Interaction Concentrator Deployment Guide

Interaction Concentrator 8.1.4

1/3/2022

Table of Contents

ICON Deployment Guide

What is Interaction Concentrator?
Interaction Concentrator collects and stores detailed data from various sources (such as T-Server, Interaction Server, and Configuration Server) in a contact center that is empowered with Genesys software. Downstream reporting systems can access Interaction Concentrator data in near real time.

This guide includes the following information:

Features and Functionality

Overview information about architecture, components, and other concepts for ICON.

Features and Functionality—A survey of what Interaction Concentrator does.

Components and Their Functions—Introduces the ICON Server and IDB.

Planning Your Deployment

Prerequisites and Considerations lists compatible Genesys components, prerequisites, and deployment considerations.

This section also includes information about supported deployment scenarios and the `role` option.

Configuring and Installing

Deploying Interaction Concentrator includes deployment instructions, additional configuration needed for various types of deployments (such as multimedia, Outbound, and LRM), and references to the relevant configuration options.

Additional Information

Troubleshooting

Starting and Stopping

Attached Data Samples—Attached Data Specification File and Sample Script for Custom Attached Data

About Interaction Concentrator

This page highlights major ICON features and functionality.

> **Tip**
> For detailed information about many of the features and functions below, see the Interaction Concentrator 8.1 User's Guide.

- For both voice and multimedia, including Genesys eServices (chat and e-mail) and 3rd Party Media, captures and stores information about:
  - The current contact center configuration (objects and associations), and preserves information about deleted configuration objects and terminated associations.
  - Active and completed voice interactions, including switch, DN, time, and routing information about calls and parties. Interaction Concentrator uses a globally unique call identifier.
  - Detailed information about multimedia interactions (e-mail, non-SIP chat, and custom-designed media such as fax and web forms).
  - SIP chat interactions.
  - Agent states and login sessions, for agents handling voice as well as multimedia (e-mail, non-SIP chat, other third-party media) and SIP chat interactions.
  - Detailed information about virtual queue usage in interaction processing, including reporting on a wide range of detailed routing results for interactions that are distributed from virtual queues.
  - Detailed information about interactions that are generated in a network-based contact solution.
  - Detailed information about interactions that are generated in a network call parking environment.
  - Attached data and captures the history of attached data changes.
- For voice interactions:
  - The option to report after-call work (ACW) for the first interaction associated with ACW, as well as the option to suppress the interruption of the ACW and NotReady agent states by interactions coming to, or produced by, the agent.
  - Custom agent states.
  - Detailed information about virtual routing point (VRP) usage in call processing.
  - Customized attached data processing.
- Captures and stores detailed information about outbound campaigns, including:
  - History of campaign processing.
  - History of chain processing.
  - Precalculated metrics provided by OCS.
- Provides a configurable filtering mechanism for certain types of data, to enable the optimization of

database size and performance.

- Provides the ability to resynchronize the configuration data in IDB with Configuration Database on demand.
- Supports high availability (HA) of all types of data through the use of parallel ICON instances, each with its own instance of IDB, in combination with supplementary data that provides information about the availability and reliability of the data stored in IDB.
- Supports near–real-time, intraday reporting by writing data to IDB as soon as the data is available (as opposed to after the interaction is completed).
- Provides a sophisticated recognition mechanism, utilizing Inter-Site Call Linkage (IS-Links), to process multi-site interactions and produce complete data for reporting across sites. Interaction Concentrator provides a stored procedure to merge the interaction records for multi-site interactions.
- Supports multibyte character encoding.
- Stores time information in both Greenwich Mean Time (GMT), as a datetime data type, and Coordinated Universal Time (UTC) seconds, as an integer data type.
  - Obtains the time information from the timestamps of the data provider events (for example, T-Server TEvents), in the form of UTC seconds.
- Provides mechanisms to purge voice and multimedia interactions, agent login session data, attached data, and OCS data that is stored in IDB.
- Supports connections secured by means of Transport Layer Security (TLS) and TLS Federal Information Processing Standards (FIPS) between Interaction Concentrator and the servers to which it connects.
- For enhanced data security, supports encrypted RDBMSs and hiding of TEvent attached data in logs. For more information on these features, see the Genesys 8.1 Security Deployment Guide.

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.

Supported Deployment Scenarios

The architectural choice for your contact center depends on your resources and reporting requirements. In fact, you can tailor the basic scenarios described in this section so that they fit the needs of your contact center at the lowest cost. For example, you can deploy a single instance of ICON for a subset of T-Servers (as opposed to one instance of ICON for each instance of T-Server). Alternatively, you can keep data for a certain site in a separate IDB, if it is not necessary to include data from this site in a consolidated report.

The downstream reporting application might affect your choice of deployment architecture. For example, in deployments that include both voice and multimedia interactions, Genesys Info Mart requires that you use separate ICON applications to process each type of data, and that you store voice and multimedia data in separate IDBs.

> **Important**
>
> Interaction Concentrator does not support deployments in which two ICON instances are configured for the same role, connect to the same T-Server or set of T-Servers, and write data to the same IDB. For more information about the ICON roles and the rules governing role assignments, see ICON Roles.

[+] Diagram conventions used in the deployment scenario diagrams

To simplify the deployment diagrams in this section, please note the following conventions:

- DB Server, which enables a connection between ICON and IDB, is omitted from the diagrams for the various deployment scenarios, even though it is required in the overall architecture (as shown in Basic Architecture).
- Storage of configuration data is not shown, even though it is required in actual deployments.
- Storage of outbound-related call data is not shown because it is optional. It is described separately in Integrating with Outbound Contact.
- Interaction Server is not shown for deployments that include eServices. Notes in the text indicate the deployment scenarios that are suitable for multimedia environments. In these environments, the Interaction Server occupies the same position in the architecture as a T-Server.

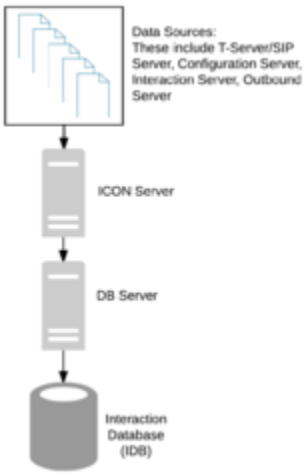
Basic Architecture

Basic Architecture

Operating on top of Genesys Framework, the Interaction Concentrator product consists of a server application called Interaction Concentrator (ICON) and a database called Interaction Database (IDB). The server receives data from the data sources such as Configuration Server, T-Server, or particular Genesys applications; it then stores this data into IDB through Genesys DB Server.

The ICON server interfaces with:

- T-Server or Interaction Server to collect data. For more information on this topic, see the "How Interaction Concentrator Works" chapter in the Interaction Concentrator 8.1 User's Guide.
- Solution Control Interface (Local Control Agent [LCA]), to control when the ICON server starts and stops.
- Configuration Server, to read Interaction Concentrator application configuration options and other configuration objects and options that affect Interaction Concentrator functionality. (This interface is logically separate from ICON's connection to Configuration Server as a source of data about contact center resources.)
- Message Server, to log messages to the Central Logger.



The figure above depicts the basic ICON architecture, omitting most of the Framework components for the sake of simplicity.

Single-Site Deployments

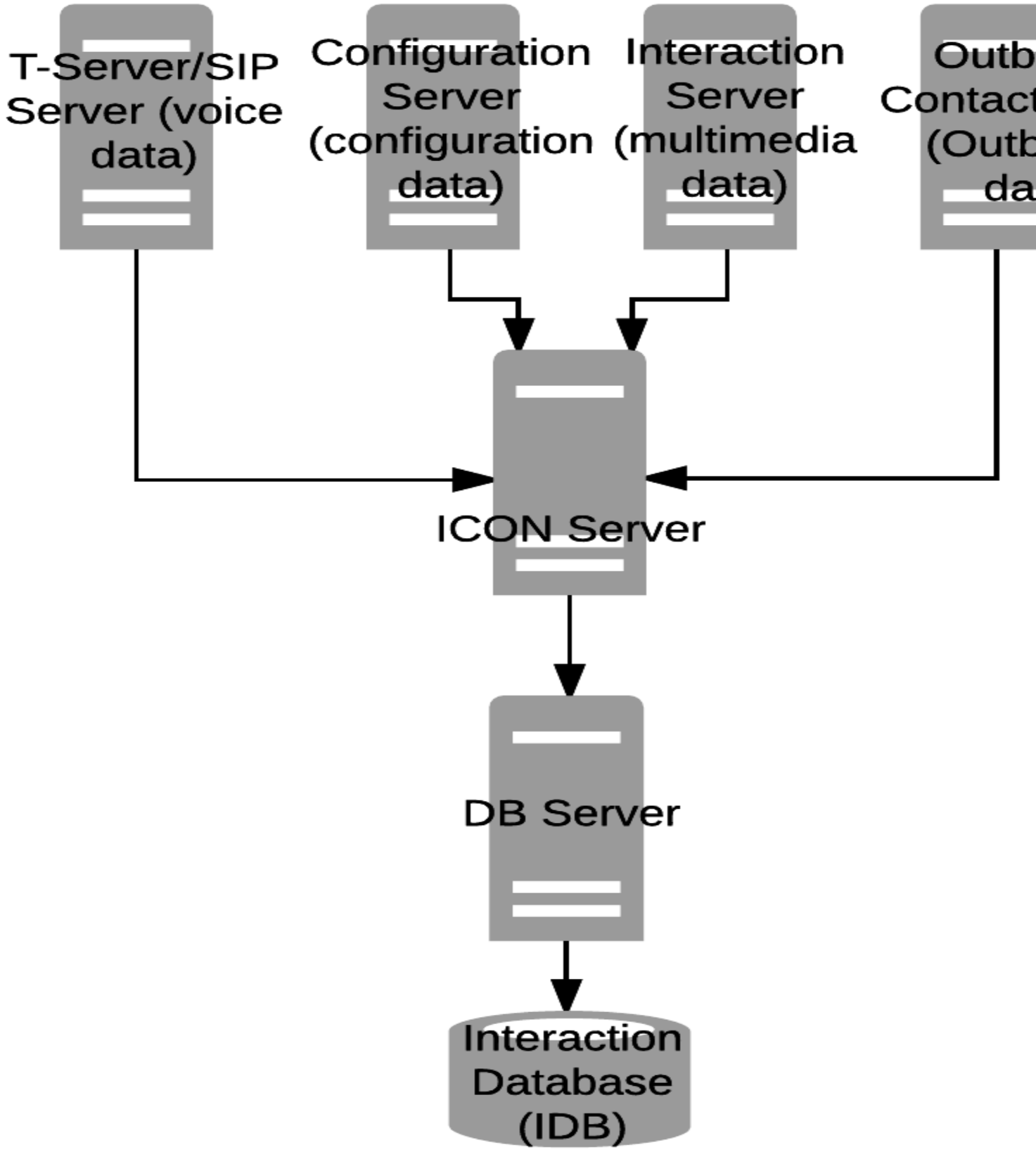
Single-Site Deployments

In a single-site contact center, the following approaches to Interaction Concentrator deployment are typical:

- A single ICON and a single IDB
- Multiple instances of ICON with different roles writing to a single IDB
- Multiple instances of ICON with different roles writing to multiple instances of IDB

One ICON and One IDB

The simplest deployment scenario, which is suitable for smaller, single-site contact centers, consists of a single ICON instance that stores all data into a single IDB instance. Deployments with multiple instances of ICON and multiple instances of IDB are straightforward extensions of this model.


T-Server/SIP Server (voice data)
Configuration Server (configuration data)
Interaction Server (multimedia data)
Outb
Contact
(Outb
da
ICON Server
DB Server
Interaction Database (IDB)

One ICON and One IDB

The Figure above illustrates the deployment for voice interactions. This type of deployment is also suitable for multimedia environments, in which case the Interaction Server occupies the same position in the architecture as T-Server.

Multi-Site Deployments

Multi-Site Deployments

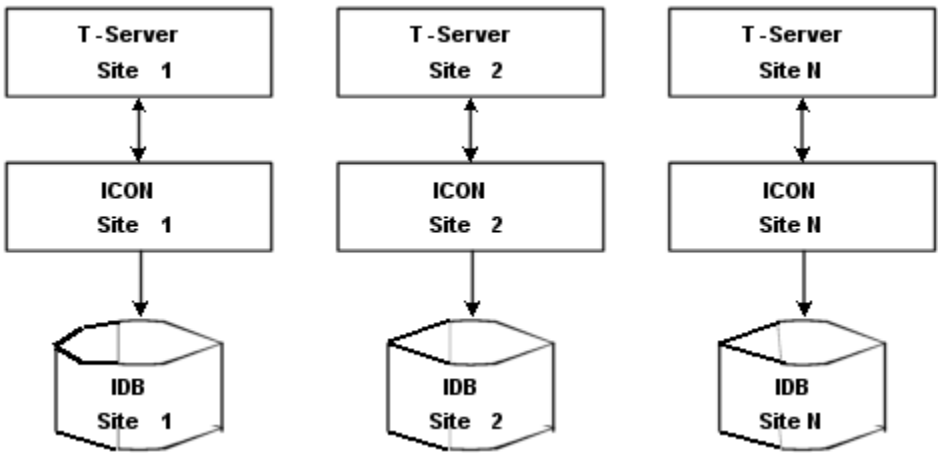
In a multi-site contact center, approaches to Interaction Concentrator deployment vary, depending on network delays between sites, the need for across-the-sites reporting, and other considerations. The following is the basic list of deployments to consider:

- A single ICON instance and a single IDB instance per site
- A single ICON instance and a single, centralized IDB for the entire contact center
- Multiple ICON instances and a single, centralized IDB for the entire contact center

One ICON and One IDB per Site

In a multi-site deployment with a single instance of ICON and a single instance of IDB in each site, each IDB is populated independently from the others with CTI-related data from the T-Server that serves that site.

The graphic illustrates the deployment for voice interactions. This type of deployment is also suitable for multimedia environments—the Interaction Server occupies the same position in the architecture as T-Server. Genesys recommends that you include only one Interaction Server in your deployment.



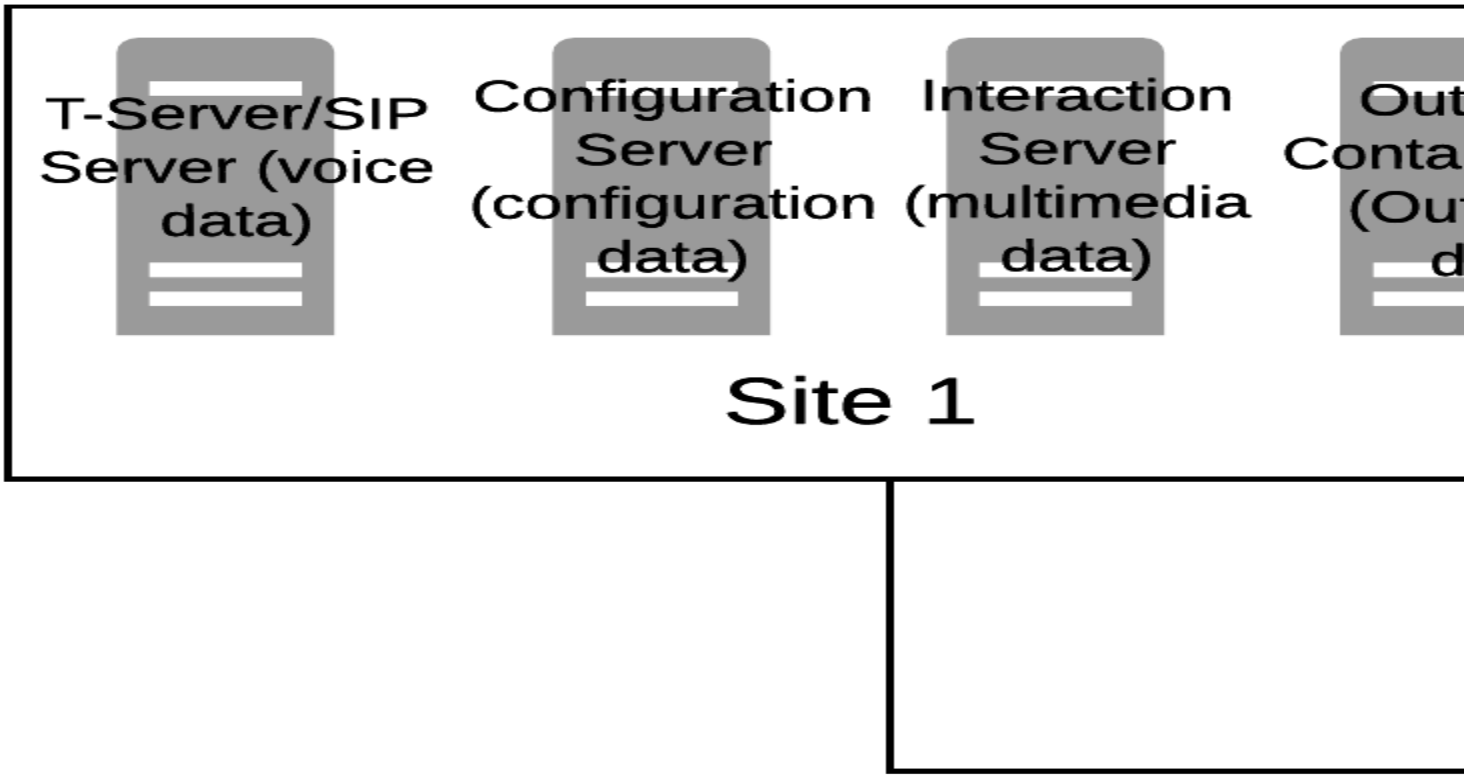
Multi-Site Deployment: Independent IDB Instances

Although the data for a particular site is readily available, this deployment does not provide across-the-sites reporting data for the entire contact center. Merging of data between IDBs is the

responsibility of the downstream reporting application. Reporting of multi-site calls is also limited by the visibility of those calls at a particular site.

One ICON and One IDB per Contact Center

In a multi-site deployment with a single ICON instance and a single, centralized IDB instance, call details from all contact center sites come into the same database through the same ICON.


T-Server/SIP Server (voice data)
Configuration Server (configuration data)
Interaction Server (multimedia data)
Site 1

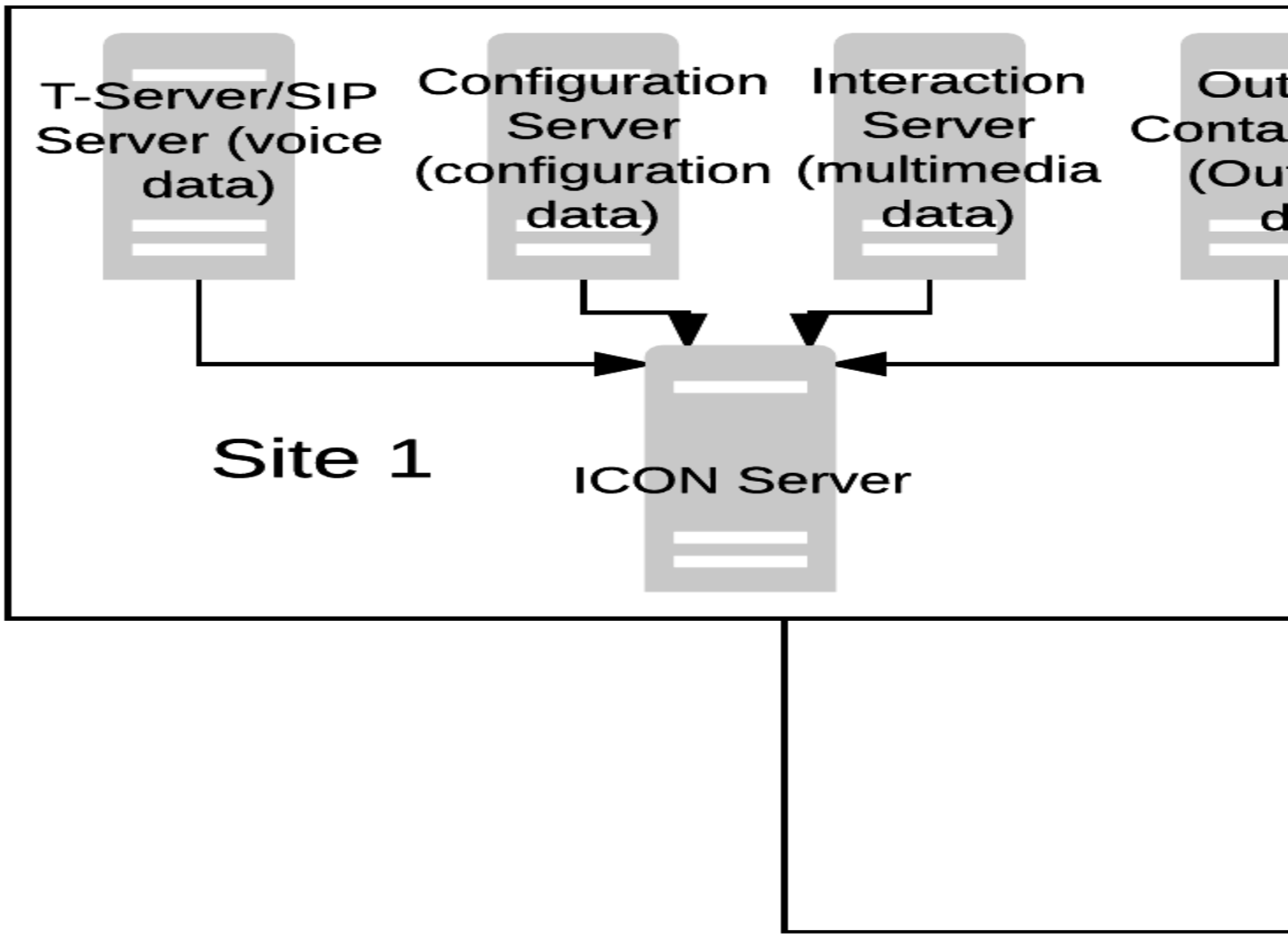
Multi-Site Deployment: A Single ICON and a Centralized IDB Instance

This scenario helps you avoid the need to merge data from different databases. However, note the following considerations:

- You must regularly run the Interaction Concentrator intra-IDB merge stored procedure to ensure correct reporting of multi-site calls (see the information about gsysIRMerge and gsysIRMerge2 in the chapter about special stored procedures in the Interaction Concentrator 8.1 User's Guide).
- Network delays might impact the timeliness of data availability.
- ICON performance is negatively affected during high-peak hours, when each T-Server handles high call volume.

Multiple ICONs and One IDB per Contact Center

In a multi-site deployment with a separate ICON instance at each site and a single, centralized IDB instance, interaction details from all contact center sites come into the same database through separate ICONs.


T-Server/SIP Server (voice data)
Configuration Server (configuration data)
Interaction Server (multimedia data)
Site 1
ICON Server

Multi-Site Deployment: Multiple ICONs and a Centralized IDB Instance

> **Warning**
> This functionality is not supported for eServices (email and chat) or 3rd Party Media. Although Interaction Concentrator supports this deployment scenario, Genesys Info Mart does not. Genesys Info Mart requires that each instance of ICON must store data only in an instance of IDB not connected to any other ICON instance.

Like the scenario of one ICON and one IDB for the contact center (see See Multi-Site Deployment: A Single ICON and a Centralized IDB Instance), this deployment provides the benefit of recording all contact center data in the same database.

However, this scenario provides the additional benefit of improved ICON performance, because a single ICON instance does not require a connection to every T-Server in the contact center. In addition, because T-Server and ICON instances are co-located at a particular site, network delays between these components are minimal.

Nevertheless, the effectiveness of data storage to IDB still depends on network delays between a given ICON instance and IDB, as well as on the performance of your RDBMS. Also, to ensure data correctness for multi-site calls, you must regularly run the Interaction Concentrator intra-IDB merge stored procedure (see the information about gsysIRMerge and gsysIRMerge2 in the chapter about special stored procedures in the Interaction Concentrator 8.1 User's Guide).

Network Deployments

Network Deployments

In a network configuration, a number of Premise T-Server applications are connected to a Network T-Server. The ICON instance connects to the Premise T-Server and Network T-Server applications.

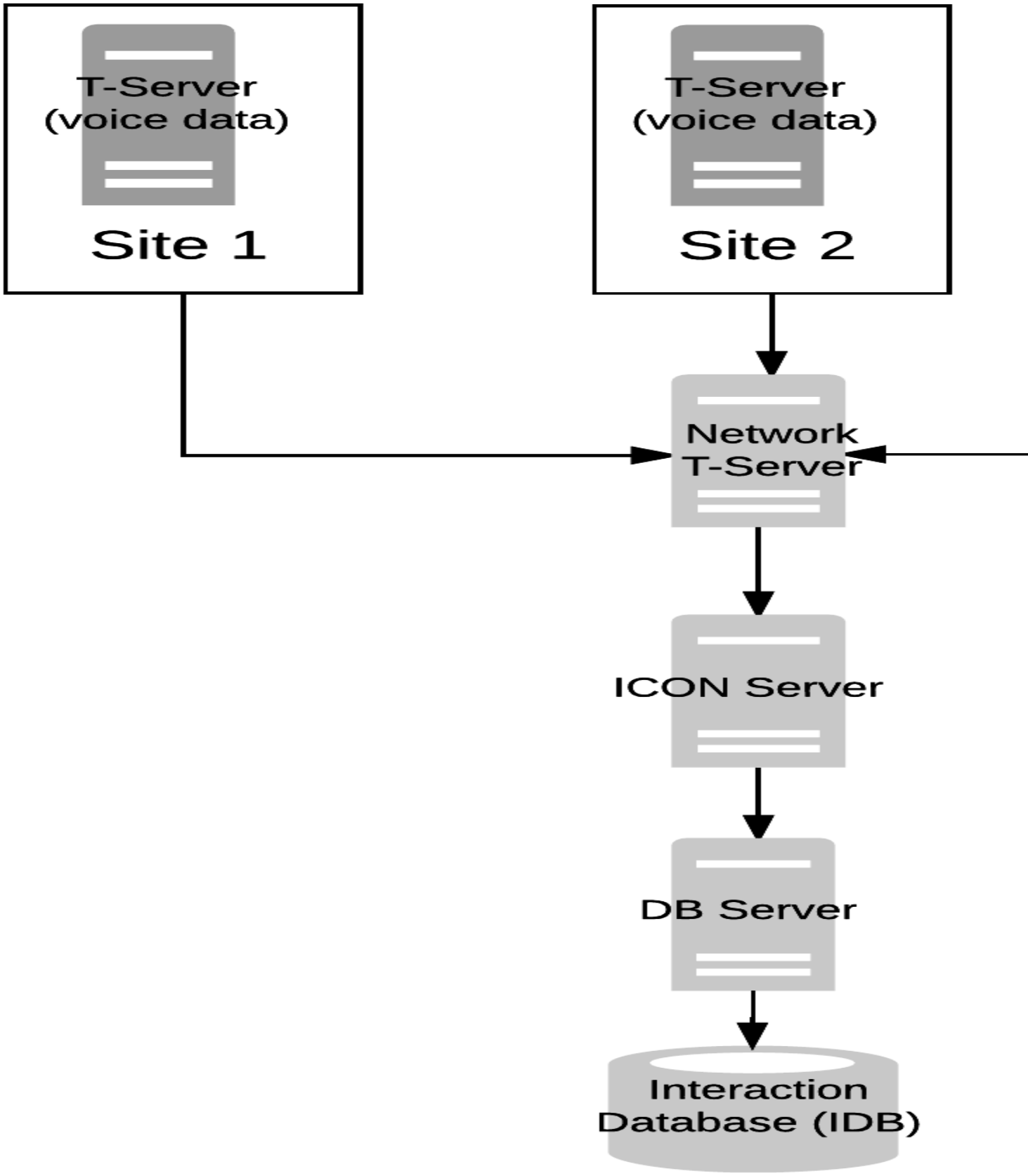
Interaction Concentrator supports the following Network T-Server and IVR T-Server deployments:

- A single ICON connected to a single Network T-Server for each network switch
- A single ICON connected to multiple Network T-Servers for each network switch, operating in load-balancing mode
- A single ICON connected to multiple IVR T-Servers for each IVR switch, operating in load-balancing mode

One Network T-Server per Network Switch

This configuration is applicable for both single- and multi-site deployments, in which there are single or multiple Network T-Servers, and each Network T-Server is connected to a separate switch (in other words, the network switch and Network T-Server are not working in load-balancing mode). Each ICON instance can be connected to multiple Network T-Server applications and multiple Premise T-Server

applications.

