ADMINISTRATOR & USER MANUAL

ComV Tables

End-to-End Installation, Configuration, and Operations Guide

For Salesforce administrators, integration owners, and business users

Purpose

Install and configure ComV Tables, generate a reusable Lightning table component without custom development, deploy it in Lightning App Builder, and operate it safely in a subscriber org.

Package version: 1.10 | Namespace: ComvTable | Source API version: 67.0

Updated September 22, 2026

Contents

- 1. Application overview
- 2. Roles, prerequisites, and installation planning
- 3. Subscriber-org post-install configuration
- 4. First-time validation and health check
- 5. Create a ComV Table configuration
- 6. Add the generated table to Lightning pages
- 7. Runtime feature guide
- 8. Administration and lifecycle operations
- 9. Troubleshooting
- 10. Go-live acceptance checklist
- Appendices: field reference, naming rules, security, and sources

How to use this guide

Administrators should complete Sections 2 through 4 before any table configuration is created. Configuration owners then use Sections 5 and 6. Business users can use Section 7 as a runtime quick reference. Keep Sections 8 through 10 available for support and release management.

Role	Primary responsibility
Salesforce administrator	Install the package, deploy My Domain, assign access, configure authentication, and validate setup.
Integration owner	Own the integration user, subscriber External Client App, certificate lifecycle, and OAuth policy.
ComV Tables configurator	Create and maintain ComV Table Config records and verify generated components.
Lightning page administrator	Place generated components on record, app, or home pages and activate them.
Business user	Search, filter, sort, edit, export, and invoke configured actions within assigned access.

Document conventions

- Setup paths use the Lightning Experience Setup menu and Quick Find search.
- Names shown in monospace-like text, such as ComVTable_Tooling_API, are exact API names unless marked as examples.

- Subscriber org means the Salesforce org where the managed package is installed.
- Validate all changes in a sandbox before production deployment.

1. Application overview

ComV Tables is a Salesforce Lightning application that lets an administrator define a configurable data table from object and field metadata. Saving a ComV Table Config record triggers an asynchronous Tooling API process that creates an Apex controller and an exposed Lightning Web Component in the subscriber org. The generated component uses the packaged table runtime to display data according to the saved configuration.

What the application provides

- Object and field selection driven by Salesforce Schema describe information.
- A preview before saving the table configuration.
- Generated Apex and LWC artifacts for Lightning record, app, and home pages.
- Optional pagination, sorting, global search, column search, column show/hide, filters, inline editing, row actions, related-list context, Flow launch, row selection, and CSV export.
- A setup health check for object access, Named Credential availability, and Tooling API connectivity.

End-to-end flow

- Install the managed package and complete subscriber-specific security and endpoint setup.
- Assign the packaged ComvTable Permissions permission set to administrators or approved configurators.
- Run Setup Health and correct every failed item.
- Create and preview a ComV Table Config record.
- Save the record; the asynchronous Tooling API job generates an Apex class and Lightning component.
- Review Deployment Status / Exception Log, then place the component in Lightning App Builder.
- Activate the page, test with a non-admin user, and move the configuration through change control.

Current authentication architecture

Version 1.10 uses the packaged Named Credential ComVTable_Tooling_API and External Credential ComVTable_Tooling_API_Ext. Do not store consumer keys or secrets in ComV Table custom metadata. A Remote Site Setting is not required for callouts made through the configured Named Credential.

Generated artifact behavior

For each configuration, the app generates an Apex controller and an LWC bundle. The component is exposed for lightning__RecordPage, lightning__AppPage, and lightning__HomePage. Generated names are effectively immutable from the ComV Table Config user interface after creation. Updates to supported configuration fields change runtime behavior; they do not safely rename existing metadata artifacts.

2. Roles, prerequisites, and installation planning

Org prerequisites

- Lightning Experience is enabled and My Domain is deployed to users.
- The target edition supports Apex, Lightning Web Components, API access, Named Credentials, External Credentials, and Tooling API operations.
- An integration user is available. Use a dedicated, active user instead of a personal administrator account.
- A subscriber-owned External Client App is approved for OAuth JWT Bearer Flow.
- A certificate lifecycle owner and renewal process are defined.
- A sandbox is available for installation rehearsal and acceptance testing.

Certificate prerequisite for packaged JWT credentials

Important before installation

Salesforce does not include JWT signing certificates in packages. Recreate the referenced certificate in the subscriber org before installing when possible. The expected certificate API name is ComVTable_Tooling_JWT. If the package is already installed, create or select the certificate and then edit the External Credential to reference it.

