Common Data Structure User Guide

This guide explains how your project data is organized, stored, and managed in Agilebars, Timebars, and Costbars applications. Understanding the data structure will help you make the most of features like spreadsheet sync, backups, and cloud publishing.

Overview: Where Your Data Lives

The Timebars applications follow a **"client-first" architecture**, which means:

- **Your data lives in your browser** using IndexedDB technology
- **No internet required** for full functionality—work completely offline
- **You control your data** with multiple export/import options
- **Cloud publishing is optional**—you choose when to share or sync

Three Data Storage Layers

Your data exists in up to three places, depending on how you use the application:

Storage Layer	Location	Purpose	User Control
IndexedDB	Your browser	Real-time work, immediate edits	Direct (via UI)
Spreadsheet Files	Your computer	Bulk editing, backups, offline work	Complete
Cloud Storage	Timebars Cloud	Team sharing, cross-device sync	User-initiated

Data Flow:

```
Your Browser (IndexedDB)
    ↓ ↑
[You control all movement]
    ↓ ↑
Excel/JSON Files ↔ Cloud Dashboard
```
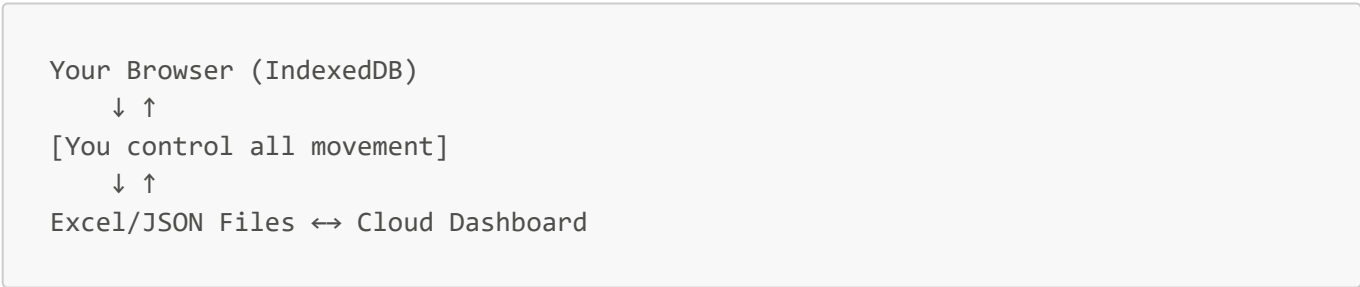
Understanding the Data Hierarchy

Timebars and Costbars support flexible hierarchies, while Agilebars uses a simple 2-tier structure. The hierarchy is maintained automatically as you drag and drop bars on the Canvas.

Color-Coded Levels

Each hierarchy level has a distinct color to make it easy to recognize:

- **L1 - Dark Brown**: Portfolio (or Program)
- **L2 - Green**: Project (always a Project in all products)
- **L3 - Orange**: Sub-Project or Work Package

- **L4 - Blue**: Task (always a Task in all products, includes milestones)
- **L5 - Gold**: Resource Allocation (Timebars and Costbars only)

Hierarchy Options

The applications support three hierarchy configurations:

5-Level Hierarchy (Maximum Flexibility)

```
L1 - Portfolio (Dark Brown)
  L2 - Project (Green)
    L3 - Sub-Project (Orange)
      L4 - Task (Blue)
        L5 - Allocation (Gold)
```

Use When: Managing large portfolios with multiple programs and complex sub-projects.

4-Level Hierarchy (Medium)

```
L1 - Portfolio (Dark Brown)
  L2 - Project (Green)
    L3 - Task (Blue)
      L4 - Allocation (Gold)
```

Use When: Small to medium projects where sub-projects aren't needed.

3-Level Hierarchy (Minimum)

```
L1 - Project (Green)
  L2 - Task (Blue)
    L3 - Allocation (Gold)
```

Use When: Simple projects without portfolio management.

2-Level Hierarchy (Agilebars Only)

```
L1 - Project (Green)
  L2 - Task (Blue)
```

Use When: Agile sprint management where allocations are managed differently.

Note: Many built-in rules prevent violations of these hierarchies. For example, you cannot drop an L4 Task on an L3 Sub-Project in a 4-level hierarchy.

