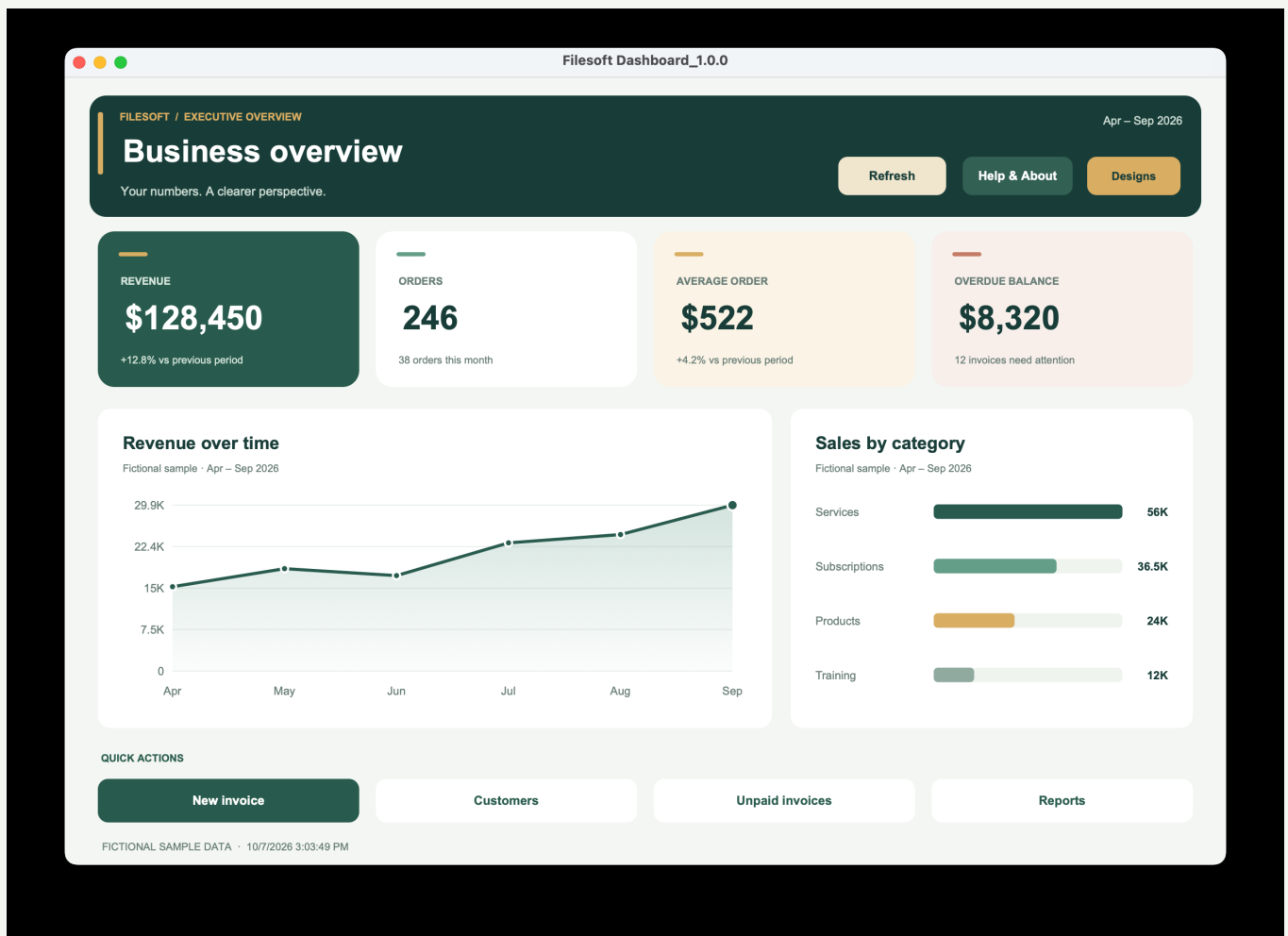


Your data. Four perspectives.

An illustrated setup guide for your existing FileMaker solution



Connect once. Choose your workspace.

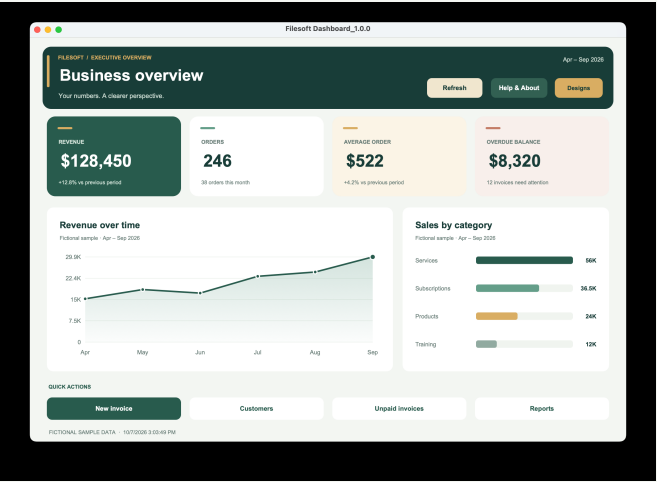
Four KPI cards, two embedded charts and four action slots share one data provider. A FileMaker developer maps your existing tables and scripts; the dashboard does not automatically discover your business model.

