FlightBrief: Administrator Guide

Applies to FlightBrief 1.0.0.

FlightBrief writes a brief of a Salesforce org for the AI assistants and people who build in it. This guide covers installing it, deciding who generates and who reads briefs, setting up the deep scan, and the administrator's tools: company standards, questions for your team, publishing and schedules.

Contents

1. What it does
2. Install
3. Brief levels
4. Set up the deep scan
5. Generate and use a brief
6. Publish a brief for the team
7. Company standards
8. Schedule briefs
9. Questions for your team
10. What a brief can see
11. Uninstall
12. Support

1. What it does

FlightBrief generates a Markdown brief of a Salesforce org: its schema and how objects relate, how to query them, the Apex conventions in use and the Flow standards. People hand the brief to an AI assistant before it builds in the org, or give it to a new admin or developer.

FlightBrief reads metadata only, never record data, and only what the person running it can already see. Nothing leaves Salesforce: the brief is saved as a file in the org.

FlightBrief is free. Support is by email at support@adeptapps.net.

2. Install

1. Install the package from its install link, for admins only.

2. Assign the permission sets: - **FlightBrief User** to everyone who will generate briefs. It opens the FlightBrief app and tab and lets the user see their own briefs. It grants no system permission. - **FlightBrief Admin** to the administrators who will set up the deep scan, maintain the company standards and publish briefs for the team. It adds the FlightBrief Setup tab. Give them FlightBrief User too, so they can generate. The deep scan setup steps also need Customize Application. - **FlightBrief Reader** to everyone who should read the brief an administrator published, without generating their own: view, copy and download only.
3. Open the App Launcher and choose **FlightBrief**.

A Surface brief works right away. A Deep brief needs the one-time setup below.

Prefer to watch? Installing, the three setup steps and sharing a brief with your team are shown in a short video (2 min 52 s, on YouTube).

Who generates

Many orgs let only administrators generate: give FlightBrief User and FlightBrief Admin to the administrators, and FlightBrief Reader to the developers and admins who use an AI assistant. The administrators review a brief, then publish it for the team (see "Publish a brief for the team"). This also keeps the deep scan's sign-in to the administrators.

Orgs where developers work in their own sandboxes can give those developers FlightBrief User instead, so they generate briefs of the sandbox themselves.

A brief covers what the person who generated it can see. A published brief is read by people who may see less, so it can name objects, fields or classes they cannot open. It never holds record data.

3. Brief levels

- **Surface**: schema and relationships, custom metadata and custom settings, Apex conventions, triggers and Flows. Works right away. Custom metadata and custom setting records are named, never their values.
- **Deep**: everything in Surface, plus validation rules with their conditions, the fields each class and Flow uses, and what each Flow does. Email addresses and record ids in rule conditions are replaced with placeholders. It reads these through the org's own Tooling API, signed in as the user running the brief, so it needs an administrator's setup once and each user's sign-in once.

Under the Deep option the FlightBrief page says whether the deep scan is ready and, if not, whose step is missing: not set up yet (an administrator's step), no access yet (the user needs the FlightBrief Deep Scan permission set) or not signed in yet (the user's step).

4. Set up the deep scan

Open the **FlightBrief Setup** tab; its **Deep scan** tab holds these steps. (Its other tabs are **Company standards**, **Questions for your team** and **Schedule**, described below.) Each step shows Done or Not done yet; press Refresh after working in Setup.

1. **Start setup, then create the OAuth app.** Press **Start setup**. FlightBrief creates the sign-in (an External Auth Identity Provider named FlightBrief Login) and shows its callback URL. Copy it. In Setup, open **External Client App Manager** and create a new app: - Enable OAuth and paste the callback URL. - Add the scopes "Manage user data via APIs (api)" and "Perform requests at any time (refresh_token, offline_access)". - Leave Require PKCE on. - After saving, open the app's OAuth settings and copy its consumer key and consumer secret.
2. **Connect FlightBrief to the app.** Paste the consumer key and secret and press **Connect**. FlightBrief creates the External Credential and the Named Credential, both named FlightBrief Org, and the **FlightBrief Deep Scan** permission set, and assigns that permission set to you. The secret goes straight to Salesforce's credential store; FlightBrief never keeps it.
3. **Enable callouts.** In Setup, open **Named Credentials**, edit **FlightBrief Org** (the one without a namespace prefix) and tick **Enabled for Callouts**. Salesforce leaves this step to an administrator.
4. **Give users access.** Assign **FlightBrief Deep Scan** to everyone who will run deep scans. Each of them signs in once: in their personal settings, open **External Credentials**, find **FlightBrief Org** and choose **Allow Access**. The FlightBrief page links there when a user still needs to sign in.