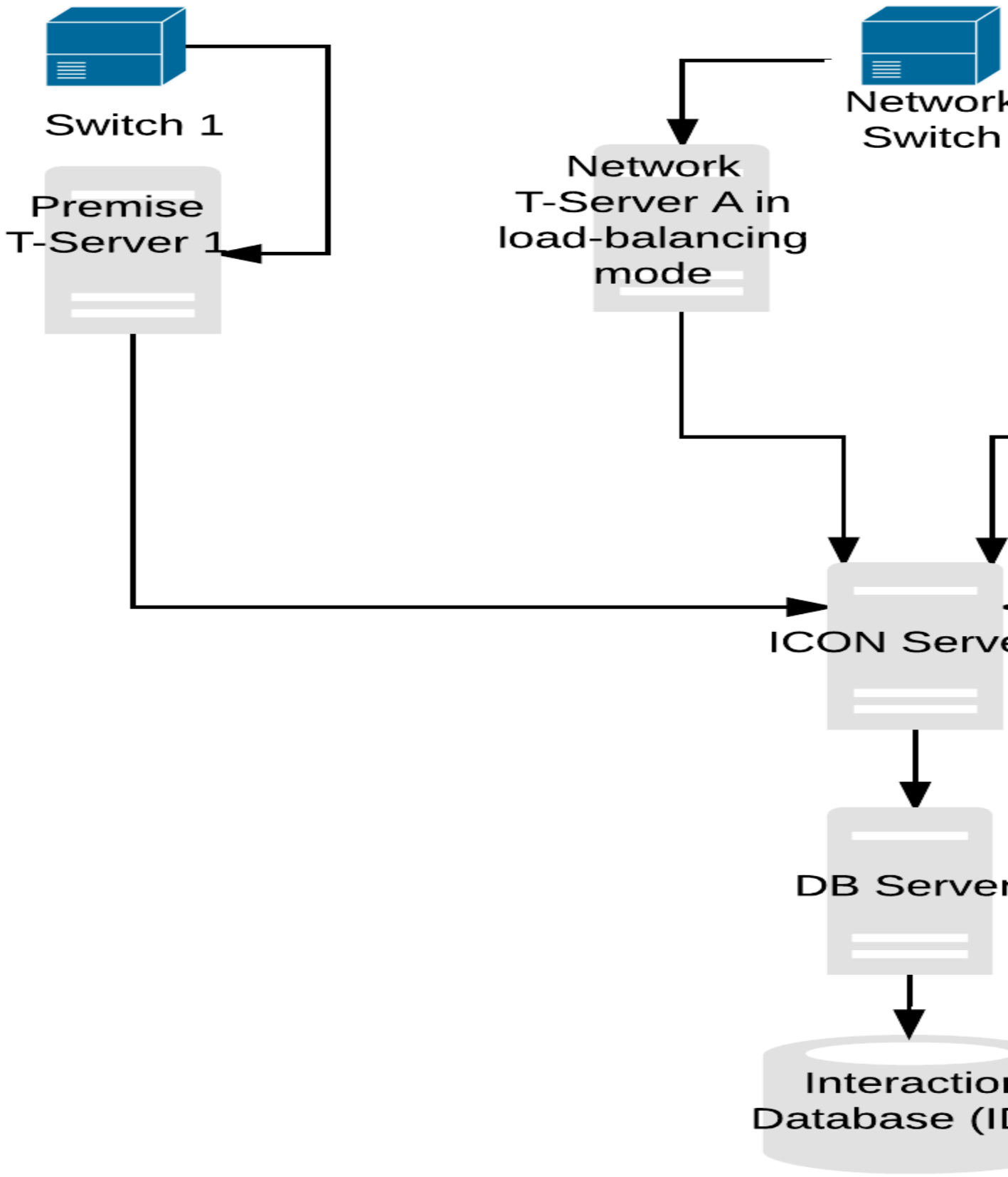

T-Server
(voice data)
Site 1
T-Server
(voice data)
Site 2
Network
T-Server
ICON Server
DB Server
Interaction
Database (IDB)

Network Deployment: A Single Network T-Server per Network Switch

> **Important**
> This Figure does not show the interconnections among T-Server objects that are required for a Network T-Server deployment.

Multiple Network T-Servers per Switch (Load-Balancing Configuration)

If the switch and multiple Network T-Servers have been configured to work in load-balancing mode, ICON supports deployments in which a single ICON instance connects to multiple Network T-Servers that serve the same switch.


Switch 1
Premise
T-Server 1
Network
T-Server A in
load-balancing
mode
Network
Switch
ICON Serve
DB Server
Interactio
Database (I

Multiple T-Servers per ICON-IDB Pair Deployment: Multiple Network T-Servers per Switch (Load-Balancing Configuration)

> **Important**
>
> This Figure does not show the interconnections among T-Server objects that are required for a Network T-Server deployment.

When multiple Network T-Servers serve the same switch in load-balancing mode, the ICON instance creates and maintains a separate connection for each Network T-Server. ICON monitors interaction activity on the switch through the notifications it receives from each Network T-Server.

A configuration option, `switch-multi-links-enabled` (renamed in release 8.1 from `load-balancing-on-network-switch`) , must be set on the Switch object in the Configuration Layer in order for ICON to identify whether the switch and related Network T-Servers are operating in load-balancing mode. For more information about this option, see switch-multi-links-enabled.

Deployments with Network T-Servers in load-balancing mode or with IVR T-Servers in load-balancing mode (described below) are the only configurations in which ICON supports separate connections to multiple T-Servers serving the same switch.

Multiple IVR T-Servers per Switch (Load-Balancing Configuration)

If a switch and multiple IVR T-Servers in an “in-front” configuration have been configured to work in load-balancing mode, ICON supports deployments in which a single ICON instance connects to multiple IVR T-Servers that serve the same switch.

In such a deployment, the ICON instance creates and maintains a separate connection for each IVR T-Server. ICON monitors interaction activity on the switch through the notifications it receives from each IVR T-Server.

A configuration option, `switch-multi-links-enabled`, must be set on the Switch object in the Configuration Layer in order for ICON to identify whether the switch and related IVR T-Servers are operating in load-balancing mode. For more information about this option, see switch-multi-links-enabled.

> **Important**
>
> Deployments with Network T-Servers in load-balancing mode (as described in Multiple Network T-Servers per Switch (Load-Balancing Configuration), above, or with IVR T-Servers in load-balancing mode are the only configurations in which ICON supports separate connections to multiple T-Servers serving the same switch.

ICON Roles

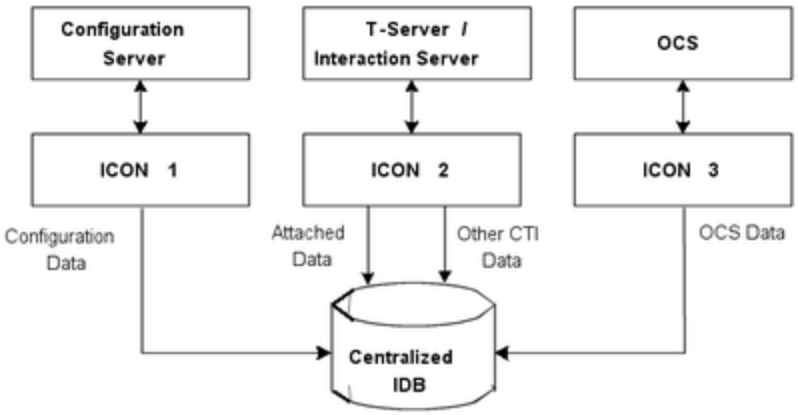
In a contact center that has a large Genesys configuration environment and/or that processes high call volumes, possibly with large amounts of attached data, you can improve Interaction Concentrator performance by deploying multiple ICON instances, each of which collects data only of a certain type.

About Roles

The following are the possible types of data that you can request a given ICON instance to store, in any combination:

- Configuration information—An ICON instance stores the initial contact center configuration state and a history of configuration changes that it retrieves from Configuration Server. Depending on the deployment scenario, the ICON instance can store configuration information about the contact center as a whole or, in a multi-tenant configuration environment, about individual tenants.
- Interaction-related and party-related information—An ICON instance can store T-Server data that pertains to calls and the parties (connections) associated with those calls. In a multimedia deployment, ICON stores similar Interaction Server data about multimedia interactions (e-mail, chat, and 3rd Party Media).

  The role that enables ICON to capture interaction-related and party-related information is the `gcc` role (for more information, see the description of the `role` option). Regardless of whether you have configured the ICON instance to perform this role, if T-Server or Interaction Server is present on the Connections tab of the ICON Application object, ICON will perform aspects of the `gcc` role, which is required for internal processing in connection with other roles. However, for an ICON instance to be able to store interaction and party information from T-Server or Interaction Server, it must have the `gcc` role defined.
- Agent state and login session state information—An ICON instance can store T-Server and, if applicable, Interaction Server data that pertains to agent states and agent login sessions.
- Attached data information—An ICON instance can store T-Server and, if applicable, Interaction Server data that pertains to the attached data that is associated with interactions.
- Outbound calls information—In an environment with the Genesys Outbound Contact, an ICON instance can store OCS data that pertains to outbound calls and campaigns.
- License reporting information—In an environment with License Reporting Manager, you can configure a specific instance of ICON to store license reporting data.



In the example shown in the figure above, the ICON instance named ICON 1 handles only the history of configuration changes (configuration data) from Configuration Server. The instance named ICON 2 handles the business data that agents attach to calls and that T-Server includes in TEvents that pertain to those calls (attached data), as well as any other CTI-related data from the same T-Server or Interaction Server. Finally, the instance named ICON 3 handles OCS data only.

Recommended Role Assignments

Genesys recommends the following role assignments in an environment with multiple ICON instances:

- Do not distribute call-related and party-related information, agent state and login-session state information, and attached data information among separate ICON instances. Assign a single ICON instance to write these three data types from a single T-Server.
- In large configuration environments, Genesys recommends dedicating one of the ICON instances to process configuration data (role = `cfg`) and disabling configuration data processing in the other ICON instances (role = `~cfg`). This improves ICON performance on startup, because the initial configuration loading stage can take quite a long time.
- If multiple ICON instances are writing data to the same IDB, ensure that you assign only one ICON instance to write configuration data to the IDB. See Rules and Restrictions.
- It is preferable not to combine the `lrm` role with any other in the same instance of ICON. If necessary, you can combine the `lrm` and `cfg` roles, but do not combine the `lrm` role with any other than `cfg`.

For the values that enable role assignments, see the description of the role configuration option.

Rules and Restrictions

When you assign ICON roles, observe the following restrictions:

- Two or more instances of ICON that perform the same role(s) cannot store information from the same data source(s) to the same IDB(s).

  For example, if you have two ICONs, each configured to perform the `gcc`, `gud`, and `gls` roles, they can write to the same IDB only if they are connected to different T-Servers or Interaction Servers.

  Conversely, if you have two ICONs, each configured to perform the `gcc`, `gud`, and `gls` roles, they can be connected to the same set of T-Servers or Interaction Servers only if they write to different IDBs.

- Two or more instances of ICON that perform the `cfg` role cannot store configuration information to the same IDB(s).

- Be aware that the default value of the `role` option is `all`.

- If you have more than one instance of ICON writing to the same IDB, you must configure the ICON applications so that only one ICON performs the `cfg` role.

Role assignments must be configured using only lower case (for example, `cfg`). ICON interprets uppercase (`CFG`) or mixed case (`Cfg`) settings as invalid and defaults to the `all` role.

Planning Your Deployment

This chapter lists the prerequisites for Interaction Concentrator deployment. It also provides other primary information that you need in order to plan an Interaction Concentrator (ICON) installation, including information about compatibility with other Genesys components.

Compatibility

Compatibility

This section lists the various Genesys components with which Interaction Concentrator release 8.1 is compatible. For information about supported operating systems and relational database management systems (RDBMSs), see the Genesys Supported Operating Environment Reference Guide. Note that Oracle versions 8.1 and earlier are not supported.

The table below lists the Genesys product components with which Interaction Concentrator operates. Refer to Interaction Concentrator 8.1 Release Note for any updates to the release requirements for the various components.

Area of Functionality	Component/Product
Configuration Layer	Configuration Server release 7.2 or higher • DB Server release 7.2 or higher **Note:** Configuration Server 7.2 does not support all ICON functionality. In particular, the configuration synchronization feature requires Configuration Server release 7.5 or higher; do not attempt to use this feature with Configuration Server release 7.2.
Management Layer	Message Server release 7.1 or higher • Local Control Agent release 7.1 or higher
T-Server	T-Server release 7.2 or higher **Note:** The feature to determine which party released the call requires T-Server release 8.0 or higher, and is supported for the Alcatel A4400/OXE switch and Avaya switches (requires Avaya Communication Manager 8.0.101.05 or higher).
eServices (formerly Multimedia or Multi-Channel Routing (MCR))	eServices Interaction Server release 7.5 or higher **Note:** To configure the total number of keep-in-memory interactions that can reside concurrently in an interaction queue or interaction workbin requires Interaction Server release 7.6.1 or higher.

Area of Functionality	Component/Product
Outbound Contact	Outbound Contact release 7.2 or higher **Note:** If you use OCS 7.2 as the data source, attached data is not automatically linked to the call record. You must specially configure attached data to be linked.
Universal Routing	Universal Routing Server release 7.2 or higher **Note:** For virtual queue reporting, the minimum required version of Universal Routing Server is release 7.2.001.11.

Prerequisites

Prerequisites

Interaction Concentrator has important specific requirements noted in the following subsections. Before you install Interaction Concentrator, review the requirements and recommendations in these subsections.

[+] Hosting

Genesys recommends that you or your IT specialist assign host computers to Genesys software before you start a Genesys installation. Keep in mind the following restrictions:

- Do not install all Genesys server applications on the same host computer.
- When installing multiple server applications on the same host computer, prevent all of them, except Configuration Server, from using swap space.

See the “Network Locations for Framework Components” chapter of the Framework Deployment Guide and the “Architecture” chapter of the Framework Management Layer User’s Guide for information about the optimal locations for:

- Configuration Layer components
- Management Layer components
- T-Server

For Interaction Concentrator and its DB Server, observe the following recommendations:

- Install DB Server on the same computer as the Interaction Database (IDB).
- For better data reliability install ICON as close to T-Server as possible. Ideally, ICON should be on same physical host as the T-Server or Interaction Server for which it gathers data. This prevents network disruptions within the LAN from making an impact on data quality. If you are running a high availability environment, the second instance of ICON should be on a different host but in same subnet as the T-Server or Interaction Server.

Time Synchronization Among Hosts

In an environment with either a single ICON instance or multiple ICON instances operating with multiple T-Servers, synchronize the system time on the T-Server host computers to one second or better.

[+] Genesys Framework

Deploy the Genesys Framework components *before* you deploy Interaction Concentrator.

Configuration Layer

At the very least, you must set up the Configuration Layer of Genesys Framework. You cannot configure Interaction Concentrator components without the Configuration Layer. This layer contains DB Server, Configuration Server, Configuration Database, Configuration Manager, and, optionally, Deployment Wizards.

Management Layer

If you intend to monitor or control Interaction Concentrator and its DB Server through the Management Layer, you must also configure and install Management Layer components—in particular, LCA, Message Server, Solution Control Server (SCS), and SCI.

To monitor the status of Interaction Concentrator components through the Management Layer, you must load an LCA instance on every host that is running ICON and DB Server instances. Without LCA, the Management Layer cannot monitor the status of these components. If you do not use the Management Layer, you do not need LCA.

> **Important**
>
> For information about, and deployment instructions for, these Framework components, see the Framework Deployment Guide and the Framework Management Layer User's Guide.

[+] Telephony Objects

Create configuration objects for every PBX about which you want Interaction Concentrator to store data.

Use Configuration Manager to configure telephony objects, including a Switching Office object and Switch object for the PBX, and one DN (Directory Number) object for each user's telephone number.

- For configuration settings that are specific to Interaction Concentrator, see Switch Configuration Options and DN Configuration Options.

[+] T-Server

If you intend to collect computer-telephony integration (CTI)–related (call) reporting data, configure and install a T-Server application for your particular PBX, if it is not yet deployed. Make sure that the Switch object that this T-Server will serve is specified in the T-Server Application Properties dialog box.

In a multi-site environment, deploy one T-Server application for each PBX.

All T-Servers and, if applicable, Interaction Servers of type T-Server from which an ICON instance should collect data must be listed among the ICON Application object's connections.

> **Important**
>
> For information about, and deployment instructions for, telephony objects and T-Server, see the Framework T-Server Deployment Guide for your particular T-Server.

[+] Outbound Contact

If you intend to collect data about outbound campaigns, deploy components of Genesys Outbound Contact, release 7.2 or higher. If you have an earlier release of Outbound Contact, upgrade to release 7.2 before you deploy Interaction Concentrator.

To provide outbound information to ICON, at least one OCS application must exist and be properly configured. All OCS instances from which an ICON instance should collect data must be listed among the ICON Application object's connections.

For deployment instructions for Outbound Contact components, see the Outbound Contact Deployment Guide. For Outbound Contact migration instructions, see the Genesys Migration Guide. For recommendations on how to enable outbound reporting in Interaction Concentrator, refer to the "Integrating with Outbound Contact" chapter in the Interaction Concentrator 8.1 User's Guide.

[+] Universal Routing

If you intend to collect data about virtual queues, deploy components of Universal Routing, release 7.2 or higher, that support virtual queue functionality. If you have an earlier release of Universal Routing, upgrade to a release that supports virtual queue functionality.

Interaction Concentrator requires Universal Routing Server (URS) release 7.2.001.11 higher for virtual queue reporting.

In order to provide virtual queue information to Interaction Concentrator, at least one URS application must exist. Interaction Concentrator functionality related to writing extended routing results from virtual queues into IDB requires Universal Routing Server (URS) release 7.6 or higher. To enable extraction of this extended routing information, you must also set the values of the **report_reasons** and **report_targets** configuration options in URS to `true`.

For deployment instructions for Universal Routing components, see the Universal Routing 8.1

Deployment Guide. For Universal Routing migration instructions, see the Genesys Migration Guide. For recommendations on how to enable virtual queue reporting in Interaction Concentrator, refer to the chapter about monitoring virtual queues in the Interaction Concentrator 8.1 User's Guide.

Interaction Concentrator functionality related to storing virtual queue history in IDB requires URS release 8.1 or higher.

[+] Multimedia

If you intend to collect interaction, agent state, and agent login session data for eServices (e-mail and chat) and 3rd Party Media interactions, configure and install an Interaction Server application for your multimedia switch, if it is not yet deployed. Make sure the Switch object this Interaction Server will serve is specified in the Interaction Server Application Properties dialog box.

In a multi-site environment, deploy one Interaction Server application for each multimedia switch.

The Interaction Server from which an ICON instance should collect data must be listed among the connections of the ICON Application object. However, ICON cannot connect directly to applications of type Interaction Server. If necessary, you must configure a dummy Interaction Server Application of type T-Server, designate the multimedia switch that Interaction Server uses as a switch for the dummy application, and connect the ICON Application to the T-Server–type Interaction Server. For more information, see Storing Multimedia Data. The functionality introduced in ICON release 7.6.1 to support a large number of concurrently active multimedia interactions requires Interaction Server release 7.6.1 or higher.

For information about—and deployment instructions for—Interaction Server, see the eServices Deployment Guide. For recommendations on how to enable multimedia reporting in Interaction Concentrator, refer to the chapter about integrating with Genesys eServices and 3rd Party Media in the Interaction Concentrator 8.1 User's Guide.

[+] Interaction Database

Interaction Concentrator uses IDB to store reporting data. At least one IDB instance is required, which can be running on any Genesys-supported RDBMS except Sybase and Informix. Oracle versions 8.1 and earlier are also not supported. For a list of Genesys-supported RDBMSs, see the Genesys Supported Operating Environment Reference Guide. For the changes in RDBMS support introduced with Interaction Concentrator 8.1, see the Interaction Concentrator 8.1 Release Note.

When planning an installation, observe the following recommendations for IDB:

- Review the information about the IDB structure in the chapter about IDB schema in the Interaction Concentrator 8.1 User's Guide.
- Estimate IDB size, using the Interaction Concentrator 8.1 Database Size Estimator. For more information, see Database Sizing.

[+] Security Features

Interaction Concentrator supports the following security features:

- Encrypted RDBMSs
- Hiding TEvent attached data from logs.

Support for Secure Connections

- Starting with release 8.1.1, Interaction Concentrator supports Transport Layer Security (TLS) and TLS-FIPS connections.
- Starting with release 8.1.2, Interaction Concentrator Supports client-side port definition, to provide secured connections to T-Server, SIP Server, Configuration Server, and Message Server.

On Windows platforms, support for TLS is integrated into the operating system, and there are no additional requirements to enable Interaction Concentrator to support it. On UNIX-based platforms, you must install the Genesys Security Pack on the Interaction Concentrator host.

For details on all of the supported security features, see the Genesys 8.1 Security Deployment Guide.

General Considerations

General Considerations

Interaction Concentrator is flexible enough to fit any contact center. When planning a deployment, evaluate your environment and your reporting needs. Review the main deployment scenarios in Deployment Scenarios, and answer the following questions:

- How many ICON servers do you need, and what data should each ICON instance handle? From what sources will the data come to a given ICON instance?
- How many Interaction Databases do you need, and what data will each IDB instance store? Which ICON instances will store the data into a particular IDB instance? In the case of multiple IDB instances, will you need to deploy a centralized IDB, and, if so, from which subset of IDBs will data be merged into the centralized IDB? How often should the merge procedure be run?
- How many DB Server applications do you need, if you deploy multiple IDB instances? Will any of these DB Server instances handle database requests for servers other than ICON?
- How many Database Access Point (DAP) applications do you need? What data will each particular ICON instance store through each DAP?

The answers to these questions will help you determine the Interaction Concentrator deployment topology and the main configuration settings for all components.

Database Sizing

Database Sizing

The size of your IDB depends on your deployment scenario, including such factors as typical call flows, attached data storage, values configured for storing outbound data in custom or secure fields, and the amount of time that records will be retained in the database.

Genesys provides an interactive tool to help you estimate the required size of your IDB. This tool, the Interaction Concentrator 8.1 Database Size Estimator, is a Microsoft Excel spreadsheet.

For more information about database sizing and deployment guidelines, see the Interaction Concentrator section in the Genesys Hardware Sizing Guide. The Genesys Info Mart section of the Genesys Hardware Sizing Guide can also provide helpful sizing and performance information for Interaction Concentrator.

Deploying Interaction Concentrator

This section describes the recommended deployment sequence for Interaction Concentrator. It also describes how to deploy each component in your Genesys environment, and includes special configuration for specific and environments, as well as a complete list of the Interaction Concentrator configuration options with detailed explanation of how they affect ICON functioning.

- Configuring and Installing Interaction Concentrator
- Special Configuration Requirements
- Configuration Options

> **Important**
>
> The information in this section is primarily directed towards first-time deployments of Interaction Concentrator. If you are migrating from an earlier release of Interaction Concentrator, ensure that you review the Interaction Concentrator chapters in the Genesys Migration Guide before you perform any installation procedures.

Configuration and Installation

This section provides instructions for configuring and installing Interaction Concentrator, the Interaction Database (IDB), DB Server, and a Database Access Point (DAP).

> **Important**
> DB Server and the DAP are Framework components required to use IDB.

Deployment Order

Recommended Deployment Order

Before you deploy Interaction Concentrator, review Planning Your Deployment, and ensure that you have accounted for all prerequisites for the installation.

Then deploy Interaction Concentrator in the following order:

[+] 1. Host configuration objects

Use Configuration Manager to configure a Host configuration object for the computers on which the DB Server and Interaction Concentrator (ICON) server applications will reside.

For information about Genesys configuration objects, see the Framework Configuration Manager Help.

[+] 2. Telephony objects

Use Configuration Manager to make any modifications to the telephony objects on which ICON will report, including the Switch object for the PBX and any DN (Directory Number) objects that are configured for this Switch object.

For configuration settings that are specific to ICON,see the sections on Switch Configuration Options and DN Configuration Options.

[+] 3. DB Server

> **Tip**
> If you decide to use a DB Server that serves another application for Interaction Concentrator storage purposes, you can skip this step.

To configure and install a DB Server that will handle ICON requests for IDB data storage, use standard deployment instructions from the Framework Deployment Guide. (See also the DB Server tab for the major steps in the procedure.)

For performance reasons, Genesys recommends that you set up the DB Server on the same host as the RDBMS server. In an environment with multiple IDB instances at separate sites, deploy one DB Server per IDB. In an environment with multiple IDB instances at the same site, deploying one DB Server for all IDB instances is sufficient.

[+] 4. Interaction Concentrator

Configure and install Interaction Concentrator.

In an environment with multiple ICON instances, repeat the configuration and installation steps for each ICON instance, making the necessary adjustments when you configure ICON connections and configuration options.

[+] 5. Interaction Database

Install a database for ICON data on one of the supported RDBMSs and initialize IDB. In an environment with multiple IDB instances, repeat the steps to create IDB for each IDB instance.

[+] 6. Database Access Point

1. Configure a DAP Application object that specifies IDB connection parameters.
2. In an environment with multiple ICON instances, each of which stores data to its own IDB instance, repeat the steps to create a separate DAP for each IDB instance. If you decide to write different types of data from a single ICON instance to different databases, also configure a separate DAP for each database.
3. Ensure that the **role** settings that you specify for the DAP are consistent with the **role** settings specified for the ICON instance that it serves.

DB Server

Deploying DB Server

After you configure Host objects for Interaction Concentrator components, deploy as many instances of DB Server as you need using the following procedure:

1. Import the application template for DB Server.
2. Configure an Application object for DB Server.
3. Install DB Server on its host.
4. Set up any environment variables that are specific to your RDBMS type.

For detailed instructions, see the Framework DB Server User's Guide.

ICON

Deploying ICON

This section explains how to deploy as many instances of Interaction Concentrator as you need on either a UNIX or a Windows operating system.

> **Important**
>
> Interaction Concentrator supports silent installation on Windows platforms. Also known as a silent setup, it enables you to install Interaction Concentrator without having to monitor the setup or provide input via dialog boxes. For instructions, see "Appendix E: Silent Setup" in the Framework 8.1 Deployment Guide.

Environment Assumptions
The instructions in this section assume that you are creating new Application objects under the Environment folder in Configuration Manager, in either a single-tenant or multi-tenant configuration environment. To create Application objects under a particular Tenant folder in a multi-tenant configuration environment, replace the word *Environment* with the name of your Tenant folder in the configuration instructions.

[+] 1. Import the Application Template

Before you can configure an Application object for Interaction Concentrator, you must import its application template. The application template provides a majority of the configuration options, as well as the default values for them. You can use this Application Template to create as many Application objects of the same type as you need.

1. Open the Configuration Manager main window.
2. Select the Environment > Application Templates folder.

3. From the File menu, select Import Application Template.
4. In the Look In box, click the down arrow.
5. Locate the Interaction Concentrator 8.1 product CD, and open the TEMPLATES folder.
6. Select the template file for Interaction Concentrator; it is called Interaction_Concentrator_81x.apd.
7. Click **Open** to open the Properties dialog box for the template.
8. Make any changes that you require.
9. Click **OK** to save the template and close the Properties dialog box.

The next step is to configure an Interaction Concentrator Application object.

[+] 2. Create an ICON Application Object

After you import the application template, you can create and configure an Application object for your Interaction Concentrator by using Configuration Manager.

1. Open the Configuration Manager main window.
2. Select the Environment > Applications folder.
3. From the File menu, select New > Application.
4. From the available application templates in the Browse dialog box, select the template that you imported for Interaction Concentrator.
5. On each of the Application tabs, enter the settings appropriate for your environment.

[+] 3. Configure the General tab

In the Properties dialog box, click the General tab, and then enter a name for this application.

> **Tip**
> The application template provides information about the application type and version. Interaction Concentrator uses the Call Concentrator application type for its Application object in the Configuration Layer.

[+] 4. Configure the Server Info tab

Click the Server Info tab, and then specify the following properties:

- Host—Enter the name or IP address of the computer on which you want to install and/or run this server.

> **Important**
> In IPv6 deployments, you cannot set the IP address of the host—only IPv4 addresses can be set for the host. Therefore, enter the *name* of the host instead.

