

Automate SharePoint Folders and File Uploads with Salesforce Flow

Worked Salesforce Flow SharePoint patterns for Accounts, Opportunities, Cases and custom objects

What does Salesforce Flow SharePoint automation cover?

CloudFiles is Document Automation for Salesforce. CloudFiles Document Management adds Flow actions that turn Salesforce Flow into a SharePoint folder automation engine: you can automate SharePoint folder creation from Opportunities, automate SharePoint folder setup for new Accounts, automate file uploads from Opportunities to SharePoint, and automate record-linked folder structures on any standard or custom object. The same actions work against AWS S3, Google Drive and Azure Blob, so one Flow design serves every storage you connect.

This guide covers what a Flow needs before you build, the actions available, and three worked patterns with numbered steps.

What do I need before building a Salesforce Flow SharePoint automation?

- **A connection that runs automation.** Flow automation runs as a SharePoint service principal or an integration user. A service principal authenticates as the CloudFiles app in your Microsoft tenant, uses no license, has no password or MFA to break, and shows an "Active for automations" badge when it is ready.
- **Registered SharePoint sites.** Your Microsoft 365 admin grants CloudFiles access site by site, and each new site needs its own grant before a Flow can use it.
- **Location IDs.** In the Content Library, "View Metadata" shows the Library, Site Id, Drive Id and Folder Id for any location, each with a copy button. Paste these into your Flow actions.
- **Least privilege on the Salesforce side.** Assign a least-privilege Salesforce permission set.
- **Conflict behavior.** Per library, choose what happens when a file or folder already exists: Rename, Replace, No Op or Fail.

Users working in the widget and Content Library act through their own storage connection. Flows act as the automation identity, so they keep running regardless of who edits the record.

Which Flow actions does CloudFiles add?

All actions are available in Salesforce Flow and from Apex.

Area	Actions
Folders	Create Folder (sync and async), Get Connected Folder, Create SharePoint Site, Create SharePoint Drive, Create Microsoft Team
Files	Copy Salesforce Files, Create Salesforce File(s), Move Resource, Copy Resources, Delete Resource, Create File Request
Metadata	Update Drive Columns, Set/Get Resource Property, Search by Resource Metadata
Permissions	Update Permissions, Create SharePoint Site Group, Update SharePoint Site Group Members
Sharing and Experience Cloud	Create Link, Update Link Settings, Share with Community User, Remove from Community, Get Shared Resources

Synchronous and asynchronous actions

About a third of the actions are asynchronous. They return nothing to the Flow that calls them; completion arrives later as a CloudFiles event, which a record-triggered or platform-event Flow handles. Put any step that depends on the result, such as removing a Salesforce copy or notifying a user, in that completion handler.

The "file attached" event setting ships off. Leave it off unless a Flow needs to react to each attachment, because turning file-attached events off reduces API calls.

Pattern 1: How do I automate SharePoint folder setup for new Accounts?

A pattern we see in IT services: every new Account gets its own SharePoint location, with a standard folder layout and access that follows the account team.

1. Create a record-triggered Flow on Account that runs when a record is created.
2. Add **Create Folder**. Set the Site Id and Drive Id you copied from View Metadata, and name the folder from the Account Name. Use the async variant for high-volume loads.
3. Add further **Create Folder** calls for the subfolders your team uses, such as Contracts, Onboarding and Invoices. This is how you automate record-linked folder structures: every Account starts with the same layout.
4. Add **Update Drive Columns** to stamp Account fields, such as industry and region, onto SharePoint columns.
5. Add **Update Permissions** to grant the Account owner write access by Salesforce user.
6. For key accounts that need their own site, use **Create SharePoint Site** and **Create SharePoint Drive** instead of a folder, then **Create SharePoint Site Group** and **Update SharePoint Site Group Members** so membership follows owner, team and territory.
7. If the account team collaborates in Microsoft Teams, you can Create Microsoft Team from Salesforce Flow in the same run.

The folder link survives record renames, so the Account keeps its folder when its name changes.

Pattern 2: How do I automate SharePoint folder creation from Opportunities and file uploads when a deal closes?

A pattern we see in energy development and private equity: each Opportunity gets a deal folder under its Account, and documents are filed when the deal closes.

When the Opportunity is created

1. Create a record-triggered Flow on Opportunity that runs when a record is created.
2. Add **Get Connected Folder** for the parent Account to find the Account folder.
3. Add **Create Folder** inside it, named from the Opportunity Name, plus any template subfolders such as Proposal, Legal and Diligence.
4. Add **Set Resource Property** to record the Opportunity stage on the folder.

When the stage changes to Closed Won or Closed Lost

1. Create a record-triggered Flow on Opportunity that runs when StageName changes to Closed Won or Closed Lost.
2. Add **Get Connected Folder** for the Opportunity.
3. On Closed Won, add **Copy Salesforce Files** to automate file uploads from Opportunities to SharePoint: the signed contract, quotes and email attachments land in the deal folder.
4. Add **Update Drive Columns** to stamp the close date, amount and stage onto SharePoint columns.
5. Add **Create Link** with an expiry and a view limit, and **Update Link Settings** to require email verification, if the customer needs a copy.
6. On Closed Lost, add **Move Resource** to move the folder into an archive library, and **Update Permissions** to revoke write access.
7. If you copy files asynchronously and plan to remove the Salesforce copies, do that removal in a separate Flow that handles the CloudFiles completion event.

Pattern 3: How do I automate a Case folder on status change?

A pattern we see in utilities and support teams: a Case gets a folder only when it needs one, and the customer can add files to it.

1. Create a record-triggered Flow on Case that runs when Status changes to a value such as Escalated.
2. Add **Create Folder** (async) under the Account's folder, named from the Case Number.
3. In a platform-event Flow that handles the completion event, add **Copy Salesforce Files** to copy files already attached to the Case into the new folder.
4. Add **Create File Request** so the customer can upload screenshots and log files straight into the Case folder, with no account in SharePoint, AWS S3 or Azure Blob.

5. For Experience Cloud portals, add **Share with Community User** so the portal user sees the folder on their Case.
6. When Status changes to Closed, add **Update Permissions** to set external access to read-only, or **Remove from Community** to withdraw it.

How do I automate record-linked folder structures on custom objects?

Every action above works on custom objects. For example, a legal team can create a folder structure on a custom Matter object, or a housing agency can create folders on an Application object when its status changes. Use a scheduled Flow with **Get Connected Folder** to find records that have no folder yet and backfill them.

How does SharePoint folder automation keep permissions in step?

Trigger a Flow on owner or team change and call **Update Permissions**, which grants or revokes read or write access by email or by Salesforce user or group. Site-group actions do the same at site level. Permissions and metadata flow from Salesforce to SharePoint; files and folders are bi-directional, so files added in SharePoint appear on the record once the folder is linked.

Security

- ISO/IEC 27001:2022 certified, with ISO/IEC 27017:2015 and ISO/IEC 27018:2019.
- Independently audited for SOC 2 Type II.
- Compliant with HIPAA (BAA available). Compliant with GDPR.
- Your files stay in your own storage. CloudFiles transfers them over encrypted connections and does not re-host them.

Get started

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