When steps 1 to 3 are done, the tab says the deep scan is set up and lists each step on one line. Its last line is about your own sign-in only; every user signs in for themselves. **Show steps** opens the full steps again.

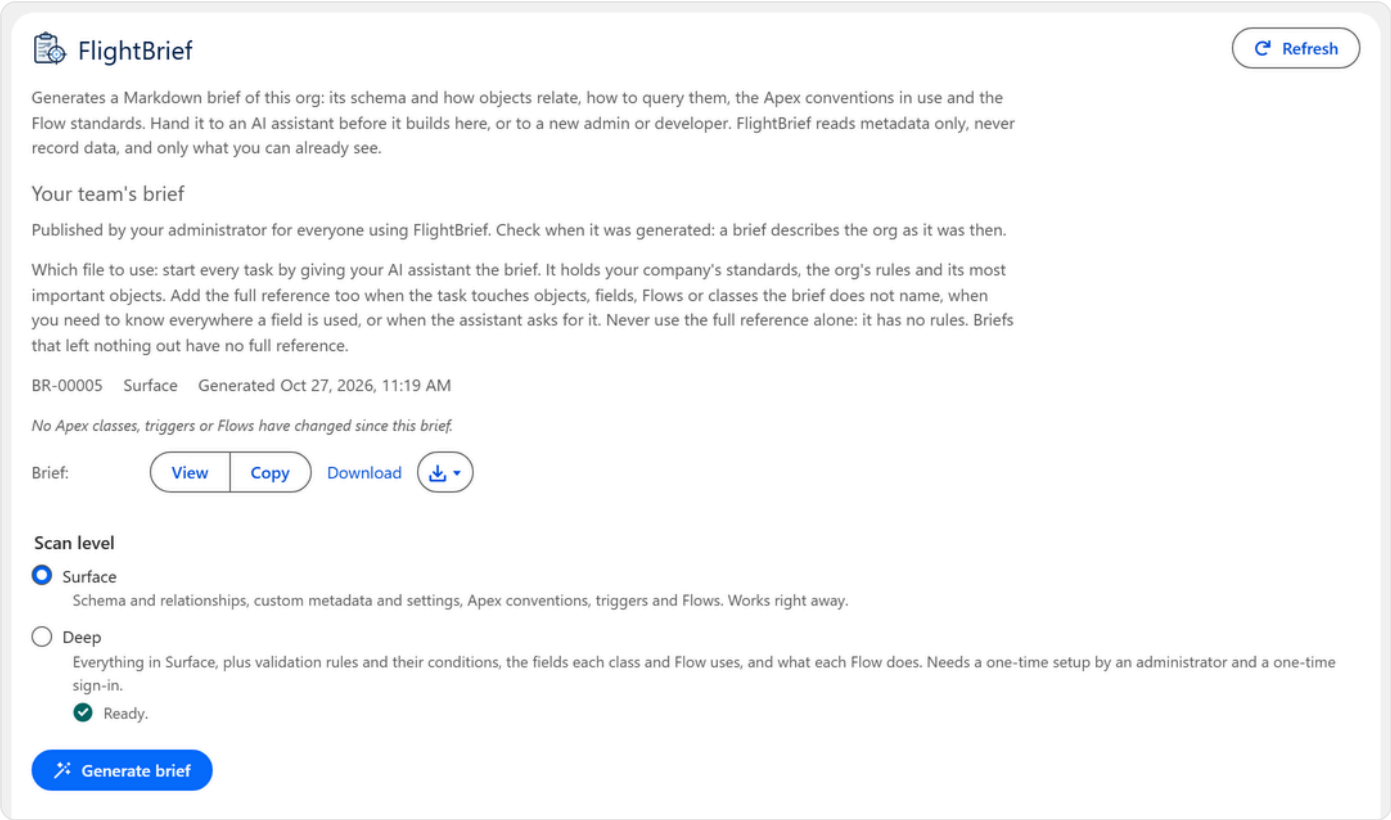
To change the OAuth app later, for example after rotating its consumer secret or creating a new External Client App, press **Replace consumer key and secret**, paste the new values and press **Save**, then confirm. Nothing else is recreated. If you create a new app, give it the same callback URL; the tab still shows it.