- Communication Port—Enter a numeric value for a port that is not used by another application. Valid values are in the range of 1–65,535. ICON uses this value as the default listening port for the web interface connection.

[+] 5. Configure the Start Info tab

> **Tip**
> The properties you define here are updated automatically during the installation procedure.

Click the Start Info tab, and then specify the following properties:

- Working Directory—Enter the full path to the directory from which the application starts.
- Command Line—Enter the command line that is used to start the application.
- Command Line Arguments—Enter any additional command-line parameters that are used to start the application. For information about command-line parameters, see "Command-Line Parameters" on the Starting and Stopping page.

[+] 6. Configure the Options tab

Click the Options tab, and then specify or change the values of the configuration options, as suitable for your deployment.

- For information about specific configuration requirements to enable ICON to capture and store various types of data (such as voice, multimedia, Outbound, and so on), see Special Configuration Requirements.
- For information about the entire set of ICON configuration options, see Configuration Options.

Role Option

- If you have already deployed another ICON instance that writes to the same IDB, make sure that only one of the ICON applications is configured to store configuration data and the history of configuration changes. For all other instances, deactivate the configuration data storage by configuring one of the following values for the **role** option in the **callconcentrator** section:
  - Explicitly specify one or more values other than `cfg`.
  - Exclude the `cfg` value by using the tilde symbol (~) (that is, set the **role** option to `~cfg`).

- For more information about the **role** option, see the role configuration option description on the Configuration Options page. For more information about role assignments and restrictions for the `cfg` role, see "Recommended Role Assignments" on the ICON Roles page.

Log Options

- Configure both ICON-specific log options and common log options in the log-related configuration sections. For option descriptions, see the log options section on the Configuration Options page and the Framework Configuration Options Reference Manual.
  - If the Interaction Concentrator working directory differs from the directory to which the application is installed, configure an option named **messagefile** in the log section. As the value of this option, specify the full path to the application-specific log messages file (`icon.lms`). Otherwise, ICON will be unable to generate its specific log events.

HTTP Listener

- Configure an HTTP listener by creating a **listeners** section and specifying the appropriate option. For option descriptions, see the listeners section on the Configuration Options page.

[+] 7. Configure the Tenants tab

> **Tip**
> The Tenants tab is displayed only in a multi-tenant environment.

- Click the Tenants tab, and then click **Add** to add all tenants that this ICON application will serve.

It is important to add *all* tenants from whose resources (switches, DNs, agents, and, if applicable, Outbound Contact objects) ICON will collect data.

- If this ICON instance is required to monitor the objects that are configured under the Environment folder, assign the Environment tenant among the other tenants.

[+] 8. Configure the Connections tab
>

Click the Connections tab, and then add the following connections: {{NoteFormat|For any connections between the ICON instance and its data sources, you can configure the connection to use Advanced Disconnect Detection Protocol (ADDP). To enable ADDP for a connection, specify addp as the Connection Protocol when you configure the connection between the Applications, and set the values for the Local Timeout, Remote Timeout, and Trace Mode properties. For more information, see the Framework Deployment Guide.

- If this ICON instance is configured to process CTI-related data, add a connection to T-Server. Note the following special requirements:

- If you have a simple multi-site topology that includes one ICON instance and multiple T-Servers, add a connection to each T-Server. Each T-Server Application object must have a Switch object assigned to it.
- In a deployment with Network T-Server, add a connection to the Network T-Server. The Network T-Server Application object must have a network Switch object assigned to it.
- In a deployment in which the Network T-Servers function in load-balancing mode, add a connection to each Network T-Server. For more information, see "Multiple Network T-Servers per Switch (Load-Balancing Configuration)" on the Supported Deployment Scenarios page.

Important

Starting in release 8.1.4, if host or port information changes for any T-Server listed on the Connections tab, Interaction Concentrator dynamically reconnects using the new connection parameters. While ICON disconnects from the prior host/port and connects to the new one, there might be a brief gap in data received from the T-Server.
In releases 8.1.0 and 8.1.1, you must restart ICON for updates to listed T-Servers to take effect.

- If this ICON instance is configured to process multimedia interactions reported by Interaction Server, add a connection to Interaction Server. The Interaction Server must be of type T-Server (for details on configuring Interaction Server correctly, see the information about ICON connections in the section, "Storing Multimedia Data" on the Special Configuration Requirements page.

Important

If you are installing ICON in a deployment that will use Genesys Info Mart to report on both voice and multimedia interactions, you must create separate ICON instances and separate IDBs for the voice and multimedia data.

- If this ICON instance is configured to process outbound data in an environment with Genesys Outbound Contact, add a connection to one or more OCS applications.
- Add a connection to the Configuration Server application (named `confserv`). You can enable ADDP for connections to Configuration Server in the same way as for other connections.
  - Genesys recommends that, at a minimum, you add a connection to Configuration Server in the ICON instance that is configured to process configuration data (`cfg` role). However, to minimize the number of unidentified call segments that might occur as a result of missing configuration data, Genesys recommends that you configure an ADDP connection to Configuration Server in each ICON Application in your deployment, regardless of the role of the ICON Application.
- If you installed the Management Layer, add a connection to Message Server, in order to provide alarm signaling and centralized logging capabilities. You can add a connection to Message Server for all or a set of Application objects after you configure them. To launch a wizard that configures connections for multiple Application objects, select two or more Application objects, right-click, and then select Manage Connections from the shortcut menu. For more information, see the Framework Configuration Manager Help.
- After you configure one or more DAP Application objects (see the DAP tab on the Configuration and Installation page), add to the ICON Connections tab any DAP Application objects through which this

ICON instance will access IDBs.

- In high availability (HA) environments using primary and backup pairs of servers, the servers listed on the Connections tab are handled as *primary*. To specify the backup servers for any primary servers, open the Application object for the primary server and add the backup server on the primary server's Server Info tab. Click OK to save your changes, then close the Properties dialog box.

IDB

Creating IDB

You can use any of the supported RDBMSs to host your IDB. Ask your Database Administrator to create a new database for each IDB instance that you intend to deploy for ICON data storage. Then, initialize each IDB instance, using the instructions in this section. {{NoteFormat|The user account that is created for IDB must have permissions to create database objects such as tables, stored procedures, and sequences.

The IDB initialization scripts create default (empty) custom dispatchers without first dropping any existing stored procedures named `gudCustDISP1` and `gudCustDISP2`. This is to decrease the risk of overwriting customer-created stored procedures. However, if the `gudCustDISP1` and `gudCustDISP2` custom dispatcher stored procedures already exist in IDB, the `CoreSchema_<db_type>.sql` script returns an error, which is safe to ignore.

Genesys provides the `SampleProc_<db_type>.sql` script to help you understand how you can modify the stored procedures for customized attached data processing. Do not execute the sample script during installation. For more information about configuring your ICON application to support customized attached data processing, see the Attached Data tab on the Special Configuration Requirements page. For an example of a script to create a custom dispatcher stored procedure and custom storage table, see Sample Script for Custom Attached Data.

[+] Initialization Scripts

After you install the ICON application, the `scripts` subfolder in the directory to which you installed ICON contains a set of initialization, migration, and sample scripts for each RDBMS type. See Initialization Scripts on the IDB tab of Configuration and Installation for a list of these scripts and the purpose of each. In the script names, `<db_type>` is a placeholder for the specific RDBMS type (`db2`, `mssql`, `postgre`, or `ora` [for Oracle]).

> **Warning**
>
> If you are migrating from an existing IDB, do not simply apply all the scripts listed under Initialization Scripts on the IDB tab of Configuration and Installation. To avoid damaging or erasing existing data, follow the migration procedures that are described in the Interaction Concentrator chapters in the Genesys Migration Guide and in the Interaction Concentrator 8.1 Deployment Procedure.

If you are running Genesys Info Mart 8.1.1 or 8.1.0, see Scripts Required for Environments Running Genesys Info Mart 8.1.1 or 8.1.0 below.

Table of Initialization Scripts

Script Name	Description
CoreSchema_<db_type>.sql	(For initial installation only) Creates the core IDB tables and indexes.
Upgrade_<target_database_version>_<db_type>.sql	(For migration only) Upgrades the IDB schema. The scripts you must execute depend on the releases from which and to which you are upgrading. For more information, see the instructions in the Interaction Concentrator chapters in the Genesys Migration Guide or the Interaction Concentrator 8.1 Deployment Procedure.
CoreProcedures_<db_type>.sql	Creates the database schema–specific set of stored procedures that implement core ICON functionality, including the merge procedures and the separate procedures to purge different types of data.
Purge2_<db_type>.sql	Creates the gsysPurge80 or gsysPurge81 stored procedure.The version created corresponds to the release of Interaction Concentrator you are installing.
Wrapper_for_<schema version>_<db_type>.sql	Links generically named merge and purge procedures to the equivalent, schema-specific stored procedures in the new set.
drop_<schema version>_<db_type>.sql	(Optional, for migration only) Removes the set of stored procedures for the specified Interaction Concentrator schema version.
SampleProc_<db_type>.sql	Serves as a sample script, illustrating how to create a custom attached data storage table and modify the custom dispatcher stored procedures.
CoreSchemaPart_ora.sql	(Optional, for Oracle RDBMSs only) Creates tables, sequences, and indexes for use in a partitioned schema.
PurgePart_ora.sql	(Optional, for Oracle RDBMSs only) Creates the stored procedure required to purge a partitioned Oracle IDB.

Scripts Required for Environments Running Genesys Info Mart 8.1.1 or 8.1.0

For environments that include Genesys Info Mart 8.1.1 or an earlier 8.1.x release, run the appropriate script as described in this section every time you migrate to a new release of Interaction Concentrator. (Starting with release 8.1.4, Genesys Info Mart automatically runs the scripts when

required.)

> **Tip**
> For the location of the scripts and detailed instructions, see the Genesys Info Mart 8.1 Deployment Procedure or the Genesys Info Mart 8.1 Deployment Guide.

- For a Voice details IDB, use `update_idb_for_gim.sql`.
- For a Multimedia details IDB, use `update_idb_for_gim_mm.sql`.
- For a Configuration details or an Outbound Contact details IDB, use either `update_idb_for_gim.sql` or `update_idb_for_gim_mm.sql`.

[+] Initializing IDB

For a first-time initialization of IDB, follow the RDBMS-specific instructions in the following procedures:

Initializing IDB on DB2

To initialize IDB by running the initialization scripts provided for a DB2 database:

1. Go to the directory into which you installed ICON.
2. Go to the `scripts\db2` subdirectory.
3. Execute the following scripts in the order shown:
   1. `CoreSchema_db2.sql`
   2. `CoreProcedures_db2.sql`
   3. `Purge2_db2.sql`—This script is optional. Execute this script if you want to use the `gsysPurge80/gsysPurge81` stored procedure in your deployment.
   4. Wrapper_for_<schema version>_db2.sql—Execute this script if your deployment will use the `gsysIRMerge` or `gsysIRMerge2` merge procedure, or if your deployment will use the `gsysPurgeIR`, `gsysPurgeUDH`, `gsysPurgeLS`, or `gsysPurgeOS` purge procedures.
4. To execute the scripts:
   - Insert the following command line at the beginning of each script, providing appropriate values for the placeholders:

     ```
     db2 connect to <dbname> user <user> using <password>
     ```

   - Use the following command line to load each initialization script:

     ```
     db2 -w -td@ -f<script_name>
     ```

Initializing IDB on Microsoft SQL

To initialize IDB by running the initialization scripts provided for a Microsoft SQL database:

1. Go to the directory into which you installed ICON.
2. Go to the `scripts\mssql` subdirectory.
3. Execute the following scripts in the order shown:
    1. `CoreSchema_mssql.sql`
    2. `CoreProcedures_mssql.sql`
    3. `Purge2_mssql.sql`—This script is optional. Execute this script if you want to use the `gsysPurge80/gsysPurge81` stored procedure in your deployment.
    4. `Wrapper_for_<schema version>_mssql.sql`—Execute this script if your deployment will use the `gsysIRMerge` or `gsysIRMerge2` merge procedure, or if your deployment will use the `gsysPurgeIR`, `gsysPurgeUDH`, `gsysPurgeLS`, or `gsysPurgeOS` purge procedures.
4. To execute the scripts, use the following command line to load each initialization script, providing appropriate values for the placeholders:

```
sqlcmd -S <dbms_server> -d <dbname> -U <user> -P <password> -i <script_name>
```

where `sqlcmd` is `isql.exe` or `osql.exe`

Initializing IDB on Oracle

To initialize IDB by running the initialization scripts provided for an Oracle database:

1. Go to the directory into which you installed ICON.
2. Go to the `scripts\oracle` subdirectory.
3. Execute the following scripts in the order shown, unless you are creating a partitioned Oracle database, in which case use the list of scripts that follows the standard set.
    - To execute the scripts, log in to the `sqlplus` command processor, and type the following at the command prompt:

```
@ <script_name>
```

    - For a standard Oracle IDB, run the following scripts:

    1. `CoreSchema_ora.sql`
    2. `CoreProcedures_ora.sql`
    3. `Purge2_ora.sql`—This script is optional. Execute this script if you want to use the `gsysPurge80/gsysPurge81` stored procedure in your deployment.
    4. `Wrapper_for_<schema version>_ora.sql`—Execute this script if your deployment will use the `gsysIRMerge` or `gsysIRMerge2` merge procedure, or if your deployment will use the `gsysPurgeIR`, `gsysPurgeUDH`, `gsysPurgeLS`, or `gsysPurgeOS` purge procedures.

    - If you are using partitioning on Oracle, run the following scripts instead:

1. `CoreSchemaPart_ora.sql` (instead of `CoreSchema_ora.sql`)
2. `CoreProcedures_ora.sql`
3. `PurgePart_ora.sql` (instead of `Purge2_ora.sql`).Execute this script if you want to purge a partitioned IDB by truncating partitions. This purge method can speed up the purge process in large deployments.

- For detailed information on partitioning, a list of the tables that can be partitioned, and how purging works in a partitioned IDB, see "Purging by Truncating Partitions" section in the "Using Special Stored Procedures" of the Interaction Concentrator 8.1 User's Guide.
- For instructions on starting to use a partitioned IDB, see the Interaction Concentrator section of the Genesys Migration Guide.

Initializing IDB on PostgreSQL

To initialize IDB by running the initialization scripts provided for a PostgreSQL database:

1. Go to the directory into which you installed ICON.
2. Go to the `scripts\postgre` subdirectory.
3. Execute the following scripts in the order shown:
   1. `CoreSchema_postgre.sql`
   2. `CoreProcedures_postgre.sql`
   3. `Purge2_postgre.sql`—This script is optional. Execute this script if you want to use the `gsysPurge81` stored procedure in your deployment.
   4. `Wrapper_for_<schema version>_postgre.sql`—Execute this script if your deployment will use the `gsysPurgeIR`, `gsysPurgeUDH`, `gsysPurgeLS`, or `gsysPurge0S` purge procedures.
4. To execute the scripts, use the following command line to load each initialization script, providing appropriate values for the placeholders:

```
psql -h <dbms_server> -U <user> --dbname=<dbname> --file=<script_name>
```

> **Important**
>
> PostgreSQL is supported for use with Genesys Info Mart 8.x, which does not use the Interaction Concentrator merge stored procedures.

DAP

Configuring a DAP

If you are unsure how to use Configuration Manager to create a new DAP Application object, refer to one of the following sources:

- Appendix A, "Standard Configuration Procedure" of the Framework Deployment Guide
- Framework DB Server User's Guide

In addition to the standard configuration steps, complete the following procedure:

1. On the General tab, when you specify the application name, keep in mind that the DAP can have the same name as the database itself. However, if you are using multiple access points to the same database, make their names unique.
2. On the General tab, click **Browse** to locate the DB Server through which this database is to be accessed. This must be the DB Server that is either deployed or reused for ICON purposes (see the DB Server tab on the Configuration and Installation page).
3. Do *not* select the JDBC Connection check box, because it does not apply to database connections through DB Server.
4. On the DB Info tab, specify the properties as follows:
   - DBMS Name—The name or alias that identifies the RDBMS that handles IDB. The value of this option is communicated to DB Server so that it connects to the correct RDBMS:
     - For Oracle, set the value to the name of the Listener service (also known as a database alias).
     - For Microsoft SQL, set the value to the name of the SQL server (usually the same as the host name of the computer on which Microsoft SQL runs).
     - For PostgreSQL, set this value to the SQL server name (usually the same as the host name of the computer where PostgreSQL runs), as configured in the DB Server Application.
     - For DB2, set the value to the name or alias name of the database, as specified in the db2 client configuration.
   - DBMS Type—The type of RDBMS that handles IDB. You must set a value for this property.
   - Database Name—The name of the IDB instance to be accessed, as it is specified in the RDBMS that handles this database. You must set a value for this property, unless you specify oracle or db2 for DBMS Type. For Microsoft SQL and PostgreSQL, the value is the name of the database to which the client will connect.
   - User Name—The user name for accessing IDB, as established in the SQL server. You must set a value for this property.
   - Password—The password for accessing IDB, as established in the SQL server.
   - Re-enter Password—Confirmation for the value that you entered for Password.
   - Case Conversion—The case conversion method for key names of key-value lists that come from DB Server. This value specifies whether, and how, a client application converts the field names of a database table when it receives data from DB Server. This parameter does not affect the values of key-value lists that come from DB Server—the actual data is presented exactly as it appears in the database tables.
     - `upper`—Field names are converted into uppercase.
     - `lower`—Field names are converted into lowercase.

- any—Field names are not converted. Use the default value (any), unless Genesys Customer Care directs you to do otherwise.

5. When configuring a DAP Application object for IDB, do *not* configure any properties on the JDBC Info tab.
6. If you intend to use multiple database access points to write different types of ICON data to different databases, specify which type(s) of data this particular database access point must handle. To do so,perform the following steps:
   1. On the Options tab, create a section named `callconcentrator`.
   2. Within the `callconcentrator` section, create a configuration option named `role`.
   3. Set the option value to indicate the types of data that will be stored through this DAP. For more information, see description of the `role` option under ICON Role in the callconcentrator section of the Configuration Options page. The `role` option values must be lower-case (for example, `cfg`). ICON interprets role option values in uppercase (`CFG`) or mixed case (`Cfg`) as invalid and defaults to the `all` role.
      - For optimal performance, Genesys recommends the following sets of values for a given database access point:
        - `gcc,gud,gls`
        - `cfg`
        - `gos`
7. After you configure a DAP Application object, add it to the Connections tab of the ICON application that will use this DAP as an interface to IDB.

Special Configuration Requirements

This section describes how to configure the Interaction Concentrator (ICON) Application object and other applications in the Genesys Configuration Layer in order to make various kinds of data available in the Interaction Database (IDB).

Click the tabs to view instructions for storing the various sorts of data you might require.

Voice

Storing Voice Data

In order to store voice interaction, agent state, and login session data in IDB, certain configuration settings are required in the Genesys Configuration Layer. This section describes the configuration settings that are required on the ICON `Application` object.

Connections:
To enable ICON to receive voice data and store it in IDB, you must configure ICON connections to appropriate T-Server instances.

Configuring for Voice Data

Any ICON `Application` object that has a configured connection to T-Server processes voice interaction data, regardless of the role that has been configured for the ICON application. However, to enable ICON to store interaction-related and party-related data for voice calls in IDB, you must configure the `gcc` role for the ICON application and associated Database Access Point (DAP).

To capture other types of data for voice objects and interactions, you must configure the appropriate values for the role configuration option.

To enable ICON to identify the party that initiated release of a call, in deployments that support this functionality, set the value of the store-releasing-party option to 1.

Filtering Data

To improve Interaction Concentrator performance, consider excluding certain types of data from IDB storage. Review the filtering options in the filter-data section, and set appropriate values for your deployment.

If your deployment utilizes the feature to identify which party initiated release of a call, be aware that

certain ICON filtering options can effectively disable this functionality.

For call-based reporting, the `call-metrics` option, in the `filter-data` configuration section, must be set to `0` (the default). Otherwise, ICON does not write any data to the `G_CALL_STAT` table. The following options in the `filter-data` configuration section affect storage of information in the `G_PARTY_STAT` table:

- `acd-party-history`
- `acd-party-metrics`
- `external-party`
- `observer-party`

If you want to implement DN-based reporting on the parties that initiated release of calls, Genesys recommends that you retain the default values for these options, so that you do not filter party information. For more information about using this feature, see the section about populating voice data in the Interaction Concentrator 8.1 User's Guide.

Multimedia

Storing Multimedia Data

In order to store multimedia interaction, agent state, and login session data in IDB, certain configuration settings are required in the Genesys Configuration Layer. This section describes the configuration settings that are required on the ICON `Application` object.

For more information about multimedia data in Interaction Concentrator, see the chapter about integrating with Genesys eServices/Multimedia and 3rd Party Media in the Interaction Concentrator 8.1 User's Guide.

[+] ICON Application

To enable ICON to receive eServices and 3rd Party Media data and store it in IDB, you must configure ICON connections to appropriate Interaction Server instances.

Connections

ICON cannot connect directly to an `Application` object of type Interaction Server. Instead, an `Application` object of type T-Server must represent Interaction Server.

You must perform the Interaction Server configuration differently depending on your environment, as follows:

- In a single-tenant environment or an environment with a single Interaction Server for each tenant, create a single application of type T-Server for each Interaction Server.

- In an environment with an Interaction Server that serves multiple tenants, you must create for each Interaction Server:
  - One application of the Interaction Server type (which can accommodate multiple Tenants).
  - As many applications of the T-Server type as there are tenants served by this Interaction Server, one for each tenant. Configure these applications using the actual Interaction Server host and port settings, following the instructions below.

To have ICON recognize connect to Interaction Server, execute the following steps:

1. Create an application with application type T-Server in Configuration Server. This might be either in addition to or instead of an Interaction Server-type application, depending on your environment (see the explanation in the bullet-points above).
2. On the Server Info tab of this application, specify the host and port parameters for your Interaction Server. If you are using both an Interaction Server-type application and a T-Server-type application, the host and port parameters must be identical.
3. Designate the multimedia switch that Interaction Server uses as a switch for the T-Server-type application.
4. Add the T-Server-type application to the Connections tab of the ICON application (instead of the Interaction Server application).
5. If ICON is already running, restart it.

> **Important**
>
> Starting in release 8.1.2, if host or port information changes for any T-Server listed on the Connections tab, Interaction Concentrator dynamically reconnects using the new connection parameters.
>
> While ICON disconnects from the prior host/port and connects to the new one, there might be a brief gap in data received from the T-Server.
>
> In releases 8.1.0 and 8.1.1, you must restart ICON for updates to listed T-Servers to take effect.

[+] Other Configuration for Multimedia

There are no special requirements for other ICON `Application` object configuration options. The type of data that ICON captures for multimedia objects and interactions depends on the `role` configuration option that you configure for the ICON instance.

[+] Configuring for 3rd Party Media

To enable ICON to store information about 3rd Party Media interactions in IDB, you must configure the mcr-om-processing configuration option.

Attached Data

Storing Attached Data

Attached data refers to the interaction-related business data that is sent by T–Server or Interaction Server as key–value pairs (KVPs) in the `UserData`, `Extensions`, or `Reasons` attributes in TEvents.

Configuring Interaction Concentrator to store attached data in IDB is a two–part process:

1. Specify the attached data key configuration file, which maps the key–value pairs (KVPs) in reporting event attributes to IDB tables and fields. For more information, see the Attached Data Specification File.
2. Specify the attached data configuration settings in the Genesys Configuration Layer.

- For more information about attached data in Interaction Concentrator, see the chapter about processing attached data in the Interaction Concentrator 8.1 User's Guide.
- For information about configuring Interaction Concentrator to store user data from EventUserEvents that are distributed by T–Server or Interaction Server from other client applications (for example, from an agent desktop application), see the Storing Agent State and Login Session Data tab on this Special Configuration Requirements page.

[+] ICON Application

This section describes the configuration settings that are available on the ICON Application object.

ICON Role Configuration Option

For every ICON instance that must store attached data, make sure that the `role` option on the `Options` tab of the ICON `Application` object includes gud in the list of values. If you deploy a single ICON instance for the entire contact center, you can keep the default value (`all`). For more information, see the description of the `role` configuration option under ICON Role in the callconcentrator section of the Configuration Options page.

Attached Data Specification File

The `adata–spec–name` ICON configuration option enables you to point ICON to a different attached data specification file:

Attached Data Configuration Options

The following ICON configuration options enable you to specify what attached data ICON should store, and in what manner:

- adata–default–storage
- adata–extensions–history
- adata–reasons–history
- adata–userdata–history

Review the descriptions and values for the configuration options under Attached Data in the callconcentrator section of the Configuration Options page. Select the appropriate values for your environment, and make related configuration changes on the `Options` tab of the ICON `Application` object.

Custom Dispatcher Configuration Options

The following ICON configuration options enable you to specify how the custom dispatcher will process attached data:

- gud–cust–disp
- gud–cust–disp–groups

Review the descriptions and values for the configuration options under Custom Dispatcher in the callconcentrator section of the Configuration Options page. . Select the appropriate values for your environment, and make related configuration changes on the `Options` tab of the ICON `Application` object.

[+] Overview: Attached Data Specification File

If you require ICON to store attached data in IDB, create an attached data specification for ICON to use. The attached data specification is an XML file stored in the installation directory that you specify when you install the Interaction Concentrator application.

- For the XML schema definition for your attached data specification, see the Schema Definition tab of the Attached Data Specification File page.

> **Important**
>
> If you change the XML file, you must restart ICON in order for the changes to take effect.

For sample attached data specifications, see:

- Sample Basic Attached Data Specification
- Sample Specification for Multimedia Attached Data
- Sample Specification for Customized Attached Data

In this section, unlike in the rest of this document, angle brackets indicate required syntax elements and do not indicate placeholders. *Italics* indicate placeholder text.

This section provides information about the following topics:

[+] Parser Limitations

The ICON XML parser imposes the following limitations:

- ICON ignores unknown attributes if they are present in the specification. When parsing the XML specification, ICON checks only for missing attributes.
- The ICON XML parser does not support namespaces.
- ICON ignores duplicate keys. Only the first occurrence of a key name is used to update the specified field in the database table.

[+] Attribute Values

This section describes the attributes that are used in the XML schema definition.

History Types

The following values can be used as history types:

none	No value for a given key is recorded in IDB.
first	Only the first value for a given key is recorded in IDB.
last	Only the last value for a given key is recorded in IDB.
all	Every change in value for a given key is recorded in IDB. This value applies only to keys that are configured to be stored in the history tables.

Storage Types The table below shows the IDB table in which each attribute is stored.

Attribute Name	IDB Table Name
public	G_USERDATA_HISTORY
secure	G_SECURE_USERDATA_HISTORY
call	G_CALL_USERDATA
call-cust	G_CALL_USERDATA_CUST
call-cust1	G_CALL_USERDATA_CUST1
call-cust2	G_CALL_USERDATA_CUST2
mcr-f	GM_F_USERDATA
mcr-l	GM_L_USERDATA
user_supplied_name, such as `cust-disp-group-n`	Customer-defined name, as specified in the custom dispatcher.

Data Source Types

The following table shows the TEvent attribute from which each attribute is derived.

Attribute Name	TEvent Attribute Name
reasons	AttributeReasons
extensions	AttributeExtensions
userdata	AttributeUserData

[+] IDB Fields

The mapping between the field attribute (the logical key name) in the attached data specification and fields in the IDB tables is predefined. This section describes the predefined IDB fields for:

- Voice attached data
- Multimedia-specific attached data
- Custom attached data

Predefined IDB Columns—Voice

For voice calls, the table below shows the predefined IDB field in which each attribute is stored in the G_CALL_USERDATA table.

Attribute Name	G_CALL_USERDATA Field
customer-segment	G_CUSTOMER_SEGMENT
service-type	G_SERVICE_TYPE
service-subtype	G_SERVICE_SUBTYPE
business-result	G_BUSINESS_RESULT
customer-id	CUSTOMER_ID
transaction-id	TRANSACTION_ID
cause-id	CAUSE_ID
account-id	ACCOUNT_ID
destination-id	DESTINATION_ID
target-id	TARGET_ID

Predefined IDB Columns—Multimedia

For eServices and 3rd Party Media interactions, the following table shows the predefined IDB fields in the GM_F_USERDATA and GM_L_USERDATA tables in which multimedia-specific attributes are stored. All the IDB fields listed in this table can be used for customer-defined keys.

