

User Scenarios

Origo Cloud Events Core

AppSource Validation Document

Publisher	Origo
App Name	Origo Cloud Events Core
Version	28.0.0.0
Platform	Business Central Cloud (v28.0+)
Supported Locales	en-US, is-IS
Document Date	June 2026

Table of Contents

Scenario 1: Installation and Initial Setup	
Scenario 2: Discovering Available Message Types (Metadata)	
Scenario 3: Retrieving Customer Credit Limit Information	
Scenario 4: Retrieving Record Data via API	
Scenario 5: Queuing an Asynchronous Message	
Scenario 6: Field Access Restrictions (Security)	

Scenario 7: Licensing and Quota Management

Scenario 8: Error Handling — Invalid Message Type

Scenario 9: Error Handling — Missing Required Parameters

Scenario 10: Permission Verification

Scenario 11: Uninstallation

Overview

Origo Cloud Events Core provides a structured, CloudEvents-compliant API framework for Business Central. It enables external applications and integrations to execute business operations (read data, post documents, manage records) through a standardized message-based API. The extension supports both synchronous and asynchronous (queued) processing, field-level security restrictions, extensible message type implementations, and a message-quota licensing model.

Key Features Covered

- Cloud Events API (REST/OData) with Task and Queue endpoints
- 80+ built-in message types across Sales, Data, Metadata, Finance, Inventory, Approval, and more
- Interface-based extensible architecture for custom implementations
- Field-level access restrictions for data security
- Asynchronous background processing with retry and cancellation
- Message-quota licensing (User pool and App Registration pool)
- Retention policies for message log cleanup

Test Environment Requirements

All scenarios can be executed on a standard Business Central online sandbox with the CRONUS demonstration company. No external services, credentials, or third-party integrations are required. The API calls in these scenarios use the built-in BC web client UI and the Cloud Event Messages page to demonstrate functionality without requiring a separate API client tool.

Scenario 1: Installation and Initial Setup

Area: Installation & Activation

Setup

1. Start with a Business Central online sandbox environment using the CRONUS demonstration company.
2. Ensure the **Origo Cloud Events Core** extension is installed (via Extension Management).

Steps

1. Search for **"Extension Management"** in the Business Central search bar.
2. Verify that **"Origo Cloud Events Core"** appears in the list with Status = Installed.
3. Search for **"Cloud Events Setup"** in the Business Central search bar.
4. Verify the Cloud Events Setup page opens without error.
5. Observe the **"Available Message Types"** count displayed on the setup page.
6. Choose the **"Messages"** action group, then choose **"Cloud Event Messages"** to open the messages list.
7. Verify the Cloud Event Messages page opens (it will be empty on a fresh install).
8. Navigate back to the Cloud Events Setup page.
9. Observe the **"Task API Url"** and **"Queue API Url"** fields — these display the API endpoint addresses.

Expected Results

Step	Expected Result
2	Extension is listed with status "Installed" and version 28.0.0.0 (or later).
4	Setup page opens showing configuration sections: General, MCP Chat, and Cloud Event Messages.
5	Available Message Types count is greater than 0 (typically 80+).
7	Cloud Event Messages page opens with no errors, showing an empty list.
9	Both API URL fields display valid URLs containing <code>/api/origo/cloudEvent/v1.0/</code> .

Notes

- On first install, a trial license of 1,000 messages per pool (User and App Registration) is automatically provisioned.
- Retention policies for Cloud Event Messages, Integrations, and Delete Log are automatically registered during installation.

Scenario 2: Discovering Available Message Types (Metadata)

Area: Core Functionality

Setup

1. Ensure Scenario 1 has been completed (extension installed, setup accessible).

Steps

1. Search for **"Cloud Events Setup"** in the Business Central search bar.
2. On the Cloud Events Setup page, choose the **"Messages"** action group, then choose **"Cloud Event Messages"**.
3. On the Cloud Event Messages page, choose the **"Request"** action group, then choose **"Create Request"** (or the equivalent new message action).
4. In the message editor, set **"Cloud Event Message Type"** to `Help.MessageTypes.Get`.
5. Set **"Source"** to `AppSource Validation Test`.
6. Choose the **"Process"** action to execute the message.
7. After processing completes, observe the **"Status"** field on the message line.
8. Choose the **"Response"** action to view the response data.

Expected Results

Step	Expected Result
4	The message type dropdown includes <code>Help.MessageTypes.Get</code> among many available types.
7	Status shows "Completed" (or equivalent success status).
8	Response data contains a JSON array listing all available message types with their names, categories, and descriptions. The list includes types like <code>Help.Tables.Get</code> , <code>Data.Records.Get</code> , <code>Customer.CreditLimit.Get</code> , etc.