After a replace, every user's sign-in stops working, although Salesforce still lists it. Each user opens **External Credentials** in their personal settings, presses **Revoke** for **FlightBrief Org**, then **Allow Access**, and logs in again. Until then their Deep option says they need to sign in.

The credentials, the permission set and the External Client App belong to your org, not to the package. You can edit or remove them like any others. If one of them breaks, delete it in Setup and run the steps again: each step creates only what is missing.

5. Generate and use a brief

On the FlightBrief page, choose Surface or Deep and press **Generate brief**. The brief is built in the background; the list shows its progress and refreshes on its own. A surface brief takes a minute or two. A deep brief reads each validation rule and Flow one at a time, so in an org with hundreds of them it can take 10 minutes or more. You can leave the page and come back; the brief keeps running. While a brief is running, Generate brief waits for it to finish.

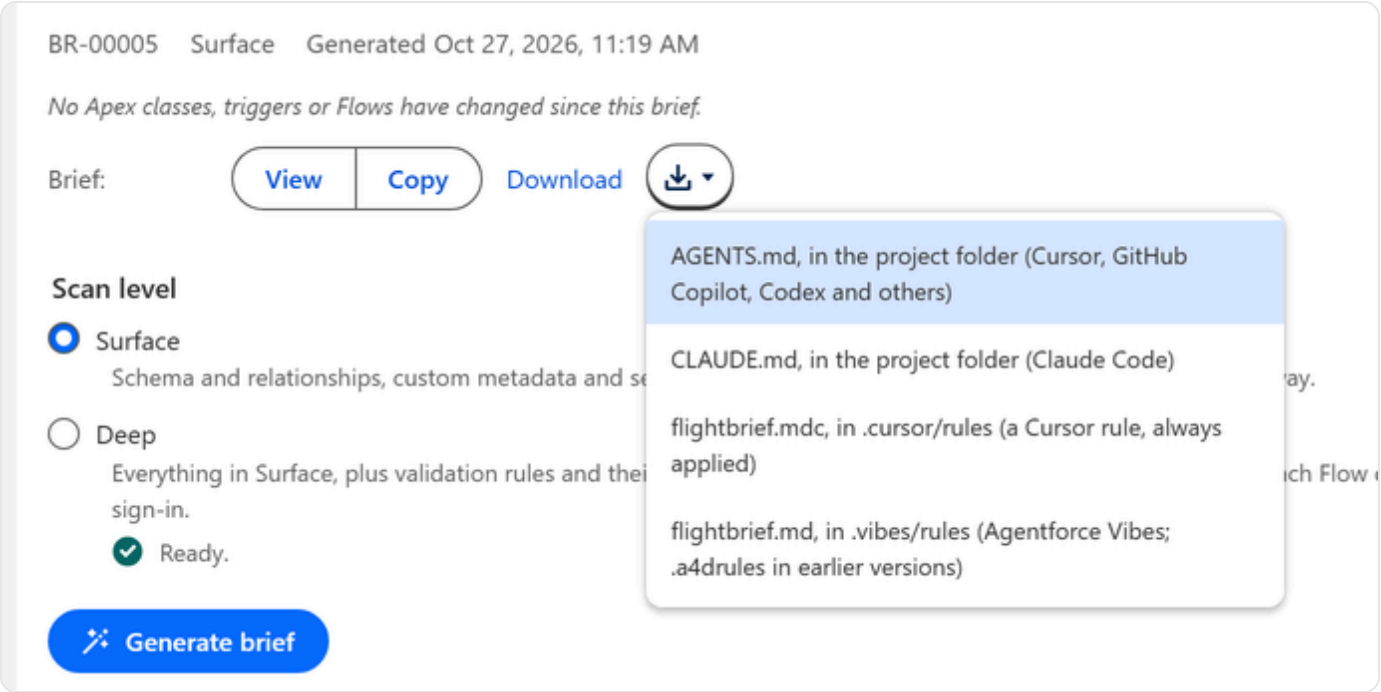


The FlightBrief page: the team's brief at the top, then the scan level and Generate brief.

To stop a brief that is queued or running, press **Cancel** on its row and confirm. It stops at its next step, usually within a minute, and shows as Canceled; you can then delete it and generate a new one. Salesforce sometimes takes several minutes to start a queued brief, especially in a sandbox; Cancel is the way out if one never starts.

Each finished brief has **View**, **Copy** and **Download**.