Important

- In this table, the **Key Name** and **Field** columns refer to Attached Data Specification attributes.
- For any field attributes marked with an asterisk (*), if it is not mapped to a customer-defined key in the attached data specification file, the IDB field will be populated with the value of the predefined key.

Predefined Key Name	Key Name	Field	IDB Field
GM_F_USERDATA Table			
FromPersonal	MyKeyName (customer-defined)	mcr-from-name	G_FROM_NAME
	MyKeyName (customer-defined)	mcr-called-back	G_CALLED_BACK
Subject	MyKeyName (customer-defined)	*mcr-subject	G_SUBJECT
Origination_Source	MyKeyName (customer-defined)	*mcr-origin-source	G_ORIGIN_SOURCE
FromAddress	MyKeyName (customer-defined)	*mcr-from-address	G_FROM_ADDRESS
	MyKeyName (customer-defined)	mcr-reserved-1 through mcr-reserved-4	G_RESERVED1 through G_RESERVED4
GM_L_USERDATA Table			
	MyKeyName (customer-defined)	mcr-suggested-response	G_S_RESPONSE
	MyKeyName (customer-defined)	mcr-auto-response	G_A_RESPONSE
	MyKeyName (customer-defined)	mcr-auto-ack	G_A_ACK
ContactId	MyKeyName (customer-defined)	mcr-ucs-contact-id	G_UCS_CONTACT_ID

Predefined IDB Columns—Custom Fields

ICON creates the IDB `G_CALL_USERDATA_CUST*` fields in the `G_CALL_USERDATA_CUST*` tables for the custom attributes that you might use in your attached data specification. You can use these fields for both voice and multimedia interactions.

- The `G_CALL_USERDATA_CUST*` fields are named `CUST_DATA_1, CUST_DATA_2, CUST_DATA_3,` and so on to `CUST_DATA_19.`

- As needed, you can use the following attribute names, which correspond to the similarly-numbered G_CALL_USERDATA_CUST* fields: cust-data-1, cust-data-2, cust-data-3, and so on to cust-data-19.

[+] Universal Routing Server Attached Data

Universal Routing Server (URS) distributes a standard set of attached data that usually exceeds reporting requirements for actual deployments.

To improve performance and conserve database resources, ICON does not store values for these keys in the IDB history tables by default, regardless of the value that you specify for the adata-userdata-history option (under Attached Data in the callconcentrator section of the Configuration Options page. If you require some or all of the following keys to be stored, explicitly define the respective keys in your attached data specification.

Depending on whether you specify the URS keys in the <public> or <secure> sections of the attached data specification, the KVP data will be stored in the KeyName, Value, and, if you also specify the id attribute, KEYID fields in the G_USERDATA_HISTORY or the G_SECURE_USERDATA_HISTORY table.

For an example of an attached data specification that includes URS attached data keys, see the Basic Sample tab on the Attached Data Specification File page.

> **Important**
>
> As a result of separate ICON processing, the value of any key that is marked with an asterisk (*) in the tables below is stored in the G_ROUTE_RESULT table by default. You must nevertheless include this key in the attached data specification file if you want the key value to be stored in the user data history tables.

source="userdata"		
CBR-Interaction_cost	RTargetAgentGroup	RTargetUsed/RTargetType
CBR-IT-path_DBIDs	RTargetRuleSelected	RTargetAgSelDBID
CBR-actual_volume	*RTargetObjectSelected	CustomerSegment
CBR-contract_DBIDs	*RTargetTypeSelected	RTargetPlSelDBID
RStrategyDBID	*RTargetAgentSelected	RRequestedSkills
ServiceType	*RTargetPlaceSelected	RTargetPlaceGroup
ServiceObjective	*RStrategyName	RTargetCampaignGroup
RVQID	*RRequestedSkillCombination	RouterData70
RTargetObjSelDBID	*RTenant	
RTargetRequested	RTargetUsed/RTargetName	

source="reasons"		
RTR	RTargetObjSelDBID	RTargetUsed/RTargetName
CBR-Interaction_cost	*RTargetRuleSelected	RTargetUsed/RTargetType

source="reasons"		
CBR-IT-path_DBIDs	*RTargetObjectSelected	RTargetAgSelDBID
CBR-actual_volume	*RTargetTypeSelected	CustomerSegment
CBR-contract_DBIDs	*RTargetAgentSelected	RTargetPlSelDBID
RStrategyDBID	*RTargetPlaceSelected	RRequestedSkills
ServiceType	*RStrategyName	RTargetPlaceGroup
ServiceObjective	*RRequestedSkillCombination	RTargetCampaignGroup
RVQID	*RTenant	RouterData70

source="extensions"		
Reasons/RTR	*Reasons/ RRequestedSkillCombination	RTargetUsed/RTargetName
Reasons/ServiceType	*Reasons/RTenant	RTargetUsed/RTargetType
Reasons/ServiceObjective	Reasons/RTargetAgSelDBID	Reasons/RTargetUsed
Reasons/RVQID	Reasons/CustomerSegment	Reasons/RTargetUsed/ RTargetName
Reasons/RTargetObjSelDBID	Reasons/RTargetPlSelDBID	Reasons/RTargetUsed/ RTargetType
Reasons/RStrategyDBID	Reasons/RRequestedSkills	Reasons/CBR-IT-path_DBIDs
*Reasons/RTargetRuleSelected	Reasons/RTargetPlaceGroup	Reasons/CBR-Interaction_cost
*Reasons/RTargetObjectSelected	Reasons/RTargetCampaignGroup	Reasons/CBR-actual_volume
*Reasons/RTargetTypeSelected	Reasons/RouterData70	Reasons/CBR-contract_DBIDs
*Reasons/RTargetAgentSelected	ReportingEventSequenceNumber	Reasons/RTR
*Reasons/RTargetPlaceSelected	Reasons	Reasons/RTargetAgentGroup
*Reasons/RStrategyName	RTargetUsed	Reasons/RTargetRequested

Virtual Queues

Storing Virtual Queue Data and Extended Route Results

This section provides information about the configuration settings in the Genesys Configuration Layer that are related to virtual queue functionality. The default configuration settings enable the storage of virtual queue data, provided that your releases of both Interaction Concentrator and URS support virtual queue functionality.

Configuration settings on the ICON `Application` object, the virtual queue `DN` object, and the `Switch` object enable you to manipulate virtual queue monitoring in the following ways:

- Change the storage mode of Interaction Concentrator.
- Disable monitoring and data storage for a particular virtual queue.
- Disable monitoring and data storage at the switch level—that is, for all virtual queues that belong to a particular switch.

For more information about virtual queue data in Interaction Concentrator, see the chapter about monitoring virtual queues and routing points in the Interaction Concentrator 8.1 User's Guide.

[+] Universal Routing Server

Although a URS release that supports virtual queue functionality is necessary in order to enable virtual queue monitoring in Interaction Concentrator, no special configuration is required on the URS side.

Beginning in release 7.6, URS provides additional information to ICON regarding the reason for routing an interaction using the `AttributeReason` of routing events. URS can also attach information to interactions about the targets for which it is waiting. (For more information, see the section about monitoring route results on routing points in the Interaction Concentrator 8.1 User's Guide.) To make this information available to Interaction Concentrator for downstream reporting purposes, set the following configuration options to `true` on the URS `Application` object:

- `report_reasons`
- `report_targets`

For more information about these URS configuration options, see the Universal Routing Reference Manual.

[+] ICON Application

The default settings enable ICON to receive virtual queue data and store it in IDB.

Connections

Although a URS release that supports virtual queue functionality is necessary in order to enable virtual queue monitoring in Interaction Concentrator, ICON receives the data from T–Server or Interaction Server. Therefore, no connection to URS is required.

vq–write–mode

The `vq–write–mode` configuration option enables you to switch the storage mode for virtual queue data, if necessary. For descriptions of the storage modes, see the chapter about monitoring virtual queues and routing points in the Interaction Concentrator 8.1 User's Guide. You configure the `vq–write–mode` option in the `callconcentrator` section on the `Options` tab of the ICON `Application` object.

extended–route–result

The `extended–route–result` configuration option specifies whether ICON stores extended routing results (from URS) in IDB. You configure `extended–route–result` in the `callconcentrator` section

on the Options tab of the ICON Application object.

> **Important**
>
> - To store extended route results in IDB, ICON requires URS release 7.6 and Interaction Server release 7.6.000.18 (or higher).
> - Interaction Concentrator functionality related to storing Virtual Queues history in the G_ROUTE_RES_VQ_HIST table requires URS release 8.1 or higher.

For more information about these options, see the configuration options under Virtual Queue in the callconcentrator section on the Configuration Options page.

[+] Virtual Queue DN

Unless you need to disable monitoring and data storage for a particular virtual queue, no configuration is necessary for the DN object that represents this virtual queue in the Configuration Layer.

monitor

The monitor configuration option enables you to turn off ICON monitoring and data storage for a particular virtual queue, if necessary. If the option is set to 0, ICON does not register with T–Server to receive events that pertain to this virtual queue. You configure the monitor option in the gts section on the Annex tab of the DN object that is configured for this virtual queue. For more information about this option, see the DN tab on the Configuration Options page.

[+] Switch

Unless you need to disable virtual queue monitoring and data storage for a particular switch, no configuration is necessary for the corresponding Switch object in the Configuration Layer.

support–dn–type–5

The support–dn–type–5 configuration option enables you to turn off ICON monitoring and data storage for all virtual queues that belong to a particular switch, if necessary. If the option is set to 0, ICON does not register with T–Server to receive events that pertain to virtual queue DNs that belong to this switch. In this case, ICON does not process or store virtual queue–related TEvents, even if the monitor option is set to 1 for any of the virtual queues that belong to the switch. You configure the support–dn–type–5 option in the gts section on the Annex tab of the Switch object that is configured for this switch. For more information about this option, see the Switch tab on the Configuration Options page.

Agent States and Logins

Storing Agent State and Login Session Data

In order to store agent state and login session data for voice and multimedia interactions in IDB, certain configuration settings are required in the Genesys Configuration Layer.

To enable ICON to receive agent data and store it in IDB, you must configure ICON connections to appropriate T–Server and Interaction Server instances.

> **Important**
> When ICON terminates a login session as 'stuck'—that is, some issue has made it necessary to terminate the login session without receiving `EventLogout`—all active reason codes related to the terminated session are removed from the `G_AGENT_STATE_RC_A` table (stores active reason codes) and are not transferred to the `G_AGENT_STATE_RC` table (stores reason code history).

[+] ICON Application Configuration

ICON Role
For every ICON instance that must store agent state or agent login session data, make sure that the `role` option on the `Options` tab of the ICON `Application` object includes `gls` in the list of values. If you deploy a single ICON instance for the entire contact center, you can keep the default value (`all`). For more information, see the description of the `role` configuration option under ICON Role in the callconcentrator section on the Configuration Options page.

Other Options
Interaction Concentrator provides a number of options to control reporting on agent login session metrics and agent login sessions—for example, `gls–acw–first` and `gls–active–reason–codes` (see the configuration options under Agent login session metrics and Agent login session in the callconcentrator section of the Configuration Options page). Review the relevant options and set appropriate values for your deployment.

[+] Configure ICON to Use Custom States

In order for ICON to store information to support reporting about custom states and common data, you must do the following:

1. Set appropriate values for the following ICON `Application` configuration options (see the custom-states section on the Configuration Options page):
   - `AgentRecordUserTypes`, which defines the custom agent states
   - `AgentUserFields`

- `EventData`
- `GlobalData`
- `store–event–data`

2. Configure the agent desktop application to send the applicable key–value pairs (KVPs) to T–Server, so that they can be included in the UserData of EventUserEvent, as explained below.

Agent Desktop Application Configuration

ICON records the beginning and end of a custom state, based on information that it receives in the UserData of an EventUserEvent from T–Server. You must configure your agent desktop application to send T–Server the appropriate KVP information for the EventUserEvent UserData.

[+] Start Recording a Custom State

In order for ICON to start recording a custom state, the desktop application must send the following KVP:

```
Key = "<StateKeyName>", Value = "+"
```

Example

```
"AfterCallWork", "+"
```

[+] Send Custom State Data

In order for ICON to store additional information about an active custom state, the desktop application must send the following KVP:

```
Key = "<CommentKey>", Value = "<StateCode>, <Comment>"
```

You can configure more than one comment key for the same custom state. However, for each comment key, ICON can store only one value. If multiple KVPs are sent for the same comment key, ICON stores only the last value.

Example

```
"Comment", "207, This is data about the state" "Explanation", "207, This is more data about the state" "Explanation", "207, This is more, changed data about the state"
```

In this example, ICON will store the following values:

- In the Comment field for state 207: `This is data about the state`
- In the Explanation field for state 207: `This is more, changed data about the state`

[+] Stop Recording a Custom State

In order for ICON to stop recording a custom state, the desktop application must send the following KVP:

`Key = "<StateKeyName>", Value = "–"`

Example

"AfterCallWork", "–"

[+] Use Multiple Custom States at Once

For each type of custom state, only one state can be active for a DN at any one time. However, ICON can simultaneously handle multiple different states independently. For example, two different states can be active on one DN, with different data corresponding to each. ICON does not support duplicate key names in attached data; KVPs with the same key name should not be sent in one EventUserEvent (as shown in the following Example).

Example

```
"AfterCallWork", "=" "Break", "=" "Comment", "207, This is data about the call"
"Comment", "208, This is data about the break" "Break", "–" "AfterCallWork", "–"
```

In the example above, ICON will store the key value only for the custom state `"207, This is data about the call"`.

Outbound Contact

Storing Outbound Contact Data

In order to store Outbound Contact data in IDB for reporting purposes, certain configuration settings are required in the Genesys Configuration Layer, both for certain Outbound-related configuration objects and for the ICON `Application` object.

For more information about Outbound Contact Server (OCS) data in Interaction Concentrator, see the chapter about integrating with Outbound Contact in the Interaction Concentrator 8.1 User's Guide.

Outbound Contact Configuration

Special configuration of the items listed below is required in order to enable OCS to process and send data to ICON about the content of the fields in calling list records.

[+] Field Object

The two Field–level configuration options (described in detail below) control whether ICON will receive and store field values:

- `icon_attribute`
- `send_attribute`

> **Important**
>
> Interaction Concentrator reads `Field` object configuration information only at startup. No real–time configuration changes to `Field` objects are recognized. To accept changes to Field configuration, restart Interaction Concentrator.

icon_attribute

For every `Field` configuration object that describes a single field (for example, a phone number) within a record, you must configure the `icon_attribute` option if you want that data to be stored in IDB.

To configure this option:

1. Open the `Properties` dialog box for the particular `Field` configuration object.
2. Click the `Annex` tab.
3. Create a new section named `default`, if it does not already exist.
4. Within this section, create a new option named `icon_attribute`.
5. Set the option to one of the following values:
   - 1—To store OCS mandatory fields in the `GO_RECORD` table, custom defined fields in the `GO_CUSTOM_FIELDS` table, and history of field changes in `GO_FIELDHIST` table.
   - 2—To store data as a secured field in the special `GO_SECURE_FIELDS` and `GO_SEC_FIELDHIST` IDB tables.

If you do not configure this option, or if you set its value to `0` (zero), OCS will not deliver those fields to ICON when sending reporting information, and ICON will not store the value of such fields.

send_attribute

For every user-defined or mandatory field that describes a single field (for example, a customer name) within a record, you must configure the send_attribute option if you want OCS to attach that data to outbound calls and in user events.

By default, OCS attaches the values of the mandatory fields listed in the table below. The table also shows the default key name for the attached data key–value pair.

Field	Key Name
contact_info	GSW_PHONE
chain_id	GSW_CHAIN_ID

Field	Key Name
attempt	GSW_ATTEMPTS
call_result	GSW_CALL_RESULT

If you do not configure the `send_attribute` option for other mandatory fields and for user–defined fields, OCS does not process data that is related to those `Field` objects, and accordingly ICON does not receive that data.

For more information, see the section about field–level options in the Outbound Contact 8.1 Deployment Guide. See also the section in the Outbound Contact 8.1 Reference Manual about attaching record information to desktop and OCS user events.

[+] Campaign Group Object

To enable reporting for all the activity associated with a Campaign Group, including chain activities, ensure that the Campaign Group object's configuration properties specify a valid Voice Transfer Destination DN. The DN must be located on the switch served by the T–Server to which OCS is connected, and the T–Server must have a CTI link connected with the switch.

[+] Outbound Contact Server

If you require OCS to report snapshot metrics that are based on calculations related to call times (Outbound Call Dialing Time, Outbound Call Transfer Time, and CPD Time), ensure that audit logging is enabled for the OCS `Application` object.

- To enable audit logging, set the `log_call_stats` configuration option to `true` or `yes`.

No other special configuration is required on the OCS Application object.

ICON Application

To enable ICON to receive OCS data and store it in IDB, you must configure ICON connections to appropriate OCS instances, and you must set relevant configuration options.

[+] Connections

In an environment with a single OCS instance, add the OCS `Application` object to the `Connections` tab of the ICON `Application` object.

In an environment with multiple OCS instances, decide on your deployment topology—that is, decide whether a single ICON instance will handle the data from all or a subset of OCS instances, or whether each OCS will have a dedicated ICON instance. Based on your deployment decision, add one or more OCS `Application` objects to the `Connections` tab of the ICON `Application` object that must store data from those OCS instances.

[+] ICON Role Configuration Option

For every ICON instance that must store outbound data, make sure that the `role` option on the `Options` tab of the ICON `Application` object includes `gos` in the list of values. If you deploy a single ICON instance for the entire contact center, you can keep the default value (`all`). For more information, see the description of the `role` configuration option under ICON Role in the callconcentrator section of the Configuration Options page.

If you store different types of data to different IDBs, make sure that `gos` is also specified for the `role` option on the `Options` tab of the appropriate Database Access Point (DAP). Configure this option on the `Options` tab of the `Application` object for the DAP that your ICON instance uses to store outbound data to IDB.

[+] OCS–Specific ICON Options

The following ICON configuration options enable you to specify what outbound data ICON should store, and in what manner:

- `gos-write-duplicate-metrics`
- `gos-write-metrics`
- `gos-write-metrics-only`

Review the descriptions and values for the `gos-write-*` configuration options (under Outbound metrics in the callconcentrator section on the Configuration Options page). Select the appropriate values for your environment, and then make the appropriate configuration changes on the `Options` tab of the ICON `Application` object.

Multi-Tenant Considerations

In multi-tenant environments, the OCS-related objects that the ICON instance monitors may be configured under various tenants. Ensure that you assign all related tenants to the ICON application.

[+] Multi-Tenant Example

For example, you might create an Outbound `Calling List` object under a tenant called `Outbound`, and have the calling list use fields that you created as `Field` objects under the `Environment` tenant. To enable ICON to process OCS data related to the Outbound `Calling List`:

1. Configure the required `Field` objects under the `Environment` tenant.
   - Configure the `icon_attribute` option for all the fields that you want ICON to store.
   - Configure the `send_attribute` option for all the fields that you want ICON to store.
2. Add both the `Environment` tenant and the `Outbound` tenant on the `Tenants` tab of the ICON `Application`.

License Reporting Manager

Storing License Reporting Data

If you are running Genesys License Reporting Manager (LRM), ICON enables you to store your LRM-specific data.

- This section explains the ICON-specific aspects of configuring ICON to work with LRM. For detailed information about LRM, see the Genesys License Reporting Manager documentation.

Configuring ICON to work with LRM requires that you set the appropriate value for the ICON role configuration option.

[+] Configuring ICON to store LRM data

To configure ICON, perform the following steps:

1. Designate the instance of ICON that will be used for LRM.
2. Set the value of the `role` option to `lrm`.
   Ideally, the ICON you use for LRM data should not have any other role. If necessary, however, you can set the LRM instance of ICON for both the `lrm` and `cfg` (Configuration data) roles.
3. Start (or if applicable, restart) ICON to have the `role` setting take effect.

[+] Configuration Notes

- If the `role` value is set to `all`, ICON stores LRM data. However, if you require *only* LRM data, setting the value to `all` results in the accumulation of large quantities of unusable data. Genesys recommends that you explicitly set the value to `lrm` to collect License Reporting data.
- Role assignments must be configured using only lower case (for example, `lrm`). ICON interprets uppercase (LRM) or mixed case (Lrm) settings as invalid and defaults to the `all` role.
- When `role=lrm`, ICON:
  - Does not write data for `gcc` or `gud` providers.
  - Disregards data filtering options.
  - Enforces having the `use–dss–monitor` option set to `true`.
- You can switch ICON to or from the `lrm` role at any time by changing the setting for the `role` option. Restart ICON to have the change take effect.

Important

It is not advisable to change roles without careful planning. ICON stores the data associated with a role only when it is configured with that role. For example, if you set an instance of ICON to collect LRM data, then change the role so it is no longer set to `lrm`, and then later change it back again, you will probably have a window of time during which there is no LRM data stored because the previous role may not have required ICON to collect the data necessary for LRM reporting.

- If you are using Genesys Info Mart, do not try to connect it to the ICON and the associated IDB that stores the LRM data. The LRM–specific ICON–IDB set stores data in a specific subset of tables. As a result, Genesys Info Mart will fail to start when it finds the tables from which it extracts data to be empty.
- HA is supported just as for any other ICON role.

Partitioned IDB

Deploying a Partitioned Oracle IDB

In environments with large amounts of data to maintain, you can choose to create a partitioned Oracle IDB, which you can then purge efficiently by truncating entire partitions using the `purgePartitions811` stored procedure. During this purge, all records in the purged partitions—both terminated and non-terminated—are truncated unconditionally.

Important

- If you need to purge only non-terminated records, use the `GSYSPurge80/gsysPurge81` purge procedure instead.
- This partitioning and purge functionality is supported only for Oracle 11g and higher.
- Genesys strongly recommends that you do not use this purge mechanism for long–lived data types, such as multimedia. When used with long–lived data types, you might encounter situations in which some of the data for a still–active interaction is purged.

[+] Overview

The procedure for deploying the purge–by–truncating–partitions functionality is outlined below.

> **Warning**
> Do not try to change the number of partitions without consulting your Genesys representative for guidance.

1. Start with a new Oracle database. There is no migration from a non-partitioned to a partitioned IDB.
2. Determine the number of partitions necessary for your environment. When you run the SQL scripts to initialize your database, the number of partitions is set and cannot be changed afterwards.
   The default number of partitions is fourteen. In the majority of cases, fourteen partitions is expected to be satisfactory. In the rare case that you believe you require a different number of partitions, discuss your requirements with your Genesys representative.
3. Run the SQL scripts to create your partitioned IDB, using the standard procedure given on the IDB tab on the Configuration and Installation page. However, the scripts used to create a partitioned IDB differ from those used for non–partitioned IDBs, so be sure to see “Creating Your Database” (below) for a list of the correct scripts.
4. Have an instance of Interaction Concentrator write data to the existing (non–partitioned) IDB and a redundant instance that collects identical data write to the new (partitioned) IDB until you are certain that the partitioned IDB contains all the data you require. At that point, you can transition entirely to the partitioned IDB.

[+] Creating Your Database

To create a partitioned Oracle IDB, follow the standard instructions on the IDB tab on the Configuration and Installation page, but run the following scripts rather than the scripts used for a non–partitioned IDB. These two initialization scripts create a new partitioned IDB:

- `CoreSchemaPart_ora.sql`
- `CoreProcedures_ora.sql`

The following initialization script sets up the stored procedure used to purge the partitioned Oracle IDB:

- `PurgePart_ora.sql`

As noted above, there is no migration path from a non–partitioned to a partitioned database. To migrate, Genesys recommends running your non–partitioned and partitioned databases in parallel until all required data appears in the partitioned database.

[+] About Partitioning

By default, the number of partitions is fourteen, with each partition equivalent to one day. Data is written into the partitions in sequence, starting with Partition 1 on Day 1, Partition 2 on Day 2, and so on, circling back to Partition 1 on Day 15.

As with all purge methods, only operational tables are purged. Special tables used for internal data storage and retrieval are neither partitioned nor purged.

The tables that are available for partitioning include the `gsys_partition` field, which must be configured to contain the UTC value taken from the `created_ts` field. This parameter is set using the `partition-type` configuration option (see it under Partitioning in the callconcentrator section of the Configuration Options page).

Each partitioned table also includes the virtual `GSYS_SHORT_DAY` column, based on value of the `gsys_partition` field.

[+] Purging

You perform the purge by executing the `purgePartitions811` stored procedure, which truncates all partitions except for the number you specify in the `days–to–keep` parameter of the SQL statement.

Instructions for how to run the `purgePartitions811` procedure, how to schedule it, and all other operational considerations are documented in the "Purging by Truncating Partitions" section the "Using Special Stored Procedures" chapter in the Interaction Concentrator 8.1 User's Guide.

[+] Partitioned Tables

The following tables are partitioned by the `CoreSchemaPart_ora.sql` script:

G_AGENT_STATE_HISTORY	G_IR_HISTORY	GO_CHAINREC_HIST
G_AGENT_STATE_RC	G_IS_LINK	GO_CUSTOM_FIELDS
G_CALL	G_IS_LINK_HISTORY	GO_FIELDHIST
G_CALL_HISTORY	G_LOGIN_SESSION	GO_METRICS
G_CALL_STAT	G_PARTY	GO_RECORD
G_CALL_USERDATA	G_PARTY_HISTORY	GO_SEC_FIELDHIST
G_CALL_USERDATA_CUST	G_PARTY_STAT	GO_SECURE_FIELDS
G_CALL_USERDATA_CUST1	G_ROUTE_RESULT	GOX_CHAIN_CALL
G_CALL_USERDATA_CUST2	G_USERDATA_HISTORY	GS_AGENT_STAT
G_CUSTOM_DATA_P	G_VIRTUAL_QUEUE	GS_AGENT_STAT_WM
G_CUSTOM_DATA_S	GM_F_USERDATA	GX_SESSION_ENDPOINT
G_CUSTOM_STATES	GM_L_USERDATA	G_ROUTE_RES_VQ_HIST
G_DND_HISTORY	GO_CAMPAIGNHISTORY	G_SECURE_USERDATA_HISTORY
G_IR	GO_CHAIN	

High Availability

Configuring for High Availability

The High Availability (HA) model used in Interaction Concentrator differs significantly from the Genesys standard HA model implemented in a majority of Genesys servers. Before you configure your ICON HA deployment, review the information in the Interaction Concentrator 8.1 User's Guide about implementing HA in Interaction Concentrator.

In an HA deployment, observe the following rules:

- You must set configuration options in both Interaction Concentrator `Application` objects exactly the same. Because this is not a typical redundant pair from the Genesys perspective, Configuration Server does not automatically synchronize the configuration options for two ICON applications.
  For example, to configure your redundant ICON applications to store voice interaction data in a pair of HA IDBs:
  - In both ICON Application objects, set the `role` option so that it contains `gcc` and `gud`. This enables both ICON applications to store call–related and attached data.
  - For any configuration options that affect the data populated by those roles, set the same option values in both ICON applications. For example, the two applications must use the same ICON configuration options for virtual queue monitoring, storage of attached data, and so on.
- For more information about setting configuration options, refer to the other pages in this section.
- You must configure a connection to the same T–Server or Interaction Server in both ICON Application objects.
- You must create two identical IDBs. Genesys recommends using two databases located on different hosts, but having the same RDBMS type and version number, to host the HA pair of IDBs.
- You must configure a DAP for each ICON to access its IDB.

For more information about configuring applications and connections in Configuration Manager, see the Framework Deployment Guide.

Configuration Options

This section describes the configuration options that you can set for effective operation of your Interaction Concentrator (ICON) Application object.

Interaction Concentrator related options are set in the following Application objects:

- Interaction Concentrator
- Switch
- DN
- Script
- DAP

> **Important**
> In addition to the configuration options described in this chapter, Interaction Concentrator supports the common log options that are described in the Framework Configuration Options Reference Manual.

Interaction Concentrator Options

This section describes the options that you define on the Options tab of your Interaction Concentrator (ICON) Application object.

You do not have to configure any options in order to start Interaction Concentrator. However, the available Configuration options can greatly affect the performance and effectiveness of your Interaction Concentrator application. Review the information about all of the configuration options for the ICON Application object in order to identify and configure the settings that are appropriate to your environment.