The Data Model: What's Stored

Your data is organized into **data stores** (similar to database tables). There are two categories:

User-Controlled Data Stores

These stores contain data you create and manage:

1. tbAdminPanel - Application Settings

- **What it stores:** Timescale settings, zoom levels, visible hierarchy levels, product license info
- **Examples:** Time Scale Start Date, Report Date, Weekly/Monthly view, Dark mode setting
- **User control:** Modified through Tools > Canvas Settings and other configuration dialogs

2. tbTimebars - Core Project Data

- **What it stores:** All bars on the Canvas—portfolios, projects, tasks, milestones, allocations
- **Examples:** Bar names, start/finish dates, costs, hours, owner, status, hierarchy
- **User control:** Created and edited via Canvas, Bar Creator, and flyout forms

3. tbMetaData - Extended Metadata

- **What it stores:** Over 100 additional fields for tagging and categorizing bars
- **Examples:** Priority, health indicators, risk scores, department, investment category
- **User control:** Edited via Core Report, Metadata View, and Spreadsheet Sync
- **Picklists:** Most fields use picklists defined in tbTags store

4. tbResources - Resource Pool

- **What it stores:** People or things that can be allocated to tasks
- **Examples:** Resource names, roles, skills, departments, locations, pay rates, availability
- **User control:** Managed via Spreadsheet Sync or Resource Listing report

5. tbTags - Picklists and Lookup Values

- **What it stores:** All dropdown list values (picklists) used throughout the application
- **Examples:** Status values (Active, On Hold, Complete), Priority levels (High, Medium, Low)
- **User control:** Edited via Editable Grids (Reports > Data Management Grids > Tags tab)
- **Purpose:** Provides consistent terminology and enables filtering/grouping

6. tbFields - Field Definitions

- **What it stores:** Metadata about all fields in the application
- **Examples:** Field names, data types, whether field is required, default values
- **User control:** Advanced users can add custom fields by editing this store

- **Purpose:** Enables dynamic form generation (FOCD - Field-Oriented Control Design)

7. tbCoreReport - Core Report Configuration

- **What it stores:** Which fields to show in the Core Report for each hierarchy level (L1-L5)
- **Examples:** Show "Cost" for L2 Projects, hide "Hours" for L1 Portfolios
- **User control:** Configured via Spreadsheet template (Core Report worksheet)

System-Generated Data Stores

These stores are automatically calculated from your user-controlled data. You don't edit them directly—they're regenerated as needed.

1. tbBaseline - Baseline Snapshots

- **What it stores:** Copies of tbTimebars data when you set baselines
- **Purpose:** Compare current plan against original plan for variance reporting
- **Generated when:** You click "Set Baseline" button on any bar

2. tbMdJoined - Combined View

- **What it stores:** Joins tbTimebars + tbMetaData into single records (150 columns total)
- **Purpose:** Simplifies reporting—all data for one bar in one place
- **Generated when:** Canvas loads, or when you click "Recalculate All"

3. tbResCalcs - Burndown Chart Data

- **What it stores:** Time-phased hours and costs for burndown chart calculations
- **Purpose:** Powers Agilebars burndown charts
- **Generated when:** You run burndown chart calculation

4. tbResCalcs2 - Resource Demand Data

- **What it stores:** Weekly resource demand by role, skill, location, department, project
- **Purpose:** Powers resource allocation bar charts (stacked charts)
- **Generated when:** Canvas loads with allocations, or "Recalculate All"

5. tbResCalcsUsage - Resource Availability

- **What it stores:** Weekly availability of resources
- **Purpose:** Resource capacity planning
- **Generated when:** Resource data changes

6. tbCharts - Chart Post-Processing

- **What it stores:** Processed data optimized for chart rendering
- **Purpose:** Faster chart loading in Cloud Dashboard
- **Generated when:** You publish to Cloud Dashboard

How Data is Stored in Your Browser

IndexedDB Technology

What is IndexedDB?

