ConnectHooks

Customer Technical Guide

Architecture, Configuration, Data Model & Security

Package	Salesforce Managed Package
Namespace	xccch
Version	1.0

Purpose of this edition

This guide is intended for ConnectHooks customers and Salesforce administrators who want to understand how the solution is configured and how it behaves technically, without needing code-level implementation details.

Customer-focused scope

Internal Apex class names, method signatures, trigger-handler details, LWC implementation names, SOQL statements and code-level execution paths are intentionally omitted. The guide retains the package architecture, setup flow, objects and fields, custom settings, custom metadata, security controls, mapping behavior and operational limits that customers may need to understand or administer.

1. What Is ConnectHooks?

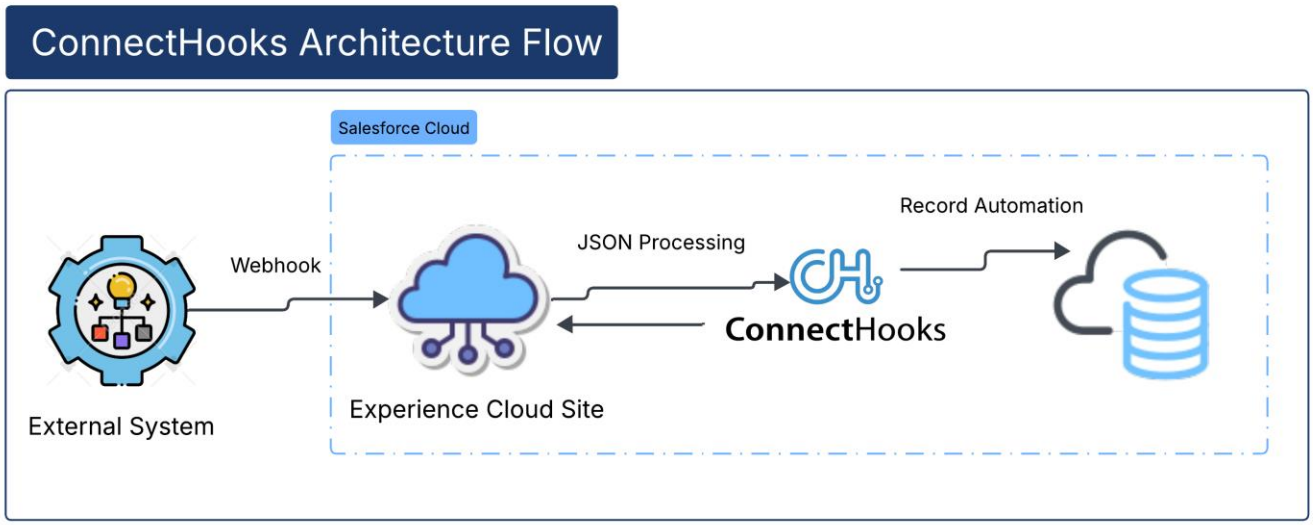
ConnectHooks is a Salesforce managed package that gives external systems - forms, applications, services, or any HTTP client - a secure endpoint for posting JSON data into Salesforce. ConnectHooks maps values from the incoming JSON to Salesforce fields and creates records automatically, without requiring custom integration code for each webhook.

At a high level, ConnectHooks combines webhook ingestion, field mapping and Salesforce record creation inside the Salesforce environment.

Capability	What it does
Webhook Endpoint	Receives HTTP POST requests from external systems.
API Key Authentication	Validates an API key supplied by the calling system before processing.
Secure URL Code	Uses a unique secure code in the webhook URL as an additional request factor.
Rate Limiting	Limits requests per webhook within a configured time window.
JSON Schema Check	Optionally checks required fields before mapping begins.
Field Mapping	Maps JSON values to Salesforce fields using JSONPath expressions.
Record Creation	Creates Salesforce records dynamically from configured mappings.
Array Support	Creates multiple records from JSON arrays in a single request.
Parent-Child Mapping	Links related parent and child records created from the same payload.
Optional Custom Processing	Allows advanced processing to replace the standard mapping path for selected webhooks.
Submission Audit	Can retain the raw payload and processing details in Salesforce.
File Attachments	Supports saving files from incoming submissions as Salesforce Files.