Recommended integration-user permissions

The integration user authenticates back into the same subscriber org and creates Apex and LWC metadata through Tooling API. Grant only the permissions required by your org policy, then confirm with the health check and a sandbox generation test.

Permission / access	Why it is needed	Guidance
API Enabled	Allows OAuth API access.	Required.
Author Apex	Allows creation of the generated Apex class.	Required for generation.
Customize Application	Supports creation and update of Lightning metadata.	Commonly required; validate in your org.
Modify Metadata Through Metadata API Functions	Controls metadata modifications through APIs in orgs where this permission is exposed.	Grant if required by org policy or generation errors.
View Setup and Configuration	Allows relevant setup and metadata visibility.	Recommended for the dedicated integration user.
ComvTable Permissions	Provides packaged class, app, object, and External Credential principal access.	Assign to the configuration user and to the callout-running user as applicable.

Install the package

- Sign in to the target sandbox or production org with an authorized installer account.
- Open the package installation URL and select the access option required by your governance policy.
- Review package components, approve third-party access only if presented and understood, and start installation.
- Wait for the installation-complete email and verify the package appears under Setup > Installed Packages.
- Do not assume post-install automation completed subscriber setup. Version 1.10 requires the steps in Section 3.

3. Subscriber-org post-install configuration

Complete in order

The generated component process depends on My Domain, the integration identity, OAuth JWT configuration, the External Credential, the Named Credential endpoint, and permission assignments. Complete all items before creating the first configuration.

3.1 Deploy My Domain

- Go to Setup > My Domain.
- Register a unique domain if the org does not already have one.
- After provisioning completes, deploy the domain to users.
- Record the canonical domain URL, for example https://acme.my.salesforce.com. Do not use a Lightning URL ending in lightning.force.com.

3.2 Create or identify the integration user

- Create a dedicated integration user with an API-capable license, or select an existing governed integration identity.
- Assign the minimum permissions listed in Section 2 and ensure the user is active.
- Record the exact Salesforce username. This becomes the JWT subject (sub) claim.
- Confirm login-hour, IP, session, and External Client App policies will not block JWT Bearer authentication.
- 3.2 Enable Lightning Web Security: go to Setup > Session Settings, select Use Lightning Web Security for Lightning web components and Aura components, and save. This is required because generated subscriber components reference packaged ComV Table base Lightning web components.

3.3 Create the signing certificate

- Go to Setup > Certificate and Key Management.
- Create a self-signed certificate or import an approved CA-signed certificate.
- Use the API name **ComVTable_Tooling_JWT** when the packaged External Credential expects that reference, or update the External Credential to the certificate selected by your security team.
- Download or retain the public certificate for External Client App configuration. Protect the private key according to organizational policy.
- Record the expiration date and create a renewal alert well before expiry.

3.4 Configure the External Client App for JWT Bearer Flow

ComV Tables version 1.10 uses a subscriber-owned External Client App (ECA) for OAuth JWT Bearer authentication. A new Connected App is not required for this architecture.

Why an External Client App is used instead of a Connected App

Salesforce restricted creation of new Connected Apps beginning with Spring '26 and recommends External Client Apps for new integrations. Existing Connected Apps can continue to operate, but ECA is the supported forward-looking model.

ECA provides a closed-by-default security model, fuller Metadata API support, improved first- and second-generation packaging, and a clean separation between developer-controlled settings and subscriber-admin policies.

A subscriber-owned ECA is appropriate here because the integration username, certificate, Consumer Key, authorization policy, and certificate-rotation lifecycle belong to the subscriber org. No subscriber private key or bearer token is distributed in the managed package.

Clarification: 'Connected App is no longer required' does not mean Salesforce immediately disabled all existing Connected Apps. It means ComV Tables no longer depends on one, and new implementations should use the ECA configuration described below.