- Built-in browser database (works in Chrome, Firefox, Edge, Safari)
- Stores structured data with indexes for fast searching
- Supports transactions (all-or-nothing operations)
- Persists data even when you close the browser
- Typically allows 50MB to 500MB+ per site

Database Details:

- **Database Name:** `timebarDB`
- **Current Version:** 7 (automatically upgrades when needed)
- **Storage Location:** Browser's IndexedDB partition
- **Typical Size:** ~5-20MB for 100-500 bars, up to 100MB for large portfolios

What This Means for You:

- Data survives browser restarts
- Data is isolated per browser (Chrome data separate from Firefox data)
- Data is specific to the domain (tb.timebars.com vs cb.timebars.com)
- Clearing browser cache may delete data—**always backup first!**

Data Import and Export

Spreadsheet Sync

The most powerful way to bulk edit your data is with the **Spreadsheet Sync** feature.

How It Works

1. **Export from App:**

   - Click Hamburger Icon > Export to CSV
   - 13 CSV files download to your browser's Downloads folder
   - Each file corresponds to one data store

2. **Import to Spreadsheet:**

   - Open the **tbClientMaster.xlsm** template (download from Admin Menu)
   - Click "Import All" button on Setup worksheet
   - Data loads into worksheet tabs (Timebars, MetaData, Resources, etc.)

3. **Edit in Spreadsheet:**

   - Use Excel features: formulas, copy-paste, fill-down, filters
   - Add new rows (tasks, resources)

- Modify existing data
- Use dropdown lists for picklist fields

4. **Export from Spreadsheet:**

   - Spreadsheet automatically exports CSVs to configured location
   - Save the spreadsheet file

5. **Import to App:**

   - Drag and drop the spreadsheet file onto the Canvas
   - App reads data and updates IndexedDB
   - Canvas refreshes with updated data

Spreadsheet Structure:

- **Row 1:** Database field names (e.g., `tbID`, `tbName`, `tbStart`)
- **Row 3:** User-friendly labels (e.g., "ID", "Name", "Start Date")
- **Row 4+:** Your data

Important Notes:

- System-generated stores (tbBaseline, tbMdJoined, etc.) are not in the spreadsheet—they regenerate automatically
- Save spreadsheet before dragging back to app
- App automatically creates backup before importing spreadsheet

---

JSON Backup Files

For complete system backups, use JSON format.

Full Backup

- Click **Hamburger Icon > Full Backup**
- Creates file: `tbFullBackup_2025-03-06_16-16-05.js`
- Includes ALL data stores (user-controlled AND system-generated)
- Timestamp in filename: `_YYYY-MM-DD_HH-MM-SS`

Restore from Backup

- Drag and drop backup file onto Canvas
- App loads all data and returns to exact state when backup was taken
- **Warning:** Overwrites current data—make a backup first if needed!

JSON Export (Individual Stores)

- Click **Hamburger Icon > Export to JSON**
- Creates separate JSON files for each store
- Useful for custom integrations, AI agents, or external tools

---

CSV File Import

You can import individual CSV files:

1. **From Spreadsheet:**

   - Export single worksheet as CSV from Excel
   - Drag and drop CSV onto Canvas
   - App imports to corresponding store

2. **From External Systems:**

   - Format CSV with Row 1 = field names
   - Match field names exactly to tbTimebars, tbMetaData, etc.
   - Drag and drop onto Canvas

Browser Download Location

Your exported files save to your browser's download folder. Here's how to find or change it:

Default Download Locations

- **Windows:** `C:\Users\<YourUsername>\Downloads`
- **Mac:** `/Users/<YourUsername>/Downloads`
- **Linux:** `/home/<YourUsername>/Downloads`

Changing Download Location

Google Chrome:

1. Settings > Advanced > Downloads
2. Click "Change" to set new folder
3. Optional: Enable "Ask where to save each file"

Mozilla Firefox:

1. Settings > Files and Applications
2. Under Downloads, choose folder or "Always ask"

Microsoft Edge:

1. Settings > Downloads
2. Click "Change" to set new folder

Safari (Mac):

1. Preferences (Cmd + ,)
2. General tab > File download location

Advanced Data Operations

Duplicate Bars and Children

To copy a bar with all its children:

1. Choose **Hamburger Icon > Bulk Manage Bars**
2. Enter the **Bar ID** (find bottom-left of bar on Canvas)
3. Click **"Duplicate Bars and Children"**
4. Duplicates appear on Canvas near original
   - Names prefixed with "duplicate"
   - IDs suffixed with "-d"