2. High-Level Architecture

ConnectHooks is designed around three practical layers: configuration, runtime processing and shared platform services. The architecture diagram below shows the external-to-Salesforce flow.



Layer 1 - Configuration

- An administrator completes the ConnectHooks setup wizard and chooses the Salesforce public site used by the package.
- The administrator creates webhooks and defines how JSON fields map to Salesforce objects and fields.
- Configuration is stored in ConnectHooks custom objects, custom settings and custom metadata.

Layer 2 - Runtime Processing

- An external system posts JSON to the webhook endpoint exposed through the Salesforce site.
- ConnectHooks validates the webhook identity and credentials, applies configured limits and validation, evaluates mappings, then creates or updates Salesforce records.
- A structured HTTP response is returned to the calling system.

Layer 3 - Salesforce Platform Services

- Salesforce objects store webhook definitions, mapping definitions, schemas and submission audit information.
- Custom settings store org-level configuration and per-webhook runtime state such as rate-limit tracking.
- Custom metadata stores package-wide operational limits and system configuration.
- Salesforce Files is used when file or raw-payload storage is enabled.

3. First-Time Setup and Webhook Configuration

After package installation, ConnectHooks initializes the base configuration. The administrator then completes the ConnectHooks Config setup flow. Steps 1-6 are completed once for the org; webhook creation is repeated whenever a new external form, application or service needs its own endpoint.

Step	Configuration step	What happens
1	Create Public Site	Create the Salesforce public site that will expose the webhook endpoint. This is a manual Salesforce Setup step.
2	Select Public Site	Choose the site that ConnectHooks will use. The selected site name is stored in ConnectHooks_Config_Settings__c.
3	Assign Public Site User Permissions	Assign the package permission set that gives the site guest user the minimum access required to receive and process ConnectHooks requests.
4	Create Files Library	Create the Salesforce Files library used by ConnectHooks when incoming submissions need to store files.
5	Enable JSON Submission Storage	Choose whether raw JSON payloads are retained for submission audit and troubleshooting.
6	Generate JSONPath API Key	Register the org with the ConnectHooks managed backend used by the JSONPath Generator feature and store the returned key in org configuration.
7	Create Webhook	Create a named webhook. ConnectHooks generates a secure URL code and an API key and exposes the complete endpoint URL for use by the external system.

Webhook endpoint structure

Each webhook has its own name, secure URL code and API key. The endpoint follows this pattern:

```
{Salesforce Site URL}/services/apexrest/xccch/ConnectHooksSubmissionService/{webhookName}/{secureCode}
```

Credential handling

The secure URL code is embedded in the endpoint URL. The API key is supplied by the external caller, preferably through the x-api-key HTTP header. Both must be valid for the request to proceed.

Field mapping configuration

The Mapping UI lets an administrator define how an incoming JSON document becomes one or more Salesforce records. The saved configuration includes:

- Target Salesforce objects for the webhook.
- Parent-child structure between mapped Salesforce objects, when applicable.
- Salesforce field API names and the JSONPath expression used to obtain each value.
- An optional JSON schema associated with the webhook for required-field validation.

Mapping definitions are stored as data in Salesforce rather than being hard-coded. Changing a mapping therefore does not require a code deployment.

4. Runtime Processing Flow

When a webhook request is received, ConnectHooks processes it in a consistent sequence. The following view is intentionally implementation-neutral and focuses on behavior that matters to administrators and integrators.

Step	Stage	Customer-visible behavior
1	Identify the webhook	Reads the webhook name and secure code from the endpoint URL. Invalid URL structure is rejected.
2	Parse the request	Reads the JSON body. Empty or invalid input is rejected before mapping.
3	Read the API key	Looks for the API key in the x-api-key header and also supports designated root-level JSON key names.