Required OAuth and security configuration

- OAuth and callback: enable OAuth and use a non-operational callback URL such as https://example.com/oauth/callback. JWT Bearer Flow does not redirect to this URL; it is present only because the field is required.
- Scopes: use Manage user data via APIs (api). The comvtrailhead ECA also contains RefreshToken for its existing approval configuration; review whether that unused scope can be removed under least-privilege policy.
- Flows: enable JWT Bearer Flow and Issue JSON Web Token (JWT)-based access tokens for named users. Upload the public certificate matching ComVTable_Tooling_JWT. Leave Client Credentials, Authorization Code and Credentials, Device, and Token Exchange flows disabled.
- PKCE: require Proof Key for Code Exchange (PKCE). PKCE is enabled for compliance and protects supported authorization-code flows; it does not change the active JWT Bearer Flow because JWT does not exchange an authorization code.
- Refresh-token controls: enable Refresh Token Rotation and enforce the idle refresh-token time-to-live. Set the refresh-token policy to expire after 30 days of inactivity (SpecificInactivity / 30 Days).
- Refresh Token IP Allowlist: JWT Bearer Flow never issues a refresh token, so the active ComV Tables integration has no refresh-token caller IP. Salesforce does not assign a small, static egress IP to an individual org. Do not enter the org hostname IP, a workstation address, an all-address range, or the complete Hyperforce range. If a future external backend uses refresh tokens, enable the control with that backend's real static public NAT/egress range. Record reviewer confirmation that this control is not applicable to the current JWT-only flow.
- User authorization: set Permitted Users to Admin approved users are pre-authorized and approve only the dedicated integration user through the minimum required profile or permission set.

- After saving the ECA, copy its Consumer Key. This public client identifier becomes the JWT issuer (iss) claim in the External Credential.
- External Credential update: go to Setup > Named Credentials > External Credentials, open ComVTable Tooling API External Credential, select the JWT signing certificate, paste the Consumer Key into iss, set sub to the exact approved integration username, confirm the audience and token endpoint, and save.

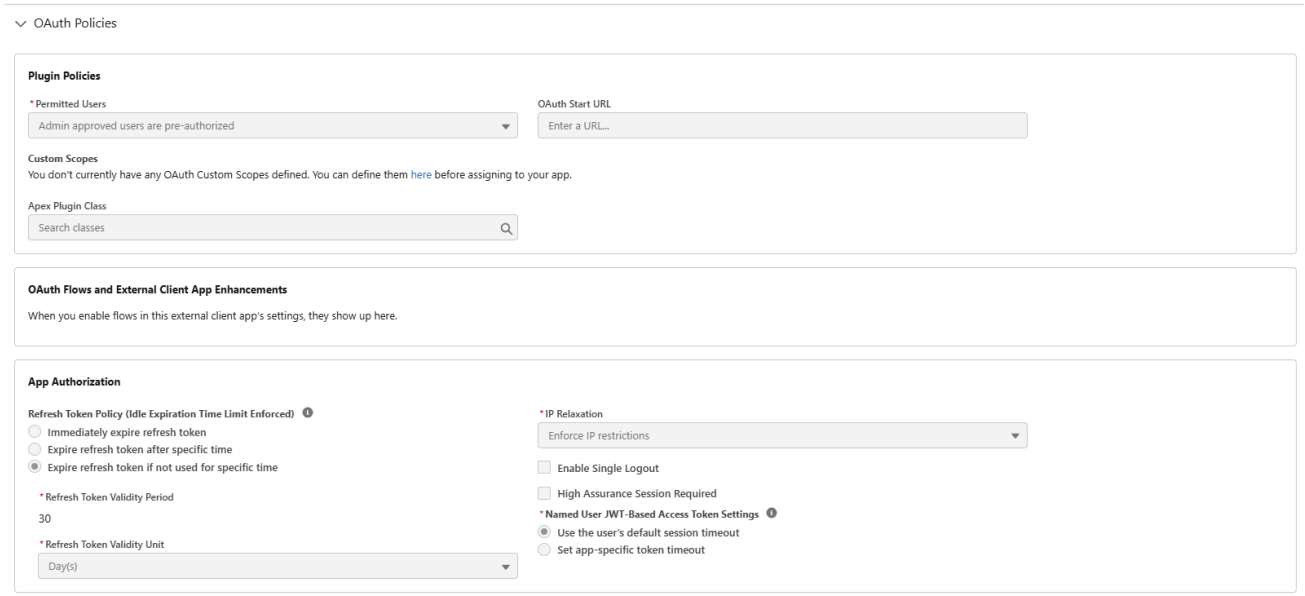


Figure 3. Existing Salesforce Setup evidence: OAuth Policies enforce the 30-day idle refresh-token policy and admin-approved user controls.