- **Brief**: give this to your AI assistant at the start of every task. It holds your company's standards, the org's rules and its most important objects, sized to fit an assistant's working memory.
- **Full reference**: saved only when the brief had to shorten long lists. It holds every object, field, validation rule, Flow and class the scan found. When nothing was shortened, the page says the brief is complete and there is no reference.



The menu beside Download saves the brief as the file an AI tool reads from your project folder.

Which file to use

- **Always the brief.** Start every task with it.
- **Add the full reference** when the task touches objects, fields, Flows or classes the brief does not name; when you need to know everywhere a field is used, for example before changing or deleting it; or when the assistant asks for it. The brief tells the assistant to ask rather than guess names. Add the whole file, or only the part for the objects your task involves.
- **Never the full reference alone.** It has no rules or standards, so an assistant given only the reference builds without them.

The download button beside the brief's Download link saves it as the file an AI coding tool reads automatically from your Salesforce project, so every session starts with the brief:

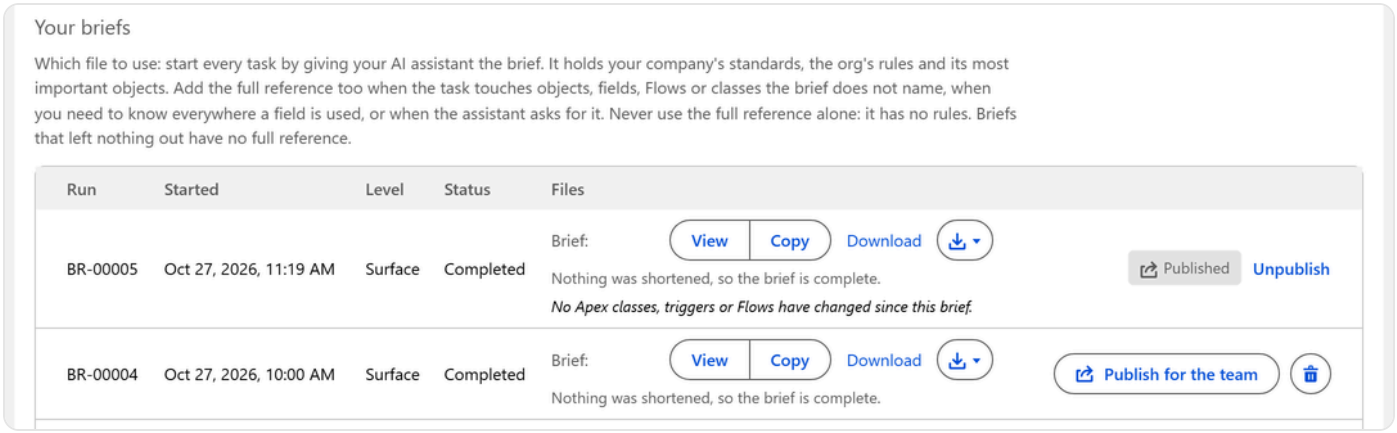
- **AGENTS.md**, in the project folder: Cursor, GitHub Copilot, Codex and others.
- **CLAUDE.md**, in the project folder: Claude Code.
- **flightbrief.mdc**, in `.cursor/rules`: a Cursor rule, always applied.
- **flightbrief.md**, in `.vibes/rules`: an Agentforce Vibes rule (`.a4drules` in earlier versions).

