

Cal.diy connector changelog

Use this page to decide whether a Cal.diy connector change requires a new artifact, migration, or caller update. It records only changes that can be established from the reviewed source range.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/cal-diy/changelog.md

Release status: the reviewed snapshot is not a tagged release. The latest tag in the reviewed repository is `v1.0.0`; the pinned documentation revision is five commits later and still declares workspace version `1.0.0`. Never use the version string alone to identify these different source states.

Identify the exact version

Identity	Published <code>v1.0.0</code> baseline	Reviewed source snapshot
Source commit	<code>034a520608eea44e6b14e61c34734bd402d989c0</code>	<code>97be86e9efedf07ecf1783b03800f683f107fb04</code>
Source position	Tagged	Five commits after the tag
Publication status	Published tag in the reviewed repository	Unreleased source; no later tag or changelog entry
Workspace and image version	<code>1.0.0</code>	<code>1.0.0</code>
Native AIP version	<code>1.0</code>	<code>1.0</code>
Cal.diy business capabilities	81	81
Target connector placement	Bundled release topology	Separately built and admitted standalone host

An operator record for the reviewed snapshot needs the full source commit, source-tree state, immutable image digest, manifest digest, connector version ID, release ID, and upstream identity. A tag or `1.0.0` image label cannot substitute for that tuple.

Reviewed source snapshot

The entries below describe the Cal.diy-relevant difference between `v1.0.0` and `97be86e9`. They are source changes, not proof that an artifact was built, published, deployed, or qualified.

Added

- A dedicated `aip-host-cal-diy` binary and common standalone-host image path.
- Registry admission, tenant-scoped routing, replica leases, health reporting,

drain, and recovery integration in the connector-fleet topology.

- Cluster-safe webhook replay fencing through durable profile-state compare-and-set.

- A standalone-host webhook ingestion boundary that persists mapped events

before acknowledging a provider delivery.

- Product-image and controlled-fleet qualification procedures for the Cal.diy

host.

Changed

- Manifest compatibility metadata now reports 81 business operations, 38

operator-only route exclusions, and two deprecated-alias exclusions.

- The trusted credential issuer is exposed as the stable

`cal_diy_deployment` connector constant and remains part of the immutable contract.

- Runtime ownership moves from an in-process connector inside the bundled

daemon to one admitted standalone host per Cal.diy account boundary.

- Webhook replay state in the standalone topology is shared across replicas;

it is not an in-memory, per-process duplicate fence.

The 81 capability IDs are not a post-tag expansion. The material change is how that stable business catalog is packaged, admitted, routed, and recovered.

Security boundary made explicit

The reviewed connector inventories 38 provider routes that remain in an operator control plane. They cover credential, OAuth, browser callback, external integration, and ownership-verification workflows. The connector does not make those routes agent-callable merely because they exist upstream.

This inventory documents an existing trust boundary. It does not add 38 capabilities and does not authorize direct calls around the connector.

Deprecated or excluded

Surface	Current status	Replacement
Singular GET <code>/v2/calendars/{calendar}/event/{event_uid}</code>	Excluded deprecated provider alias	<code>cap:cal_diy:calendar.event.get</code> on the plural <code>/events/</code> route
Singular PATCH <code>/v2/calendars/{calendar}/event/{event_uid}</code>	Excluded deprecated provider alias	<code>cap:cal_diy:calendar.event.update</code> on the plural <code>/events/</code> route
Booking-create <code>meetingUrl</code>	Accepted by the schema but marked deprecated	Structured <code>location</code> with the appropriate location variant
Free-form booking-create location string	Accepted for compatibility but marked deprecated	A supported structured <code>location</code> object

The source defines no removal date for either deprecated input form. Avoid new use, migrate stored templates and callers, and treat a future removal as a schema and compatibility change requiring its own release entry.

Apply the required migration

Change encountered	Required action	Do not do
Bundled connector to standalone host	Follow the bounded bundled-to-fleet migration; preserve tenant, account, credential, idempotency, approval, transaction, provider-effect, webhook, and recovery evidence	Start both routes for the same mutations without explicit fencing
New standalone artifact at the same workspace version	Admit a new immutable connector version and bind its exact source, image, manifest, and release identities	Replace an existing version record or identify it only as 1.0.0
Singular calendar-event provider alias	Select the canonical AIP capability and its published schema	Rewrite only the path text while bypassing AIP governance
Deprecated <code>meetingUrl</code> or free-form location	Convert the caller to a structured <code>location</code> input and test the exact schema	Depend on an undocumented removal date
Provider revision or API-version pin changes	Update source-owned mappings, regenerate contract artifacts, and requalify the affected operations	Override <code>cal-api-version</code> at runtime to conceal drift

The process migration must not be combined with capability renames or schema changes. Stabilize the standalone topology first, accept its evidence, and then introduce a separately versioned contract change.

Retain the compatibility pins

The reviewed snapshot selects these values in source:

Surface	Pin
Connector ID	<code>cal-diy</code>
Connector profile	<code>aip.connector.cal_diy.v1</code>
Manifest format	<code>aip-manifest/v1</code>
Cal.diy upstream commit	<code>f00434927386c9ecdcbd7e6c5f82d22044a245bc</code>
General and booking API	<code>2024-08-13</code>
Slot API	<code>2024-09-04</code>
Event-type API	<code>2024-06-14</code>
Private-link controller default	<code>2024-04-15</code>
Schedule API	<code>2024-06-11</code>
Webhook payload and header version	<code>2021-10-20</code>
Standalone version-ID shape	<code>cver_cal_diy_1_0_0_<release-id></code>

These pins describe the connector contract, not every Cal.diy deployment. Changing a provider deployment without changing the connector source does not silently establish compatibility with the new deployment.

Decide when a new connector version is required

Create and admit a new immutable connector version when a change affects any of these properties:

- capability IDs, schemas, contracts, risk, approval, retry, or transaction

semantics;

- provider method, path, API version, request partition, or response mapping;
- credential issuer, required scopes, tenant or account binding, or secret

resolution;

- webhook signature, version, replay scope, destination policy, or event

mapping;

- artifact bytes, dependencies, manifest digest, upstream baseline, or host

lifecycle behavior.

A documentation clarification alone does not require a connector version. It still needs the source revision that proves the corrected statement.

Record future entries completely

Each future entry should state:

1. release or explicit unreleased status and UTC review date;
2. full source commit and tree state;
3. connector, image, manifest, catalog, and upstream identities;
4. added, changed, fixed, security-relevant, deprecated, and removed surfaces;
5. caller, operator, storage, webhook, and provider migration actions;
6. rollback and compatibility boundary;
7. exact qualification result and retained evidence, or `not_run`;
8. deprecation introduction, replacement, and removal date only when source or

release policy defines them.

Do not backfill release history from filenames, mutable tags, or source presence. Mark unknown history as unknown.

Related documentation

- [AIP release notes](#)
- [Migrate bundled connectors to the fleet](#)
- [Operator-only and excluded Cal.diy routes](#)
- [Cal.diy API versions and provider routes](#)
- [Qualify the Cal.diy connector](#)