

Monitoring and Log Retention

Use versioned execution evidence, attempt history, classification and separate retention controls to operate Connector 2.0.

Find an execution in the central monitor

1. Record the application client, UTC/time-zone context, URL path, user, object and the X-CGR-Message-ID response header when present.
2. Open /CGR/MONI with display permission (authorization object /CGR/MONI, activity 03). Use user, object and time filters to narrow the result.
3. Match the message ID and verify target system/client, actual execution user, method, HTTP status and execution state.
4. Open request/response details, Classification and Attempt history as needed. Keep HTTP status, technical execution and business result distinct.

The list is paged and reads header information without loading every payload. ALV sorting and filtering apply to the current page. Caller IP is the observed connection peer and may identify a proxy; older records can have no value. Add the column through Change Layout if a saved layout hides it.

Understand the stored evidence

Component	Meaning
Header / MESSAGE_ID	Attempt identity, technical lifecycle, object, user, timestamps, channel and chain links
C: normalized context	Full URI/query, typed execution inputs, routing, commit setting and observed execution identity
R: original request	Original HTTP body bytes when supplied, including an explicitly empty POST body; a bodyless GET has no invented R
O: output	Backend execution output or detailed failure diagnostic; not the Gateway outer OData entity

Supported generic SICF executions and Gateway function GET/POST and table/report GET use the versioned path. Metadata and requests rejected before this execution boundary keep lightweight legacy audit behavior. Authentication, license or object denial can occur before a versioned record exists; the absence of a record does not prove that no request reached SAP.

A start record is durably logged before dispatch. Logging uses a separate database transaction boundary, so committing a log does not deliberately commit the caller's business work. An uncertain log start prevents dispatch; uncertain finalization preserves the obtained business result and raises a warning. Do not repeat the business call to repair logging.

Payload viewing and size bounds

Structured JSON details expand objects, arrays, parameter values and fields. Stored strings, leading zeros, Unicode and unknown fields remain as stored. The viewer can show current Dictionary descriptions without retyping the recorded values.

Limit	Effect
16 MiB per persisted payload component	Hard capture/storage ceiling; this is not an API total-row limit
1 MiB payload-reader boundary	Oversized payloads are refused in full rather than displayed as a misleading shortened copy
500 complete records per table	A display limit; a notice identifies omitted records. Each selected row includes all nested data
JSON depth 64	Unsupported/deep or malformed JSON uses the appropriate bounded text/hex fallback or refusal

Unavailable, expired, binary or stored-truncated data is labeled. A storage truncation differs from a display row limit. Viewing a payload does not extend its retention. Legacy record editing rules do not make versioned context or output editable.

Technical execution and business classification

SUCCEEDED means the target returned and serialization completed. RUNNING or UNKNOWN means the execution needs investigation; it is not permission to retry. Optional classification uses the captured version of an explicit FUNCTION profile and complete output evidence.

Configured BAPI_RETURN evidence	Business result
Any valid TYPE A, E or X	BUSINESS_ERROR
Otherwise any W	WARNING
Otherwise explicit S	SUCCESS
Empty, blank or information-only RETURN	UNKNOWN; no conclusive success evidence
Missing/invalid/incomplete evidence	UNKNOWN with a reason

Classification accepts complete output up to 1 MiB, depth 64, 8192 JSON nodes and 1024 selected RETURN rows. Exceeding a bound yields UNKNOWN rather than classifying a prefix. TABLE/REPORT business outcome is NOT_APPLICABLE. HTTP status and business commit/rollback policy are unchanged. A recorded FUNCTION BUSINESS_ERROR displays a red icon even with HTTP 200; the status selection filter still filters HTTP status.

Rerun a function deliberately

1. Select exactly one eligible completed FUNCTION attempt and choose Rerun function. Review function, source attempt, chain, target, current operator, commit setting and draft revision.
2. Use Input (F6) to compare saved and draft JSON. Apply and review validates corrections without calling the function. The correction editor refuses input above 262144 characters.
3. Review the accepted revision, then choose Execute once (F8) and confirm that business changes can

repeat. Enter and editor actions do not execute.

4. Inspect the new immutable child attempt and output. Original evidence remains unchanged; Attempt history relates the attempts.

The current operator needs display and execute activity 16, plus correction activity 02 for revised input. Current license, object grants, interface, evidence and target are rechecked. Expired/corrupt/stale input or a RUNNING/UNKNOWN chain blocks execution. TABLE/REPORT, bulk and automatic reruns are unavailable. Historical user permissions are never reused.

Agree retention and stored deadlines

Component	Default retention
Completed headers	90 days
Successful context / original input	7 days
Failed context / original input	30 days
Successful response	1 day
Failed response	30 days

Settings are client-specific /CGR/CONN_CFG, CONFIG_ID GLOBAL. Completed records retain their saved expiry dates. An expiry date is the last retained UTC day; deletion eligibility starts only when it is before today. RUNNING/UNKNOWN chains remain protected unless the separate scheduled-purge unresolved policy is explicitly selected.

Choose the correct cleanup tool

Tool	Purpose and controls
/CGR/RETENTION_PREVIEW	Read-only metadata candidates with bounded pages; it does not reserve rows or certify later deletion
/CGR/RET_ADMIN and /CGR/RET_CLEANUP	Client opt-in plus foreground one-page manual cleanup. Execute defaults off; each page requires confirmation
/CGR/RET_PURGE	Separate full-scan versioned-chain purge, preview by default; optional SM36 background variant explicitly enables Execute
/CGR/DELETE_LOG	Legacy-only log cleanup; not the versioned retention-policy tool

Scheduled purge checks the whole chain and commits each eligible chain separately. Include unresolved RUNNING/UNKNOWN is off by default; if selected, a finite fallback uses the current header policy and latest activity. Review that policy before enabling it. Locked, malformed, oversized or ambiguous chains remain for review; an end-of-scan result does not certify that all expired data was removed.

Preview requires monitor activity 03; Execute requires administer activity 70 as well. Importing code does not create a job. Review the preview, save the explicit selection variant and schedule /CGR/RET_PURGE in SM36 only after policy approval. Check actual counts and review samples in SM37. Earlier chains can already be committed if a later chain fails.

Notification evidence is separate

Use `/CGR/NTF_QUEUE` for captured intents and `/CGR/NTF_DLV_VIEW` for delivery state. Notification retention is independent of execution-log retention. SUBMITTED means local SAPconnect acceptance, not SMTP or mailbox delivery; inspect SOST and the Basis mail route. Uncertain submissions are not automatically retried. Deleting an execution log does not delete the captured mail intent.

CGR SOFTWARE

cgrsoft.com · sales@cgrsoft.com

Use these release 2.0 instructions with the supplied release delivery notes. Release 1.3 has a separate documentation and PDF set.