



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Interaction Concentrator Deployment Guide

Components and Their Functions

5/9/2025

Components and Their Functions

Interaction Concentrator consists of a server, known as the ICON Server or simply ICON, and the Interaction Database (IDB).

ICON

ICON Functions

This section provides a high-level overview of ICON data-handling.

Data preprocessing

ICON preprocesses events received from Configuration Server, T-Server, Interaction Server, and Outbound Contact Server (OCS), according to the role configured for the ICON instance. Preprocessing occurs in the in-memory queue (accumulator).

You can configure the size of the in-memory queue or the interval at which data is written from it to the persistent queue. You can also configure the total number of keep-in-memory interactions that can reside concurrently in an interaction queue or interaction workbin. (This functionality requires Interaction Server release 7.6.1 or higher.)

For more information about the relevant configurations, refer to the [in-memory queue options](#).

Data preparation and writing

Once data is processed in the in-memory queue, ICON performs the following functions:

- Writes the data from the in-memory queue to the persistent queue and persistent caches.
- Manages the data in the persistent queue and persistent caches.
- Writes data from the persistent queue into IDB.
- Writes data from the persistent cache for configuration data (`cfg-sync.db`) into IDB.

For detailed information about the configuration options that determine ICON functionality and performance, see [Configuration Options](#).

The persistent queue (PQ) file

The persistent queue is a file that ICON creates and uses to store data before writing it to IDB. The persistent queue also stores information about requests to write data to IDB. Data in the persistent queue survives a shutdown and restart of ICON. The size of the persistent queue is not formally limited by ICON, but the operating system may impose some limitations.

Important

To reduce the possibility that Interaction Concentrator might lose connection with the PQ file you are required to locate it on a local drive rather than a network or removable drive.

Each ICON instance creates its own persistent queue file (default name `icon.pq`), which stores data for all the roles that are configured for that ICON. For more information, see [ICON Roles](#).

Persistent Queue Configuration Options

ICON configuration options enable you to specify:

- The file name of the persistent queue.
- The frequency (in terms of number of committed transactions) with which ICON clears data out of the persistent queue.
- Thresholds for environment failure alarms.
- The alarm thresholds can be used to monitor ICON performance.
- Persistent queue behavior at startup.

For more information, see [persistent queue configuration options](#).

Persistent caches for Configuration and Login Session data

In addition to the regular persistent queue, the ICON instance that performs the `cfg` role creates and maintains a persistent cache for configuration data. The name of the persistent cache for configuration data is `cfg-sync.db` and it cannot be changed.

The `cfg-sync.db` persistent cache plays an important role in maintaining IDB synchronization with the Configuration Database. ICON keeps a timestamp in the persistent cache for configuration data changes and, on startup, requests from Configuration Server all configuration changes that occurred after that timestamp.

- For more information about how the persistent queue and the `cfg-sync.db` persistent cache work to maintain up-to-date configuration information, see the section about populating configuration data in the [Interaction Concentrator User's Guide](#).

- For recommendations about best practices regarding synchronization, see the chapter about resynchronization in the [Interaction Concentrator User's Guide](#).

Persistent Cache for Agent Login Session Data

In addition to the regular persistent queue, the ICON instances that perform the gcc, gls, and gud roles create and maintain a persistent cache for agent login session data. In High Availability (HA) deployments, ICON uses this cache to prevent duplicate storage of agent login sessions in IDB and to prevent stuck login sessions. For more information, see the chapter about agent states and login sessions in the [Interaction Concentrator User's Guide](#).

A configuration option, `agent-pstorage-name` (see `agent-pstorage-name` Option), enables you to specify the name of this persistent cache. The default file name is `apstorage.db`.

IDB

Interaction Database (IDB)

The Interaction Database (IDB) stores data about contact center interactions and resources at a granular level of detail. IDB is a database optimized for storage (in other words, primarily for inserting data). Interaction Concentrator itself does not provide a reporting facility. You can use IDB as a consistent and reliable data source for downstream reporting applications.

- For a high-level description of the IDB architecture, see the chapter about IDB schema in the [Interaction Concentrator 8.1 User's Guide](#).
- For a complete table structure and descriptions of all IDB tables and fields, see the Interaction Concentrator 8.1 Physical Data Model document for your particular relational database management system (RDBMS).

Stored Procedures

Interaction Concentrator uses a number of stored procedures. Most of these are entirely internal to Interaction Concentrator functioning. Therefore detailed information about them is not relevant for end users.

Most stored procedure names start with a schema-specific prefix, so that they constitute a schema-specific package. Each ICON 8.1.x version works only with the stored procedures package for the associated schema. This streamlines future migration by reducing the number and combinations of scripts that must be executed to upgrade the required stored procedures. A wrapper script links the stored procedures that are exposed for end-user use to the equivalent stored procedures in each

schema-specific set.

- For more information about these stored procedures, refer to the chapter about special stored procedures in the [Interaction Concentrator 8.1 User's Guide](#).

The following stored procedures are exposed for end-user use and require user input or action:

[+] Merge Procedures

gsysIRMerge and **gsysIRMerge2**—The merge procedure that finalizes data processing of closed single-site and multi-site interactions. **gsysIRMergeReset**—The procedure that resets the merge procedure to recover from a failed state.

[+] Purge Procedures

There are a number of alternative procedures for purging IDB.

Important

These procedures do not purge all tables. For a list of tables that are purged, see Table 26 in Chapter 16, “Using Special Stored Procedures,” of the [Interaction Concentrator 8.1 User's Guide](#).

- **gsysPurgeIR**, **gsysPurgeUDH**, **gsysPurgeLS**, and **gsysPurgeOS**—The procedures that safely purge voice interactions, user data history, agent login session, and Outbound Contact data, respectively, from IDB.
- **gsysPurge80/gsysPurge81**—Alternative procedures that safely purge voice and multimedia interactions; attached data; agent login session data; and Outbound Contact data from IDB. The version of this purge procedure corresponds with the Interaction Concentrator release you are using.
- **purgePartitions811**—In a partitioned IDB, this purge procedure clears the database by truncating all except the specified number of days/partitions in all affected tables. It also retains a default additional partition for “tomorrow.”

Important

- The **purgePartitions811** procedure is supported only for IDBs running Oracle 11 or higher. Genesys recommends that you do not use the **purgePartitions811 *purge** procedure on IDBs containing long-living data, such as multimedia IDBs.

[+] Time-Setting

gsysInitTimeCode—The stored procedure that populates the **G_TIMECODE** table to enable time-interval reporting.

[+] Custom Dispatchers

gudCustDISP1 and gudCustDISP2—The stored procedures that customize attached data processing.