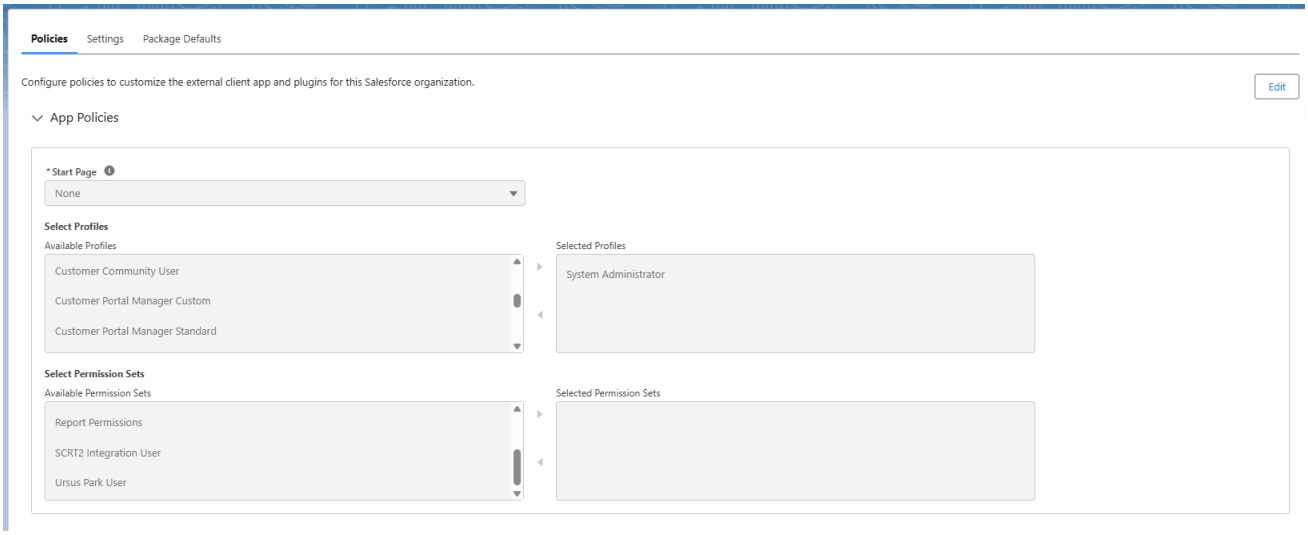


Figure 4. Existing Salesforce Setup evidence: App Policies restrict the ECA to the approved integration profile or permission set.

3.5 Configure the packaged External Credential

Go to Setup > Named Credentials > External Credentials and open ComVTable Tooling API External Credential. Verify or update the following subscriber-specific values.

Setting	Subscriber value
Authentication protocol / flow	OAuth 2.0 with JWT Bearer Flow
Signing certificate	ComVTable_Tooling_JWT or the subscriber-approved replacement
iss claim	Consumer Key from the subscriber External Client App
sub claim	Exact username of the active integration user
aud claim	https://login.salesforce.com for production; use the org-approved sandbox audience when applicable
Identity Provider URL	Production: https://login.salesforce.com/services/oauth2/token; sandbox: use the org-approved test or My Domain token endpoint
Principal	Named Principal: ComVTable_Tooling_Principal
JWT algorithm	RS256

3.6 Configure the Named Credential endpoint

- Go to Setup > Named Credentials and open ComVTable Tooling API.
- Set URL to the subscriber org My Domain base URL, for example https://acme.my.salesforce.com.
- Confirm the linked External Credential is ComVTable_Tooling_API_Ext.
- Keep Generate Authorization Header enabled.
- Keep the Named Credential enabled and save.

Do not copy the packaging-org endpoint

The source package may contain a development-org URL as a packaging default. Every subscriber must replace it with that subscriber org's canonical My Domain URL.

3.7 Assign packaged permissions

- Go to Setup > Permission Sets > ComvTable Permissions.
- Confirm External Credential Principal Access includes ComVTable_Tooling_API_Ext - ComVTable_Tooling_Principal.
- Assign the permission set to approved ComV Tables administrators/configurators and to the effective callout user if required by your execution model.

- Confirm these users also have access to the target objects and fields they will configure or display.

3.8 Verify application access

- Open App Launcher and confirm ComV Tables is available.
- Open the ComV Table Configs tab and confirm the user can create a record.
- Confirm the user can read the target object and all selected fields.
- Do not create the first production configuration until Setup Health succeeds.

4. First-time validation and health check

Run Setup Health

- Open App Launcher > ComV Tables.
- Open ComV Table Configs and start a new configuration.
- Select Health Check.
- Review every item in Setup Health. The Tooling API Callout check must report success before component generation is attempted.
- If a check fails, correct the related permission, credential, endpoint, certificate, or OAuth claim, then run the check again.