Configuration Server recognizes the following sections for ICON Application objects:

- callconcentrator Section
- custom-states Section
- dbw-error-reactions Section
- filter-data Section
- listeners Section
- user_named_section Section
- log Section

callconcentrator Section

For ease of reference, the options in this section are grouped by area of functionality (presented in alphabetical order) as follows:

[+] Agent login session metrics

gls-acw-first Option
gls-stats-update Option
gls-stats-update-delta Option
gls-store-event-seq Option

gls-acw-first

Specifies which interaction ICON associates with after-call work (ACW). This option is configured in the ICON Application, or in the Switch Application, or both. If it is set only in the ICON Application, it applies to all switches ICON is configured to monitor. If any Switch is set with a value different from that set in the ICON Application, the Switch value takes precedence.

By default, ICON associates after-call work metrics with the voice interaction that immediately precedes the completion of the after-call work (the last voice interaction).

Setting this option to `true` enables ICON to associate after-call work with the voice interaction that most recently changed the agent's state from NotBusy to Busy (the first voice interaction). In this case, subsequent voice interactions that occur during the period of after-call work are considered as related to ACW processing and should not interrupt measurement of ACW-related metrics.

When the agent logs out, changes his or her state to Ready, or goes NotReady for any reason other than to perform after-call work, ICON reports the end of the current ACW state.

- Configured in: ICON Application, `[callconcentrator]` Section; Switch Application, Annex tab, `[gts]` Section

ICON Application Settings:

- Default value: `false`
- Valid values:
  - `false`—ICON associates the last voice interaction with after-call work.
  - `true`—ICON associates the first voice interaction with after-call work.
- Changes take effect: After restart

Switch Application Settings:

> **Important**
>
> To associate the first ACW value, specify the value of this option on the Switch Application. A change to the setting of this option on the ICON Application does not propagate to SIP switches.

- Default value: `-1`
- Valid values:
  - `-1`—ICON uses the value of the `gls-acw-first` option specified in the ICON Application object. If no value is set at the application level, ICON associates the last voice interaction with after-call work.
  - `0`—ICON associates the last voice interaction with after-call work.
  - `1`—ICON associates the first voice interaction with after-call work.

- Changes take effect: After restart

> **Important**
>
> For SIP switches, the default value results in the same functionality as setting the option to `0`.

gls-stats-update

Specifies whether agent metrics (such as the duration of a particular agent state) are updated in the GS_AGENT_STAT and GS_AGENT_STAT_WM IDB tables as the agent login session progresses. By default, ICON stores agent metrics only after an agent login session ends.

- Configured in: ICON Application, [callconcentrator] Section
- Default value: false
- Valid values:
  - true—Agent metrics (such as a state duration) are updated dynamically in IDB.
  - false—Agent metrics (such as a state duration) are stored in IDB after a login session ends.
- Changes take effect: After restart

gls-stats-update-delta

Specifies the minimum change, in seconds, in the duration of an agent state that causes an updated metric's value to be stored in IDB. ICON processes this option only if you set the `gls-stats-update` option to `true`.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `10`
- Valid values: Any integer
- Changes take effect: After restart

gls-store-event-seq

Specifies whether ICON stores event sequence numbers when events related to an agent login session trigger creation of new records in the following IDB tables:

- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- G_DND_HISTORY

By default, ICON retrieves event sequence numbers from T-Server or Interaction Server events and stores the numbers along with new records in the above tables. To provide event sequence numbers with multimedia events, Interaction Server release 7.6 or higher is required.

- Configured in: ICON Application, [callconcentrator] Section
- Default value: 1
- Valid values:
  - 0—ICON does not store a sequence number of the event that triggered a new record in an agent-related table.
  - 1—ICON stores a sequence number of the event that triggered a new record in an agent-related table.
- Changes take effect: After restart

[+] Agent login session

gls-active-reason-codes Option
gls-enforce-reason-code Option

gls-active-reason-codes

Specifies whether ICON captures and stores the values of active Agent state reason codes.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `false`
- Valid values:
  - `true`—ICON captures active Agent state reason codes, and temporarily stores the values in the `G_Agent_State_RC_A` table in IDB. When the reason code is terminated, ICON deletes the record from the `G_Agent_State_RC_A` table, and it creates a new record for the terminated reason code in the `G_Agent_State_RC` table, which stores the values of reason codes that have been changed or terminated.
  - `false`—ICON stores information about Agent state reason codes only when the reason code is changed or terminated. The information is stored in the `G_Agent_State_RC` table.
- Changes take effect: After restart

gls-enforce-reason-code

Enables you to control whether software(SW) and hardware (HW) reason code changes are processed separately for separate devices in multi-device login sessions. (Multi-device login sessions refers to scenarios in which an agent logs in to a DN and to one or more queues at the same time.)

Specifies whether changes to the HW reason code or the SW reason code for a particular device will affect the HW reason code, SW reason code, or both types of reason code on all other devices on which an agent is logged in.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—Both types of reason code changes are processed independently for each device. A new HW or SW reason code does not terminate the previous HW and SW reason codes for all other devices. For example, for an agent logged in to DN1 and Queue1, a change of HW reason code on Queue1 does not affect the SW reason code on Queue1 or either type of reason code on DN1.
  - 1—Only HW reason code changes are enforced on all devices. A new HW reason code becomes active on the device for which it is reported and terminates the previous HW reason codes for all other devices.

    For example, a change of HW reason code on Queue1 terminates the HW reason code on DN1, but it does not affect the SW reason code on Queue1 or DN1; by contrast, a change of SW reason code on Queue1 does not affect the SW reason code on DN1 or the HW reason code on any device.
  - 2—Only SW reason code changes are enforced on all devices. A new SW reason code becomes active on the device for which it is reported and terminates the previous SW reason codes for all other devices.
  - 3—HW or SW reason code changes are enforced on all devices. A new HW reason code becomes active on the device for which it is reported and terminates all previous HW and SW reason codes for all other devices; similarly, a new SW reason code becomes active on the device for which it is reported and terminates all previous HW and SW reason codes for all other devices. A simultaneous change of both HW and SW reason codes on a device makes only the new SW reason code active on the device for which these reason codes are reported, terminates the HW reason code on this device, and terminates all previous HW and SW reason codes for all other devices.

    For example, a change of HW reason code on Queue1 terminates the HW reason code on DN1 and also terminates the SW reason codes on Queue1 and DN1.
- Changes take effect: After restart

[+] Attached data

adata-default-storage Option

adata-default-storage

Specifies the default destination for storing attached data, for a key that is not included in the XML specification file specified by the `adata-spec-name` option value. ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or gud.

This option applies to voice and multimedia interactions.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `public`
- Valid values: `public, secure`
- Changes take effect: After restart

> **Important**
>
> For descriptions of these values, see “Storage Types” on the Attached Data tab of the Special Configuration Requirements page.

adata-extensions-history

Specifies what changes to a key's value must be recorded in IDB for a key that originates from the Extensions TEvent attribute but that is not included in the XML specification file specified by the `adata-spec-name` option value. ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud`.

This option applies to voice and multimedia interactions.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `none`
- Valid values:
  - `none`—No value for a given key is recorded in IDB.
  - `first`—Only the first value for a given key is recorded in IDB.
  - `last`—Only the last value for a given key is recorded in IDB.
  - `all`—Every change in value for a given key is recorded in IDB.
- Changes take effect: After restart

adata-reasons-history

Specifies what changes to a key's value must be recorded in IDB for a key that originates from the Reasons TEvent attribute but that is not included in the XML specification file specified by the `adata-spec-name` option value. ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud`.

This option applies to voice interactions only.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `none`
- Valid values:
  - `none`—No value for a given key is recorded in IDB.
  - `first`—Only the first value for a given key is recorded in IDB.
  - `last`—Only the last value for a given key is recorded in IDB.
  - `all`—Every change in value for a given key is recorded in IDB.
- Changes take effect: After restart

adata-spec-name

Specifies the name of the XML file that contains the specification of attached data. ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or gud.

For more information about the specification, see Attached Data Specification File.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `ccon_adata_spec.xml`
- Valid values: Any valid name
- Changes take effect: After restart

adata-userdata-history

Specifies what changes to a key's value must be recorded in IDB for a key that originates from the UserData reporting event attribute, but that is not included in the XML specification file specified by the `adata-spec-name` option value. ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud`.

This option applies to voice and multimedia interactions.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `none`
- Valid values:
  - `none`—No value for a given key is recorded in IDB.
  - `first`—Only the first value for a given key is recorded in IDB.
  - `last`—Only the last value for a given key is recorded in IDB.
  - `all`—Every change in value for a given key is recorded in IDB.
- Changes take effect: After restart

cseq-adjustment

This option is available in release 8.1.000.37 and higher. Improves sequence tracking for user data in the G_USERDATA_HISTORY table,which enables downstream reporting applications, such as Genesys Info Mart, to correctly associate user data with interaction activity when user data updates occur within the same second that the call is transferred or terminated.

To preserve compatibility with legacy behavior, cseq-adjustment enables you to control whether ICON will implement the improved behavior regarding populating the CSEQ field in the G_USERDATA_HISTORY table.

- Configured in: ICON Application, [callconcentrator] Section
- Default value: 0
- Valid values:
  - 0—Compatibility mode. Preserves legacy behavior, which includes the following known issue:

The value set in the CSEQ field in the G_USERDATA_HISTORY table sometimes behaves inconsistently. In some scenarios, the last CSEQ value is recorded, in other scenarios the next CSEQ value is recorded. This prevents Genesys Info Mart from consistently associating user data with the correct INTERACTION_RESOURCE_FACT if both a user data update and the end of the IRF record occur during the same second. (ER# 312034811)

-
  - 1—Preserves legacy behavior, but corrects the known issue. ICON stores correct values in the CSEQ field in the G_USERDATA_HISTORY table for all scenarios.
  - 2—Compatibility mode for Genesys Info Mart 8.x releases. ICON modifies the value of the CSEQ field in the G_USERDATA_HISTORY table to match the behavior that Genesys Info Mart 8.x expects.
- Changes take effect: Immediately

suppress-user-data

Specifies whether the switch instructs T-Server to propagate attached data only when the attached data changes. This optimizes ICON processing of attached data by reducing network traffic.

This option can be set at the level of the Switch or the ICON application. ICON automatically detects the Switch-level option setting. If the Switch-level option is set to the (default) value of 1 (unchanged attached data suppressed), T-Server TEvents are optimized for all ICON applications that connect to the T-Servers for that Switch. In this case, the Switch-level option setting overrides any ICON-level settings of `0` (unchanged attached data not suppressed). If the Switch-level option is set to `0`, an application-level setting of 1 will override it.

- Configured in: ICON Application, `[callconcentrator]` Section; Switch Application, Annex tab, `[gts]` Section
- Default value: 1
- Valid values:
  - `0`—Unchanged attached user data is not suppressed.
  - 1—Unchanged attached user data is suppressed.
- Changes take effect: After restart

[+] Configuration

cfg-annex Option

cfg-annex

Enables ICON to store data in the GC_ANNEX table, which stores changes to certain options on the Annex tabs of Person, Agent Group, DN, DN Group, and Switch objects. This data enables Genesys Interactive Insights 8.1.4 and higher to control visibility of certain data and reports based on attributes such as geographical location, business line, or organization structure. This data is stored only when ICON has the `cfg` role and the value for this option set to 1.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—ICON does not process changes to the specified Annex tab options.
  - 1—ICON processes changes to the specified Annex tab options and stores the data to IDB.
- Changes take effect: After restart

[+] Custom dispatcher

gud-cust-disp Option
gud-cust-disp-groups Option

gud-cust-disp

Specifies whether ICON calls a custom stored procedure to handle attached data and store the information in custom tables. ICON starts executing the new custom dispatcher as soon as the new configuration option value is set. Processing of interaction information stored in the persistent queue that was begun by the old custom dispatcher is handled in IDB by the old custom dispatcher.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—ICON does not call a custom dispatcher.
  - `1`—ICON calls the `gudCustDisp1` stored procedure.
  - `2`—ICON calls the `gudCustDisp2` stored procedure.
- Changes take effect: Immediately

> **Important**
>
> For more information, refer to the section about custom dispatchers in the Interaction Concentrator 8.1 User's Guide.

gud-cust-disp-groups

Specifies the maximum number of key groups that ICON can process. If you code more than the maximum number of groups in the XML file, ICON ignores the extra key groups and does not provide data to the active custom dispatcher.

Key names that you specify must be unique both within and across key groups. The maximum number of keys that you can specify for any particular key group is limited to 34 (17 key-value pairs for string values, and 17 for integer values).

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: 16
- Valid values: `0`–255 (`0` indicates that ICON will not process any group)
- Changes take effect: After restart

[+] Database writing

dbw-request-tout Option
dbw-seq-step Option
dbw-seq-tout Option

dbw-request-tout

Specifies the amount of time, in seconds, that ICON waits for the completion of a database writing transaction. If a transaction is not completed when this interval expires, ICON generates an error message and forces the transaction to be rolled back.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `600`
- Valid values: Any integer
- Changes take effect: Immediately

Examples:

- `dbw-request-tout = 30`
- `dbw-request-tout = 120`

dbw-seq-step

Specifies the reservation value that ICON uses when updating the counter in the SEQCOUNTER field of the G_PROV_CONTROL table. At startup, ICON reads the initial counter value (*M*) from the G_PROV_CONTROL table, increments the counter in every database transaction, and writes the new value into the GSYS_SEQ or GSYS_USEQ field of the tables that are participating in the transaction.

ICON updates the value of the SEQCOUNTER field in the G_PROV_CONTROL table as follows:

1. During the first database transaction after startup, ICON inserts the sum (*L*=*M*+*N*) of the initial counter value (*M*) and the reservation value specified by the dbw-seq-step option (*N*). For example, if the initial value that ICON retrieves at startup is 700, and if you keep the default value of 500 for the dbw-seq-step option, ICON writes 1200 during the first transaction.
2. During the next *N–1* transactions, ICON does not update the G_PROV_CONTROL table, but updates only those tables that are participating in the transactions.
3. During the *Nth* transaction, ICON inserts into the G_PROV_CONTROL table a new value (*K*=*L*+*N*) that is the sum of the current counter value (*L*) set in Step 1 and the reservation value (N). Continuing the example from Step 1, during the 500th transaction, ICON writes the new counter value of 1700.
4. During each subsequent *Nth* transaction, ICON uses the same logic to update the value of the SEQCOUNTER field in the G_PROV_CONTROL table.

- Configured in: ICON Application, [callconcentrator] Section
- Default value: 500
- Valid values: Any integer
- Changes take effect: After restart

Examples:

- dbw-seq-step = 100
- dbw-seq-step = 300

dbw-seq-tout

Specifies the amount of time, in seconds, after which ICON writes the current value of the transaction counter to the `G_PROV_ CONTROL.SEQCURRENT` field. The ICON merge procedure relies on this field for the detection of newly-updated records. (For more information, see the section about the merge stored procedure in the Interaction Concentrator 8.1 User's Guide.)

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `60`
- Valid values: Any integer
- Changes take effect: Immediately

Examples:

- `dbw-seq-tout = 30`
- `dbw-seq-tout = 120`

[+] High Availability

use-dss-monitor Option
dss-no-data-tout Option

use-dss-monitor

This option controls whether user data and call-termination timestamps are stored independently in IDB.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `false`
- Valid values:
  - `true`—ICON does not synchronize user data and call-termination timestamps in IDB. As a result, user data and call-termination data are stored independently in IDB.
  - `false`—ICON synchronizes user data with call-termination data or the call-termination data is updated only after user data is stored in IDB. Also, ICON does not write data to the `G_DSS_*_PROVIDER` tables.
- Changes take effect: After restart

> **Important**
>
> If you want the `G_DSS_*_PROVIDER` tables to be populated, you must set the value to `true`.

dss-no-data-tout

Specifies the time interval in seconds after which, if no new data has been written to the persistent queue, Interaction Concentrator creates a “no data” record for the applicable provider and updates the `NODATA_IUTC` field in the applicable `G_DSS_*_PROVIDER` table.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `300`
- Valid values: Any integer in the range of `60` to `86400`
- Changes take effect: After restart

The NoData indication enables you to distinguish cases in which there was no data from those in which a connection problem prevented the data from being properly recorded.

[+] HTTP Connection

http-protocol-enabled Option

http-protocol-enabled

Enables or disables the HTTP connection to the main ICON port.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `true`
- Valid values:
  - `true`—The HTTP connection on the main ICON application port is enabled. This is the same functionality as that available in releases prior to 8.1.x.
  - `false`—The HTTP connection on the main ICON application port is disabled.
- Changes take effect: After restart

[+] ICON Role

role Option

role

Specifies the type of data that this ICON instance processes and stores in IDB.

> **Important**
> Role assignments must be configured using only lower case (for example, `cfg`). ICON interprets uppercase (`CFG`) or mixed case (`Cfg`) settings as invalid and defaults to the `all` role.

- Configured in: ICON Application, `[callconcentrator]` Section; DAP Application, `[callconcentrator]` Section (for details relevant to setting the `role` option in the DAP Application, scroll to the end of this page)
- Default value: `all`
- Valid values: A comma-separated list including any of the following:
  - `all`—Stores all types of data.
  - `cfg`—Stores the initial configuration state and a history of configuration changes retrieved from Configuration Server.
  - `gcc`—Stores interaction-related and party-related information—that is, T-Server and Interaction Server data that pertains to voice and multimedia interactions, and the parties associated with those interactions.
  - `gls`—Stores T-Server and Interaction Server data that pertains to agent states and agent login sessions.
  - `gud`—Stores T-Server and Interaction Server data that pertains to the attached data associated with calls.
  - `gos`—In an environment with the Outbound Contact solution, stores OCS data that pertains to outbound calls and campaigns.
  - `lrm`—In an environment with License Reporting Manager, stores license reporting data.
- Changes take effect: After restart

> **Important**
> All ICON instances are assigned a predefined role, `svc`, to store service information about the ICON instance, for identification purposes, in IDB. The `svc` role cannot be turned off, and you do not need to specify it in the option value.

Any combination of the valid values can be used. Prefixing an option value with a tilde (~) excludes that type of data from ICON processing, and includes all other types. For example, the value `~cfg` deactivates ICON processing of configuration data, and activates processing and storage of all other

types of data. Ensure that the role that you specify for the ICON instance is consistent with the role that you specify for the DAP (see below for DAP-specific considerations).

Examples of correct settings:

- `role = cfg,gcc,gud`
- `role = all`
- `role = gcc,gud,gls,gos`
- `role = ~cfg`

(The last two examples are equivalent.)

DAP-Specific Notes
All types of ICON data go through the same DAP in the following cases:

- No `role` option is defined for the DAP.
- The `role` option is defined, and its value is explicitly set to `all`.
- You specified only one DAP Application object on the **Connections** tab of the ICON Application object.

> **Important**
>
> - Regardless of whether a given DAP handles all types of ICON data or a subset of them, a separate database connection is opened for each type of data.
> - Ensure that the role that you specify for the DAP is consistent with the role that you specify for the ICON instance.
> - A DAP cannot be assigned the `lrm` role. If you do so, it is ignored and the default value (`all`) is used.

[+] IDB

db-schema-name Option

db-schema-name

Specifies the database schema name ICON will use when the RDBMS requires an explicit schema name to be specified when executing stored procedures. For information about what the term schema name means and for any delimiters that the RDBMS syntax requires, see the vendor documentation for your RDBMS.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: Empty
- Valid values: Any string
- Changes take effect: After restart

[+] In-memory queue

acc-proc-tout Option
acc-queue-lifespan Option
acc-queue-size Option

acc-proc-tout

Specifies the interval, in milliseconds, at which ICON scans its in-memory queue in order to determine whether the timeout set by the `acc-queue-lifespan` option has expired.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `3000`
- Valid values: Any positive integer
- Changes take effect: Immediately

acc-queue-lifespan

Specifies the interval, in seconds, at which ICON accumulates records in its in-memory queue before writing them to a persistent queue (as the first stage of serialization). The process of writing to a persistent queue is triggered when the limit set either by this option or by the `acc-queue-size` option is exceeded.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: 5
- Valid values: Any positive integer
- Changes take effect: Immediately

acc-queue-size

Specifies the maximum number of serialization records that ICON keeps in the in-memory queue before writing them to a persistent queue (as the first stage of serialization). The process of writing to a persistent queue is triggered when the limit set either by this option or by the `acc-queue-lifespan` option is exceeded. This option also defines the size of a database-writing transaction.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `500`
- Valid values: Any positive integer
- Changes take effect: Immediately

[+] Multimedia

calls-in-the-past Option
om-force-adata Option
mcr-om-processing Option

calls-in-the-past

Specifies whether ICON stores data for multimedia interactions that begin while ICON is down, or while ICON has no connection to Interaction Server. The data stored for reconstructed interactions is the same as the data stored for the interactions that ICON tracks from their beginning.

> **Important**
>
> ICON cannot restore a correct timestamp of interaction record creation, or the information about previous parties, or the first values of user data keys.
>
> In releases from 8.1.000.14 through 8.1.100.34, setting the `om-memory-optimization` option to `true` causes old interactions to be re-created in spite of setting the `calls-in-the-past` option to `false`. In release 8.1.100.36 and higher, the `calls-in-the-past` option is no longer overridden by setting the `om-memory-optimization` option to `true`.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `false`
- Valid values:
  - `true`—ICON reconstructs operational data about a multimedia interaction that is already in progress when ICON receives one or more of the following reporting events from Interaction Server:
    - EventPlacedInQueue
    - EventPlacedInWorkbin
    - EventAgentInvited
    - EventPartyAdded
  - `false`—ICON does not record data for multimedia interactions that begin while ICON is down, or while ICON has no connection to Interaction Server.
- Changes take effect: After restart

om-force-adata

For deployments that have been configured to report data for multimedia interactions that started in the past (the `calls-in-the-past` configuration option has been set to `true`), specifies whether ICON stores a UserData snapshot that corresponds to the interaction-related data.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `false`
- Valid values:
  - `true`—If the `calls-in-the-past` configuration option has also been set to `true`, ICON stores a UserData snapshot in the `GM_F_USERDATA` table for interactions created in the past.
  - `false`—If the `calls-in-the-past` configuration option has been set to `true`, ICON does not store a UserData snapshot when it restores a Multimedia interaction that was created in the past.
- Changes take effect: After restart

> **Important**
>
> When the first event relative to the interaction is EventProcessingStopped, ICON does not restore the interaction. Nevertheless, if the `om-force-adata` option is set to `true`, ICON stores the data in the `GM_F_USERDATA` table.

mcr-om-processing

Specifies whether ICON stores information about 3rd Party Media interactions in IDB. By default, ICON processes interactions other than chat, e-mail, or voice and stores the type of media in special fields of the following tables:

- GX_SESSION_ENDPOINT
- G_AGENT_STATE_HISTORY
- GS_AGENT_STAT
- G_AGENT_STATE_RC
- G_CALL

When this option is set to 0, ICON processes neither interactions nor agent data for 3rd Party Media.

- Configured in: ICON Application, [callconcentrator] Section
- Default value: 1
- Valid values:
  - 0—ICON does not store data in IDB about interactions other than chat, e-mail, or voice.
  - 1—ICON stores information in IDB about 3rd Party Media interactions.
- Changes take effect: After restart

> **Important**
>
> For more information about 3rd Party Media support, refer to the chapter about integrating with Genesys eServices/Multimedia and 3rd Party Media in the Interaction Concentrator 8.1 User's Guide.

[+] Operational memory

om-check-filter-flag Option
om-max-in-memory Option
om-memory-optimization Option

om-check-filter-flag

Specifies whether or not ICON stores strategy activity according to the value of the `om-activity-report` configuration option that is defined in the Script object (of type simple routing). If the value is set to `0`, ICON stores all strategy activity regardless of the value of the `om-activity-report`.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `1`
- Valid values: `0, 1`
- Changes take effect: After restart

om-max-in-memory

Specifies the maximum number of keep-in-memory interactions placed in queues or interaction workbins (in units of one thousand).

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `100`
- Valid values: `1–2,000` (in units of one thousand)
- Changes take effect: After restart

Keep the default value unless you are advised otherwise by Genesys Customer Care. If you need to change the option's value, use the following formula to calculate an approximate value for this option:

- `Size of available operational memory (K)/((1,000 + size of user data (K))*2)`
  where:

  size of user data = average size of the interaction user data that is attached to the interaction in Interaction Server.

> **Warning**
> An incorrect value for this option can affect ICON performance, or cause ICON to stop processing interactions.

om-memory-optimization

Specifies whether memory usage will be optimized.

> **Important**
> In releases from 8.1.000.14 through 8.1.100.34, setting the `om-memory-optimization` option to `true` causes old interactions to be re-created in spite of setting the `calls-in-the-past` option to `false`. In release 8.1.100.36 and higher, the `calls-in-the-past` option is no longer overridden by setting the `om-memory-optimization` option to `true`.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `false`
- Valid values:
  - `true`—ICON optimizes memory usage according to the values that are set for the `om-max-in-memory` and `om-memory-clean` options.
  - `false`—Preserves legacy behavior (prior to ICON release 7.6.1).
- Changes take effect: After restart

[+] Outbound metrics

gos-write-duplicate-metrics Option
gos-write-metrics Option
gos-write-metrics-only Option

gos-write-duplicate-metrics

Specifies whether all metrics related to active outbound objects are stored in IDB exactly as Outbound Contact Server (OCS) provides them, or whether ICON filters out duplicate metrics. ICON identifies active outbound objects by CampaignGUID, ChainGUID, and CallAttemptGUID.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—ICON does not subsequently write the same precalculated OCS metric after it is stored in IDB.
  - 1—ICON writes all metrics related to active objects, exactly as OCS provides them, without filtering out possible duplicate metrics.
- Changes take effect: After restart

> **Important**
>
> For more information about outbound-related metrics, refer to the chapter about integrating with Outbound Contact in the Interaction Concentrator 8.1 User's Guide

gos-write-metrics

Specifies whether ICON writes any precalculated OCS metrics to IDB.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: 1
- Valid values:
  - `0`—ICON does not store any precalculated metrics that OCS provides.
  - 1—ICON stores precalculated metrics that OCS provides.
- Changes take effect: After restart

gos-write-metrics-only

Specifies whether ICON excludes from database storage all outbound data except precalculated metrics.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—ICON stores both OCS data and precalculated OCS metrics, regardless of the value of the `gos-write-metrics` option.
  - `1`—Provided that the `gos-write-metrics` option is also set to 1, ICON stores only precalculated metrics.
- Changes take effect: After restart

[+] Partitioning

partition-type Option

partition-type

Dynamically specifies the content of the `gsys_partition` field in IDB tables that contain this field.

- Configured in: ICON Applicaion, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - 0—For all interactions, the `gsys_partition` field contains the date, in the YYYYMMDD format, from the `created_ts` field.
  - 1—For all interactions, the `gsys_partition` field contains the UTC value from the `created_ts` field.
  - 2—For multimedia interactions:
    - In the `G_IR`, `G_IR_HISTORY`, `G_CALL`, and `G_CALL_HISTORY` tables, the `gsys_partition` field contains the UTC value from the `attr_itx_submitted_at` attribute in the Interaction Server EventInteractionSubmitted event.
    - In the `G_AGENT_STATE_RC`, `G_CALL_STAT`, and `GM_L_USERDATA` tables, the `gsys_partition` field contains the timestamp of interaction termination.
    - In all other tables, the `gsys_partition` field contains the UTC value from the `created_ts` field.
- Changes take effect: Immediately

> **Important**
>
> For voice interactions, setting `partition-type=2` has the same effect as setting `partition-type=1`.
>
> If you are partitioning an Oracle database, you must set the `partition-type` value to 2.

[+] Persistent queue

agent-pstorage-name Option
pq-backlog-alarm-threshold Option
pq-backlog-clearance-threshold Option
pq-dbname Option
pq-purge-number Option
pq-startup-check Option
pq-startup-purge Option

agent-pstorage-name

Specifies the name of the persistent cache file that ICON creates and uses to store information about agent login sessions before writing the information to IDB.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `apstorage.db`
- Valid values: Any valid file name
- Changes take effect: After restart

pq-backlog-alarm-threshold

Specifies the maximum number of records allowed to be pending in the persistent queue for submission to IDB. When the threshold is reached, ICON generates log message 25025.

The purpose of the option is to enable an alarm to be generated when the number of records not submitted to IDB is unacceptably high because of some failure in the environment. The following are examples of environment failure:

- The database is not available, or it is not responding to ICON requests.
- The load on the ICON server is too high.
- The ICON process has not been suitably configured (for example, large quantities of expensive attached data are being stored).
- The network is slow.
- The load on the RDBMS is too high.
- There is an overall system overload.

