

Twenty connector changelog

Use this page to decide whether a Twenty connector change requires a new artifact, admission record, migration, caller update, or qualification result. It records only changes established from the reviewed Git history.

SECTION	Connectors
TYPE	Connector
PROTOCOL	AIP 1.0
SOURCE	docs/connectors/twenty/changelog.md

Release status: the published `v1.0.0` tag predates the Twenty connector.

The reviewed Twenty code is an unreleased source snapshot that still declares workspace version `1.0.0`.

Identify the exact source state

Identity	Published AIP <code>1.0.0</code>	First Twenty source	Reviewed Twenty source
Source commit	<code>034a520608eea44e6b14e61c34734bd402d989c0</code>	<code>417e17672510f81975c035d437e5a3f58ee53574</code>	<code>97be86e9efedf07ecf1783b03800f683f107fb04</code>
Source time	2026-07-21T10:49:24Z	2026-07-25T19:48:45Z	2026-07-26T10:41:19Z
Git position	Tagged <code>v1.0.0</code>	Five-commit review range, before hardening	<code>v1.0.0-5-g97be86e</code>
Publication status	Published AIP Core tag	Unreleased source	Unreleased source
Workspace version	<code>1.0.0</code>	<code>1.0.0</code>	<code>1.0.0</code>
Twenty connector crate	Absent	Present	Present
Twenty host crate	Absent	Present	Present
Twenty operation count	None	22	22

No tag in the reviewed repository contains `97be86e9`. The version string `1.0.0` therefore cannot identify a Twenty artifact.

A deployable record needs full source, image, manifest, connector-version, release, provider, configuration, and qualification identities.

First Twenty source baseline

Commit `417e17672510f81975c035d437e5a3f58ee53574` adds the Twenty connector and standalone host. This commit is a source-history milestone, not a separately published semantic release.

Added connector contract

- connector ID `twenty` and profile `aip.connector.twenty.v1`;
- exact provider mapping revision

`96a24563674313a3071d359bfccaf33d5e130ab8`;

- 22 fixed capabilities across records, metadata, and OpenAPI documents;
- twenty tool capabilities and two resource capabilities;
- eight low-risk reads and fourteen governed mutations;
- required approval and idempotency for every mutation;
- deterministic UUID injection and forced upsert for record creation;
- non-empty bulk filters and explicit permanent-delete behavior;
- fixed provider origin, Bearer authentication, redirect rejection, and bounds;
- structured provider errors, retry metadata, and mutation uncertainty;
- signed webhook mapping, workspace validation, and nonce replay fencing;
- eight source tests for catalogue, guards, idempotency, webhook, replay, and errors.

Added standalone host

- `aip-host-twenty` on the common connector-host lifecycle;
- immutable type, version, instance, replica, tenant, and artifact identity;
- owner-controlled API-token, webhook-secret, and operation-allowlist files;
- durable PostgreSQL runtime, replay, Event, and publication state;
- optional `POST /webhooks/twenty` ingress;
- instance-scoped shared replay state;
- local Event persistence before central publication;
- product-image build and inspection support for `aip-host-twenty`.

The first source baseline already contains the final 22-operation connector library shape reviewed here.

Reviewed hardening snapshot

Commit `97be86e9efedf07ecf1783b03800f683f107fb04` is the documentation baseline. The Twenty connector library itself has no diff from `417e1767` in this range.

Changed webhook publication

The standalone host changes central Event publication from a direct publish call to durable enqueue:

- an accepted Event enters the connector-host publication outbox;
- successful HTTP output reports `central_delivery` state;
- central acknowledgement and durable local queueing remain distinguishable;
- outbox storage failure returns `webhook.storage_unavailable`;
- a temporary central delivery failure no longer requires the provider to

create a replacement event.

This change affects recovery semantics even though the Twenty connector library is unchanged. Adopt it only as a newly built and admitted host artifact.

Changed shared host and build surfaces

The reviewed commit also hardens common connector-host lifecycle behavior used by Twenty. It changes shared host, bootstrap, remote, fleet, and failure-matrix code in the same source snapshot.

The generic host Dockerfile changes its target-cache identity to share one locked release tree across workspace binaries. That build optimization does not define runtime compatibility, but a rebuilt image still needs an immutable artifact identity and supply-chain evidence.

Unchanged connector pins

Surface	Reviewed value
Connector ID	twenty
Connector profile	aip.connector.twenty.v1
Native AIP version	1.0
Workspace version	1.0.0
Twenty mapping revision	96a24563674313a3071d359bfccaf33d5e130ab8
Capability count	22
Reads and mutations	8 reads, 14 mutations
Provider request maximum	16 MiB
Default provider response maximum	64 MiB
Configurable response maximum	256 MiB
Webhook body maximum	4 MiB
Webhook clock window	Five minutes
Webhook live-nonce capacity	100,000

Changing a limit in deployment within the allowed range does not change the source pin. It does change configuration identity and qualification scope.

