task	Where to go
View executive summary	/dashboard
Browse all reports	/dashboard/reports
Create a dashboard source	/dashboard/pubsets
Switch active source	/dashboard/sources
View resource cost charts	/dashboard/visualizations/cost-charts
View resource usage charts	/dashboard/visualizations/usage-charts
View burndown charts	/dashboard/visualizations/burndown
Explore project hierarchy	/dashboard/drilldown/cards
View shop/Gantt schedule	/dashboard/facilities
Refresh resource chart data	/dashboard/settings/preprocess
Manage user roles	Strapi Admin → Content Manager → Users

Roles at a glance

Role	Create Sources	See All Pubsets	See Own Pubsets	See Granted Pubsets
Administrator	✓	✓	✓	✓
Project Manager	✓	—	✓	✓
Team Member	—	—	✓	✓
No role	—	—	—	—

Related Help Topics

- [Personal Dashboard Guide](#) — Single-pubset dashboard with zero configuration
- [Cloud Publishing Guide](#) — How to publish data to the cloud and manage pubsets
- [Local Reports Guide](#) — Reports available within client applications
- [Data Synchronization, Backup, Recovery and Retention Guide](#) — Understanding the data model

- [User Interface Guide](#) — Navigate the client apps
-

Support

Need Help?

- Knowledge Base: www.timebars.com/knowledgebase
 - Email Support: jcox@tbcox.com
 - Phone: (613) 255-5374
-

For technical issues, contact your system administrator or the Timebars support team.