START HERE / STEP 01

Quick Setup Guide

Install CGR Connector, configure access, and verify your first API request.

PRODUCT 2.0 / SOFTWARE COMPONENT CGR 200

For your SAP Basis administrator and Connector setup administrator. This guide covers the generic **FUNCTION, TABLE and REPORT** APIs in a same-system installation.

Before you begin

- Obtain the CGR installation or upgrade package, its delivery notes and a customer license text file for `SAP_CONNECTOR` and the target SAP system.
- Basis confirms platform prerequisites, SPAM/SAINT level, installation permissions and the change window. Agree the application client, HTTPS address and named integration user.
- Use **client 000 for the SAINT installation**. Perform license and Connector configuration in the **application client** that will receive API calls.

Install the add-on with SAINT

1. Log on to client **000** as the Basis installation administrator and run `SAINT`.
2. Load the CGR-provided archive using **Installation Package > Load Packages > From Front End** (wording varies by SAP release), or have Basis load it from the application server.
3. Start installation and select the CGR package. For an existing installation, use the matching upgrade package specified in the delivery notes.
4. Review the calculated queue and prerequisite checks, choose the installation start options, and complete the import. Resolve any reported errors before continuing.
5. Confirm completion and check that installed component **CGR** shows release **200**, or the exact release stated in the supplied delivery notes.

> **Use the supplied package type.** These SAINT steps apply to a SAP add-on installation or upgrade package. If CGR supplied transport files instead, Basis imports those through the transport procedure in the delivery notes. A source ZIP is not a SAINT package.

SAP reference: SAP Add-On Installation Tool guide, installing/upgrading add-ons and loading installation packages. The supplied CGR delivery notes define the supported platform and package prerequisites.

STEP 02 / APPLICATION CLIENT

Install the license

Transaction /CGR/LICENSE | Command field: /n/CGR/LICENSE

1. Open `/CGR/LICENSE` in SAP GUI in the intended application client.
2. Enter **SAP_CONNECTOR** in Product. Select the CGR license **.txt** file using the License File field's browse/value help (**F4**).
3. Choose **Execute (F8)**.

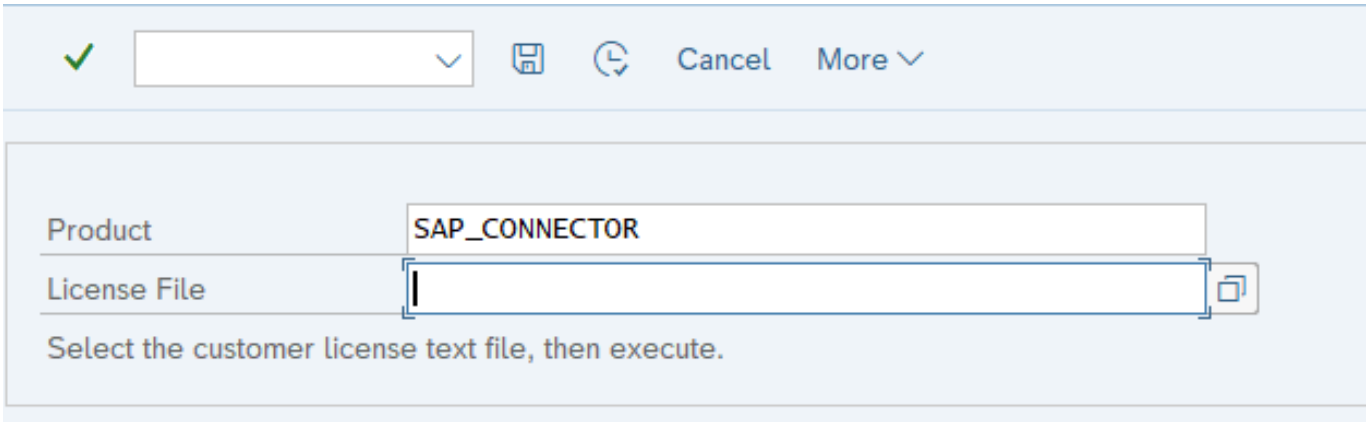

Figure 1. License installation screen. Select the license text file supplied by CGR.

