

A Costing Sheet in Every New Project Folder, for Construction and Manufacturing

When CloudFiles creates the project folder in SharePoint, the costing sheet is generated from the Opportunity and placed inside it.

The situation: the folder appears, the costing sheet does not

Construction contractors, engineering firms and manufacturers run each won deal as a project with a standard folder structure: drawings, contracts, purchasing, site records and costing. With CloudFiles Document Management, that structure is copied from a folder template in SharePoint as soon as the project starts, so every project looks the same. The first document the project team needs in that folder is the costing sheet, built from the quantities, costs and prices on the Opportunity. Estimators usually rebuild it by hand, copying lines from Salesforce into a spreadsheet.

CloudFiles Document Generation, part of CloudFiles Document Automation for Salesforce, closes the gap. When the project folder is created, a flow generates the costing sheet from the Opportunity and writes it into the new folder, so the folder arrives with its first document already inside.

How do you automate document generation from Opportunity templates?

1. The object and its data

The master record is the Opportunity. The template reaches:

- Opportunity fields: project name, customer, site address, close date, estimator.
- The opportunity line items: product or work package, product family, quantity, unit cost and sales price.
- Account and owner details through relationship paths.

2. The template, built in Excel

The costing sheet is an Excel template built with the CloudFiles Office add-in, because the estimators work in Excel and want to keep working in it. Excel templates support fields and tables:

- A header block of fields: project, customer, site and dates.
- A table that repeats one row per opportunity line item, sorted by product family, with quantity, unit cost and price.

- A summary table built from a Record Collection resource grouped by product family, with SUM of cost and price for each family.
- Number and currency patterns with group and decimal separators, and locale formatting.

Conditions and loops are Word features, so the Excel template keeps to fields and tables, which is exactly what a costing sheet needs. Build, preview against a real Opportunity, then publish.

3. The trigger: a CloudFiles event

The trigger is not a button and not a stage change. It is a CloudFiles event: the project folder being copied from the folder template in SharePoint. A flow listens for that event and starts as soon as the new folder exists.

4. The Flow action

The flow calls the Generate Document action for the Opportunity with the costing template, and sets the destination to the new project folder. The output is XLSX, so the estimators open a live workbook. Teams that also want a fixed record of the original costing can generate a PDF copy from the same template in the same flow.

5. Where the file goes

The costing sheet is written into the project's folder in SharePoint, in the costing subfolder or wherever the folder template expects it. It sits next to the drawings and contracts from the first minute of the project. The same pattern works with OneDrive, Google Drive, Box or Dropbox.

What else can the new folder receive?

The same event can drop more than one document into the folder. Common additions:

- A **Word** project kickoff letter to the customer, with conditional sections for the type of work.
- A **PowerPoint** kickoff deck for the internal team, with fields and pictures such as the site photo and the customer's logo.
- A **PDF** scope summary for the site manager.

Each is a separate published template, and each generation uses one credit.

Why project teams choose this pattern

- The costing sheet is ready when the folder is, so no project starts from a blank spreadsheet.
- Estimators keep working in Excel, and the template itself is an Excel file they can read and adjust.
- Generation and document storage come from one vendor in one package, so the event that creates the folder and the action that fills it work together without extra integration.
- If the team later adds a signed contract to the same folder, CloudFiles DocuSign signing returns the signed file to the record or its folder.

What to set up

- Install CloudFiles from AgentExchange with Document Management connected to SharePoint, and set up the project folder template.
- Install the CloudFiles Office add-in from Microsoft AppSource and build the costing template in Excel.
- Preview against an Opportunity with a full set of line items, then publish.
- Build the flow on the CloudFiles folder-copied event, call Generate Document for the Opportunity, and set the new folder as the destination.
- Add any further Word, PowerPoint or PDF templates the new folder should receive.
- Test the whole chain in a sandbox, then move the templates and flow to production.

Get started

Install CloudFiles from AgentExchange and start a free 14-day trial, or book a demo at cloudfiles.io.