Health item	Expected result	If it fails
Configuration object access	User can access ComV Table Config records.	Assign ComvTable Permissions and verify CRUD/FLS.
Named Credential configured	ComVTable_Tooling_API resolves.	Verify the packaged credential exists and is enabled.
Tooling API callout	A query against ApexClass returns successfully.	Check endpoint, JWT certificate, iss/sub/aud, OAuth authorization, API permission, and principal access.

Perform a sandbox smoke test

- Create a simple Account table with Id and Name columns, a page size of 5, and no optional actions.
- Use Preview Table to verify readable data is returned in user mode.
- Save the configuration and wait for the asynchronous generation process.
- Open the saved record and review Deployment Status / Exception Log.

- Confirm the generated Apex class and LWC bundle appear in Setup or through Salesforce CLI metadata retrieval.
- Place the component on a test Lightning page and validate it as a non-admin user.

Security behavior

Preview and runtime queries enforce object/field access and use user-mode query behavior. Successful administrator testing does not prove that business users can see the same records or fields; always test with representative permission sets and sharing.

When setup is considered complete

- My Domain is deployed.
- Named Credential URL points to the subscriber org.
- External Credential claims and certificate belong to the subscriber.
- Integration user is active and authorized.
- ComvTable Permissions is assigned to approved users.
- Health Check and sandbox component generation both succeed.

5. Create a ComV Table configuration

Start a configuration

- Open App Launcher > ComV Tables > ComV Table Configs.
- Select New.
- Enter the table label and valid metadata names for the LWC component and Apex controller.
- Select the Salesforce object. Object, LWC component, and Apex controller names cannot be safely changed after generation.
- Select the columns to display. Only readable fields should be offered or retained.
- Enable optional features and select their eligible fields.
- Select Preview Table and resolve any access or configuration errors.
- Save only after the preview and naming validation succeed.

Core configuration fields

Field	Purpose	Practical guidance
Table Label Name	User-facing table heading.	Use a clear business label.
LWC Component Name	Base metadata name for the generated component.	Start lowercase; use letters, numbers, and single underscores; no spaces.
Apex Controller Name	Base metadata name for the generated controller.	Start with a letter; use letters, numbers, and underscores.
Object API Name	Source object queried by the runtime.	Choose once; treat as immutable.
Column Fields Name	Fields displayed as columns.	Include only fields required for the use case.
Component Description	Administrative description.	Record audience, page, owner, and intended use.

Feature configuration

Feature	Enable and configure
Row selection	Select Row Action enables selectable table rows.
Inline editing	Enable Inline Edit and select updateable fields in Fields Inline Edit.
Sorting	Enable Column Sorting (shown as “Shorting” in some labels) and select sortable fields.
Column search	Enable Column Search and select searchable fields.
Column menu	Enable Column Show/Hide and define default fields.
Pagination	Enable Pagination and set Pagination Offset from 1 through 500.
Global search	Enable Global Search and select eligible text/search fields.
Global filter	Enable Global Filter and select filterable fields.
Row actions / Flow	Enable Row Actions and optionally provide a Flow API Name. The selected record Id is passed as recordId.
CSV export	Enable Export CSV for client-side export of table data.
Related list mode	Enable Related List Mode and select the lookup/master-detail Parent Field API Name.
Advanced filters	Use Add Filter to define field/operator/value rules stored as JSON.

Preview and save

Preview returns up to five records and is designed to catch invalid object/field selection and access issues before generation. After Save, generation runs asynchronously. Refresh the saved record until Deployment Status / Exception Log confirms success or displays an actionable error.

Naming is a lifecycle decision

Renaming the record label is different from renaming generated metadata. If a metadata name must change, treat it as a controlled replacement: create a new configuration, migrate Lightning pages, test, and retire the old artifacts.

6. Add the generated table to Lightning pages

Confirm generation succeeded

- Deployment Status / Exception Log reports Tooling API component creation completed without an ApexClass or LightningComponentBundle error.
- The generated component appears in Lightning App Builder under Custom components after metadata propagation completes.
- The generated Apex class and LWC use the expected names from the configuration.

Add to a record page

- Open a representative record, select Setup > Edit Page, or open Lightning App Builder from Setup.
- Drag the generated custom component onto the page.
- For Related List Mode, place it on the parent object record page and confirm Parent Field API Name references the child object relationship field.
- Set component visibility rules if required.
- Save and activate the page for the intended apps, profiles, and form factors.
- Test with records that have zero, one, and many related rows.

Add to an app or home page

- Open Setup > Lightning App Builder and create or edit an App Page or Home Page.
- Add the generated component and arrange the page for the expected table width.
- Save, activate, and assign the page.
- Test search, filters, pagination, actions, and responsive behavior.