Confirm installation

The result should say **License installed successfully** or **License replaced successfully**. Verify the displayed product, system identifier and validity date.

Field	Value
Product	`SAP_CONNECTOR`
License File	The original CGR-issued text file containing the complete `CGR1:` license key.

Install the license in every application client that will use the Connector. The license record is stored per client. Uploading a replacement updates the existing license for that product.

> **For a new license:** give CGR the target licensing system identifier. Basis can obtain it from `SLIC_GET_SYSTEM_ID`, output `SYSTEMID`. Use the complete value; the ordinary three-character SAP SID may be different.

STEP 03 / APPLICATION CLIENT

Configure the main object types

Transaction /CGR/BO | Command field: /n/CGR/BO

Open `/CGR/BO`, choose **New Entries** if required, and create or verify the three rows below. Save the configuration and record it in a customizing request if your landscape prompts for one.

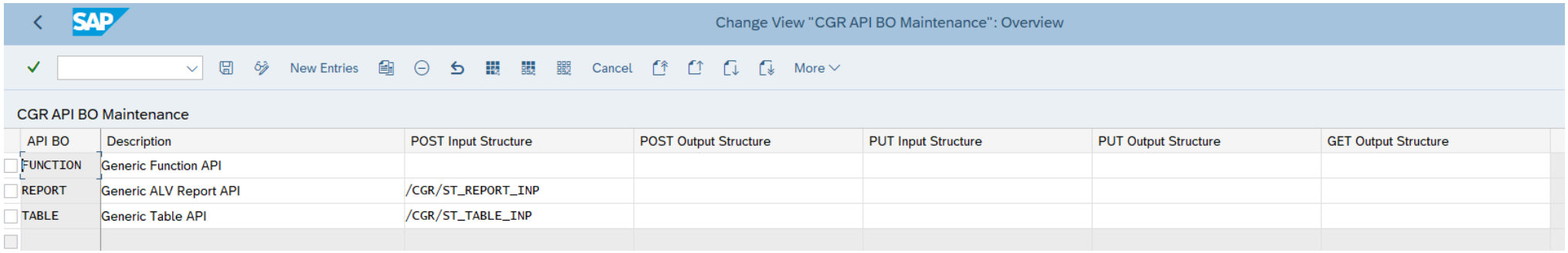

Figure 2. Main object type configuration. Blank structure cells are intentional.

API BO	Description	POST Input Structure
`FUNCTION`	Generic Function API	Leave blank
`REPORT`	Generic ALV Report API	`/CGR/ST_REPORT_INP`
`TABLE`	Generic Table API	`/CGR/ST_TABLE_INP`

Leave POST Output, PUT Input, PUT Output and GET Output Structure blank for these rows, as shown. This catalog describes the object types and interfaces. **User access is granted separately in /CGR/API_AUTH.** Removing a catalog row is not a method of revoking API access.

STEP 04 / APPLICATION CLIENT

Grant user access

Transaction /CGR/API_AUTH | Command field: /n/CGR/API_AUTH

Use a named SAP integration user created by your Basis/security team. In `/CGR/API_AUTH`, choose **New Entries**, enter the permitted objects and save. Maintain the entries in the **same client** used by the API. Use uppercase SAP user and object names.

Specific object examples

CGR_API is an example user name; replace it with your actual integration user. Add only the rows your integration requires.

User	API BO	Object Name
CGR_API	TABLE	MARA
CGR_API	TABLE	KNA1
CGR_API	FUNCTION	BAPI_PO_GETDETAIL1

For a report, use `REPORT` and its exact ABAP program name in Object Name.