Notes

- The `Help.*` message types are license-exempt — they do not consume quota.
- This demonstrates the self-discovery capability: external integrations can query available functionality at runtime.

Scenario 3: Retrieving Customer Credit Limit Information

Area: Core Functionality (Sales)

Setup

1. Ensure the CRONUS demonstration company has customer data (Customer "10000" - Adatum Corporation, or similar default customer).
2. Navigate to **"Customers"** and note a customer number that has a Credit Limit set (e.g., customer "10000").

Steps

1. Search for **"Cloud Event Messages"** in the Business Central search bar.
2. Choose the action to create a new message request.
3. Set **"Cloud Event Message Type"** to `Customer.CreditLimit.Get`.
4. Set **"Source"** to `AppSource Validation Test`.
5. Set **"Cloud Event Message Subject"** to `10000` (or the noted customer number).
6. Choose the **"Process"** action to execute the message.
7. Observe the **"Status"** field — it should show success.
8. Observe the **"Response Time"** field — it should show processing duration.
9. Choose the **"Response"** action to view the response data.

Expected Results

Step	Expected Result
7	Status shows "Completed" .
8	Response Time shows a value (typically under 1 second).
9	Response JSON contains fields: <code>status</code> ("Success"), <code>customerNo</code> , <code>customerName</code> , <code>creditLimitLCY</code> , <code>balanceLCY</code> , <code>remainingCreditLimitLCY</code> , and credit status indicators.

Notes

- The credit limit calculation uses the implementation configured in Cloud Events Setup (default: standard BC calculation).
- This message type consumes one message from the User quota pool.

Scenario 4: Retrieving Record Data via API

Area: Core Functionality (Data)

Setup

1. Ensure the CRONUS demonstration company has item records (standard demo data is sufficient).

Steps

1. Search for **"Cloud Event Messages"** in the Business Central search bar.
2. Choose the action to create a new message request.
3. Set **"Cloud Event Message Type"** to `Data.Records.Get`.
4. Set **"Source"** to `AppSource Validation Test`.
5. Set **"Cloud Event Message Subject"** to `Item` (the table name).
6. In the **"Data"** (request payload) field, enter: `{"top": 5}`
7. Choose the **"Process"** action to execute the message.
8. Observe the **"Status"** field.
9. Choose the **"Response"** action to view the response data.

Expected Results

Step	Expected Result
8	Status shows "Completed" .
9	Response JSON contains a <code>data</code> array with up to 5 item records. Each record includes field values like <code>No_</code> , <code>Description</code> , <code>UnitPrice</code> , etc., following the data shipping standard format.

Notes

- The `top` parameter limits the number of records returned — essential for performance with large tables.
- Field names in the response follow BC naming with dots replaced by underscores (e.g., `"No."` becomes `"No_"`).
- Field access restrictions (if configured) automatically exclude restricted fields from the response.

Scenario 5: Queuing an Asynchronous Message

Area: Core Functionality (Queue/Background Processing)

Setup

1. Ensure the extension is installed and the Cloud Events Setup is accessible.

Steps

1. Search for "**Cloud Event Messages**" in the Business Central search bar.
2. Choose the action to create a new message request.
3. Set "**Cloud Event Message Type**" to `Help.Tables.Get`.
4. Set "**Source**" to `AppSource Validation Test`.
5. Choose the "**Queue**" action (instead of "Process") to schedule the message for background processing.
6. Observe the "**Status**" field immediately after queuing — it should show a queued/pending state.
7. Wait a few seconds, then refresh the page (or press F5).
8. Observe the "**Status**" field — it should now show success (completed by the background task).
9. Observe the "**Response Data Length**" field — it should show a value greater than 0.
10. Choose the "**Response**" action to view the response data.

Expected Results

Step	Expected Result
6	Status shows a pending/queued state indicating the task has been scheduled.
8	After refresh, status shows " Completed " — the background task processed the message.
9	Response Data Length is greater than 0.
10	Response contains a JSON array of all available tables in the database (table ID, name, caption, namespace).

Notes

- Background processing uses BC Task Scheduler. Processing time depends on system load.
- If the background task has not completed after 30 seconds, refresh again — sandbox environments may have slower task processing.

- `Help.*` types are license-exempt and ideal for testing the queue mechanism.

Scenario 6: Field Access Restrictions (Security)

Area: Data Integrity / Security

Setup

1. Search for **"Cloud Events Setup"** in the Business Central search bar.
2. Choose the **"Field Restrictions"** action to open the Cloud Events Field Accesses page.