Before you begin: use a copy of your solution, keep the original release files, and decide which invoices, dates and currency your dashboard represents.

Owner-tested in FileMaker 19 and 26 on macOS; exact tested point releases were not restated. The 1.0.1 add-on was exported in 19.6.3. Windows, Go, WebDirect and other versions are not verified. Screenshots show the unchanged 1.0.0 demo.

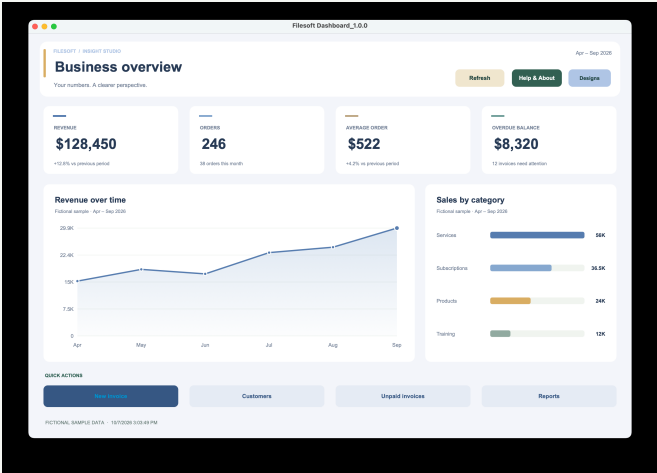
Choose your design

The same figures, actions and refresh timestamp in every layout



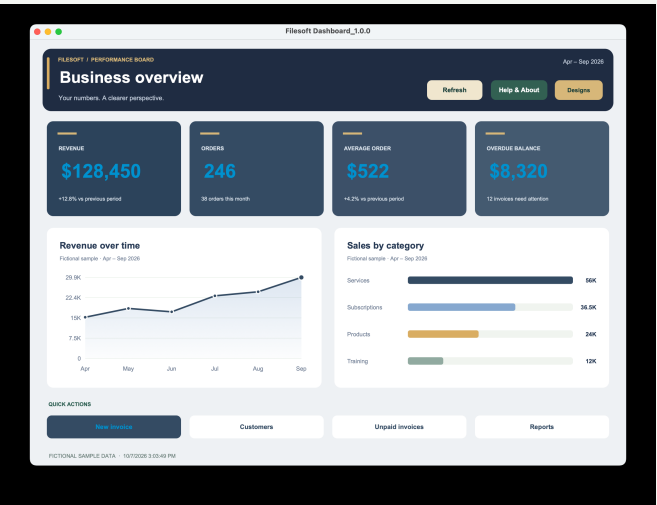
Executive Overview

1120 x 760 pt | Forest green and gold; a featured KPI.



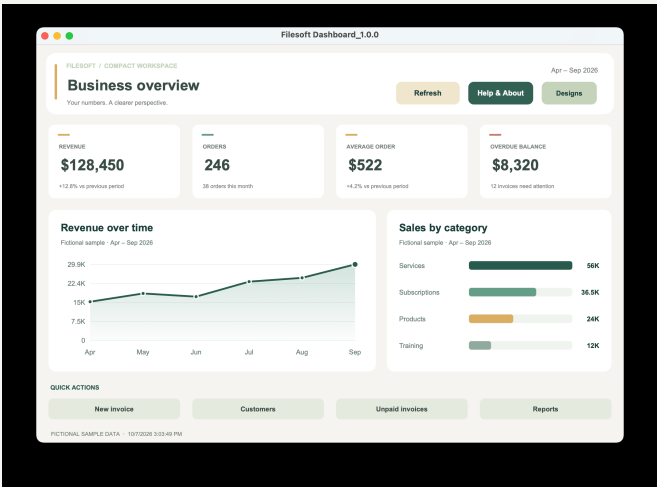
Insight Studio

1280 x 860 pt | Airy blue layout; more chart space.



Performance Board

1120 x 800 pt | Strong KPI cards; equal chart panels.



Compact Workspace

960 x 650 pt | A smaller desktop composition.

Click **Designs** to open the chooser. Selection opens a supplied layout; it does not rearrange a group already placed on a customer layout. Design choice is navigation, not a saved user preference.