Use Cases:

- Template projects to reuse
- Create multiple similar tasks
- Testing "what-if" scenarios

Transfer Bars Between Products

Move bars from one product to another (e.g., Timebars to Agilebars):

1. Choose **Hamburger Icon > Bulk Manage Bars**
2. Enter the **Bar ID** to transfer
3. Enter optional **prefix** for bar ID (helps avoid duplicate IDs)
4. Click **"Transfer Bars and Children"**
5. Check Downloads folder for file: `transferBars20250307-112848.json`
6. Open the target product (e.g., Agilebars)
7. Drag and drop the transfer file onto Canvas

Use Cases:

- Waterfall project becomes Agile (Timebars → Agilebars)
- Agile work package becomes portfolio item (Agilebars → Costbars)
- Share project templates across products

Delete Database and Start Over

To completely reset the application:

1. Choose **Hamburger Icon > Delete Database**
2. **Warning:** This destroys ALL data permanently!
3. Browser history remains (page visits), but data is gone
4. Refresh browser to rebuild database with fresh Demo Data

When to Use:

- Starting completely fresh
- Troubleshooting corrupted data
- Switching to different project

Always backup first!

File Importer (Alternative to Drag-Drop)

If you don't like drag-and-drop, use the File Importer:

1. Choose **Hamburger Icon > File Importer**
2. Click **Browse** button
3. Select file(s) from your computer
4. App imports selected files

Supported Files:

- Spreadsheet (.xlsx, .xlsm)
- JSON backups (.js, .json)
- CSV files (.csv)

Automatic Backups

The application automatically creates backups before major operations:

- **Before Spreadsheet Sync:** Backup created in Downloads folder
- **Before Demo Data Load:** Backup created automatically
- **Before Delete Database:** Backup offered (optional)

Backup Naming: `tbFullBackup_YYYY-MM-DD_HH-MM-SS.js`

Tip: Keep recent backups organized in a "Timebars Backups" folder on your computer.

Data Validation and Integrity

Built-In Data Validation

The application validates data in multiple places:

1. User Interface:

- Required fields enforced
- Date pickers ensure valid dates
- Dropdown lists enforce picklist values
- Numeric fields reject non-numbers

2. Spreadsheet Import:

- Field names validated against schema
- Data types checked (dates, numbers, text)
- Foreign key references verified (e.g., resource exists)
- Unique constraints enforced (e.g., no duplicate bar IDs)
- Pre-import validation report shows errors before importing

3. Referential Integrity:

- Deleting a parent bar deletes all children (cascade delete)
- Orphaned records detected and cleaned up
- Foreign key relationships maintained

4. Transaction Atomicity:

- Multi-store operations are all-or-nothing
- If any error occurs, entire operation rolls back
- Ensures data consistency

Cloud Publishing and Synchronization

For team collaboration and cross-device access, you can optionally publish data to the Timebars Cloud.

How Cloud Publishing Works

Step 1: Publish to Cloud

1. Main Menu > Publish
2. Log in to cloud service (secondary login for security)
3. Click "Activate" (first time only)
4. Click "Re-Publish!" on PubSet with blue background
5. Data uploads to Strapi Cloud storage

Step 2: Access Cloud Dashboard

- Visit www.timebars.com/dashboard
- View reports, charts, visualizations
- Share with team members (same subscription)

Step 3: Re-Hydrate on Another Device

1. Open same app on different device (e.g., iPad)
2. Main Menu > Publish
3. Log in to cloud service
4. Click "Re-Hydrate!" on PubSet
5. Data downloads to device

Cloud Storage Details:

- **Technology:** Strapi CMS with PostgreSQL database
- **Security:** HTTPS encryption, JWT authentication
- **Control:** User-initiated only, never automatic
- **PubSets:** Up to 10 separate datasets (depends on subscription)
- **Dashboard PubSet:** Blue background—updates Cloud Dashboard

Data Flow:

```
Device A: Publish → Cloud Storage ← Re-Hydrate: Device B
                          ↓
                   Cloud Dashboard
                   (Reports & Charts)
```

See Cloud Publishing Guide for complete details.