Wildcard examples

User	API BO	Object Name	Grants access to
CGR_API	TABLE	*	All tables
CGR_API	FUNCTION	*	All functions
CGR_ADMIN	*	*	All object types and objects

These are alternatives illustrating broad grants, not a recommended default configuration.

> **Matching rules:** an exact user is required. Use a complete `*` for Object Name, or `*` in both API BO and Object Name. Patterns such as `BAPI_*`, `MA*`, a wildcard User, or `*` API BO with a specific object name are not supported.

No matching grant means access is denied. There are no deny overrides. If a wildcard grant remains, deleting a narrower row does not revoke that access.

Basis must also assign the appropriate SAP roles for the chosen interfaces and business operations. This allowlist does not grant configuration-maintenance or SAP transaction permissions.

STEP 05 / CONNECTIVITY AND VERIFICATION

Run the first API request

Basis configures HTTPS and the service; the integration user performs the test.

Activate and check the HTTP service

1. In `SICF`, locate **default_host > cgr > api**. Activate the service and any required parent nodes. Confirm handler `/CGR/CL_API_HANDLER`.
2. Confirm the HTTPS host, port, certificate, application client and approved authentication method. Ensure calls run as the intended integration user.
3. For a basic same-system installation, keep the default local routing. Separate backend systems require additional Basis configuration and a review of the backend execution user.

The paths `/func`, `/table` and `/report` are routed beneath `/cgr/api`; they are not separate SICF services.

Read one material number

First grant the integration user `TABLE / MARA` as shown on page 4. In your HTTP client, select the approved authentication method and enter that user's credentials. Send the following **GET** request over HTTPS:

```
GET /cgr/api/table?id=MARA&top=1&select=MATNR&sap-client=100 HTTP/1.1
Host: sap.example.com
Accept: application/json
```

Replace sap.example.com, its HTTPS port if required, and client 100 with your own values. The example host and client are placeholders.

Verify the result and monitor entry

- Expect **HTTP 200** with JSON. A populated table should return at most one material number; an empty `[]` can be valid if there is no data.
- Copy the response header `X-CGR-Message-ID`. Open `/n/CGR/MONI`, find the request by time, user and object, then match the message ID.
- Verify the recorded user is the intended integration user. Check that reading a known existing table with no exact or wildcard grant is denied.

The monitoring user needs monitor display permission: authorization object `/CGR/MONI`, activity `03`. Use a fresh authenticated HTTP session when testing another user or changed permissions.

STEP 06 / HANDOVER

Finish setup and troubleshoot

Complete the basic connection first, then enable optional operational features.

Quick troubleshooting

Symptom	Check
License rejected	Correct product, target licensing system identifier and validity date; complete original text file; license installed in the API's client.
HTTP 401	HTTP credentials, selected client, SAP user status and service authentication.
HTTP 404	Service path and JSON response text. Connector license or object authorization failures can also return 404; do not assume the service is missing.
Access still works after a row is removed	A remaining wildcard grant, the client used, and the actual logged-on user. Retest with a fresh HTTP session.
No monitor entry	Confirm the request reached SAP and passed the checks before execution. Also review the monitor's client, user/time filters and display permissions.
Function returns a business error	Inspect its returned messages. HTTP 200 confirms an HTTP response, but does not by itself establish a successful business result.

Optional follow-up configuration

Notifications and digests: configure classification, rules and recipients, then enable preparation, sending and scheduling. The interval is configurable (default: 15 minutes). Basis configures SAPconnect/SMTP; basic setup alone does not enable email.

Retention: agree payload visibility and retention periods. Review a purge preview before scheduling cleanup.

OData or a remote backend: add Gateway or RFC configuration when required. This guide's first test uses the local SICF API.

Ready for handover

Record the installed release, application client, HTTPS endpoint, integration user, approved object grants, license expiry and first successful message ID. Verify both an allowed and a denied request.

For CGR support, include the system/client, transaction or URL path, timestamp, message ID and exact error. Exclude passwords, authorization headers, session cookies and license keys.