The draggable group uses Executive Overview. To embed another design, copy its complete group in Layout mode and preserve its object names. The layouts have fixed dimensions.

Install and see the example

Start in a disposable copy before connecting business data

1. Register the package

Open Filesoft Dashboard_1.0.1.fmaddon with your intended FileMaker version. If the entry is missing, save your work, quit that version and reopen it before checking the chooser.

2. Install into the target file

In Layout mode, open the Objects pane, select Add-ons, then Install Add-on. Choose Filesoft Dashboard 1.0.1. Installation adds its scripts, layouts and FSDASH_Context table shell.

3. Place the dashboard

Drag the installed component onto a desktop Form View layout with at least 1120 x 760 points of clear space. Keep it outside portals; use one placed dashboard per layout.

4. Load the example

Return to Browse mode and click Refresh. The fictional-data badge should remain visible for sample data. Try Designs and Help & About, then Back to dashboard.

Where to work

Script Workspace: FMDASH > Dashboard > Controller contains FSDASH | Data Provider and FSDASH | Action Router. Button entry scripts are in UI. Help has its own UI and Controller folders.

Manage Layouts: FMDASH > Desktop contains Dashboard, Choosers and Help. Scripts and layouts are hidden from user menus; use the dashboard buttons to navigate.

Installing a new package does not migrate a dashboard already installed in a host file. Preserve customized components and records; do not uninstall a populated component as an upgrade shortcut.

Connect your figures

Your provider returns JSON; the dashboard validates and displays it

Stage	Responsibility
Your tables / host scripts	Choose the reporting scope and collect authorized figures.
FSDASH Data Provider	Return one complete JSON object using Exit Script.
FSDASH Render	Validate the payload, retain it in this session and refresh the visible objects.
Four designs	Read the same snapshot; changing designs does not fetch data.

Three decisions before coding

Scope: define the date range, tenant/team/customer, posted/void status and currency. Apply the same scope to every query. ExecuteSQL examples read accessible records independently of the current found set.

Meaning: distinguish net invoiced, cash received and outstanding balance. The supplied invoice example uses invoice-date totals before tax; it is not cash revenue or a full receivables report.

Presentation: format KPI values as strings in your provider. Chart values remain raw numbers. Build JSON with JSONSetElement so quotes and special characters are escaped correctly.

Preserve the caller

Read without changing the caller's layout, found set, active record or open transaction. Keep the caller's privileges. Do not enable full-access execution to expose restricted totals. Return an empty/invalid payload on failure so the existing snapshot is retained with an error.

Use **sample=false** only for real figures. The period label is descriptive; there is no built-in date-filter editor. Your provider or existing host UI owns the filters.

The JSON contract

Version 1: exact counts for cards and actions; bounded chart arrays

Key	Required value
version / sample	Number 1 / JSON Boolean true or false
title, subtitle, period	Strings
kpis	Exactly four objects: label, value and note strings
actions	Exactly four label strings; fixed slots 1, 2, 3, 4
trend	title string; labels string array; values number array; 0-24 points
breakdown	title string; labels string array; values number array; 0-8 categories

Each chart needs matching label/value counts. Empty arrays mean no data. Zero and negative values are valid. Supply points in display order; the dashboard does not sort them. Use one unit per chart.

A complete, minimal payload

```
{
  "version": 1, "sample": true,
  "title": "Invoice overview", "subtitle": "Example only",
  "period": "September 2026",
  "kpis": [
    {"label": "NET INVOICED", "value": "1200", "note": "Before tax"},
    {"label": "INVOICES", "value": "3", "note": "In period"},
    {"label": "AVERAGE", "value": "400", "note": "Per invoice"},
    {"label": "OVERDUE", "value": "200", "note": "Due today"}
  ],
  "actions": ["New invoice", "Customers", "Unpaid", "Reports"],
  "trend": { "title": "Net invoiced", "labels": ["Sep"],
    "values": [1200] },
  "breakdown": { "title": "By category", "labels": ["Services"],
    "values": [1200] }
}
```

Suggested label lengths: title 36, subtitle 80, KPI label 22, value 12, note 32 and action label 22 characters. Check all designs with your actual formats.

Example: existing invoices

A complete provider script outline is included as editable text

Open **Invoice Provider Example.txt**. It is a developer adaptation example, not an installer or an automatically imported script. Map its identifiers and policies first, then recreate the steps inside your provider and verify in your host solution.

Assumed field in Invoices	Meaning / required type
InvoiceDate / DueDate	Date fields, not timestamps or formatted text
NetTotal / BalanceDue	Number fields; valid numeric amounts in one currency
IsVoided	Number: 0 for included invoices, 1 for void invoices
Category	One category per invoice; text

What the example calculates

The last six completed calendar months supply net invoiced, invoice count and average invoice value. Monthly sums cover the same period. Services, Subscriptions and Products are explicit categories; Other includes all remaining or blank categories.