Access model after activation

Component visibility does not grant data access. Runtime results remain constrained by licenses, object permissions, field-level security, sharing rules, restriction rules, and record access. Inline edits and Flow actions additionally require update access and Flow permissions.

Test persona	Validate
Administrator	Configuration, health check, component generation, and App Builder placement.
Standard business user	Expected rows and fields, search/filter results, pagination, and permitted actions.
Restricted user	No exposure of inaccessible objects, fields, or records; clear error behavior.
Mobile user, if supported	Page layout, table usability, scrolling, and action behavior.

7. Runtime feature guide

Feature	User action	Expected behavior
Sort	Select an enabled column header.	Rows reorder using the configured sortable field.
Column search	Enter a value in an enabled column search control.	The displayed dataset narrows for that column.
Global search	Enter a search term in the global search control.	Configured searchable fields are evaluated.
Global filter	Select a configured field and filter value.	Records are constrained by the chosen filter.
Advanced filters	Use the administrator-defined filter rules.	Runtime query applies validated field/operator/value criteria.
Column show/hide	Open the column menu and select fields.	Visible columns change while the configured defaults remain available.
Pagination	Use next/previous controls.	Rows are displayed in the configured page size.
Inline edit	Edit an enabled cell and save.	Permitted fields update; errors are returned for validation or access failures.
Row selection	Select one or more rows.	Selected rows are available to configured actions.
Row action / Flow	Choose the configured action.	The selected record Id is passed to the Flow variable recordId.
CSV export	Select export.	The current table data is downloaded as comv-table-export.csv.

User tips

- Clear filters before reporting missing data.
- If a field is visible but not editable, confirm both the table configuration and Salesforce field-level/update permissions.
- Exported data reflects the table dataset available to the user; handle downloads according to data-classification policy.
- Report the table label, page URL, user, time, action, and exact error message when requesting support.

Flow contract

When Row Actions are enabled with a Flow API Name, the configured Flow must exist, be active, and accept an input variable named recordId. Test the Flow independently and from the table. Ensure users have permission to run the Flow and to access every object and field the Flow uses.

8. Administration and lifecycle operations

Change an existing configuration

- Treat Object API Name, LWC Component Name, and Apex Controller Name as immutable after creation.
- Preview supported field and feature changes before saving.
- Regression-test all Lightning pages using the generated component.
- Document configuration ownership, business purpose, and release ticket in Component Description.

Retire a configuration safely

- Inventory every Lightning page that uses the generated component.
- Remove the component from pages and activate the revised pages.
- Delete the generated LWC bundle and Apex controller through an approved metadata deployment process, such as Salesforce CLI or DevOps tooling.
- After metadata removal is confirmed, delete or archive the ComV Table Config record according to policy.
- Validate that no page, Flow, permission set, or deployment manifest retains a reference.

Do not delete the record first

Deleting a ComV Table Config record does not automatically delete the generated metadata. Removing the record first can leave an unmanaged component that no longer has a valid runtime configuration.

Certificate rotation

- Create or import the replacement certificate before the current certificate expires.
- Add the new public certificate to the External Client App according to its rotation capabilities.
- Update the External Credential signing-certificate reference during an approved maintenance window.
- Run Health Check and generate or update a sandbox test component.
- Retire the old certificate only after successful validation and rollback-window completion.

Upgrade readiness

- Read release notes and test package upgrades in a full or partial sandbox.
- Back up ComV Table Config data and retrieve generated Apex/LWC metadata before upgrading.
- Re-run Health Check after every package upgrade, domain change, certificate rotation, integration-user change, or OAuth policy change.
- Retest at least one generated table for each major feature used in production.

Operational monitoring

- Review Exception Log after every generation or update.
- Monitor integration-user login history and OAuth failures.
- Track certificate expiration and integration-user license/status.
- Audit permission-set assignments and External Client App policies periodically.

9. Troubleshooting

Symptom	Likely cause	Resolution
Health Check: Tooling API callout fails	Incorrect endpoint, JWT claims, certificate, principal access, OAuth policy, or inactive user.	Verify Sections 3.2-3.7; review login history; retest.
invalid_grant / user has not approved this consumer	JWT subject is not authorized by the External Client App policy.	Pre-authorize the integration user or correct the External Client App profile/permission-set policy.