Data Ownership and Privacy

You Own Your Data

Complete Control:

- Data resides in YOUR browser by default
- No server storage unless YOU choose to publish
- Export anytime in standard formats (Excel, JSON)
- Delete anytime with full database wipe
- No vendor lock-in—standard file formats ensure portability

Privacy Guarantees:

- Application works fully offline—no internet required
- No cookies stored in your browser
- No automatic data collection or telemetry
- Cloud publishing is opt-in, not required

Security:

- HTTPS/TLS encryption for all cloud communication (bank-level)
- JWT token authentication for cloud access
- Passwords encrypted before storage
- Cloud data isolated by user account

Troubleshooting Data Issues

Data Not Appearing After Browser Restart

Cause: Browser cache cleared or IndexedDB deleted

Solution:

1. Restore from most recent backup file
2. Drag-drop backup onto Canvas
3. Consider enabling browser setting to NOT clear cache on exit

---

Spreadsheet Import Fails with Validation Errors

Cause: Data doesn't match expected format or field names changed

Solution:

1. Check Row 1 field names match exactly (case-sensitive)
2. Review validation error messages
3. Fix errors in spreadsheet
4. Use "Partial Import" option to import only valid rows

Cloud Re-Hydrate Overwrites Recent Local Changes

Cause: Re-Hydrate replaces ALL local data with cloud data

Solution:

1. **Before Re-Hydrate:** Always backup local data first
2. Compare timestamps: Check "Last Published" date on PubSet
3. If local data is newer, publish first, then re-hydrate

Browser Shows "Storage Quota Exceeded" Error

Cause: IndexedDB storage limit reached (rare, usually 50MB+)

Solution:

1. Export data to spreadsheet or JSON
2. Delete database to free space
3. Import only essential data back
4. Consider splitting large portfolios across multiple products

---

Data Appears Corrupted or Inconsistent

Cause: Browser crash during write operation, or manual editing of IndexedDB

Solution:

1. Click **Recalculate All** (Right-click Canvas > Recalculate All)
2. Regenerates all system-generated stores
3. If still broken, restore from backup

Best Practices

Regular Backups

Recommended Schedule:

- **Daily:** If actively working on critical projects
- **Weekly:** For stable, ongoing work

- **Before Major Changes:** Always backup before:
    - Spreadsheet sync
    - Bulk operations
    - Demo data loads
    - Version upgrades

Backup Strategy:

1. Create "Timebars Backups" folder on your computer
2. Organize by date: `2025-03-06_ProjectName_Backup.js`
3. Keep at least 3-5 recent backups
4. Archive older backups monthly

Spreadsheet Sync Workflow

Best Practice:

1. Export CSV files from app
2. Import to spreadsheet, make changes
3. Save spreadsheet with descriptive name
4. **Test import on demo data first** (if major changes)
5. Backup before importing to production data
6. Drag spreadsheet to Canvas
7. Verify data looks correct
8. Keep spreadsheet file as documentation

Multi-Device Usage

Best Practice:

1. Designate one device as "primary"
2. Always publish from primary after changes
3. Always re-hydrate on secondary before working
4. Avoid editing on multiple devices simultaneously
5. Establish team convention: "Last person to edit publishes"

Data Hygiene

Keep Data Clean:

- Delete test/demo bars before production use
- Archive completed projects (or delete if not needed for historical reference)
- Maintain consistent naming conventions
- Use tags/metadata consistently across team
- Periodically review and clean up unused resources

Related Help Topics

- [User Interface Guide](#) - How to create, edit, and manage bars
- [Cloud Publishing Guide](#) - Publishing to Timebars Cloud
- [Cloud Reports Guide](#) - Using Cloud Dashboard
- [Local Reports Guide](#) - Reports in the app

Summary

Key Takeaways:

1. **Data Lives in Your Browser** - IndexedDB stores everything locally
2. **You Control Movement** - Nothing syncs automatically; you decide
3. **Three Storage Options** - Browser, Spreadsheet, Cloud (all optional)
4. **Flexible Hierarchies** - 2-5 levels depending on your needs
5. **Two Store Types** - User-controlled (you edit) and System-generated (automatic)
6. **Multiple Import/Export** - Spreadsheet, JSON, CSV—choose what works
7. **Always Backup** - No undo, so backup before major operations
8. **Offline-First** - Full functionality without internet

Your data, your control, your way.