

SQL Change Guard Installation and First Use Guide

This guide describes, in order, every step from installing SQL Change Guard in an empty environment to executing the first change request and verifying its evidence file. The details of each step and the integrations (corporate directory, corporate sign-in, ITSM, SIEM) are in the Installation and Operations Guide; the preparation list is in the Installation Prerequisites.

1. Prerequisites and supported environments

Component	Requirement
Application server	Windows Server (4 cores, 8 GB memory recommended), IIS, ASP.NET Core Hosting Bundle (.NET 10), IIS URL Rewrite module. API, Worker and web interface can run on the same server
Application database	SQL Server 2022 or later (Express included). Older versions are not supported because audit records are kept in SQL Server's append-only ledger table
Governed databases	SQL Server, PostgreSQL, Oracle (the types enabled in the licence). The minimum supported target versions are confirmed with the product team before installation
Browser	Current Edge, Chrome or Firefox
Network	443 from the browser to the web interface and the API; 1433 from the API and the Worker to the application database; the database ports from the API and the Worker to the governed databases
Ledger digest store	A folder where the API application pool identity can create and read new files but cannot change or delete them (WORM share, object-locked storage). The API writes the digest in a round started by the Worker. The database administrator must not be able to write to it

Ready before installation: servers, DNS names and HTTPS certificates, database accounts, the licence file, the evidence signing key (RSA, Installation and Operations Guide section 2.3).

2. SQL Server requirements and permissions

2.1 Application database

There are two installation paths. The choice is made with the Database:AutoMigrate setting of the API.

Path	Who installs	Account permission
A (default, AutoMigrate = true)	The product, at first start	db_owner on the database during installation. If the database does not exist, the product creates it; this also needs dbcreator. After installation the permission can be reduced to the path B permissions
B (AutoMigrate = false)	The DBA, with scripts	The least privilege below

Permissions for the application account on path B (sysadmin and db_owner are not granted):

```
USE [SQLChangeGuard];  
  
CREATE USER [application_account] FOR LOGIN [application_account];  
  
ALTER ROLE db_datareader ADD MEMBER [application_account];  
ALTER ROLE db_datawriter ADD MEMBER [application_account];  
  
GRANT EXECUTE ON SCHEMA::dbo TO [application_account];  
  
ALTER ROLE evidence_api_writer ADD MEMBER [application_account];  
ALTER ROLE evidence_worker_writer ADD MEMBER [application_account];  
  
GRANT VIEW LEDGER CONTENT TO [application_account];  
  
GRANT VIEW DEFINITION ON SCHEMA::evidence TO [application_account];  
  
DENY UPDATE, DELETE ON dbo.TableWriteLogs TO [application_account];
```

The API and the Worker can use the same account; this is the default. If wanted, the Worker gets a separate account: the API account is added only to evidence_api_writer, the Worker account only to evidence_worker_writer, and the other permissions are given to both. The database then refuses user decisions such as approval, rejection and reconciliation decisions, and sign-in events, written by the Worker account.

2.2 Governed databases

On every target server an account for the product is created. It must be able to perform the operations in approved scripts; the product also reads object definitions to generate rollback scripts and compare objects (VIEW DEFINITION on SQL Server). The account is entered on the product's Servers screen and the password is stored encrypted.

3. Creating the database and installing the schema

3.1 Path A: the product installs

1. Create the empty database (or give the account dbcreator and the product creates it).
2. When the API starts for the first time (section 4), the log shows [Schema] Applying and [Schema] Completed lines for each script. The product also turns on the snapshot isolation needed for ledger verification: ALLOW_SNAPSHOT_ISOLATION was turned on.

3.2 Path B: the DBA installs

1. The scripts are in the API folder under DbScripts\sqlserver: 001_Tables.sql, 002_SeedData.sql, 003_StoredProcedures.sql, 004_TableWriteLogs.sql. They are applied in order:

```
sqlcmd -S SQLSERVER -d SQLChangeGuard -I -f 65001 -b -i 001_Tables.sql
```

```
sqlcmd -S SQLSERVER -d SQLChangeGuard -I -f 65001 -b -i 002_SeedData.sql
```

```
sqlcmd -S SQLSERVER -d SQLChangeGuard -I -f 65001 -b -i 003_StoredProcedures.sql
```

```
sqlcmd -S SQLSERVER -d SQLChangeGuard -I -f 65001 -b -i 004_TableWriteLogs.sql
```

