

Error Handling and Troubleshooting

Trace the correct layer, preserve evidence and resolve uncertain outcomes before repeating a business operation.

Start with the exact response and context

1. Record system/client, time and time zone, HTTP method, path, exact response status/body and X-CGR-Message-ID when present.
2. Inspect X-CGR-Log-Warning and correlate the attempt in /CGR/MONI. Verify the actual user, backend, object and technical execution state.
3. Inspect the returned function messages or Classification separately from HTTP status. HTTP 200 does not prove that the requested business operation succeeded.
4. Correct the identified configuration or input. Repeat only a minimal read-only test unless the business effect of a previous updating call has been reconciled.

Use HTTP status as a clue, not the whole diagnosis

Status or symptom	What to check
400 / invalid request	Required id/variant, duplicate controls or JSON keys, active parameter types, field names, paging and supported filter grammar
401 at HTTP logon	Credentials, chosen application client, SAP user status, Basic service logon or bearer token validity
Authorization / scope rejection	Actual backend identity, OAuth scopes and SAP roles, then exact /CGR/API_AUTH matching in the API client
404	Read the JSON/body: a missing service, object denial, license denial or unavailable object can appear as 404; do not infer only a URL problem
405	Unsupported method, including non-GET metadata or Gateway TABLE/REPORT POST
500 / runtime failure	Execution diagnostic, /CGR/MONI, backend runtime/lock/database evidence and ST22 as appropriate
Timeout / lost response	Investigate business state and RUNNING/UNKNOWN evidence before retrying

Framework responses can occur before Connector execution. A Gateway function POST can return HTTP 201 while its technical execution log uses 200. Preserve the actual status and body rather than forcing every issue into a single status-to-cause mapping.

License, access and client problems

- License rejected: verify product SAP_CONNECTOR, full licensing system identifier, validity and the complete original CGR-issued text file. Install it in the application client used by the request.
- Request denied: check the actual logged-on user and client. /CGR/API_AUTH requires exact user

matching and an exact object or supported complete wildcard.

- Access remains after deleting a row: another wildcard grant can still allow it. Partial patterns such as BAPI_* are not supported, and there are no deny overrides.
- Changed user or roles seem ignored: retest with a fresh authenticated HTTP session and confirm the user recorded by SAP.
- Metadata denied: object metadata uses authorization too; a valid HTTP login does not grant object definitions.

Service and backend routing

For SICF, verify default_host > cgr > api is active and uses /CGR/CL_API_HANDLER. /func, /table and /report are routed resource paths below it. Confirm HTTPS, port and application client. For OData, verify /CGR/CONN_SRV registration, ICF activation, system alias, token scope and Gateway error evidence.

For a SICF backend, verify BACKEND in /CGR/RFC_DEST or the selected sap-system alias, RFC connectivity and the observed execution identity. A destination mapping to a different user is rejected by the current policy. OData runs in the backend selected by Gateway; do not add separate Connector RFC rerouting as a workaround.

Typed input and query failures

- Request metadata with cgr-metadata=true on GET to confirm the authorized active definition without executing it.
- Use the correct POST shape: SICF NAME/VALUE array, Gateway parameter-name object. Send valid UTF-8; explicitly supported UTF-16 must be correctly identified.
- Remove unknown, duplicate and output-only parameters. Use objects for structures and arrays for table values; preserve identifier leading zeros as strings.
- For TABLE, validate every projected/ordered field and use the supported filter grammar. Raw SQL, subqueries and field-to-field comparisons are not accepted.
- For REPORT, verify the active executable program, approved ALV output and existing target-client variant. Gateway requires a valid variant.
- Use positive top and nonnegative skip. top=0 requests unlimited results; it is not a protective cap.

Business result differs from technical success

Inspect the returned RETURN structures. Optional BAPI_RETURN classification uses an exact function/output profile captured when the attempt starts. A/E/X yields BUSINESS_ERROR, W yields WARNING and explicit S can yield SUCCESS; empty or information-only output is inconclusive. Missing configuration, malformed/incomplete output or evaluation bounds produce UNKNOWN.

Changing a profile does not reclassify history. TABLE and REPORT business outcomes are NOT_APPLICABLE. A red business-error icon with HTTP 200 can be correct: the function completed but reported a business rejection. Classification never requests rollback or makes a call safe to repeat.

Missing, expired or incomplete monitor evidence

No entry: check the client, time/user filters, monitor authority and whether the request reached the versioned execution boundary. Authentication, license, metadata and early validation paths may use different audit behavior. No entry is not proof that no request arrived.

Payload unavailable: inspect the stated reason. Successful response retention defaults to one day, earlier than input or header retention. The viewer has a 1 MiB boundary and 500 complete records per table; persisted components have a separate 16 MiB ceiling. A display notice, stored truncation and expired storage are different conditions.

Log warning after execution: preserve the obtained business response and the warning. Finalization uncertainty does not authorize a second business dispatch. RUNNING/UNKNOWN chains require investigation rather than automatic retry or cleanup.

Manual rerun is blocked

Rerun requires one eligible completed FUNCTION attempt, retained valid input, a consistent completed chain, the current operator's display/execute rights and current license/object grants. Correction needs activity 02 and an exact draft revision. The editor limit is 262144 characters. Stale, corrupt, expired or unresolved chains fail closed. TABLE/REPORT and bulk reruns are unavailable.

Review the accepted draft and use Execute once deliberately. A previous function may have committed internally or caused an external effect; replay can duplicate it even if a previous HTTP response was lost.

Notification or digest did not arrive

1. Check that the new attempt captured a matching enabled rule and, for business outcomes, the correct classification profile. A later configuration change does not rewrite an old intent.
2. Inspect /CGR/NTF_QUEUE and /CGR/NTF_DLV_VIEW. Capture, preparation and sending have separate controls and revisions; installation enables none of them.
3. Review prospective UTC admission/sending cutoffs. Older backlog is deliberately excluded. Check digest interval and the scheduled job in SM37.
4. If SUBMITTED, inspect SOST and the Basis SAPconnect/SMTP route. Local submission is not mailbox delivery.
5. If submission is uncertain, preserve the ledger and native receipt evidence. Do not automatically reset or resend it.

Escalate with useful evidence

Send CGR the installed release, system/client, transaction or URL path, timestamp/time zone, message/attempt ID and exact redacted diagnostic. Include the smallest reproducible request and expected result where safe. Exclude passwords, authorization headers, cookies, tokens and license keys. For ambiguous updates, include the business document/reconciliation evidence before considering another execution.

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.