Step	Stage	Customer-visible behavior
4	Authenticate and apply limits	Requires an active webhook, a matching secure URL code and a matching API key, then applies the configured per-webhook rate limit.
5	Validate required fields	If a JSON schema is configured, ConnectHooks checks required fields before processing mappings.
6	Choose processing path	Uses the standard mapping engine unless optional custom processing is enabled for that webhook.
7	Load mappings	Loads the target objects, parent-child hierarchy and field-level JSONPath expressions saved for the webhook.
8	Create or update records	Evaluates JSONPath values, converts them to Salesforce-compatible field values, creates multiple records for array mappings, links parent and child records, and uses an external ID for upsert when one is applicable; otherwise records are inserted.
9	Audit and respond	If enabled, stores submission information and raw JSON, then returns a structured result to the caller.

Typical HTTP outcomes

Status	Meaning
200	Request processed successfully.
207	Mixed result when some mapped record operations succeed and others fail.
400	Request format, JSON body, schema requirement or mapping configuration is invalid.
401	API key or secure URL code is missing or invalid.
404	The requested webhook cannot be found or is not active.
429	The webhook has exceeded its configured request limit for the current time window.

5. JSONPath and Mapping Behavior

ConnectHooks uses JSONPath expressions to select values from incoming JSON. The mapping experience supports simple properties, nested structures and collection-oriented expressions. The package supports RFC 9535 JSONPath syntax, including the examples below.

Expression	Example	Result
Root	$.name	Value of the name key at the document root.
Dot notation	$.user.email	Nested property access.
Bracket notation	$["user"]["email"]	Bracket-style nested property access.
Wildcard	$.orders[*].id	All id values in the orders array.
Array index	$.orders[0].id	The first order id.
Array slice	$.orders[0:3]	The first three array items.
Recursive descent	$..email	All email properties found at any depth.
Filter expression	$.orders[?(@.status=="active")].id	Ids for array items that satisfy the filter.

Collection mappings

When a mapping targets values from a JSON collection, ConnectHooks can create one Salesforce record per array element. Fields originating from the same array object are expected to map to the same Salesforce object so that each array element produces a coherent record.

Parent-child record creation

Mapped Salesforce objects can be arranged in a parent-child hierarchy. When both parent and child records are created from the same webhook payload, ConnectHooks links the child to the corresponding parent through the applicable Salesforce relationship.

Type-aware mapping

The Mapping UI prevents mappings that are clearly incompatible with the target Salesforce field type. Null values are not accepted as a field mapping value. This validation helps catch mapping issues during configuration rather than after a webhook is live.

Schema validation scope

In version 1.0, the configured JSON schema is used to check the schema's required fields before processing. This is a required-field presence check and should not be interpreted as a full JSON Schema validation engine.

6. Salesforce Data Model

ConnectHooks stores configuration and submission information in custom Salesforce objects. These records are part of the package data model and are useful for understanding how webhook definitions and mappings are represented in the org.

CH_Webhook__c - one record per webhook

Field	Type	Purpose
Webook_Name__c	Text	Unique webhook name used in the endpoint URL. Spaces are not allowed.
Secure_URL_Code__c	Text	Unique 48-character code embedded in the endpoint URL as an additional request factor.
Enc_Api_Key__c	EncryptedText	48-character webhook API key stored in an encrypted Salesforce field.
is_Active__c	Checkbox	Only active webhooks process requests.
isDelete__c	Checkbox	Soft-delete flag used to hide a webhook without removing the underlying record.
Enable_Storage_Of_Form_Submission_JSON__c	Checkbox	Controls whether raw JSON is retained for webhook submissions.
Schema__c	Long Text	Optional JSON schema used for request validation.

CH_Mapping_Object__c - one record per mapped Salesforce object

Field	Type	Purpose
Custom_Object__c	Text	API name of the target Salesforce object, for example Lead or Order__c.
Webhook__c	Lookup	Parent CH_Webhook__c record.
Parent_Mapping_Object__c	Lookup	Self-lookup that identifies the parent mapping object for hierarchical mappings.