2. Run DbScripts\DbSchemaVersions.ps1 from the API folder; review and apply the SchemaVersions_Register.sql it produces.

3. Once, when there is no other connection to the database:

```
ALTER DATABASE [SQLChangeGuard] SET READ_COMMITTED_SNAPSHOT ON WITH ROLLBACK IMMEDIATE;
```

```
ALTER DATABASE [SQLChangeGuard] SET ALLOW_SNAPSHOT_ISOLATION ON;
```

4. Grant the permissions in section 2.1.

Check that the evidence table is a ledger table:

```
SELECT SCHEMA_NAME(schema_id) AS schema_name, name, ledger_type_desc FROM sys.tables WHERE ledger_type <> 0;
```

Expected: evidence, AuditLogs, APPEND_ONLY_LEDGER_TABLE.

4. Installing the API

1. Copy the API folder to the server (for example D:\SQLChangeGuard\api).
2. In IIS create an application pool (.NET CLR version: No Managed Code) and a site whose root is the API folder. Add the HTTPS binding and certificate.
3. Give the application pool identity write permission to the logs folder, read permission to the query result file folder (for downloads), permission to create new files in and read the ledger digest store, and read permission to the private key of the evidence signing certificate.
4. In appsettings.json set at least: AllowedOrigins (the web interface address), Database:AutoMigrate, Display:TimeZone, QueryResult:OutputPath, Evidence:Signing:Enabled = true, Evidence:Signing:InstallationId, the public key or certificate thumbprint of the signing key, Evidence:Ledger:DigestOutputPath. Signing is off in the code default; while it is off, the evidence file is produced unsigned and requests are not executed on servers defined as Production.
5. Define the secrets as environment variables (setx /M in an administrator command prompt).
Required:

Environment variable	Purpose
ConnectionStrings__Default	Application database connection string
Jwt__SigningKey	Signature of session tokens (at least 32 characters)
Security__ServerPasswordKey	Encryption of server passwords (base64, 32 bytes)
Security__QrZipPasswordKey	Encryption of query result package passwords (base64, 32 bytes)
Workers__ApiKey	The Worker identifies itself to the API (same value as WorkerApi__Key on the Worker)

6. Start the application pool. Log file: logs\scg-YYYYMMDD.log. There must be no error lines.

5. Installing the Worker (Windows service)

1. Copy the Worker folder to the server (for example D:\SQLChangeGuard\worker).
2. In the Worker's appsettings.json set WorkerApi:BaseUrl to the API address. Define the WorkerApi__Key environment variable with the same value as Workers__ApiKey on the API.
3. If the Worker connects with a separate database account, define the ConnectionStrings__Worker environment variable on the Worker server. If it is not defined, the Worker uses the same account as the API and receives the connection information from the API at start-up.

4. Register and start the service (administrator command prompt):

```
sc.exe create "ChangeGuardWorker" binPath= "D:\SQLChangeGuard\worker\SCG.Workers.exe" start=
auto DisplayName= "SQL Change Guard Worker"
```

```
sc.exe start "ChangeGuardWorker"
```

5. Give the service account write permission to the logs folder and the query result folder. The Worker account needs no permission on the ledger digest store; the API writes the digest.
6. In the Worker log, see the lines Worker configuration loaded from API and Worker database connection source. After one digest interval (60 minutes by default) the line [Ledger] Digest published appears in the API log and a ledger-digest-*.json file (with a .sig next to it when signing is on) is created in the store.

6. Web interface (client)

1. Copy the web interface folder to the server.
2. In IIS create a separate site whose root is this folder; add the HTTPS binding.
3. The IIS URL Rewrite module must be installed (the web.config in the folder redirects page refreshes to index.html).
4. Check that this address is written in the API's AllowedOrigins setting.