Steps

1. On the Cloud Events Field Accesses page, choose **"New"** to create a new restriction.
2. Set **"Table No."** to 18 (Customer table).
3. Set **"Field No."** to a sensitive field number, e.g., 21 (Customer Posting Group) or another sensitive field.
4. Set **"Restriction Type"** to Read (blocks reading this field via the API).
5. Leave **"User Security ID"** as the current user (or blank for all users).
6. Close the Field Accesses page.
7. Navigate to **"Cloud Event Messages"** and create a new message.
8. Set **"Cloud Event Message Type"** to `Data.Records.Get`.
9. Set **"Source"** to `AppSource Validation Test`.
10. Set **"Cloud Event Message Subject"** to `Customer`.
11. In the request data, enter: `{"top": 1}`
12. Choose **"Process"** to execute the message.
13. Choose **"Response"** to view the response data.
14. Verify that the restricted field does NOT appear in the response JSON for the customer record.

Expected Results

Step	Expected Result
1–5	Field restriction record is created successfully without errors.
12	Message processes successfully with Status "Completed".
14	The response JSON for the customer record does NOT include the restricted field. All other unrestricted fields are present.

Cleanup

1. Navigate back to **"Cloud Events Setup"** > **"Field Restrictions"**.
2. Delete the restriction record created in this scenario.

Notes

- Field restrictions are enforced at the API layer — the BC UI is not affected.
- Restriction Type "Both" blocks both reading and writing; "Write" blocks only modifications.
- Wildcard support: Table No. = 0 restricts across all tables; Field No. = 0 restricts all fields in a table.

Scenario 7: Licensing and Quota Management

Area: Core Functionality (Licensing)

Setup

1. Ensure the extension has been freshly installed (trial license should be active with 1,000 messages per pool).

Steps

1. Search for **"Cloud Events Setup"** in the Business Central search bar.
2. Observe the licensing information displayed on the setup page (quota status or notification area).
3. Process a chargeable message (e.g., `Customer.CreditLimit.Get` as in Scenario 3).
4. Return to the Cloud Events Setup page.
5. Verify that the quota consumption reflects the processed message.
6. Choose the **"Request License"** action.
7. Verify that the license request action is available and navigates to the license request form or URL.

Expected Results

Step	Expected Result
2	Licensing section displays current quota status (trial license with 1,000 messages).
3	Message processes successfully (trial quota is available).
5	Consumed messages count has increased by 1.
7	The "Request License" action opens a license request page or navigates to the license portal URL.

Notes

- Trial provides 1,000 messages in the User pool and 1,000 in the App Registration pool.
- `Help.*` and `Webhook.*` message types are exempt from quota — they never consume messages.
- When quota is exhausted, chargeable messages return a structured error JSON with a link to request more licenses.

Scenario 8: Error Handling — Invalid Message Type

Area: Error Handling

Setup

1. Ensure access to the Cloud Event Messages page.

Steps

1. Search for **"Cloud Event Messages"** in the Business Central search bar.
2. Choose the action to create a new message request.
3. Attempt to set **"Cloud Event Message Type"** to a non-existent type name — observe that only valid message types can be selected from the dropdown.
4. Set **"Cloud Event Message Type"** to `Customer.CreditLimit.Get`.
5. Set **"Source"** to `AppSource Validation Test`.
6. Set **"Cloud Event Message Subject"** to `INVALID-CUSTOMER-99999` (a customer number that does not exist).
7. Choose **"Process"** to execute the message.
8. Observe the **"Status"** field.
9. Choose the **"Response"** action to view the response data.

Expected Results

Step	Expected Result
3	The message type field only accepts valid, registered message types. Invalid values cannot be entered.
8	Status shows an error state (the message was processed but the customer was not found).
9	Response JSON contains "status": "Error" and an "error" field with a descriptive message such as "Customer INVALID-CUSTOMER-99999 not found." The error message is user-friendly and identifies what went wrong.

Notes

- Error responses follow a consistent JSON structure: **{ "status": "Error", "error": "<message>" }**.
- Failed messages do NOT consume license quota.
- The error is captured cleanly — no unhandled BC error dialogs appear.

Scenario 9: Error Handling — Missing Required Parameters

Area: Error Handling

Setup

1. Ensure access to the Cloud Event Messages page.

Steps

1. Search for **"Cloud Event Messages"** in the Business Central search bar.
2. Choose the action to create a new message request.
3. Set **"Cloud Event Message Type"** to `Data.Records.Get`.
4. Set **"Source"** to `AppSource Validation Test`.
5. Leave the **"Cloud Event Message Subject"** field empty (no table specified).
6. Leave the request **"Data"** field empty (no table name or number in payload).
7. Choose **"Process"** to execute the message.
8. Observe the **"Status"** field.
9. Choose the **"Response"** action to view the response data.

Expected Results

Step	Expected Result
8	Status shows an error state.
9	Response JSON contains "status": "Error" with a clear error message indicating that a table must be specified (e.g., via subject field or data payload). The message identifies the missing parameter.

Notes

- All message types validate required input before processing and return structured error responses.
- No unhandled exceptions or raw error dialogs are shown to the user.