Symptom	Likely cause	Resolution
invalid assertion / signature error	Certificate mismatch, expired certificate, wrong issuer, audience, or clock-related claim.	Match the uploaded public certificate to the External Credential signing certificate and verify claims.
401 Unauthorized	Token not issued or insufficient API/OAuth scope.	Confirm api scope, integration-user status, and client policy.
403 Forbidden / insufficient access	Missing Tooling API metadata permissions.	Review Author Apex, Customize Application, metadata API permission, and org policy.
Named Credential endpoint points elsewhere	Packaging default was not replaced.	Set URL to the subscriber My Domain base URL.
Preview shows no readable fields	Field-level security or invalid field selection.	Grant appropriate read access or select eligible fields.
Generated component not visible in App Builder	Generation failed, metadata propagation delay, or wrong page type.	Check Exception Log, wait briefly, refresh App Builder, and verify target page type.
Apex class exists but LWC failed	Partial Tooling API creation or resource error.	Correct the reported bundle/resource error and use a controlled retry or replacement configuration.
Inline edit fails	Field is not configured as editable, user lacks update access, or validation rule fails.	Check configuration, FLS/CRUD, sharing, and validation message.
Flow action fails	Flow inactive, wrong API name, missing recordId input, or user lacks Flow/data access.	Activate and test the Flow; verify the input variable and permissions.
Related list displays wrong/no rows	Incorrect parent lookup field or component placed on wrong record page.	Correct Parent Field API Name and page placement.

Information to collect for support

- Subscriber org type and My Domain, excluding secrets.
- Package version and approximate installation date.
- Configuration record Id, table label, object API name, and generated metadata names.
- Full Exception Log or Health Check detail, with tokens and sensitive data removed.
- Affected username/profile/permission sets and whether an administrator can reproduce.
- Steps, timestamp, page URL, and browser console/network detail where allowed.

Never share secrets

Do not place access tokens, private keys, client secrets, session identifiers, or unredacted sensitive record data in support tickets or Exception Log fields.

10. Go-live acceptance checklist

Platform and security

- ☐ My Domain is deployed and the Named Credential URL uses the canonical subscriber domain.
- ☐ Subscriber certificate exists, is current, and has a documented owner and renewal date.
- ☐ External Credential issuer, subject, audience, token URL, certificate, and Named Principal are correct.
- ☐ Integration user is dedicated, active, least-privileged, and OAuth-authorized.
- ☐ ComvTable Permissions and External Credential principal access are assigned only to approved users.
- ☐ No credentials are stored in custom metadata, configuration records, source code, or logs.
- ☐ ECA PKCE is required and Refresh Token Rotation is enabled.
- ☐ Refresh-token policy is SpecificInactivity with a 30-day validity period.
- ☐ Refresh Token IP Allowlist is configured with the real static public egress range of any refresh-token backend, or JWT-only non-applicability is documented and accepted by the reviewer.
- ☐ Dummy callback URL is non-operational and documented as unused by JWT Bearer Flow.

Functional validation

- ☐ Setup Health passes.
- ☐ Preview works for the production target object and selected fields.
- ☐ Generated Apex and LWC artifacts complete without Exception Log errors.
- ☐ Lightning page is activated for the correct app/profile/form factor.
- ☐ Standard and restricted-user tests confirm correct data visibility.
- ☐ Enabled search, filter, sorting, pagination, editing, Flow, related-list, and export features pass.

Operations and governance

- ☐ Configuration owner, support owner, integration owner, and certificate owner are recorded.
- ☐ Generated metadata is included in backup and source-control procedures.
- ☐ A safe retirement procedure is understood.
- ☐ Upgrade, regression-test, incident, and certificate-rotation procedures are documented.
- ☐ Business users received the runtime quick reference and data-export policy.

Go-live gate

Do not approve production use until every applicable checkbox is complete and evidence is retained according to the organization's change-management policy.

Appendix A. Complete configuration field reference

Field	Type	Required / notes
Table Label Name	Text	Required; user-facing table heading.
LWC Component Name	Text	Required; treat as immutable after creation.
Apex Controller Name	Text	Required; treat as immutable after creation.
Object API Name	Text	Required; treat as immutable after creation.
Column Fields Name	Text Area	Required; selected display fields.
Component Description	Text Area	Optional administrative description.
Select Row Action	Checkbox	Enables row selection.
Inline Edit Enable	Checkbox	Enables inline editing.
Fields Inline Edit	Text Area	Fields eligible for inline edit.
Shorting Enable	Checkbox	Enables sorting; legacy field spelling retained.
Shorting Fields Name	Text Area	Sortable fields.
Column Search Enable	Checkbox	Enables per-column search.
Column Search Fields Name	Text Area	Fields with column search.
Column Show/Hide Menu Enable	Checkbox	Enables column menu.
Column Show/Hide Menu Default Fields	Text Area	Default menu/visible field selection.
Pagination Enable	Checkbox	Enables pagination.
Pagination Offset	Number	Rows per page; UI range 1-500.
Global Search Enable	Checkbox	Enables global search.
Global Search Fields Name	Text Area	Fields included in global search.
Global Filter Enable	Checkbox	Enables global filters.
Global Filter Fields Name	Text Area	Fields offered for global filtering.
Advanced Filters	Long Text Area	Validated filter-builder rules stored as JSON.
Row Actions Enable	Checkbox	Enables configured row actions.
Flow API Name	Text	Optional Flow invoked with recordId.
Export CSV Enable	Checkbox	Enables CSV export.