For a team, put the brief in the assistant's standing instructions, such as a project file or custom instructions, so every conversation starts with it, and add the reference only for the tasks that need it. Replace both when you generate a new brief.

Users see only their own briefs. **Delete** removes a brief and its files; a brief that is still running cannot be deleted.

6. Publish a brief for the team

Holders of FlightBrief Admin see **Publish for the team** under each of their own finished briefs. Publishing makes that brief, and its full reference, the team's brief: everyone with FlightBrief Reader or FlightBrief User sees it at the top of the FlightBrief page under "Your team's brief", with View, Copy and Download and when it was generated. One brief is published at a time; publishing another replaces it. FlightBrief asks before either.



Your briefs: the published one, and Publish for the team on the others.

Unpublish stops sharing it; the brief stays in your list. A published brief cannot be deleted until it is unpublished. Republish after changes to the org, so the team's brief stays current: the page shows readers when it was generated, and how many Apex classes, triggers and Flows have changed since. Your own newest brief shows the same count.

7. Company standards

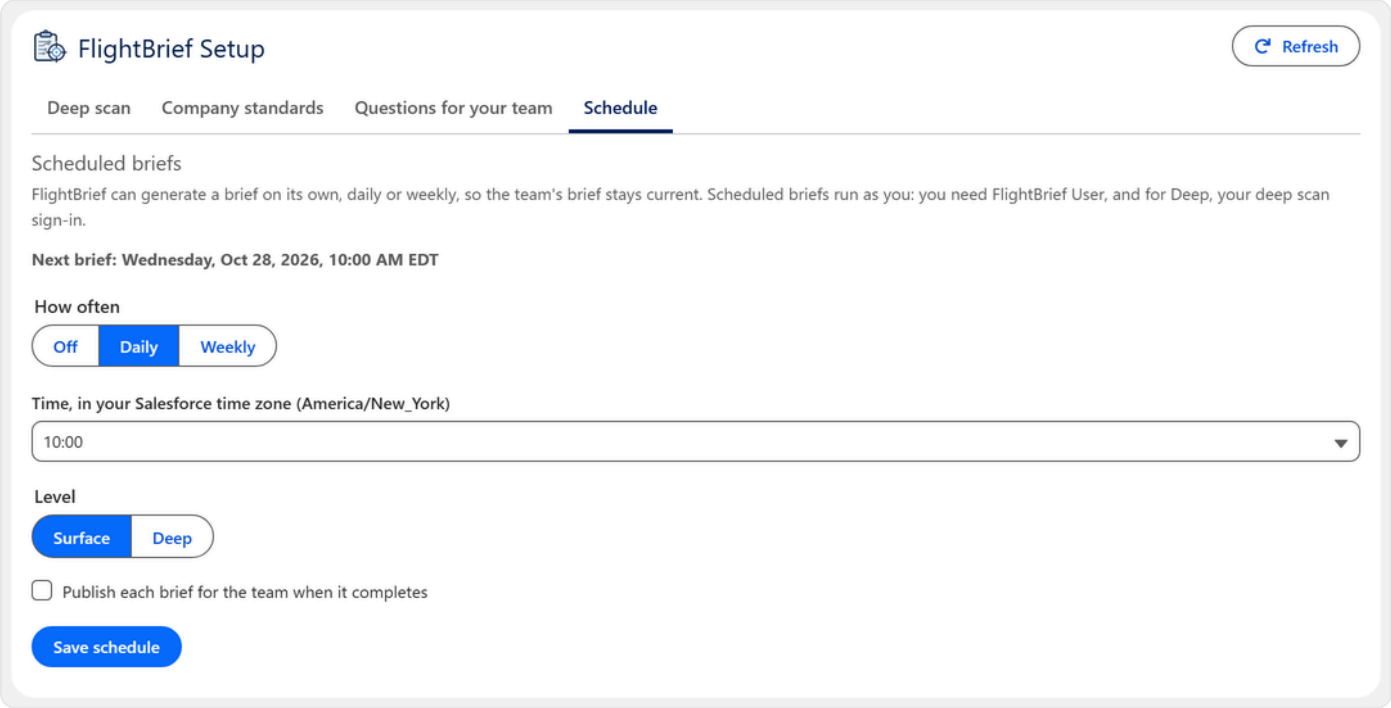
The FlightBrief Setup tab also holds your company's own Salesforce standards, declarative and programmatic. Every brief starts with them and tells the AI assistant to follow them where the code does otherwise.