Scenario 10: Permission Verification

Area: Permission Verification

Setup

1. Ensure you have access to User Management (or use two user accounts: one admin, one restricted).
2. Identify the permission sets provided by the extension:
 - **CE Full Access** — full read/write to all Cloud Events objects
 - **CE API Access** — access to API pages only
 - **CE Read All** — read-only access to Cloud Events data

Steps

1. Search for "**Permission Sets**" in the Business Central search bar.
2. Filter on Extension Name = "**Origo Cloud Events Core**".
3. Verify the following permission sets are listed:
 - CE Full Access
 - CE Force Access
 - CE API Access
 - CE API Integration
 - CE API Storage
 - CE API Translation
 - CE Approval Admin
 - CE Approval Log Read
 - CE Edit Co. Memory
 - CE Read All
 - CE G/L Posting Gate
 - CE Item Posting Gate
 - CE FA Posting Gate
 - CE Job Posting Gate
 - CE Res. Posting Gate
 - CE Whse. Posting Gate
4. Select "**CE Full Access**" and choose "**Permissions**" to view its contents.
5. Verify the permission set grants appropriate access to Cloud Events tables, codeunits, and pages.
6. Close the permission set details.
7. Search for "**Cloud Events Setup**" — verify it opens (current user has SUPER or CE Full Access).

Expected Results

Step	Expected Result
3	All listed permission sets are present and associated with the extension.
5	CE Full Access includes read/write permissions for Cloud Events tables (Message, Setup, Integration, etc.) and execute permissions for relevant codeunits.
7	Cloud Events Setup page opens successfully for a user with appropriate permissions.

Notes

- Permission sets follow the principle of least privilege — API-only users receive **CE API Access** without administrative capabilities.
- Posting Gate permission sets control which posting operations can be performed through the API (G/L, Item, FA, Job, Resource, Warehouse).
- Users without any CE permission set cannot access Cloud Events functionality.

Scenario 11: Uninstallation

Area: Uninstallation

Setup

1. Ensure the extension is installed and some message data exists (from executing previous scenarios).

Steps

1. Search for "**Extension Management**" in the Business Central search bar.
2. Locate "**Origo Cloud Events Core**" in the list.
3. Select the extension and choose "**Uninstall**".
4. On the uninstall confirmation, choose to **delete extension data** (or retain, depending on options presented).
5. Confirm the uninstallation.
6. Wait for the uninstallation to complete.
7. Search for "**Cloud Events Setup**" in the Business Central search bar.
8. Search for "**Cloud Event Messages**" in the Business Central search bar.

Expected Results

Step	Expected Result
3–6	Uninstallation completes without errors.
7	No results found — the Cloud Events Setup page is no longer available.
8	No results found — the Cloud Event Messages page is no longer available.

Notes

- Standard Business Central operations (Sales, Purchasing, Inventory, etc.) continue functioning normally after uninstallation.
- If "delete extension data" was selected, all Cloud Events message history, setup, and field restriction records are removed.
- If "retain data" was selected, the data remains in the database and will be available if the extension is reinstalled.
- No orphaned background tasks or scheduled jobs remain after uninstallation.

Appendix: Extension Information

App ID	a629b897-7541-4562-bebb-c6122f15801c
Publisher	Origo
Name	Origo Cloud Events Core
Version	28.0.0.0
Target	Cloud
Application Version	28.0.0.0
Platform Version	28.0.0.0
Runtime	17.0
ID Range	52014710 – 52015710
Supported Locales	en-US, is-IS
Privacy Statement	https://www.origo.is/um-origo/stefnur/personuverndarstefna
EULA	https://www.origo.is/skilmalar-og-oryggismal
Help URL	https://origopublic.blob.core.windows.net/help/Cloud Events/bc27/en-US/index.html

Permission Sets Summary

Permission Set	Purpose
CE Full Access	Full read/write access to all Cloud Events objects
CE Force Access	Override access for administrative operations
CE API Access	Access to Cloud Events API pages (for service connections)
CE API Integration	Access to integration-related API endpoints
CE API Storage	Access to storage API endpoints
CE API Translation	Access to translation API endpoints
CE Approval Admin	Manage approval workflows via API
CE Approval Log Read	Read-only access to approval logs

CE Edit Co. Memory	Edit company-level memory entries
CE Read All	Read-only access to all Cloud Events data
CE G/L Posting Gate	Allow G/L posting operations via API
CE Item Posting Gate	Allow Item posting operations via API
CE FA Posting Gate	Allow Fixed Asset posting operations via API
CE Job Posting Gate	Allow Job/Project posting operations via API
CE Res. Posting Gate	Allow Resource posting operations via API
CE Whse. Posting Gate	Allow Warehouse posting operations via API