CH_Mapping__c - one record per field mapping

Field	Type	Purpose
Fields__c	Text	Salesforce target field API name, for example LastName or Email.
JsonPath_Expression__c	Long Text	JSONPath expression used to extract the value from the incoming JSON.
Mapping_Object__c	Lookup	Parent CH_Mapping_Object__c record.

JSON_Schema__c - schema storage per webhook

Field	Type	Purpose
CH_Webhook__c	Lookup	Parent CH_Webhook__c record.
Schema__c	Long Text	JSON schema text associated with the webhook. The required array is used during request checking.

Form_Submission__c - submission audit record

Field	Type	Purpose
Form_Name__c	Text	Webhook name that received the submission.
Form_Submission_Unique_Id__c	Text	Unique key used to identify duplicate submissions.
Error_Details__c	Long Text	Processing error information recorded for failed requests.
SObject_Record_Id__c	Text	ID of the Salesforce record associated with the submission.
Form_Entry_Id__c	Text	External entry identifier supplied by the calling system.

7. Custom Settings and Custom Metadata

ConnectHooks uses Salesforce configuration data to separate org-specific setup, per-webhook runtime tracking and package-wide operational values from the mapping records themselves.

ConnectHooks_Config_Settings__c - hierarchy custom setting

Stores org-level configuration. The package maintains one org-level configuration record.

Field	Purpose
Public_Site_Used__c	Name of the Salesforce public site selected for ConnectHooks.
Secure_URL_Code__c	Legacy global secure code retained from the earlier global-webhook model.
Step_Create_Public_Site__c	Tracks whether the administrator completed the public-site creation setup step.
Disable_Storage_Of_Form_Submission_JSON__c	Inverted storage flag: true means JSON submission storage is disabled.
JSONPath_Gen_Sys_API_Key__c	API key used by the JSONPath generation backend integration.
AI_BackEnd_EndPoint_Url__c	Base URL of the ConnectHooks managed backend service. The package default is https://apps.getxccelerance.com/connecthooks.

ConnectHooks_Rate_Limit_Tracker__c - hierarchy custom setting

Maintains request counters per webhook for the current rate-limit window.

Field	Purpose
Total_API_Hits__c	Number of requests received by the webhook in the current time window.
Last_Reset_Time__c	Timestamp of the last counter reset, used to determine when the current window expires.

ConnectHooks_Custom_Handler__c - hierarchy custom setting

Optional configuration for webhooks that use custom processing instead of the standard mapping engine.

Field	Purpose
Handler_Class__c	Identifier of the custom processing implementation configured for the webhook.
Is_Active__c	When enabled, the standard mapping flow is bypassed and the configured custom processing path is used.

CH_Config_Meta_Store__mdt - custom metadata type

Stores package-wide operational configuration. The primary package record is named Free_Edition.

Field	Purpose
API_Rate_Limit__c	Maximum API requests per webhook within the configured time window. Default: 300.
Time_Window_Hours__c	Duration of the rate-limit window. Default: 24 hours.
Webhook_Creation_Limit__c	Maximum number of webhooks a user can create under the packaged limit. Default: 2.
Global_API_Key__c	System authentication value used when communicating with the ConnectHooks managed backend.

8. Security and Access Model

ConnectHooks uses multiple independent controls around the public webhook endpoint. The intent is that possession of the endpoint URL alone is not sufficient to submit data successfully.

Control	How it protects the flow
API Key	The caller supplies an API key. A missing or incorrect value is rejected.
Secure URL Code	Each webhook URL includes a unique secure code that must match the webhook configuration.
Active Webhook Check	Only webhooks marked active can process incoming requests.
Rate Limiting	Each webhook is limited by the configured request count and time window. Requests beyond the limit are rejected until the window resets.
Schema Requirement Check	Optional request validation can block payloads that omit required fields defined by the webhook schema.
Least-Privilege Guest Access	The public-site guest user receives only the package access required for webhook processing.
Encrypted API-Key Field	The webhook API key is stored in an EncryptedText field on CH_Webhook__c.
Configuration-Based Secrets	Keys and service URLs are stored in Salesforce custom settings/custom metadata rather than being hard-coded in application logic.