- Type or paste them into the **Standards** box, or press **Load from a .md or .txt file**. A loaded file fills the box; nothing is saved until you press **Save standards**.
- From Confluence or another wiki, copy the page text, or use a Markdown export if you have one. Headings and lists carry over. PDF and Word files are not read.
- Up to 131,072 characters fit. A brief carries the first 24,000 (about 6,000 tokens); the rest goes in the full reference, and the tab says so when your text is longer.
- To remove them, empty the box and save; FlightBrief asks first.

Holders of the FlightBrief Admin permission set can change the standards. Everyone with FlightBrief User can read them, since every brief includes them.

8. Schedule briefs

On the FlightBrief Setup tab, **Schedule** lets FlightBrief generate a brief on its own, so the team's brief stays current. Choose **Daily** or **Weekly** (and a day), a time in your own time zone, Surface or Deep, and whether to **publish each brief for the team when it completes**, then press **Save schedule**. The tab shows when the next brief will run. Choose **Off** and save to stop.

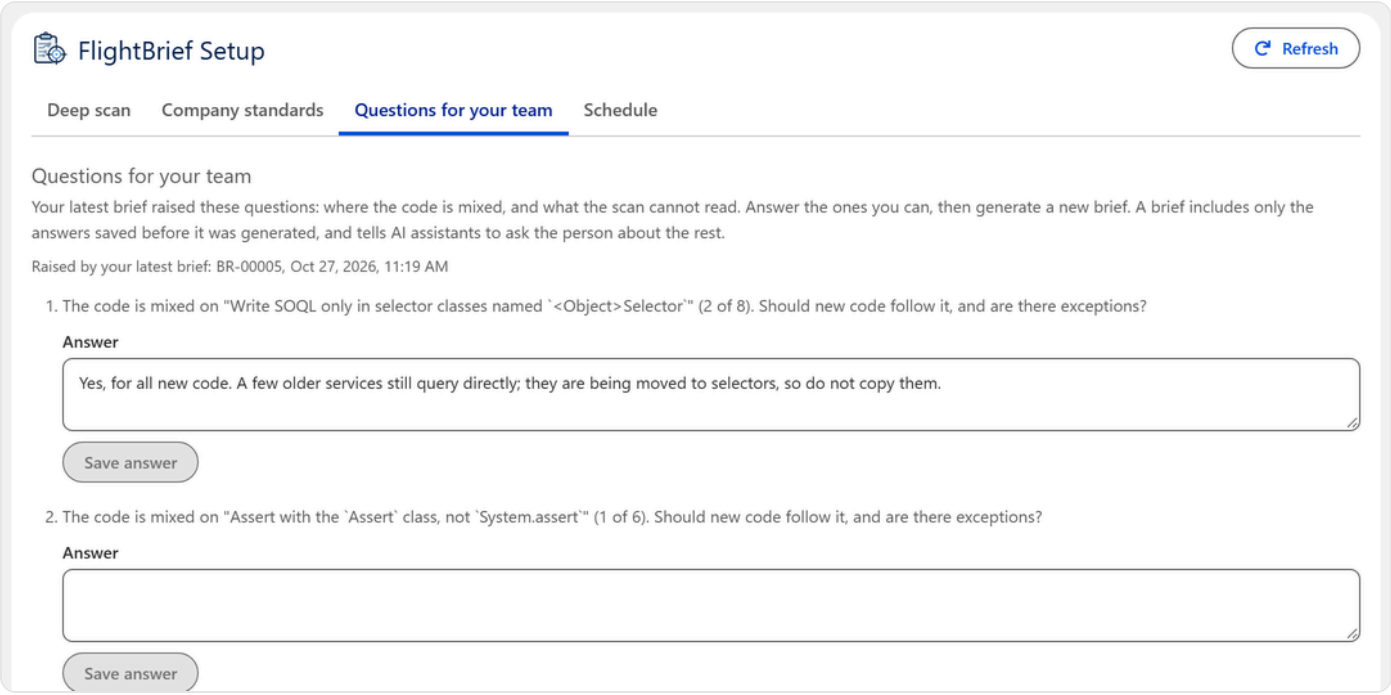


The Schedule tab, with a daily Surface brief at 10:00.

Scheduled briefs run as the administrator who saved the schedule: you need FlightBrief Admin and FlightBrief User, each brief covers what you can see, and a Deep brief needs your deep scan sign-in. If it cannot connect, that brief fails and says why. If another administrator saves the schedule, it moves to them. A brief that is still running when the next one is due is left to finish, and that run is skipped.

9. Questions for your team

Every brief raises questions the scan cannot settle: where the code is mixed, for example some classes `with sharing` and some `inherited sharing`, and what it cannot read, such as sharing rules, page layouts, large data volumes and integrations. The FlightBrief Setup tab lists the questions your latest brief raised under **Questions for your team**.



Questions for your team, with the first one answered.

Type an answer and press **Save answer**; each question is saved on its own. Then generate a new brief: a brief carries only the answers saved before it was generated, and the tab says so until you do. If your team reads a published brief, publish the new one. Every later brief carries your answers under "Answers from your team" and lists the rest under "Open questions", which tells the AI assistant to ask the person rather than guess. Empty an answer and save to open the question again. Holders of FlightBrief Admin can answer; everyone with FlightBrief User reads the answers in their briefs.