Apply the first-adoption migration

The published `v1.0.0` tag offers no Twenty connector to upgrade in place. Adopting the reviewed snapshot is a new standalone connector deployment.

Area	Required action	Unsafe shortcut
Source	Select the exact reviewed commit and reconstruct a clean locked tree	Build from an unnamed working directory
Artifact	Build, sign, scan, and admit an immutable aip-host-twenty digest	Identify the image only as 1.0.0
Manifest	Verify the 22-operation catalogue and admit an intentional subset	Enable all mutations by default
Workspace	Bind tenant, external account, provider workspace UUID, and token revision	Accept workspace identity from Action input
State	Provision durable shared PostgreSQL before registration	Use process memory for production replay state
Network	Fix provider origin, AIP ingress, control plane, and optional Event route	Expose provider credentials or arbitrary proxy routes
Governance	Add approval, stable keys, verification reads, and incident policy for mutations	Replay an ambiguous mutation with a new key
Webhooks	Provision HMAC secret, shared replay scope, local Event log, and outbox	Treat central timeout as proof of rejection
Operations	Add readiness, lease, storage, provider, Action, and outbox monitoring	Route traffic from static liveness alone
Qualification	Run exact source, image, workspace, operation, webhook, and recovery gates	Promote source presence to a qualified result

Preserve one rollback record containing the previous routing state, provider configuration, credentials, workspace fixtures, and cleanup procedure. There is no tagged Twenty artifact in `v1.0.0` to use as an implicit rollback target.

Preserve capability compatibility

Callers should discover capability contracts from the admitted manifest. Do not derive provider routes or convert arbitrary provider operations into capability IDs.

The following changes require a new immutable connector version:

- capability ID, kind, schema, risk, approval, idempotency, side effect, or retry contract;
- provider method, route, query, body, response, status, or redaction mapping;
- deterministic ID, bulk-filter, delete-mode, merge, or restore behavior;
- metadata-resource set, layout guard, webhook-secret injection, or OpenAPI path;
- provider origin, workspace binding, credential, redirect, or operation policy;
- webhook HMAC, clock, nonce, replay, Event, outbox, or response behavior;
- host lifecycle, storage schema, lease, readiness, signing, or routing behavior;
- artifact bytes, dependencies, base image, manifest digest, or upstream revision.

A documentation clarification can retain artifact identity only when it changes no executable contract, generated asset, example that defines behavior, or qualification claim.

Requalify provider changes

The upstream commit identifies the source used to derive REST and metadata mappings. It does not identify a running provider artifact by itself.

Re-audit and requalify when the provider image, hosted release, workspace schema, enabled feature set, API token scope, or endpoint behavior changes. Retain exact provider evidence with the connector result.

Do not keep the old upstream pin while promoting unreviewed provider bytes as compatible. Record compatibility as unknown until the affected operation and failure matrices pass.

Record the adoption decision

For each adopted source snapshot, retain:

```
TEXT
Release status:
AIP source revision:
Source-tree digest:
Cargo lock digest:
Host image digest:
Manifest digest:
Connector version ID:
Twenty mapping revision:
Provider artifact or release identity:
Workspace binding:
Credential revision:
Configuration revision:
Qualification run and evidence digest:
Migration reviewer:
Accepted at:
Rollback record:
```

Never record only `1.0.0`. That value describes the tagged AIP baseline, the first Twenty source commit, and the reviewed hardening snapshot.

Keep limitations explicit

At the reviewed source snapshot:

- no published tag contains the Twenty connector;
- source-owned product image checks do not prove provider behavior;
- the source-owned five-product fleet topology has no Twenty runtime service;
- provider exactly-once effects and mutation rollback are not guaranteed;
- the connector has no cancellation or streaming operation;
- direct GraphQL and arbitrary provider proxying are excluded;
- retained local evidence does not map its workspace digest to the reviewed

commit;

- current exact-artifact qualification is not established by this changelog.

These limits define release and evidence boundaries. They are not deferred features or passing qualification cases.

Maintain future entries

Add a new changelog entry only after source and artifact identities are fixed. Each entry should include:

1. release or explicit unreleased status and UTC review time;
2. full source commit and reconstructable tree identity;
3. artifact, manifest, provider, workspace, and configuration identities;
4. added, changed, fixed, security-relevant, deprecated, and removed surfaces;
5. caller, operator, storage, webhook, and provider migration actions;
6. compatibility and rollback boundaries;
7. exact qualification result or `NOT RUN`;
8. deprecation replacement and removal date only when committed policy defines them.

Do not reconstruct release history from filenames, mutable tags, or source presence. Mark unknown publication or qualification state as unknown.

Related documentation

- [Twenty connector overview](#)
- [Twenty connector configuration](#)
- [Twenty capability index](#)
- [Deploy the Twenty connector](#)
- [Qualify the Twenty connector](#)