Permission sets

Permission set	Assigned to	Purpose
ConnectHooks_Admin	Installing user and other administrators who are assigned the permission set.	Administrative access to ConnectHooks configuration, package objects and the Mapping UI.
ConnectHooks_Public_Site_User_Permissions	Salesforce site guest user.	Minimum package access required for the public endpoint to read webhook and mapping configuration and process incoming submissions.

Operational note

The webhook endpoint is intentionally public at the Salesforce Site layer so external systems can reach it, but ConnectHooks still requires webhook-specific credentials and applies package security controls before mapping or record creation.

9. Submission Audit, JSON Storage and Files

ConnectHooks can retain submission information in Salesforce to support administration and troubleshooting. When JSON storage is enabled, the raw request payload is recorded for the webhook submission and can also be saved as a Salesforce File in the package library.

- Form_Submission__c provides the package-level submission audit record.
- The audit record can retain the webhook name, a unique submission identifier, external entry identifier, resulting Salesforce record ID and error details.
- A Salesforce Files library is created during setup for ConnectHooks-managed file storage.
- Incoming file references supported by the payload can be saved and associated as Salesforce Files.

Storage choice

Raw JSON storage is configurable. Administrators can disable it at the org configuration level when they do not want ConnectHooks to retain incoming payloads for audit purposes.

10. Operational Limits and Administration

The package uses custom metadata and custom settings to enforce usage limits and maintain runtime counters. The defaults documented for version 1.0 are:

Limit	Default	Notes
Requests per webhook	300	Per configured 24-hour window.
Rate-limit window	24 hours	Counter resets after the configured window expires.
Webhook creation limit	2	Package default for the Free_Edition metadata record.

The request counter is tracked separately for each webhook. If the metadata that defines a valid rate limit is not available, the documented runtime behavior is to log the condition rather than block the request solely because the rate-limit configuration is missing.

Webhook creation limits are enforced when a new webhook is created. The documented version 1.0 behavior counts webhooks created by a user against the creation limit, including webhooks that were later deleted.

11. Advanced Processing Option

The standard ConnectHooks path is configuration-driven: JSONPath mappings create Salesforce records without custom code for each integration. For use cases that require specialized processing, a webhook can be configured to use a custom processing implementation instead of the standard mapping engine.

- Authentication, webhook identity checks and rate limiting remain part of the ConnectHooks request flow.
- The custom path is configured per webhook through ConnectHooks_Custom_Handler__c.
- When the custom path is active, the standard field-mapping record creation flow is bypassed for that webhook.

When to use it

Use the standard mapping engine for normal JSON-to-Salesforce record ingestion. Reserve custom processing for cases where business logic cannot be represented by the standard mapping model.

12. Summary

ConnectHooks provides a Salesforce-native configuration model for receiving webhook JSON and turning it into Salesforce records. Administrators configure the public site, webhook credentials, mapping definitions and optional storage once, then external systems can submit data to the webhook endpoint on an ongoing basis.

From a technical administration perspective, the most important parts of the solution are:

- Webhook configuration is represented by CH_Webhook__c records with per-webhook credentials, status and schema/storage settings.
- Mapping configuration is stored as Salesforce data in CH_Mapping_Object__c and CH_Mapping__c, including parent-child structure and JSONPath expressions.
- Org-level setup and runtime state are stored in hierarchy custom settings; package-wide limits are stored in CH_Config_Meta_Store__mdt.
- Requests pass through credential checks, active-webhook validation, rate limiting and optional required-field validation before record creation.
- Submission information and raw JSON can be retained for audit, and incoming files can be stored as Salesforce Files.