To avoid triggering the alarm because of expected fluctuations in the ICON server load, do not set the value of this option too low. The optimal value depends on your specific deployment and contact center activity profile. Genesys recommends basing the value on the average load in your contact center, calculated from reported values for Records queued in previous 15 minutes on the Database Writer performance counter page (see the chapter about monitoring Interaction Concentrator in the Interaction Concentrator 8.1 User's Guide). For example, if 100,000 records are queued during 15 minutes of average load, consider setting the `pq-backlog-alarm-threshold` value to `400,000` to cover one hour of average load and allow for some peak loads.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values: `0`–(232–1) (`0` indicates that no log message will be generated)
- Changes take effect: After restart

pq-backlog-clearance-threshold

Specifies the minimum number of records pending in the persistent queue. When this number is reached, ICON will generate message 25026, if log message 25025 was previously generated (see the `pq-backlog-alarm-threshold` option).

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: 0
- Valid values: `0`–the value set for the `pq-backlog-alarm-threshold` option (`0` indicates that no log message will be generated)
- Changes take effect: After restart

pq-dbname

Specifies the name of the persistent queue file that ICON creates and uses to store information before writing the information to IDB.

With the default setting, the file name consists of the prefix `icon_`, followed by the identifier that Configuration Server assigns to this particular ICON application (the DBID)—for example, `icon_161.pq`.

The special value `:memory:` instructs the Persistent Queue Manager to use memory as storage instead of a physical file. Using memory for persistent queue storage may improve ICON performance with regard to database writes. However, this setting increases memory consumption, and you run the increased risk of losing data in the event ICON terminates abnormally.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `icon.pq`
- Valid values: Any valid file name or `:memory:`
- Changes take effect: After restart

> **Important**
>
> Genesys recommends that this file reside locally, not on a network.
>
> Do not use the `:memory:` value if the `role` option for the ICON instance is set to `cfg`. By design, configuration synchronization requires persistent storage, so the temporary storage provided by `pd-dbname = :memory:` will generate configuration synchronization errors for an ICON configured to perform the `cfg` role.

pq-purge-number

Specifies the number of committed transactions after which ICON purges from its persistent queue the information that is already stored in IDB. For example, if the value is set to `10`, ICON performs a purge operation on its persistent queue after every ten transactions.

> **Important**
>
> This purge does not affect the PQ file size. To reduce the file size, use the `pq-startup-purge` option.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `10`
- Valid values: Any positive integer
- Changes take effect: Immediately

pq-startup-check

Specifies whether ICON checks the integrity of its persistent queue at startup. With a large-sized persistent queue file (hundreds of megabytes), the integrity check takes up to three minutes of startup time. For any integrity violations that it finds during the integrity check, ICON issues an error message, changes the extension of the corrupted queue file to `*.bak`, and creates a new database queue.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `true`
- Valid values:
  - `true`—ICON performs the startup integrity check.
  - `false`—ICON omits the startup integrity check.
- Changes take effect: After restart

pq-startup-purge

Controls the purging of the persistent queue (PQ) file, which reduces the total file size by releasing unused file space. Purging a large PQ file may take several minutes. Therefore purging is performed only at ICON initialization, before ICON is fully started and has active interactions.

> **Important**
> If ICON has unprocessed transactions in the PQ file at startup (for example, because of DBServer or database unavailability during the previous session) only space not occupied by these unprocessed transactions can be released.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—Never purge the PQ file
  - `1`—Always purge the PQ file when ICON starts up
  - Any other positive number (optionally followed by kb, mb, or gb)—Maximum file size before ICON purges the PQ file at startup. The default unit is bytes; the units should be set in lowercase. For example, you might set the value to `100 mb` or `100 kb`.
- Changes take effect: After restart

[+] Scenario recognition

advanced-ext-party-reconstruction Option
dest-busy-processing Option
store-releasing-party Option
gcti-mode-monitoring Option

advanced-ext-party-reconstruction

For environments using SIP Server, Interaction Concentrator (ICON) supports call scenarios in which a call is sent from a monitored to an unmonitored site, and no party associated with the call remains on the monitored site. In these scenarios, the external party to which the call was sent can be reconstructed and stored in IDB.

> ## Important
>
> To use this functionality, you must also set the value for the `delivered-flag` option in the `[gts]` section of the Switch object's **Annex** tab to `1`.
>
> The non-monitored external party in these call scenarios is reported on and stored in IDB in the ALERTING state. This affects the following statistics:
>
> - `G_PARTY_STAT.TT_ON_CONNECTED`—The total time, in seconds, that all parties in a call were in the CONNECTED state during the lifetime of the party.
> - `G_CALL_STAT.TT_CONNECTED`—The total time, in seconds, during which all parties in a call were simultaneously in the CONNECTED state.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
  - `0`—No external party is created.
  - `1`—Enables advanced processing to create an external party in specific call scenarios in which a call is sent from a monitored to an unmonitored site and no party associated with the call remains on the monitored site.
- Changes take effect: Immediately

The following are examples of call scenarios for which you might need Interaction Concentrator to reconstruct the external party on the unmonitored site:

- Single-step transfer to an external number.
- Single-step transfer to a Routing Point, which then routes the call to an external number.
- Redirection of a call to an external number.
- Routing of a call to an external number in such a way that no party that is associated with this call remains on the monitored site.

dest-busy-processing

This option dynamically enables you to specify how to handle EventDestinationBusy TEvents.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `false`
- Valid values:
  - `true`—ICON processing of the destination-busy event records the cause of the party state change*mdash;cceventcause is set to busy (value = 1).
  - `false`—ICON processing of the destination-busy event does not record the cause of the party state change—cceventcause is set to normal (value = 6). This preserves ICON legacy behavior (prior to release 8.0.000.37).
- Changes take effect: Immediately

store-releasing-party

For those deployments in which T-Server reports the required data, specifies whether ICON stores data in the G_CALL_STAT and G_PARTY_STAT tables in IDB to identify the party that released the call.

- Configured in: ICON Application, [callconcentrator] Section
- Default value: 0 (ICON does not store the information)
- Valid values:
  - 1/true—For terminated calls, ICON stores data about the endpoint and party that initiated termination in the G_CALL_STAT table (GSYS_EXT_VCH1 and GSYS_EXT_VGH2 fields) and G_PARTY_STAT table (GSYS_EXT_INT1 field) in IDB.
  - 0/false—ICON does not store data about the endpoint and party that released the call. In the G_CALL_STAT table, the value of the GSYS_EXT_VCH1 and GSYS_EXT_VGH2 fields is an empty string. In the G_PARTY_STAT table, the value of the GSYS_EXT_INT1 field is 0.
- Changes take effect: After restart

Important

For information about how ICON populates the values of the fields, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

In Interaction Concentrator release 8.x, this feature is supported only for the Alcatel A4400/OXE switch.

gcti-mode-monitoring

Regulates the mode that ICON uses for multi-site scenario recognition.

- Configured in: ICON Application, `[callconcentrator]` Section
- Default value: `0`
- Valid values:
- `1`—Enables new scenario recognition logic that is implemented in ICON release 7.6.000.21.
- `0`—Preserves ICON legacy behavior (prior to release 7.6.000.21).
- Changes take effect: After restart

[+] Synchronization

cfg-auto-resync Option
start-cfg-sync Option
sync-call-data-limit Option
tsync-threshold Option

cfg-auto-resync

Specifies whether ICON will automatically initiate resynchronization of configuration data between Configuration Server data and IDB with the `cfg` role when an inconsistency is detected.

- Configured in: ICON Application
- Section: `[callconcentrator]`
- Default value: `0`
- Valid values: `0, 1`
  - `0`—ICON does not initiate automatic resynchronization.
  - `1`—ICON initiates automatic resynchronization.
- Changes take effect: Immediately

start-cfg-sync

Specifies whether ICON performs synchronization of configuration data between Configuration Database and IDB. By default, ICON ignores this option.

To start data synchronization, first set the option value to `0`; then, change the option value to `1`. This action prompts ICON to start the synchronization process. Once started, the synchronization process completes regardless of the subsequent changes to the option value.

> **Important**
>
> To perform data synchronization, ICON must have a connection to Configuration Server from the moment you change the option value from `0` to `1` until the moment when data synchronization is complete.

- Configured in: ICON Application
- Section: `[callconcentrator]`
- Default value: `-1`
- Valid values:
  - `-1`—ICON ignores this option even when it is defined in the configuration.
  - `0`—ICON acknowledges that this option is specified in the configuration and waits for a notification about the option value change from `0` to `1`.
  - `1`—ICON starts the data synchronization between Configuration Database and IDB under the condition that the value changed first to `0` and then from `0` to `1` during ICON run time. The value of `1` at ICON startup *does not* trigger the synchronization of configuration data.
- Changes take effect: Immediately upon real-time notification only.

sync-call-data-limit

Specifies the maximum number of pending synchronizations for calls and attached data.

This option controls memory consumption during the process of synchronizing calls and user data. The call record is not terminated in IDB until all attached data related to that call has been written to the database. Until then, ICON keeps in memory all information that is related to the call.

If the limit is reached, no more call records are locked until the number of pending synchronizations falls below the configured limit. This situation does not produce any loss or duplication of data, but call records that have not been locked might be marked as terminated before their related attached data has been written to IDB.

> **Important**
> Genesys recommends that you do not change the default value.

- Configured in: ICON Application
- Section: `[callconcentrator]`
- Default value: `1000000`
- Valid values: `0`–(232–1) (`0` indicates that no synchronization takes place)
- Changes take effect: Immediately

tsync-threshold

Specifies the maximum time difference, in milliseconds, allowed between the ICON host and the T-Server (or, if applicable, Interaction Server) host. When the threshold is reached, ICON generates standard log message 25130. See also the `min-tsync-roundtrip` option in the Switch object.

- Configured in: ICON Application
- Section: `[callconcentrator]`
- Default value: `1000`
- Valid values: `0`–`2000` (`0` indicates that no log message is generated)
- Changes take effect: Immediately

[+] Virtual Queue

vq-write-mode Option
extended-route-result Option
route-res-vqid-hist-enabled Option
store-route-result-reliability Option

vq-write-mode

Specifies how ICON writes to IDB information about a particular association between an interaction and a virtual queue. When this option is set to `0`, ICON creates a complete IDB record when the association is terminated, as indicated by either EventDiverted or EventAbandoned. When this option is set to `1`, ICON initially creates an IDB record when the association starts, as indicated by the EventQueued TEvent; after the association is terminated, as indicated by either EventDiverted or EventAbandoned, ICON updates the existing record.

- Configured in: ICON Application
- Section: `[callconcentrator]`
- Default value: `0`
- Valid values:
  - `0`—ICON stores virtual queue–related data in one step.
  - `1`—ICON stores virtual queue–related data in two steps.
- Changes take effect: After restart

extended-route-result

Specifies whether ICON stores extended routing results—the statuses of interactions distributed by Universal Routing Server (URS)—in the `GSYS_EXT_INT1` field in the `G_ROUTE_RESULT` table. For details of the routing results stored in IDB when `extended-route-result = 0` or `1`, refer to the chapter about monitoring virtual queues and route points in the Interaction Concentrator 8.1 User's Guide.

> **Important**
>
> You must have URS 7.6 or higher to populate extended routing results. Also, the URS configuration options `report_reasons` and `report_targets` must be set to `true`.

- Configured in: ICON Application
- Section: `[callconcentrator]`
- Default value: 0, false
- Valid values:
  - `0/false`—ICON stores standard route results in the `G_VIRTUAL_QUEUE` and `G_ROUTE_RESULT` IDB tables.
  - `1/true`—ICON stores extended routing results in `G_VIRTUAL_QUEUE` and `G_ROUTE_RESULT` IDB tables.
- Changes take effect: After restart

route-res-vqid-hist-enabled

Specifies whether virtual queue (VQ) IDs associated with G_ROUTE_RESULT records are stored in the G_ROUTE_RES_VQ_HIST table.

- Configured in: ICON Application
- Section: [callconcentrator]
- Default value: false
- Valid values:
  - true—The G_ROUTE_RES_VQ_HIST table stores VQ IDs associated with G_ROUTE_RESULT records.
  - false—The G_ROUTE_RES_VQ_HIST table is not populated.
- Changes take effect: After restart

store-route-result-reliability

Determines whether the GSYS_EXT_INT1 field in the G_ROUTE_RESULT table stores a value indicating the reliability of the data received from Universal Routing Server (URS). This field is updated based on the values set in the extended-route-result and store-route-result-reliability options. If extended-route-result = true, ICON stores a reliability flag in the GSYS_EXT_INT1 field in the G_ROUTE_RESULT IDB table. If extended-route-result = false BUT store-route-result-reliability = true, ICON stores a reliability flag.

- Configured in: ICON Application
- Section: [callconcentrator]
- Default value: false/0
- Valid values: false/0; true/1
  - false/0—No value is stored in the GSYS_EXT_INT1 field.
  - true/1—ICON stores a value in the GSYS_EXT_INT1 field. For the values stores in this field and their meanings, see the *Interaction Concentrator 8.1 Physical Data Model* document for your RBDMS.
- Changes take effect: After restart.

custom-states Section

The options in this section define Interaction Concentrator support for the processing of custom agent states and custom user data.
AgentRecordUserTypes Option
AgentUserFields Option
EventData Option
GlobalData Option
store-event-data Option

AgentRecordUserTypes

Defines the custom agent states. The agent desktop application starts and ends custom agent states, and it sends the required key-value pair (KVP) data to ICON through T-Server's EventUserEvent. ICON verifies the values provided in EventUserEvent for the key names specified by this configuration option, in order to determine when custom states start (value for the configured key = "+") and finish (value = "-"). After a state is started and before it is finished, the desktop application can send data in user events, to be stored in the custom fields that correspond to the state, as specified by the AgentUserFields configuration option. For more information about ICON custom state recording, see "Using Custom States" on the Agent States and Logins tab on the Special Configuration Requirements page.

- Configured in: ICON Application

- Section: [custom-states]
- Default value: No default value
- Valid values: A comma-separated list of the custom state codes and key names in the format <StateCode>,<KeyName>. The custom state code must be a number greater than 199.
- Changes take effect: Immediately

Example:
AgentRecordUserTypes = 207,AfterCallWork,208,Break

AgentUserFields

Specifies the fields in the G_CUSTOM_STATES table in which ICON will store values (provided in the UserData section of EventUserEvent) for the specified key names, for data that was sent while the DN was in a custom agent state.

- Configured in: ICON Application
- Section: [custom-states]
- Default value: No default value
- Valid values: A comma-separated list of the data types, table field names, and key names in the format <Type>,<FieldName>,<KeyName>.
- Changes take effect: After restart

> ### Important
> All the custom data fields in the G_CUSTOM_STATES table require character-type data. Regardless of the data type that you specify in this option, ICON converts the value from the UserData KVP into a string before storing it in the custom data field that is specified for that key name. If the value of the key in the UserData KVP is KVList, ICON ignores the value.

Example:

- AgentUserFields = char,CUST_DATA_1,KeyName1,char,CUST_DATA_2,KeyName2
  The value of the key with the name KeyName1 will be stored in the CUST_DATA_1 field.

  The value of the key with the name KeyName2 will be stored in the CUST_DATA_2 field.

EventData

Specifies the list of key names for which ICON will store KVP data (provided in the UserData section of EventUserEvent) in the G_CUSTOM_DATA_S table.

- Configured in: ICON Application

- Section: [custom-states]
- Default value: No default value
- Valid values: A comma-separated list of the data types and key names in the format <Type>,<KeyName>.
- Changes take effect: Immediately

> **Important**
>
> The limit for configuration option specifications is 255 characters. If your desired EventData option specification exceeds this limit, you can specify additional options in the format EventData_X, where *X* is any integer, starting with 1. ICON recognizes all the EventData specifications as one option, and it concatenates the content of the options in sequence.
>
> The field for the key's value in the G_CUSTOM_DATA_S table requires character-type data. Regardless of the data type that you specify in this option, ICON converts the value from the UserData KVP into a string, before storing it in the G_CUSTOM_DATA_S table. If the value of the key in the UserData KVP is KVList, ICON ignores the value.
>
> Ensure that the key name you specify does not conflict with a key name specified in the GlobalData option. The key names specified in the EventData and GlobalData options must be unique.

Example:
```
EventData =char,CUSTOMER_NAME,int,CUSTOMER_PHONE
```

GlobalData

Specifies the list of key names for which ICON will store KVP data (provided in the UserData section of EventUserEvent) in the G_CUSTOM_DATA_P table. The position of the key name in the list determines the mapping to the custom data field in the G_CUSTOM_DATA_P table.

- Configured in: ICON Application
- Section: [custom-states]
- Default value: No default value
- Valid values: A comma-separated list of the data types and key names in the format <Type>,<KeyName>.
- Changes take effect: Immediately

> **Important**
>
> The limit for configuration option specifications is 255 characters. If your desired EventData option specification exceeds this limit, you can specify additional options in the format EventData_X, where *X* is any integer, starting with 1. ICON recognizes all the EventData specifications as one option, and it concatenates the content of the options in sequence.

The field for the key's value in the G_CUSTOM_DATA_P table requires character-type data. Regardless of the data type that you specify in this option, ICON converts the value from the UserData KVP into a string, before storing it in the G_CUSTOM_DATA_P table. If the value of the key in the UserData KVP is KVList, ICON ignores the value.

Ensure that the key name you specify does not conflict with a key name specified in the EventData option. The key names specified in the EventData and GlobalData options must be unique.

Example:

- GlobalData= char,CUSTOMER_NAME,int,CUSTOMER_PHONE
  The value of the key with the name CUSTOMER_NAME will be stored in the CUST_DATA_1 field.

  The value of the key with the name CUSTOMER_PHONE will be stored in the CUST_DATA_2 field.

store-event-data

Specifies what, if any, KVP data (provided in the UserData section of EventUserEvent) ICON will store in the G_CUSTOM_DATA_S table.

- Configured in: ICON Application
- Section: [custom-states]
- Default value: none
- Valid values:
  - none—ICON does not store any data.
  - all—ICON stores the values of all keys.
  - conf—ICON stores the values of the keys that are configured in the EventData option.
- Changes take effect: Immediately

Example:
```
store-event-data = conf
```

dbw-error-reactions Section

The option or options in this section define Interaction Concentrator reactions to specific database error messages. In other words, each configuration option in this section represents a rule for handling a certain database error. If Interaction Concentrator receives an database error message, it will try to find the text specified in the configuration option as a substring of the error message. If it finds this substring text, it applies the specified error reaction.

You can use the predefined uniqueness Option error reaction or configure your own options.

uniqueness

The Interaction Concentrator Application template includes a preconfigured error reaction set to ignore unique constraint violations:

- Option Name: `uniqueness`
- Option Value: `error=unique;reaction=ignore`

The default option name, `uniqueness`, is correct for Oracle RDBMSs, but not for Microsoft SQL Server, PostgreSQL, or DB2 databases. If you are using a Microsoft SQL Server, PostgreSQL, or DB2 database, consult the documentation for your RDBMS to determine the correct error value, and then configure this option accordingly. For example, if the error returned by your Microsoft SQL Server database in reaction to violation of the uniqueness constraint is `duplicate`, configure the option as follows: `error=duplicate; reaction=ignore`.

custom-error-reactions

You can specify any number of options within this section. For example, the `dbw-error-reactions` section that you define might include an option configured as follows:

```
[dbw-error-reactions]
dbw-error1=error=ORA1123-005;reaction=retry
```

To configure an error reaction, perform the following steps:

1. Create a separate option for every database error message for which a certain reaction is required.
2. Specify any meaningful name as the option name, making it unique within the `dbw-error-reactions` section; ICON does not process the name parameter.
3. Include both a database error message and the expected reaction as two parameters of the option value, in the following format:
   `error=<error_substring>;reaction=<reaction_type>`

   Where:

   - <error_substring>—The database error message or a substring of the error message that is sufficient to identify it among all database error messages. ICON selects the first option with a matching substring among all options that contain matching substrings. You can use any character and symbol in <error_substring> except the semicolon (;). A semicolon signals the end of the error parameter to ICON. If you must include a semicolon within a substring, surround the entire substring with single quotation marks (‘<error_substring>’) or double quotation marks ("<error_substring>").
   - <reaction_type>—The expected reaction to the database error message identified by <error_substring>. The reaction can be one of the following:
     - reconnect—ICON forcefully disconnects from the database and attempts to reconnect after receiving the database error message identified by <error_substring>. This reaction type is recommended for error messages related to the temporary unavailability of a database that is inadequately processed by the database server.

  - retry—ICON rolls back the current transaction, and then attempts to resubmit it after receiving the database error message identified by <error_substring>. This reaction type is recommended for error messages related to nonfatal database problems (for example, a locked table state) that tend to disappear during subsequent transaction attempts.
  - ignore—ICON rolls back the current transaction after receiving the database error message identified by <error_substring>. ICON then attempts to resubmit the failed transaction, statement by statement, ignoring the statement that caused the error. This reaction type is recommended for logical errors such as constraint violations.
- Default value: No default value
- Valid values: Any string in the following format:
  `error=<error_substring>;reaction=<reaction_type>`
- Changes take effect: Immediately

Example: The following configuration option prompts ICON to resubmit a request that previously failed with an error message that contains the substring `ORA1123-005`:

```
dbw-error1=error=ORA1123-005;reaction=retry
```

filter-data Section

To use the options listed in this section, create a section named `filter-data` in the ICON Application.

The options in this section control Interaction Concentrator output to IDB. Refer to the *Interaction Concentrator 8.1 Physical Data Model* for your RDBMS type for details about data stored in the IDB tables that are mentioned in the option descriptions. Evaluate whether your reports require each type of described data.

Important

Excluding certain types of data from IDB storage may help you save database space, and thus improve your database performance.

Important

To enable identification of the party that initiated release of a call in deployments that support this feature, ensure that the value of the `external-party`, `acd-party-metrics`, `acd-party-history`, `call-metrics`, and `observer-party` options is `0` (the default value). Otherwise, ICON will not store records in the `G_PARTY_STAT` table for the types of parties influenced by these options. As a result, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.

> For more information about the releasing-party feature, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

acd-party-history

Specifies whether ICON should exclude party history information about distribution devices such as ACD queues, Routing Points, virtual routing points, and External Routing Points from storage in IDB. By default, ICON collects party history information about distribution devices and stores this information in the G_PARTY_HISTORY IDB table. Set this option to 1 to instruct ICON not to store party history information in the G_PARTY_HISTORY table.

> **Important**
>
> The acd-party-history option applies to SIP and voice interactions only.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

> **Important**
>
> To enable identification of the party that initiated release of a call in deployments that support this feature, ensure that the value of the external-party, acd-party-metrics, acd-party-history, call-metrics, and observer-party options is 0 (the default value). Otherwise, ICON will not store records in the G_PARTY_STAT table for the types of parties influenced by these options. As a result, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.
>
> For more information about the releasing-party feature, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

acd-party-metrics

Specifies whether ICON should exclude from IDB storage party metrics for distribution devices such as ACD queues, Routing Points, virtual routing points, and External Routing Points. By default, ICON collects precalculated party metrics for distribution devices and stores this information in the G_PARTY_STAT IDB table. When set to 1, ICON does not store data in the G_PARTY_STAT table for distribution devices.

> **Important**
>
> The `acd-party-metrics` option applies to SIP and voice interactions only.

- Configured in: ICON Application
- Default value: `0`
- Valid values: `1, 0`
- Changes take effect: After restart

> **Important**
>
> To enable identification of the party that initiated release of a call in deployments that support this feature, ensure that the value of the `external-party`, `acd-party-metrics`, `acd-party-history`, `call-metrics`, and `observer-party` options is `0` (the default value). Otherwise, ICON will not store records in the G_PARTY_STAT table for the types of parties influenced by these options. As a result, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.
>
> For more information about the releasing-party feature, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

call-history

Specifies whether ICON should exclude call-history information from IDB storage. By default, ICON collects and stores call history data in the `G_CALL_HISTORY` IDB table. When set to `1`, ICON ceases writing to this table.

- Configured in: ICON Application
- Default value: `0`
- Valid values: `1, 0`
- Changes take effect: After restart

call-metrics

Specifies whether ICON should exclude call metrics from IDB storage. By default, ICON calculates call metrics and stores them in the G_CALL_STAT IDB table. When set to 1, ICON ceases writing to this table.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

Important

To enable identification of the party that initiated release of a call in deployments that support this feature, ensure that the value of the external-party, acd-party-metrics, acd-party-history, call-metrics, and observer-party options is 0 (the default value). Otherwise, ICON will not store records in the G_PARTY_STAT table for the types of parties influenced by these options. As a result, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.

For more information about the releasing-party feature, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

external-party

Specifies whether ICON should exclude external-party data from IDB storage. By default, ICON collects information about external parties (for example, interaction participants outside a given switch domain) and stores this information in the following IDB tables:

- G_PARTY
- G_PARTY_HISTORY
- G_PARTY_STAT

When set to 1, ICON collects and stores data about internal parties only (for example, interaction participants within a given switch domain).

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

> **Important**
>
> To enable identification of the party that initiated release of a call in deployments that support this feature, ensure that the value of the external-party, acd-party-metrics, acd-party-history, call-metrics, and observer-party options is 0 (the default value). Otherwise, ICON will not store records in the G_PARTY_STAT table for the types of parties influenced by these options. As a result, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.
>
> For more information about the releasing-party feature, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

gls-all

Specifies whether ICON should exclude all information about agent activity from IDB storage. By default, ICON collects information about agent activity, such as login sessions and agent states, unless certain types of data are configured to be excluded by setting one or more of the following options to 1:

- gls-ivr
- gls-no-person
- gls-queue)
- gls-wm

ICON stores this information in the following IDB tables:

- G_LOGIN_SESSION
- GX_SESSION_ENDPOINT
- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- G_DND_HISTORY
- GS_AGENT_STAT
- GS_AGENT_STAT_WM

When set to 1, ICON ceases writing to these tables. However, even when this option is set to 1, ICON continues writing agent IDs to the G_PARTY table.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

gls-ivr

Specifies whether ICON should exclude from IDB storage data about agent activity at IVR endpoints. By default, ICON collects data about agent activity when agent login sessions are initiated from IVR endpoints and stores this information in the following IDB tables:

- G_LOGIN_SESSION
- GX_SESSION_ENDPOINT
- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- G_DND_HISTORY
- GS_AGENT_STAT
- GS_AGENT_STAT_WM

When set to 1, ICON verifies whether the DN at which an agent logs in is an IVR device; in this case, ICON does not store information about this agent's activity to these tables. Furthermore, for parties associated with an IVR device, ICON does not record the agent's ID in the G_PARTY IDB table.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

> **Important**
>
> See the ivr option on the DN tab of the Configuration Options page for information on how to configure a DN as an IVR port. For a description of how ICON identifies an IVR, see the Interaction Concentrator 8.1 User's Guide.

gls-metrics

Specifies whether ICON should exclude agent states from IDB. By default, ICON collects agent states unless certain types of data are configured to be excluded by setting one or more of the following options to 1:

- `gls-all`
- `gls-ivr`
- `gls-no-person`
- `gls-queue`
- `gls-wm`

ICON stores agent states information in the following IDB tables:

- `GS_AGENT_STAT`
- `GS_AGENT_STAT_WM`

When set to 1, ICON does not store information about agent states to these tables.

- Configured in: ICON Application
- Default value: `0`
- Valid values: `1, 0`
- Changes take effect: After restart

gls-no-person

Specifies whether ICON should exclude from IDB storage data about agent activity for agents whose login ID is not associated with any Person configuration object. By default, ICON collects data about all agent activity and stores this information in the following IDB tables:

- G_LOGIN_SESSION
- GX_SESSION_ENDPOINT
- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- G_DND_HISTORY
- GS_AGENT_STAT
- GS_AGENT_STAT_WM

When set to 1, ICON verifies whether the LoginID reported in events regarding agent states is assigned to any Person object configured in the Configuration Database; if this is not the case, ICON does not store information about this agent's activity to these tables.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

gls-queue

Specifies whether ICON should filter out information from IDB storage about the queues where agents are logged in. By default, ICON collects information about agent queue(s) and stores this information in the following IDB tables:

- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- GS_AGENT_STAT
- GS_AGENT_STAT_WM
- GX_SESSION_ENDPOINT

When set to 1, ICON ceases writing queue-related data to the first four tables listed above. ICON does continue to write information to the GX_SESSION_ENDPOINT table about the queues where agents are logged in.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

gls-wm

Specifies whether ICON should exclude from IDB storage data about changes in agent work mode that do not coincide with changes in agent state. By default, ICON collects and stores data about agent work modes and changes in agents work modes in the following IDB tables:

- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- GS_AGENT_STAT_WM

When set to 1, ICON disregards information about work mode and work mode changes. It records a value of unknown in the IDB tables listed above.