10. What a brief can see

A brief covers only what its user can read, and the brief names whose access it used. Briefs from different users of the same org can therefore differ.

- **The data model** (objects, fields, relationships, record types, descriptions) follows the user's object and field access. An object or field the user cannot see is left out, and a lookup to an object they cannot see says so instead of naming it.
- **Code and automation** (Apex classes and triggers, Flows, validation rules) follow what Salesforce lets the user read as metadata. A validation rule's formula, a Flow's summary or a code example can therefore name a field the user cannot see on a record, just as it would in Setup. No record data is ever read.
- Code examples leave out comments and show every text value as `'...'`.

When the scan stops at one of its own limits (40 standard objects, records of 40 custom metadata and setting types, 2,000 permission rows, 25 pages of a deep-scan query), the brief's "Limits of this brief" section says so.

11. Uninstall

1. Ask users to delete the briefs they no longer need from the FlightBrief page. Uninstalling does not delete brief files: each stays in its owner's Files, titled "FlightBrief - " followed by the org name and date.
2. Remove every assignment of the FlightBrief User, FlightBrief Admin and FlightBrief Reader permission sets. Salesforce refuses to uninstall while any of them is assigned.
3. Uninstall the package from Setup, Installed Packages.

If you install FlightBrief again later, the deep scan works again without repeating setup: assign the permission sets, and users who signed in before do not need to sign in again.

Uninstalling leaves what setup created in your org, because it belongs to you: the FlightBrief Login identity provider, the FlightBrief Org External Credential and Named Credential, the FlightBrief Deep Scan permission set and the External Client App. Delete them in Setup if you no longer need them, in this order, since each refers to the ones after it: the Named Credential, the FlightBrief Deep Scan permission set, the External Credential, the identity provider and the External Client App.

12. Support

Email support@adeptapps.net with the FlightBrief version (Setup, Installed Packages) and what you saw. Do not send briefs or screenshots that show anything your company keeps private.

© 2026 Adept Apps LLC. Salesforce and AppExchange are trademarks of Salesforce, Inc.