Field	Type	Required / notes
Related List Mode	Checkbox	Filters by current parent record.
Parent Field API Name	Text	Lookup/master-detail field used by Related List Mode.
LWC API Name	Text	System-generated/read-only reference.
Apex Class API Name	Text	System-generated/read-only reference.
Exception Log	HTML	Generation status and error detail.

Appendix B. Naming and configuration examples

Purpose	Example
Table label	Active Accounts
LWC component name	activeAccounts
Apex controller base name	ActiveAccounts
Object API name	Account
Column fields	Name, Industry, OwnerId, LastModifiedDate
Generated LWC bundle	comvActiveAccountsTableLwcCom
Generated Apex class	ComvActiveAccountsController
Related-list example	Object: Contact; Parent Field API Name: AccountId; component placed on Account record page
Flow contract	Flow API Name: Review_Account; input variable: recordId, Text, Available for input

Naming rules

- LWC component base name: begin with a lowercase letter; use alphanumeric characters and single underscores; no spaces, double underscores, or trailing underscore.
- Apex controller base name: begin with a letter; use letters, numbers, and underscores; avoid reserved words.
- Choose names that remain meaningful because metadata renames require a replacement/migration process.

Environment endpoint examples

Environment	Named Credential base URL	OAuth token endpoint / audience
Production	https://acme.my.salesforce.com	https://login.salesforce.com/services/oauth2/token / https://login.salesforce.com
Sandbox	https://acme--uat.sandbox.my.salesforce.com	Use the subscriber-approved sandbox or My Domain token endpoint and matching audience.

Verify, do not assume

Salesforce domain formats and security policies can vary. Copy the canonical My Domain URL from the subscriber org and validate the chosen OAuth audience/token endpoint through the organization's integration standards.

Appendix C. Security and governance notes

- Use Named Credentials and External Credentials for callout authentication. Never reintroduce secret-bearing custom metadata fields.
- Keep the integration identity separate from human administrators and review its permissions regularly.
- Generated Apex and LWC metadata should enter source control after creation if the organization manages production metadata through a DevOps pipeline.
- Review advanced filters, Flow actions, CSV export, and inline editing as separate data-security capabilities during approval.
- Use Salesforce sharing, restriction rules, object permissions, and field-level security as the authoritative access controls.

Appendix D. Reference sources

All Salesforce references were cross-checked against the linked official documentation in September 2026.

- Authentication Protocols for Named Credentials
- Enable External Credential Principals
- Use OAuth JWT Bearer Flow with Named Credentials

External Client Apps overview:
https://help.salesforce.com/s/articleView?id=xcloud.external_client_apps.htm&type=5

Connected Apps and External Client Apps feature comparison:
https://help.salesforce.com/s/articleView?id=sf.connected_apps_and_external_client_apps_features.htm&type=5

OAuth authorization and Spring '26 Connected App creation restriction:
https://developer.salesforce.com/docs/platform/api-rest/guide/intro-oauth-and-connected-apps.html

Set IP Allowlist Ranges for Refresh Tokens:
https://help.salesforce.com/s/articleView?id=xcloud.set_ip_allowlist_for_refresh_tokens.htm&type=5

OAuth 2.0 JWT Bearer Flow (no refresh token issued):
https://help.salesforce.com/s/articleView?id=001115985&type=1

Glossary

Term	Meaning
External Credential	Authentication protocol, claims, signing certificate, and principal used by a Named Credential.
Named Credential	Named endpoint and authentication binding used by Apex callouts.
JWT	Signed JSON Web Token used to obtain an OAuth access token without a stored password.
Tooling API	Salesforce API used here to query and create Apex and Lightning metadata artifacts.
Subscriber org	The Salesforce org where the managed package is installed.
FLS	Field-level security.