> **Important**
> This option does not affect ICON's ability to track after-call work.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

ir-history

Specifies whether ICON should exclude data about the interaction record history from IDB storage. By default, ICON collects interaction record history and stores this information in the G_IR_HISTORY IDB table. When set to 1, ICON ceases to write data to this table.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

observer-party

Specifies whether ICON should exclude from IDB storage data related to a service observer on a call. By default, ICON collects data about every party involved with the call and stores this information in the following IDB tables:

- G_PARTY
- G_PARTY_HISTORY
- GS_PARTY_STAT

When set to 1, ICON does not store data about the party with the role Observer to these tables.

- Configured in: ICON Application
- Default value: 0
- Valid values: 1, 0
- Changes take effect: After restart

> **Important**
>
> To enable identification of the party that initiated release of a call in deployments that support this feature, ensure that the value of the external-party, acd-party-metrics, acd-party-history, call-metrics, and observer-party options is 0 (the default value). Otherwise, ICON will not store records in the G_PARTY_STAT table for the types of parties influenced by these options. As a result, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.
>
> For more information about the releasing-party feature, see the section about identifying who released the call in the Interaction Concentrator 8.1 User's Guide.

udata-history-terminated

Specifies whether ICON should exclude from IDB storage information about changes in UserData values for certain keys. When ICON is configured to store an entire history of UserData values for certain keys, ICON collects data about every change in value for those keys and, at interaction termination, stores this information in the following IDB tables:

- `G_USERDATA_HISTORY`
- `G_SECURE_USERDATA_HISTORY`

When set to `1`, ICON does not insert new records in these tables at call termination. ICON does, however, continue to write information about the creation, addition, and removal of key-value pairs to these tables.

- Configured in: ICON Application
- Default value: `0`
- Valid values: `1, 0`
- Changes take effect: After restart

listeners Section

This section refers to a separate configuration section that describes the HTTP listening port. You must name this section `listeners` in the ICON Application. The name of this option must correspond to the section name for the user named section described below.

user_named_option Option

user_named_option

You will name any option you create in this section. One option name *must* match the name of a new section you create to define the parameters for an http connection.

Because ICON processes only the name of this option, but not the value, you can use the value to enter a short description for the connection; ICON will print this description to its log.

- Configured in: ICON Application, `listeners` Section
- Default value: No default value
- Valid values: Any string
- Changes take effect: Immediately

Example:

```
[listeners]
http-9090="ICON HTTP listener"
```

Where `http-9090` is the same as the name of the section that describes the parameters of an HTTP connection at a port that ICON opens for listening.

> **Important**
>
> To enable access to the performance counters, configure an HTTP Listener option, and then configure a corresponding section, along with its `port, protocol,` and `transport` options.

user_named_section Section

This is a user-configurable section. You can choose any name for this section, provided that it matches the name that you specify for an option in the listeners section. Use the options described below to set parameters for this connection.
port Option
protocol Option
transport Option

Example:

- `[http-9090]`
- `port=9090`
- `transport=tcp`

- `protocol=http`

port

Specifies the number of the port that ICON opens for HTTP listening.

- Configured in: ICON Application, user-named Section.
- Default value: No default value
- Valid values: Any integer from 1 to 65535
- Changes take effect: After restart

> **Warning**
> The value for the port option must not coincide with the ICON Application object's communication port that is opened for client connections.

protocol

Specifies the application-level protocol for the configured listener. Change the value to `http` to enable access to interfaces that are exposed through HTTP in ICON and that display performance counters.

> **Important**
> The HTTP interface is not available by default.

- Configured in: ICON Application, user-named Section.
- Default value: `sip`
- Valid values: `http`, `sip`
- Changes take effect: Immediately

transport

Specifies the transport layer protocol for the connection between ICON and its client.

- Configured in: ICON Application, user-named Section.
- Default value: TCP (Transmission Control Protocol)
- Valid Value: TCP
- Changes take effect: Immediately

log Section

In addition to the log options that are common to all Genesys Server applications, and that are described in the Framework Configuration Options Reference Manual, Interaction Concentrator supports a number of unique log options that can help you troubleshoot various scenarios when you deploy ICON and test its functionality in your environment.

Use the log section on the **Options** tab to set all of the Interaction Concentrator log options.

The meaning of the log options valid values are as follows:

- 0—No troubleshooting-related logging
- 1—Logging of errors only
- 2—Detailed troubleshooting-related logging
- 3—Full details in troubleshooting-related logging

x-conn-debug-open Option
x-conn-debug-select Option
x-conn-debug-timers Option
x-conn-debug-write Option
x-print-attached-data Option
x-server-trace-level Option
x-server-config-trace-level Option
x-server-dbw-trace-level Option
x-server-gcti-trace-level Option
x-server-http-trace-level Option
x-server-smtp-trace-level Option
x-server-debug-level Option

x-conn-debug-open

Specifies the verbosity with which ICON logs messages related to network connections and disconnections at a transport protocol level. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to network connection messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-conn-debug-select

Specifies the verbosity with which ICON logs messages related to incoming information at a transport protocol level. This option may significantly increase log volume. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to incoming information messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-conn-debug-timers

Specifies the verbosity with which ICON logs messages related to triggering connection timers at a transport protocol level. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to connection timer–triggering messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-conn-debug-write

Specifies the verbosity with which ICON logs messages related to outgoing information at a transport protocol level. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to outgoing information messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-print-attached-data

Specifies whether userdata will be printed to the log. Genesys recommends that you do not change the default setting (`false`), because printing userdata to the log can significantly increase log size and impact system resources.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: false
- Valid values:
  - `true`—Enables printing userdata to the log.
  - `false`—Suppresses printing userdata to the log.
- Changes take effect: Immediately

x-server-trace-level

Specifies the verbosity with which ICON prints troubleshooting-related logs. This option sets the default value for all troubleshooting-related log options that are unique to ICON. That is, the value that you set for this option applies to the following function-specific options if you do not configure them:

```
x-conn-debug-open
x-conn-debug-select
x-conn-debug-timers
x-conn-debug-write
x-server-config-trace-level
x-server-dbw-trace-level
x-server-gcti-trace-level
x-server-smtp-trace-level
x-server-http-trace-level
```

If you do set a value for any of these function-specific options, and if that value differs from the `x-server-trace-level` option value, the function-specific option value supersedes the `x-server-trace-level` option value for log messages related to that particular function.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: `0`
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-server-config-trace-level

Specifies the verbosity with which ICON logs messages related to the configurations of the objects on which it relies. Messages can include configuration information about ICON's own Application object. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to configuration information messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-server-dbw-trace-level

Specifies the verbosity with which ICON logs messages related to data-writing operations with the persistent queue and IDB. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to data-writing operation messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-server-gcti-trace-level

Specifies the verbosity with which ICON logs messages related to its CTI communications. Messages can include TEvents that ICON receives from T-Server, including call-related and party-related events, and they can also include reports about CTI transactions. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to CTI communications messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-server-http-trace-level

Specifies the verbosity with which ICON logs messages related to its HTTP communications. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value that you set for this option supersedes the value set for the `x-server-trace-level` option with regard to data-writing operation messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-server-smtp-trace-level

Specifies the verbosity with which ICON logs messages related to its SMTP communications. The value `0` disables troubleshooting-related logging, and the value 3 produces the most detailed logs. Any value you set for this option supersedes the value set for the `x-server-trace-level` option with regard to data-writing operation messages.

- Configured in: ICON Application, **Options** tab, `log` section
- Default value: As specified by the `x-server-trace-level` option
- Valid values: `0|1|2|3`
- Changes take effect: Immediately

x-server-debug-level

Like the `x-server-trace-level` option, specifies the verbosity with which ICON prints troubleshooting-related logs. ICON supports both option names, but Genesys recommends using the `x-server-trace-level` option name. For more information, see the description of the `x-server-trace-level` option.

Switch Options

This section describes the configuration options that you configure on the **Annex** tab of any Switch configuration object that is monitored by a T-Server related to your Interaction Concentrator. Interaction Concentrator processes these options.

> **Important**
> The information in this section does not apply to Multimedia switches.

gts Section
All the Switch configuration options that affect Interaction Concentrator behavior are contained in a special configuration section, `gts`. If required, you must create this section on the **Annex** tab of the Switch object.

call-deletion-timeout

Specifies the amount of time, in seconds, that ICON delays call context deletion after receiving a notification that the call has been deleted in T-Server.

- Configured in: Switch Application
- Default value: `30`
- Valid values: `3–600`
- Changes take effect: Immediately

delivered-flag

Controls when an unmonitored party is reconstructed (regarding an event flow), and when a transition to the alerting state occurs for this party in the call to an external destination (regarding the switch).

- Configured in: Switch `Application` object, `Annex` tab, `[gts]` ection
- Default values:
  - `0`—(For all switches except Cisco CallManager)
  - `1`—(For Cisco CallManager)
- Valid values:
  - `0`—The alerting state is generated when EventDialing arrives.
  - `1`—The alerting state is generated when EventNetworkReached arrives.
  - `2`—The alerting state is generated when EventEstablished arrives.
  - `3`—An unmonitored party is not reconstructed.
- Changes take effect: After restart

> **Tip**
> Genesys Customer Care recommends that you set the value of this option to 3 only for a particular event flow.

emulate-event-queued-extrp

Enables the emulation of EventQueued for an External Routing Point that belongs to this switch.

> **Important**
> Generation of EventQueued for an External Routing Point depends on a particular T-Server and its switch. ICON requires this event for correct party representation in any environment.

- Configured in: Switch `Application` object, `Annex` tab, `[gts]` section
- Default value: 1
- Valid values:
  - 0—EventQueued is not emulated.
  - 1—EventQueued is emulated.
- Changes take effect: Immediately

> **Important**
> For help setting this option correctly, contact Genesys Customer Care.

emulate-event-queued-rp

Enables the emulation of EventQueued for a Routing Point that belongs to this switch.

> **Important**
> Generation of EventQueued for a Routing Point depends on a particular T-Server and its switch. ICON requires this event for correct party representation in any environment.

- Configured in: Switch Application
- Default value: 1
- Valid values:
  - `0`—EventQueued is not emulated.
  - 1—EventQueued is emulated.
- Changes take effect: Immediately

> **Important**
> For help setting this option correctly, contact Genesys Customer Care.

emulate-event-queued-rq

Enables the emulation of EventQueued for a routing queue that belongs to this switch.

> **Important**
> Generation of EventQueued for a routing queue depends on a particular T-Server and its switch. ICON requires this event for a correct party representation in any environment.

- Configured in: Switch Application
- Default value:1
- Valid values:
  - 0—EventQueued is not emulated.
  - 1—EventQueued is emulated.
- Changes take effect: Immediately

> **Important**
> For help setting this option correctly, contact Genesys Customer Care.

fix-time-stamps

Enables adjustment of timestamps when the CTI event contains an earlier timestamp than the timestamp from a previously received CTI event.

- Configured in: Switch Application
- Default value: `0`
- Valid values:
  - 0—Adjustment is disabled.
  - `Any non-zero integer`—Adjustment is enabled.
- Changes take effect: After restart

gls-acw-first

Specifies which interaction ICON associates with after-call work (ACW). This option is configured in the ICON Application, or in the Switch Application, or both. If it is set only in the ICON Application, it applies to all switches ICON is configured to monitor. If any Switch is set with a value different from that set in the ICON Application, the Switch value takes precedence.

By default, ICON associates after-call work metrics with the voice interaction that immediately precedes the completion of the after-call work (the last voice interaction).

Setting this option to `true` enables ICON to associate after-call work with the voice interaction that most recently changed the agent's state from NotBusy to Busy (the first voice interaction). In this case, subsequent voice interactions that occur during the period of after-call work are considered as related to ACW processing and should not interrupt measurement of ACW-related metrics.

When the agent logs out, changes his or her state to Ready, or goes NotReady for any reason other than to perform after-call work, ICON reports the end of the current ACW state.

- Configured in: ICON Application, `[callconcentrator]` Section; Switch Application, Annex tab, `[gts]` Section

ICON Application Settings:

- Default value: `false`
- Valid values:
  - `false`—ICON associates the last voice interaction with after-call work.
  - `true`—ICON associates the first voice interaction with after-call work.
- Changes take effect: After restart

Switch Application Settings:

> **Important**
>
> To associate the first ACW value, specify the value of this option on the Switch Application. A change to the setting of this option on the ICON Application does not propagate to SIP switches.

- Default value: `-1`
- Valid values:
  - `-1`—ICON uses the value of the `gls-acw-first` option specified in the ICON Application object. If no value is set at the application level, ICON associates the last voice interaction with after-call work.
  - `0`—ICON associates the last voice interaction with after-call work.
  - `1`—ICON associates the first voice interaction with after-call work.

- Changes take effect: After restart

> **Important**
>
> For SIP switches, the default value results in the same functionality as setting the option to `0`.

gls-associations-rule

Controls, for this switch, how ICON associates DNs with a given agent login session. You can configure DN associations in Configuration Layer in two ways:

- By adding DNs to the same Place object. (For example, a DN of Position type and DN of Extension type on the same phone set on an Avaya switch must belong to the same Place. Another example involves DNs of different media types that are included into the same Place.)
- By creating a relationship between two DNs through the Association field in the DN Properties window.

The `gls-associations-rule` option enables ICON to process signaling on the associated DNs as follows:

- With the setting of `-1`, ICON creates two separate login sessions for an agent who logs in with two different login IDs at two DNs that belong to the same place. For example, when one DN is used for multimedia interactions and another DN is used for voice interactions, ICON handles agent login sessions at these two DNs separately.
- With the setting of `0`, ICON creates a single login session for two DNs that belong to the same place when an agent logs in at one of these DNs. For example, when an agent logs in at a position DN and an extension DN exists on the same phone set, ICON maintains a single login session for these two DNs.
- With the setting of `1`, ICON creates a single login session for two DNs that are related through the Association field when an agent logs in at one of these DNs. For example, when an agent logs in to different queues from two associated DNs, ICON maintains a single login session for these two DNs.
- Configured in: Switch Application
- Default values:
  - -1—(For SIP switches)
  - 0—(For all switches except SIP)
- Valid values:
  - -1—ICON associates each DN with a separate login session.
  - 0—ICON associates a single login session with multiple DNs at a place.
  - 1—ICON associates a single login session with two DNs associated through configuration.
- Changes take effect: After restart

gls-enable-acw-busy

Specifies, for this switch, whether ICON should continue ACW and NotReady agent states when agents place or receive calls during the period of time that after-call work or NotReady agent state were invoked.

The following IDB tables are affected by this option: `G_AGENT_STATE_HISTORY`, `G_AGENT_STATE_RC`, `GS_AGENT_STAT`, `GS_AGENT_STAT_VM`. For a description of these tables, refer to the IDB schema chapter in the Interaction Concentrator 8.1 User's Guide.

- Configured in: Switch Application
- Default value: 1
- Valid values:
  - `0`—ICON continues ACW and NotReady agent states while an agent is handling another call.
  - 1—ICON interrupts ACW and NotReady agent states while the agent handles another call.
- Changes take effect: After restart

ICON recognizes completion of after-call work when any of the following occur:

- The agent logs out.
- The agent places himself/herself in Ready mode.
- The agent goes NotReady for any reason other than to perform after-call work. (This includes indirect work mode changes such as when the agent walks away from his or her desk for a period of time.)

> **Important**
>
> This option is not valid for SIP-compliant switches that handle interactions other than voice interactions.

gls-flag-on-disconnect

Specifies how ICON handles agent states when disconnecting from, and reconnecting to, T-Server.

- Configured in: Switch Application
- Default value: `0`
- Valid values:
  - `0`—When reconnecting to T-Server, ICON compares the agent state from its memory with the state from EventRegistered. If the in-memory state does not match the currently reported agent state, ICON updates the agent state in both its internal memory and IDB. When disconnecting from T-Server, ICON performs no actions specific to agent states.
  - 1—When disconnecting from T-Server, ICON closes any existing agent login sessions, and records this fact in IDB. When reconnecting to T-Server, ICON uses information from EventRegistered to start new agent login sessions, sets the current agent states, and writes this data to IDB.
  - 2—When disconnecting from T-Server, ICON does not close any existing agent login sessions. Instead, it changes agents' states to UNKNOWN, and records these new states in IDB. When reconnecting to T-Server, ICON uses information from EventRegistered to restore the current agents' states and write them to IDB.
- Changes take effect: Immediately

Important

- Genesys recommends setting this option to `0` when the switch is monitored by T-Server 7.6.
- Genesys recommends that you do not set the value of this option to 1 for deployments supporting HA of agent data. If you choose to set this option to 1, however, a limited amount of HA agent data will be available (event sequence numbers only) provided that you also set the `gls-use-ts-id` configuration option in the `[gts]` section to `0` on the Switch **Annex** tab.

gls-improve-data-for-agent

Specifies when ICON should process agent states data in two-step transfer and conference scenarios.

> **Important**
> Genesys Info Mart customers should use the default value for this option.

- Configured in: Switch Application
- Default value: `0`
- Valid values:
  - `0`—EventCallDeleted triggers agent states data processing (legacy behavior).
  - `1`—Enables ICON to process agent states data based on EventReleased and store a more accurate value of `PartyID` in the `G_AGENT_STATE_HISTORY` table when a record describes one of the following:
    - An agent state changing from Busy to another state
    - An agent state changing from Busy to Busy
    - An agent state changing to ACW
- Changes take effect: After ICON is restarted

gls-max-duration

Specifies the maximum amount of time, in hours, that an agent login session can last on a DN that belongs to this switch. Setting the option value to `0` (zero) prevents ICON from checking session durations.

- Configured in: Switch Application
- Default value: `0`
- Valid values: Any integer from `0` to `720`
- Changes take effect: Immediately

In deployments that use T-Server release 7.6 or later, ICON ignores the `gls-max-duration` option. With T-Server release 7.6 and later, T-Server generates agent login session IDs and controls the login sessions. In this case, the `gls-max-duration` option has no effect on ICON reporting.

Earlier releases of T-Server do not provide agent login session IDs. In these cases, ICON generates its own agent login session IDs, and uses the `gls-max-duration` and `gls-max-inactivity` options to help manage reporting on agent login session activity.

gls-max-inactivity

Specifies the maximum allowed inactivity period, in hours, during a single login session. ICON closes any agent login session for which no agent-related activity is detected during the specified interval. Setting the option value to `0` (zero) prevents ICON from checking inactivity durations.

- Configured in: Switch Application
- Default value: `0`
- Valid values: Any integer from `0` to `72`
- Changes take effect: Immediately

In deployments that use T-Server release 7.6 or later, ICON ignores the `gls-max-inactivity` option. With T-Server release 7.6 and later, T-Server generates agent login session IDs and controls the login sessions. In this case, the `gls-max-inactivity` option has no effect on ICON reporting.

Earlier releases of T-Server do not provide agent login session IDs. In these cases, ICON generates its own agent login session IDs, and uses the `gls-max-duration` and `gls-max-inactivity` options to help manage reporting on agent login session activity.

gls-use-ts-id

Specifies whether ICON uses the login session ID generated by T-Server (GUID) or by itself when connecting to, or disconnecting from, T-Server.

- Configured in: Switch Application
- Default value: 1
- Valid values:
  - `0`—ICON generates the login session ID itself.
  - 1—ICON uses the login session ID (GUID) generated by T-Server.
- Changes take effect: After restart

> **Important**
>
> If you set this option to `0`, make sure you also set the `gls-flag-on-disconnect` option to 1 in order to access available HA agent data.

gts-dnis-detection

Specifies how the value of DNIS is determined for outbound calls.

- Configured in: Switch Application
- Default value: `0`
- Valid values: `0, 1`
  - `0`—The DNIS is captured only from the attributeDNIS value in the TEvents related to the outbound call.
  - `1`—An extended algorithm will be used to find the value of the DNIS.
- Changes take effect: After restart

min-tsync-roundtrip

Specifies the amount of time, in milliseconds, allowed for messages sent from ICON to T-Server to be acknowledged by T-Server, for the purposes of time synchronization. All messages that are acknowledged within the specified round-trip delay are considered valid for the purposes of calculating the time difference between the ICON host and the T-Server host.

See also the `tsync-threshold` option in the ICON Application object.

- Configured in: Switch Application
- Default value: `50`
- Valid values: `0`–`500` (`0` indicates that no calculation will be performed)
- Changes take effect: Immediately

ring-divert

Controls whether ICON identifies the PARENTPARTYID and the PARENTLINKTYPE of the Ringing party in event flows in which EventRinging comes before EventDiverted or the call is routed to an external switch.

> **Tip**
>
> You can set this option in the **Annex** section either of the Switch or the DN configuration object, or both. If it is set to a valid value, the DN option overrides the value set for the Switch.

- Configured in: Switch Application, DN Application
- Default Value: `0`
- Valid Values: `0, 1`
  - `0`—ICON preserves its former behavior; that is, ICON does not identify the PARENTPARTYID or the PARENTLINKTYPE in these event flows.
  - `1`—ICON correctly sets the value of the PARENTPARTYID and the PARENTLINKTYPE.
- Changes Take Effect: If set for the Switch, changes take effect after ICON is restarted; if set for the DN, changes take effect immediately.

> **Important**
>
> Interaction Concentrator does not support event flows in which EventRinging comes before EventDiverted for two-step transfer scenarios in which the transfer is completed before the call rings on the target DN. This is the case even if you set the `ring-divert` option value to 1.

sst-options

Specifies the TEvents that ICON uses to recognize a single-step transfer, in order to ensure the correct processing of scenarios involving a single-step transfer.

- Configured in: Switch Application
- Default value: `0`
- Valid values:
  - `0`—EventReleased, followed by a corresponding EventRinging or EventQueued. Arrivals of EventReleased, EventRinging, or EventQueued trigger the recognition logic.
  - `1`—EventReleased only. Arrival of EventReleased with an additional cause attribute triggers the recognition logic.
- Changes take effect: After restart

> **Important**
>
> Set this value to 1 for:
>
> - SIP Server deployments with VoIP IVRs (GVP and third-party)
> - T-Server for Siemens HiPath 4000 CSTA III.

support-dn-type-5

Enables the processing of events that pertain to DNs of the Virtual Queue type that belong to this switch.

- Configured in: Switch Application
- Default value: 1
- Valid values:
  - 0—ICON does not process any Virtual Queue–related events for DNs that belong to this switch.
  - 1—ICON processes Virtual Queue–related events for DNs that belong to this switch.
- Changes take effect: Immediately

suppress-user-data

Specifies whether the switch instructs T-Server to propagate attached data only when the attached data changes. This optimizes ICON processing of attached data by reducing network traffic.

This option can be set at the level of the Switch or the ICON application. ICON automatically detects the Switch-level option setting. If the Switch-level option is set to the (default) value of 1 (unchanged attached data suppressed), T-Server TEvents are optimized for all ICON applications that connect to the T-Servers for that Switch. In this case, the Switch-level option setting overrides any ICON-level settings of `0` (unchanged attached data not suppressed). If the Switch-level option is set to `0`, an application-level setting of 1 will override it.

- Configured in: ICON Application, `[callconcentrator]` Section; Switch Application, Annex tab, `[gts]` Section
- Default value: 1
- Valid values:
  - `0`—Unchanged attached user data is not suppressed.
  - 1—Unchanged attached user data is suppressed.
- Changes take effect: After restart

switch-multi-links-enabled

Specifies whether this switch is working in load-balancing mode; that is, it is served by multiple Network T-Servers or IVR T-Servers. ICON uses this option to determine whether to enable connection to more than one Network T-Server or IVR T-Server serving this switch.

- Configured in: Switch Application
- Default value: `0`
- Valid values:
  - 1—A network or IVR switch in load-balancing mode.
  - Any other integer—Not a network or IVR switch in load-balancing mode.
- Changes take effect: After restart

This option should be used only in a configuration in which Network T-Servers or IVR T-Servers are working in load-balancing mode; that is, when there is no duplication in notification events received in ICON via connections to these T-Servers. Currently, load balancing mode is supported only for Network T-Servers and IVR T-Servers.

third-party-queue-in-divert

Specifies how Interaction Concentrator should process multi-queue scenarios in which a call is distributed to multiple queues simultaneously, then it is distributed from one of these queues and cleared from the remaining queues. When the `third-party-queue-in-divert` option is set to 1, Interaction Concentrator takes into account AttributeThirdPartyQueue in EventDiverted when AttributeCallState has a value of `0` in order to process interactions in the same way as for redirect scenarios.

> **Important**
>
> Currently only T-Server for Avaya Communication Manager release 7.6 and higher supplies AttributeThirdPartyQueue. For all other T-Servers, Genesys recommends that you use the default value of the `third-party-queue-in-divert` option.

- Configured in: Switch Application
- Default value: `0`
- Valid values:
  - `0`—Interaction Concentrator does not check for the presence of AttributeThirdPartyQueue in EventDiverted.
  - `1`—Interaction Concentrator checks for the presence of AttributeThirdPartyQueue in EventDiverted.
- Changes take effect: After Interaction Concentrator is restarted

> **Important**
>
> - Genesys Info Mart 7.x customers should use the default value for this option.
> - In multi-queue scenarios, distribution to external DNs is not supported.

DN Options

This section describes the ICON-related configuration options you configure on the **Annex** tab of a DN configuration object. Interaction Concentrator processes these options.

gts Section
You must name this section `gts` in the configuration.

do-not-register

Enables you to prevent ICON from registering specified DNs. ICON does not receive any events from these DNs and processes data from these DNs as external.

- Configured in: DN Application, Annex tab, `[gts]` section
- Default value: `0`
- Valid values:
  - `0`—ICON functions as in prior releases, and registers this DN.
  - `1`—ICON does not register the specified DN.
- Changes take effect: After restart

emulate-event-queued

Enables the emulation of EventQueued for this particular DN. This setting supersedes the value set in an EventQueued-related option at the Switch level.

> **Important**
> Generation of EventQueued for a Routing Point, a Routing Queue, and an External Routing Point depends on a particular T-Server and its switch. ICON requires this event for correct party representation in any environment.

- Configured in: DN Application
- Default values:
  - 0—For a Routing Queue DN
  - 1—For a Routing Point DN and an External Routing Point DN
- Valid values:
  - 0—EventQueued is not emulated.
  - 1—EventQueued is emulated.
  - Changes take effect: Immediately

ivr

Specifies whether ICON treats this DN as an IVR port. By default, ICON identifies DNs as IVR ports using one of the following criteria:

- DN has a type of Voice Treatment Port in Configuration Database.
- DN has a type of ACD Position or Extension, and it is specified as an Associated DN in the properties of the IVR port.
- Configured in: DN Application
- Default value: `0`
- Valid values:
  - `0`—ICON does not recognize this DN as an IVR port unless the DN configuration satisfies one of the above criteria.
  - 1—ICON treats this DN as an IVR port regardless of other configuration parameters specified for this DN.
- Changes take effect: Immediately

Important

- You must set the value of this option to 1 if (a) you want ICON to reliably report that the endpoint associated with a party is an IVR port and (b) the DN configuration does not satisfy either of the above criteria. Note that ICON does not limit the DNs to which you can apply this configuration value, and there are no internal checks to verify that the value is correct for your deployment. Set this configuration option carefully to avoid unwanted downstream implications. For example, if you incorrectly set this option to 1 for a non-IVR device type, ICON will report it as an IVR device, and the downstream reporting application might interpret ICON data incorrectly.
- See the `gls-ivr` option description to exclude data about agent activity associated with this IVR device from IDB storage. See also the description of the `GSYS_EXT_INT1` field in the `G_PARTY_HISTORY` table, which can be found in the *Interaction Concentrator 8.1 Physical Data Model* for your RDBMS.

monitor

Applicable to DNs of the Virtual Queue type, this option enables the processing of Virtual Queue–related events for this particular DN. This option is meaningful only when the `support-dn-type-5` configuration option is set to 1 (default) in the corresponding Switch object configuration.

- Configured in: DN Application
- Default value: 1
- Valid values:
  - `0`—ICON does not handle any Virtual Queue–related events for this DN.
  - 1—ICON processes Virtual Queue–related events for this DN.
- Changes take effect: Immediately

ring-divert

Controls whether ICON identifies the PARENTPARTYID and the PARENTLINKTYPE of the Ringing party in event flows in which EventRinging comes before EventDiverted or the call is routed to an external switch.