Overdue balance includes only invoices in that same invoice-date period with positive BalanceDue and DueDate before today. To show all receivables, deliberately change that query and label its wider scope.

Queries fail visibly

Each scalar ExecuteSQL result is checked for "?" before conversion. A query error exits with an empty payload; it never becomes a false zero. A successful SUM over no matching rows is normalized to zero. Validate missing source amounts upstream.

Do not join invoice headers to lines and then sum header totals: multiple lines would count the same invoice repeatedly. For line-category reporting, aggregate line amounts once and reconcile them with the chosen invoice totals.

The example uses quoted, literal identifiers for an explicitly assumed schema and positional parameters for values. Rename identifiers deliberately; no customer fields or SQL helper custom functions are assumed. It has not been executed against your schema.

Connect the four actions

Labels and script destinations are configured separately

Open **FSDASH | Action Router** under FMDASH > Dashboard > Controller. Keep the existing guards and four slot branches. Replace each branch's not-connected return with a native Perform Script call to your existing workflow.

Slot	Example destination
1 / New invoice	Your existing invoice creation controller
2 / Customers	Your customer list or search workflow
3 / Unpaid invoices	Your unpaid-invoice report or filtered list
4 / Reports	Your existing reports screen

Return a clear result

Success:

```
JSONSetElement ( "{}" ; "ok" ; 1 ; JSONBoolean )
```

Failure:

```
JSONSetElement ( "{}" ;  
  [ "ok" ; 0 ; JSONBoolean ] ;  
  [ "message" ; "Describe the problem" ; JSONString ] )
```

Check `Get ( LastError )` immediately after Perform Script, then read `Get ( ScriptResult )`. Translate the host script's success, failure and cancellation convention. A successful call does not prove the business operation succeeded. The accompanying Action Router Example.txt explains the adapter.

Keep UI and business logic separate

UI handlers show messages; controllers return results. Preserve the host workflow's permissions and transaction boundaries. The dashboard does not evaluate provider labels as executable script names.

After an action changes business records, return to the dashboard and Refresh. If automating that step, call **FSDASH | Refresh** while the dashboard is active and handle its JSON result. Do not overwrite existing startup or layout triggers.

Customize and verify

Keep the shared data contract while adapting the presentation

Style the native and web parts separately

Edit native fonts, fills and sizes in Layout mode. The chart palettes and typography are inside each web viewer's embedded HTML/SVG calculation; changing a FileMaker theme does not automatically restyle them. Preserve the stored calculation documentation header when editing formulas.

Keep FSDASH object names: Refresh uses them. Keep the supplied layout names: navigation uses them. Maintain readable normal, hover and pressed states, particularly on dark KPI cards and action buttons.

Understand the session snapshot

\$\$FSDASH_Data and \$\$FSDASH_Updated are global variables within one file/session. They are not saved records or per-window state. Use one active dashboard window per file/session; refresh again after reopening. Changing designs preserves the snapshot and its timestamp.

Check your integration

- Reconcile every total against the same scope in an existing report.
- Check zero invoices, no matching categories, negative adjustments and malformed payloads.
- Confirm a failed refresh retains previous figures and their old timestamp.
- Run each action as an ordinary user, including access-denied and cancellation paths.
- Check Help, Back and Designs with every layout and with unsaved edits.
- Review labels, currency formatting and chart rendering on each intended client.

Refresh and navigation refuse open edits or transactions. Save or cancel those edits explicitly; the dashboard does not commit or revert them for you.

Help, terms and references

Keep this guide with your configured solution

Symptom	Check
Not in the FM19 chooser	Use add-on 1.0.1, reopen the intended client and check registration.
Invalid data / blank figures	Complete JSON keys, correct types and equal chart-array counts.
Action not connected	Map the corresponding Action Router slot; labels are not bindings.
Script cannot be found	Native button binding and installed FSDASH scripts.
Design or Help cannot open	Open edits/transactions, layout access and preserved layout names.

Usage and delivery

Free download after name and email collection; the download link is delivered by email. Personal and internal-business use and modification are allowed. Resale and redistribution are not allowed. Support is best-effort, with no promised response times or updates. Downloading does not automatically subscribe you to marketing.

Support

filesofttools.com | support@filesofttools.com

Customization and implementation services | [Filesoft guides](#)

Technical references

Claris: [Working with add-ons](#); [ExecuteSQL](#); [JSON functions](#); [FileMaker 19 SQL Reference](#). Sources checked 7 October 2026.

Compatibility evidence: the owner confirmed add-on 1.0.1 works in FM19 and FM26. This is separate from validating your own data mapping. The invoice example is documentation code and has not been runtime-tested or installed in the delivered native files.