7. Licence and first sign-in

1. Open the web interface and sign in with the administrator account created by the installation: user name admin, password admin123. The product forces a password change at first sign-in; set the new password.
2. On Settings > License upload the content of the licence file. The product verifies the signature.
3. On Settings > Users add users (or set up the corporate directory or corporate sign-in). For the first trial at least three people are recommended: a requester (Requester or RequestManager), an approver (ChangeManager) and an executor (DbAdmin). Keep at least one local SystemManager account as the emergency account.
4. On Settings > Servers define the first target server: name, address, database type, default database, account and password, environment (Production or Test). Test the connection.
5. On Settings > Approval Policies define the policy for this server and request type (how many steps, which role). If the target server is not covered by any approval policy, the request cannot be saved (a rollback request is exempt from this rule).
6. Open Settings > Diagnostics > Governance Readiness. The rows "Evidence table is a ledger" and "Evidence write boundary" must be green. Decide on the rows marked for attention.

8. First successful request

1. Sign in as the requester and click New Request. Type: Change. Server: the server defined in section 7. Enter a ticket number if your organisation requires it. Example script:

```
USE [TestDatabase]
```

```
GO
```

```
CREATE TABLE dbo.SqlChangeGuardFirstTrial (Id INT NOT NULL PRIMARY KEY, Note NVARCHAR(100) NULL)
```

```
GO
```

2. Save. The product parses the script and shows the objects and the risk band. If a blocking standard is violated, the request is not saved and the reason is shown.

3. Sign in as the approver, open the request in Change Requests and click Approve. If the policy has more than one step, each step is approved by an authorized person.

4. Sign in as the executor and click Execute. The request is queued and the Worker executes it within seconds. The status becomes Executed; the execution log is shown in the request detail.

5. Check that the table exists in the target database.

If the request moves to Reconciliation Required: the execution reached the target but its outcome is not known for certain (for example the Worker was interrupted while running, the script ended the product's transaction itself, or on an Oracle target an error came after a data change had become permanent). The request is not executed again. A person with execute permission checks the target and, with the Record the outcome button in the request detail, states "the change is on the target" or "the change is not on the target", and writes in the note what was checked. If only part of the change is on the target, "on the target" is chosen, the note says what was applied and what was not, and the remaining work is sent in a new request opened with the "New request for the remaining work (copy)" button. The decision and the note appear in the Reconciliation Record box in the request detail and in the evidence file.

9. Generating and verifying the evidence file

1. Sign in with a user who has the Can Export Evidence File permission (for example Auditor), open the request detail and click Evidence File. EvidenceFile_CR-....zip is downloaded.

2. The ZIP contains Dossier.pdf, Dossier.json, Manifest.json, Manifest.sig, PublicKey.pem and Verify.ps1. In the folder run:

```
powershell -NoProfile -ExecutionPolicy Bypass -File Verify.ps1
```

The signature and file hashes must match.

3. The verdict on the first page of Dossier.pdf must not be red. In section "12. Field Verification" every field must be PASS; the ledger result becomes VERIFIED once a digest covering the last record has been written (before that it shows NOT_COVERED; wait for the next digest interval). In section "13. Event Time and Ledger Commit Time" every event must be TIME_OK.

If these steps are complete, the installation works: the request was recorded, the approval and execution evidence was written to the ledger, the ledger was verified against the external digest and the evidence file was produced with a signature.

10. Common problems

Symptom	Cause	Solution
The API does not start, connection error in the log	ConnectionStrings__Default is wrong or the account lacks permission	Check the connection string and the permissions in section 2.1
The API does not start, "checksum" or script error	An applied installation script was changed	Do not change the delivered scripts; contact support
The Worker does not start, "Failed to load worker configuration"	The API address is wrong or WorkerApi__Key differs from Workers__ApiKey	Check both values and the address
Saving a request is refused with "SAVE_BLOCKED_BY_STANDARD"	The script violates a blocking SQL standard	Correct the script according to the rule in the message
Ledger result UNAVAILABLE in the evidence file	ALLOW_SNAPSHOT_ISOLATION is off or the account lacks VIEW LEDGER CONTENT	Sections 3.2 and 2.1
NOT_COVERED in the evidence file	No ledger digest covers the last record yet	Wait one digest interval; see the [Ledger] Digest published line in the API log
VERIFIED_INTERNAL in the evidence file	The API cannot read the ledger digest store (Evidence:Ledger:DigestOutputPath); verification used only the digest inside the ledger	Make the path an absolute path (UNC) visible from the API server too, and give the API application pool identity read permission on the folder
State-changing operations are refused, "evidence write boundary" warning	Something was changed in the evidence schema (trigger, procedure)	Undo the change; the Governance Readiness screen shows the finding