> **Tip**
>
> You can set this option in the **Annex** section either of the Switch or the DN configuration object, or both. If it is set to a valid value, the DN option overrides the value set for the Switch.

- Configured in: Switch Application, DN Application
- Default Value: `0`
- Valid Values: `0, 1`
  - `0`—ICON preserves its former behavior; that is, ICON does not identify the PARENTPARTYID or the PARENTLINKTYPE in these event flows.
  - `1`—ICON correctly sets the value of the PARENTPARTYID and the PARENTLINKTYPE.
- Changes Take Effect: If set for the Switch, changes take effect after ICON is restarted; if set for the DN, changes take effect immediately.

> **Important**
>
> Interaction Concentrator does not support event flows in which EventRinging comes before EventDiverted for two-step transfer scenarios in which the transfer is completed before the call rings on the target DN. This is the case even if you set the `ring-divert` option value to 1.

Script Options

This section describes the ICON-related configuration options that you configure on the **Annex** tab of Script configuration objects of the interaction queue and simple routing (for a routing strategy) types. Interaction Concentrator processes these options.

callconcentrator Section
You must name this section callconcentrator in the configuration.

objects of type: interaction queue
om-memory-clean Option

objects of type: simple routing
om-activity-report Option

om-memory-clean

Specifies whether ICON immediately removes an interaction from memory when the interaction is placed in the interaction queue.

> **Important**
>
> The `om-memory-optimization` option must be set to `true` in order for `om-memory-clean` to work.

- Configured in: Script Application
- Default value: `0`
- Valid values:
  - `0`—ICON does not take any special action to remove an interaction when it is placed in the interaction queue.
  - `1`—ICON immediately removes an interaction when it is placed in the interaction queue.
  - Changes take effect: After restart

> **Important**
>
> Regardless of the value of this option, the removal of an interaction from ICON memory can be triggered by the value set for the `om-max-in-memory` option.

om-activity-report

Specifies whether ICON stores activity data that is related to a particular strategy. If the value is set to `false`, ICON does not store in IDB any data about parties for this strategy or any user data changes that are made by this strategy.

- Configured in: Script Application
- Default value: `true`
- Valid values: `true, false`
- Changes take effect: After restart

> **Important**
>
> Regardless of the value of this option, ICON will always process EventInteractionStopped events.

DAP Option

This section describes the ICON-related `role` configuration option that you configure on the **Options** tab of a Database Access Point (DAP) Application configuration object that specifies the IDB connection. Interaction Concentrator processes this option.

role

Specifies the type of data that this ICON instance processes and stores in IDB.

> **Important**
> Role assignments must be configured using only lower case (for example, `cfg`). ICON interprets uppercase (`CFG`) or mixed case (`Cfg`) settings as invalid and defaults to the `all` role.

- Configured in: ICON Application, `[callconcentrator]` Section; DAP Application, `[callconcentrator]` Section (for details relevant to setting the `role` option in the DAP Application, scroll to the end of this page)
- Default value: `all`
- Valid values: A comma-separated list including any of the following:
  - `all`—Stores all types of data.
  - `cfg`—Stores the initial configuration state and a history of configuration changes retrieved from Configuration Server.
  - `gcc`—Stores interaction-related and party-related information—that is, T-Server and Interaction Server data that pertains to voice and multimedia interactions, and the parties associated with those interactions.
  - `gls`—Stores T-Server and Interaction Server data that pertains to agent states and agent login sessions.
  - `gud`—Stores T-Server and Interaction Server data that pertains to the attached data associated with calls.
  - `gos`—In an environment with the Outbound Contact solution, stores OCS data that pertains to outbound calls and campaigns.
  - `lrm`—In an environment with License Reporting Manager, stores license reporting data.
- Changes take effect: After restart

> **Important**
> All ICON instances are assigned a predefined role, `svc`, to store service information about the ICON instance, for identification purposes, in IDB. The `svc` role cannot be turned off, and you do not need to specify it in the option value.

Any combination of the valid values can be used. Prefixing an option value with a tilde (~) excludes that type of data from ICON processing, and includes all other types. For example, the value `~cfg` deactivates ICON processing of configuration data, and activates processing and storage of all other

types of data. Ensure that the role that you specify for the ICON instance is consistent with the role that you specify for the DAP (see below for DAP-specific considerations).

Examples of correct settings:

- `role = cfg,gcc,gud`
- `role = all`
- `role = gcc,gud,gls,gos`
- `role = ~cfg`

(The last two examples are equivalent.)

DAP-Specific Notes
All types of ICON data go through the same DAP in the following cases:

- No `role` option is defined for the DAP.
- The `role` option is defined, and its value is explicitly set to `all`.
- You specified only one DAP Application object on the **Connections** tab of the ICON Application object.

> **Important**
>
> - Regardless of whether a given DAP handles all types of ICON data or a subset of them, a separate database connection is opened for each type of data.
> - Ensure that the role that you specify for the DAP is consistent with the role that you specify for the ICON instance.
> - A DAP cannot be assigned the `lrm` role. If you do so, it is ignored and the default value (`all`) is used.

Starting and Stopping

You can start and shut down Interaction Concentrator components by using the Management Layer, a startup file, a manual procedure, or Services Manager.

All of these methods usually require command-line parameters for a server application as well as an executable file name. The next section describes the command-line parameters that are common to most Genesys server applications. Subsequent sections describe the startup and shutdown procedures.

For information about using the Management Layer, startup files, and Services Manager for startup, see the Framework Deployment Guide.

Before Starting

Before You Start ICON

The following issues are important for you to consider before you attempt to start ICON.

Starting the cfg Role for an Oracle IDB

If the relational database management system (RDBMS) of the Interaction Database (IDB) that stores configuration-related data is Oracle, Genesys strongly recommends that you collect statistics on your IDB schema before you start the ICON instance that performs the `cfg` role. Collecting statistics before starting ICON significantly shortens the amount of time it takes to start up.

> **Tip**
> This applies only to the second and subsequent start-ups. The first time you start ICON in a new installation, the IDB is empty.

Verifying ICON Connections and Configuration

Before you attempt to start ICON, confirm that the connections and configuration options that have been configured for your ICON Application are correct for your deployment.

Connections

In general, do not change any connections on the Connections tab of the ICON Application during startup or runtime. Furthermore, do not disconnect from Configuration Server during startup.

- If ICON disconnects from Configuration Server during startup, ICON initialization will fail.
- If you remove or change other connections during startup, ICON might fail to initialize correctly.
- If you remove or change any connections during runtime, ICON functioning might be affected.
- You must restart Interaction Concentrator after a backup instance is configured of any application, such as OCS or T-Server, for which Interaction Concentration has a connection configured on the Connections tab. If you do not restart Interaction Concentrator, data from the affected application is not written to the database.

For more information about configuring connections, see the instructions for configuring the Connections tab in the procedure to configure the ICON Application object.

Configuration Options

Do not make changes to ICON configuration options during startup. You can make changes to ICON configuration options during runtime, but in some cases you must restart ICON for the changes to take effect. For more information, see Configuration Options.

Command-Line Parameters

Command-Line Parameters

The following startup command-line parameters are supported by Interaction Concentrator:

Parameter	Description
`-host`	The name of the host on which Configuration Server is running.
`-port`	The communication port that client applications must use to connect to Configuration Server.
`-transport-port`	The client-side port that ICON must use to connect to Configuration Server.
`-transport-address`	The TCP host address to be used for the transport-port.
`-app`	The exact name of an application as configured in the Configuration Database.
`-V`	The version of a component. Note that specifying this parameter does not start an application, but instead returns its version number. You can use either an uppercase letter (*V*) or lowercase letter (*v*).
`-lmspath`	The full path to the log messages files that an application uses to generate log events. (These files are the common file named `common.lms` and the application-specific file with the extension `*.lms`.) Use this parameter when the common and

Parameter	Description
	application-specific log message files are located in a directory other than the application's working directory—for example, when the application's working directory differs from the directory to which the application was originally installed. Note that if the full path to the executable file is specified in the startup command line (for instance, `c:\gcti\multiserver.exe`), the path that is specified for the executable file is used to locate the `*.lms` files, and the value of the `lmspath` parameter is ignored.

> **Warning**
> An application that does not locate its *.lms file at startup cannot generate application-specific log events and send them to Message Server.

Starting

Starting ICON

This section provides startup instructions for ICON server. You can start ICON in any of the following ways:

[+] Starting ICON with Solution Control Interface

Complete the following procedure to start ICON with Solution Control Interface (SCI).

Genesys recommends that the following applications be running before you start ICON:

- The DB Server that provides access to IDB.
- The relational database management system.
- T-Server.
- Outbound Contact Server, if ICON is configured to collect data from OCS.
- Interaction Server, if ICON is configured to collect data from Multimedia.

If you have configured ICON to store attached data, ensure that there is a proper attached data specification file in ICON's working directory. (By default, ICON uses the `ccon_adata_spec.xml` file.)

> **Tip**
> For a short period of time after starting or restarting, ICON may produce `[cp:...]` or `FSM` errors in the log. These errors occur when ICON encounters elements of interactions that it cannot resolve because the interactions were already in progress when ICON was started or restarted. You can safely ignore these errors.

For detailed instructions about starting the Genesys components on which Interaction Concentrator depends, see:

- Framework Deployment Guide
- Framework T-Server Deployment Guide for your particular T-Server type
- Framework DB Server User's Guide
- Outbound Contact Deployment Guide
- eServices Deployment Guide

1. On the list pane in the SCI Applications view, select your ICON Application object.
2. Do one of the following:
   - On the toolbar, click the Start button.
   - From the Action menu, select Start.
   - Right-click the Application object to access the shortcut menu, and then select Start.
3. In the confirmation box that appears, click **Yes.**

SCI starts your Interaction Concentrator application. You have completed all the steps necessary to start ICON using SCI.

[+] Starting ICON manually on UNIX

Complete the following procedure to start ICON manually on UNIX.

1. Go to the directory to which you have installed ICON.
2. Enter the name of the ICON executable, followed by the appropriate command-line parameters, using the following syntax:
   ```
   ./icon -host <hostname> -port <portno> -app <application>
   ```
   Where:
   - `hostname` is the name of the host on which Configuration Server is running.
   - `portno` is the communication port that client applications must use to connect to Configuration Server.
   - `application` is the name of the Interaction Concentrator Application object, as defined to Configuration Server.

> **Important**
> If the host name or application name contains spaces or hyphens (-), enclose them in double quotation marks.

For example, to start ICON with command-line parameters that specify the host as `cs-host`, the port as `2020`, and the name as `ICON 03`, enter the following:

```
./icon -host "cs-host" -port 2020 -app "ICON 03"
```

You have completed all the steps necessary to start ICON manually on UNIX.

[+] Starting ICON on Windows

Complete the following procedure to start ICON on Windows.

To start ICON from the `Start > Programs` menu, or from the console window.

1. Open a console window.
2. Go to the directory to which you installed Interaction Concentrator.
3. Enter the following command line:
   ```
   icon.exe -host <hostname> -port <portno> -app <application>
   ```
   Where:
   - `hostname` is the name of the host on which Configuration Server is running.
   - `portno` is the communication port that client applications must use to connect to Configuration Server.
   - `application` is the name of the Interaction Concentrator Application object, as defined to Configuration Server.

> **Important**
> If the host name or application name contains spaces or hyphens (-), enclose them in double quotation marks.

For example, to start ICON with command-line parameters that specify the host as `cs-host`, the port as `2020`, and the name as `ICON 03`, enter the following:

```
icon.exe -host "cs-host" -port 2020 -app "ICON 03"
```

You have completed all the steps necessary to start ICON on Windows.

[+] Starting ICON as a Windows Service

On Microsoft Windows platforms, by default, the installation process installs Interaction Concentrator as a Windows Service. If you stopped ICON from running as a Windows Service and need to start it again as a Windows Service, complete the following procedure.

1. Open the Windows Control Panel, and then double-click the **Services** icon. The **Services** dialog box opens.
2. In the **Services** list box, select your ICON service, and then click **Start.** (If you disabled Interaction Concentrator from operating as a Windows Service, the **Start** option for this application will not be available.)

You can install the Local Control Agent (LCA) as a Windows Service with the user interface disabled. In this case, all servers that are started through SCI are started without a console, unless you specifically select the **Allow Service to Interact with Desktop** check box for both LCA and ICON.

You have completed all the steps necessary to start ICON as a Windows service.

Stopping

Stopping ICON

To prevent ICON from self-starting, make sure that you clear the **autorestart** property in the ICON Application object.

[+] Stopping ICON with Solution Control Interface (Recommended)

If you are using LCA and SCS, complete the following procedure to stop ICON with SCI.

1. On the list pane in the SCI Applications view, select your ICON Application object.
2. Do one of the following:
   - On the toolbar, click **Stop.**
   - From the Action menu, select **Stop.**
   - Right-click the Application object to access the shortcut menu, and then select **Stop.**
3. In the confirmation box that appears, click **Yes.**

SCI stops your Interaction Concentrator application. You have completed all the steps to stop ICON using SCI.

[+] Stopping ICON on UNIX from the command line

Stop ICON on UNIX by using the following procedure.

- On the command line, enter the following:
  ```
  kill -SIGTERM <processid>
  ```

Where <processid> is the application's UNIX process ID.

You have completed all the steps to stop ICON from the command line.

[+] Stopping ICON on UNIX from the console window

- From the active console window, press **CTRL+C.**

You have completed all the steps to stop ICON from the console window.

[+] Stopping ICON on Windows from the console window

If ICON is running as an application—not as a Windows Service—stop it using the following procedure.

- From the application's console window, press **CTRL+C.**

You have completed all the steps to stop ICON from the console window.

[+] Stopping ICON as a Windows Service

If you are running ICON as a Windows Service, you should stop it only from the Services Control Manager. To stop Interaction Concentrator running as a Windows Service, use the following procedure.

1. Open the Control Panel, and then double-click the **Services** icon. The **Services** dialog box opens.
2. In the **Services** list box, select your ICON service, and then click **Stop.**

You have completed all the steps to stop ICON running as a Windows Service.

Troubleshooting

This section describes problems that you might encounter when starting or running your Interaction Concentrator (ICON) application, and how to resolve them.

Restrictions

Restrictions

To avoid a wide range of startup and runtime problems, observe the following restrictions:

- Do not disconnect ICON from Configuration Server during startup.
- Do not change any connections on the **Connections** tab of the ICON Application during runtime.

Startup Problems

Startup Problems

The following are the most common startup problems:

[+] ICON does not connect to the Configuration Server

Possible causes of this problem are as follows:

Command-line parameters on the ICON Application object's **Server Info** tab incorrectly specify the Configuration Server host and port.

- **Solution:** Correct the command-line parameters and restart the application. For more information about the command-line parameters, Command-Line Parameters.

Configuration Server is not running, or it is inaccessible over the network.

- **Solution:** Start Configuration Server or re-establish the network connection.

[+] ICON Exits at Startup

See the ICON log file for the reasons for the startup failure. Possible reasons include:

The Application name specified in the ICON startup command line does not correspond to any existing Application object in the Configuration Layer.

- **Solution:** Create the Application object. For more information about creating and configuring the ICON Application, see Deploying Interaction Concentrator.

The Application name specified in the ICON startup command line refers to an Application object that is not of the Call Concentrator application type.

- **Solution:** Remove the Application object of the incorrect type, and then use the correct template to create a new Application object of the Call Concentrator type. For more information about creating and configuring the ICON Application object, see Deploying Interaction Concentrator.

There is no assignment to a Database Access Point (DAP) Application object on the **Connections** tab of the ICON Application object.

- **Solution:** Add to the ICON Application object's **Connections** tab any DAP Application objects through which this ICON instance will access Interaction Databases (IDBs).

The DAP Application object assigned on the ICON Application object's **Connections** tab does not have an associated DB Server application.

- **Solution:** Associate a DB Server with the DAP Application object. For more information, see the DAP tab in the Configuration and Installation page.

The ICON instance has been configured to process call attached data (role = gud), but ICON cannot open the file specified in the `adata-spec-name` configuration option. The following error message in the log file indicates the existence of this condition:

```
Std 02016 Unable to open attached data file '<attached data specification file
name>', error code XXX
```

- **Solution:** Verify the following and correct as required.
  - The file specified in the `adata-spec-name` configuration option exists. If the file does not exist, create a new one or use the default attached data specification file (`ccon_adata_spec.xml`) provided in the Interaction Concentrator installation package.
  - The Interaction Concentrator user (the account under which ICON has been started) has the required permissions to read the attached data specification file.

> **Important**
>
> If you make changes to the attached data specification file, you must restart ICON for them to take effect.

The persistent queue file has become corrupted.

- **Solution:** Force ICON to create a new persistent queue file by doing one of the following:
  - Using operating system commands, move or rename the corrupted `.pq` file. On restart, ICON will create a new `.pq` file with the original file name in the original location.

- Reset the `pq-dbname` configuration option in the ICON Application object. On restart, ICON will create a new `.pq` file with the new file name in the specified location. See the `pq-dbname` configuration option under Persistent Queue in the callconcentrator section on the Configuration Options page.
  With either method, all unprocessed data in the old `.pq` file will be lost to ICON and IDB.

There is no free disk space on the disk where the `apstorage.db` file resides.

- **Solution:** Free up memory on the disk or add more disk memory. For more information about the `apstorage.db` file, see "Populating Agent Login Session Data" in the Interaction Concentrator 8.1 User's Guide.

Runtime Problems

Runtime Problems

The following are the most common runtime problems:

[+] ICON does not connect to T-Server or Interaction Server

Possible causes of this problem are as follows:

There is no assignment to the T-Server Application object or the Interaction Server Application object on the ICON Application object's **Connections** tab.

- **Solution:** Add to the ICON Application object's **Connections** tab any T-Server or Interaction Server Application objects from which this ICON instance will receive interaction-related information.

The T-Server or Interaction Server application is not running, or it is not accessible over the network.

- **Solution:** Start the application or re-establish the network connection.

The T-Server or Interaction Server Application object cannot connect to its Switch link.

- **Solution:** See the applicable troubleshooting guide for your particular T-Server or Multimedia Interaction Server.

The release of the T-Server or Interaction Server Application object is not compatible with Interaction Concentrator. T-Server release 7.2 is the minimum version required by any release of Interaction Concentrator. Multimedia Interaction Server release 7.5 is the minimum version required for Interaction Concentrator support of eServices. For more information about Interaction Concentrator compatibility and interoperability with other Genesys components, see "Compatibility" on the Planning Your Deployment page.

- **Solution:** Upgrade the T-Server or Interaction Server Application object to a compatible release.

The Switch object associated with the T-Server Application object does not have all the necessary DN

objects configured.

- **Solution:** Create the DN objects. For more information, see the *Deployment Guide* for your particular T-Server.

[+] ICON Does Not Receive Call-Related Events from T-Server

Possible causes of this problem are as follows:

ICON was not restarted after changes were made on the ICON Application object's **Connections** tab.

- Solution: Stop ICON, then restart.

ICON was not restarted after a backup instance was configured of a T-Server to which Interaction Concentrator has a connection configured on the **Connections** tab.

- **Solution:** Stop ICON, then restart.

There is no connection between the ICON Application object and T-Server.

- **Solution:** See "No Connection to T-Server or Interaction Server" on the Runtime Problems tab.

[+] ICON Does Not Write Information to the Database

Possible causes of this problem are as follows:

The database parameters are incorrectly specified on the DAP Application object. These parameters include the user name and password.

- **Solution:** Specify the correct values on the DAP Application object's **DB Info** tab, then restart ICON. For more information, see the DAP tab in the Configuration and Installation page.

DB Server is not running, or it is inaccessible over the network.

- **Solution:** Start DB Server or re-establish the network connection.

The RDBMS server is not available, or the IDB to which DB Server is trying to connect is not available.

- **Solution:** Take the necessary steps to make the database server and database available.

The DAP Application object has been configured for a role that prevents it from writing certain classes of information to the database.

- **Solution:** Reconfigure the `role` option for the DAP Application object. Restart ICON. For more information about configuring a DAP, see the DAP tab on the Configuration and Installation page, and the description of the role configuration option under ICON Role in the callconcentrator section on the Configuration Options page.

IDB has not been initialized by the Interaction Concentrator initialization scripts.

- **Solution:** Run the Interaction Concentrator initialization scripts. For more information, see the IDB tab on the Configuration and Installation page.

ICON was not restarted after changes were made on the ICON Application object's Connections tab.

- **Solution:** Stop ICON, then restart.

ICON was not restarted after a backup instance was configured of a T-Server to which Interaction Concentrator has a connection configured on the **Connections** tab.

- **Solution:** Stop ICON, then restart it.

Records are accumulating in the in-memory queue and are not being written to IDB.

- **Solution:** This might not be a problem. Configuration options control whether a size threshold or timeout triggers the transfer of records from the in-memory queue to the persistent queue, from which the records are then written to IDB. Wait for the event that triggers the transfer, and re-evaluate your configuration as necessary. For more information see the In-memory queue configuration options in the callconcentrator section on the Configuration Options page.

The program logic consistently produces an error because of incorrect RDBMS settings. For example, there may be insufficient free space available on the RDBMS for data storage, or the rollback segment may be too small.

- **Solution:** Review the error messages reported in the ICON log file. If you have configured an HTTP Listener, you can also view the error messages on the Database Writer performance counter web page (for more information, "Monitoring Interaction Concentrator" in the Interaction Concentrator 8.1 User's Guide). Provide the appropriate fix on the RDBMS side. For example, if the error messages cite no free space available for data storage, increase the table space.
  If the error was entirely related to the RDBMS problem, you do not need to restart ICON or perform any manipulation of the persistent queue (`.pq` file). However, if the `.pq` file has become corrupted and there are additional errors in the program logic, you must replace the `.pq` file.

[+] ICON Has Lost Synchronization with the Configuration Database

There are a number of reasons why ICON might lose synchronization with the Configuration Database, especially following a shutdown of ICON.

Loss of synchronization has the following impact on IDB:

- ICON fails to capture data about configuration objects created while ICON was stopped.
- ICON does not mark configuration data as deleted in cases where the applicable configuration objects were deleted while ICON was stopped.
- ICON fails to capture changes in associations between objects (while it is stopped).

Solution: If you suspect that your configuration data in IDB is inconsistent with Configuration Database, perform a manual resynchronization. See "How to Resynchronize Configuration Data" in the Interaction Concentrator 8.1 User's Guide.

[+] ICON Does Not Function Correctly

Possible causes of this problem are as follows:

A connection configured on the Connections tab of the ICON Application was removed or changed while ICON was operating.

- **Solution:** Stop ICON. Verify that the connections that have been configured on the Connections tab of the ICON Application object are as required for the deployment, then restart ICON. For more information about configuring connections, see "Configure the Connections tab" on the ICON tab of the Configuration and Installation page.

Merge Problems

Merge Problems

The most common problems encountered in executing the merge procedure (`gsysIRMerge` or `gsysIRMerge2`) are as follows:

For more information about the merge procedure, “Merge Stored Procedure”in the Interaction Concentrator 8.1 User’s Guide.

[+] Merge Procedure Does Not Complete Successfully

In general, the most likely reason the merge procedure fails is an inconsistency in IDB. The database inconsistency might be introduced by ICON, by the downstream reporting application, through manual intervention, or in some other way. For example, if ICON writes a duplicate `G_IS_LINK` record while the merge procedure is executing, the RDBMS might report a primary key violation. Describing the possible causes of this problem in detail is beyond the scope of this document.

The following tables store information about the state of the merge procedure:

- `GSYS_PENDING_IR`
- `GSYS_PENDING_LINK`
- `GSYS_SYSPROCINFO`

Solution: Review the error messages reported in the ICON log file, and take appropriate action to resolve the cause of the failure. You might also have to reset the merge procedure so that it recovers from its failed state (see “Merge Procedure Recovery” below). Then restart the merge procedure.

Merge Procedure Recovery
Interaction Concentrator provides a stored procedure, `gsysIRMergeReset`, to simplify the steps to reset the merge procedure to recover from a failed state. To invoke the procedure, use an SQL statement like the following (the exact syntax depends on the RDBMS):

```
EXEC gsysIRMergeReset
```

> **Important**
> Under some circumstances, merge procedure recovery is not required. For example, the merge procedure may fail to complete successfully as a result of a deadlock condition. In this case, no special action is required other than to run the procedure again. However, if an error is discovered in the merge procedure, execute the stored procedure to reset the merge procedure.

[+] Merge Procedure Does Not Execute

Possible causes of this problem are as follows: The stored procedure was called incorrectly.

- **Solution:** Verify the syntax of the call to execute gsysIRMerge or gsysIRMerge2, and correct the execution command as required. For more information, see "Executing the Merge Procedure" in the Interaction Concentrator 8.1 User's Guide.

There is an error in the database or in database performance that is not specifically related to the merge procedure or to ICON—for example, insufficient disk space or insufficient privileges.

- **Solution:** Review the error messages reported in the ICON log file. Provide the appropriate fix that the RDBMS requires, then restart the merge procedure.

The database error might be related to an inconsistency in IDB, in the sense that it was exposed or induced by an inconsistency in IDB, or resulted in an inconsistency in IDB. In these cases, reset the merge procedure (see "Merge Procedure Recovery" above), then restart the merge procedure. If the merge procedure still fails to execute, contact Genesys Customer Care.

[+] Merge Procedure Performance Is Slow or Unstable

Possible causes of this problem are as follows:

On a DB2 platform, default values of certain database parameters result in an excessive number of deadlocks.

- **Solution:** Contact Genesys Customer Care for assistance with database locking issues.

There is an inconsistency in IDB that does not cause the merge procedure to fail, but that significantly interferes with merge procedure performance.

- **Solution:** Reset the merge procedure (see "Merge Procedure Recovery," above), then restart the merge procedure. If the problem persists, review database settings and try general database tuning adjustments. If the problem still persists, contact Genesys Customer Care.

Attached Data Specification File

This section presents the XML schema definition for processing key-value pairs (KVPs) from the attached data that T-Server or, if applicable, Interaction Server provides with TEvents. If you require ICON to store attached data in IDB, create an attached data specification for ICON to use, based on the information in this section.

> **Important**
>
> If you change the XML file, you must restart ICON in order for the changes to take effect.

Schema Definition

Schema Definition

The following is the XML schema definition for your attached data specification.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsd:schema targetNamespace="http://www.genesyslab.com/standards/icon/ed1"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:icon="http://www.genesyslab.com/standards/
icon/ed1" elementFormDefault="qualified" attributeFormDefault="unqualified">
<xsd:annotation>
  <xsd:documentation>Attached data configuration specification</xsd:documentation>
</xsd:annotation>

<xsd:simpleType name="AdataHistoryType">
<xsd:restriction base="xsd:string">
                  <xsd:enumeration value="none" />
<xsd:enumeration value="first" />
<xsd:enumeration value="last" />
<xsd:enumeration value="all" />
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="AdataCallHistoryType">
<xsd:restriction base="xsd:string">
<xsd:enumeration value="first" />
<xsd:enumeration value="last" />
        </xsd:restriction>
 </xsd:simpleType>

<xsd:simpleType name="AdataCallFieldType">
<xsd:restriction base="xsd:string">
                  <xsd:enumeration value="customer-segment" />
<xsd:enumeration value="service-type" />
                  <xsd:enumeration value="service-subtype" />
```

```
                  <xsd:enumeration value="busines-result" />
                  <xsd:enumeration value="customer-id" />
                  <xsd:enumeration value="transaction-id" />
                  <xsd:enumeration value="cause-id" />
                  <xsd:enumeration value="account-id" />
                  <xsd:enumeration value="destination-id" />
                  <xsd:enumeration value="target-id" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="AdataCallCustFieldType">
<xsd:restriction base="xsd:string">
                  <xsd:enumeration value="cust-data-1" />
                  <xsd:enumeration value="cust-data-2" />
                  <xsd:enumeration value="cust-data-3" />
                  <xsd:enumeration value="cust-data-4" />
                  <xsd:enumeration value="cust-data-5" />
                  <xsd:enumeration value="cust-data-6" />
                  <xsd:enumeration value="cust-data-7" />
                  <xsd:enumeration value="cust-data-8" />
                  <xsd:enumeration value="cust-data-8" />
                  <xsd:enumeration value="cust-data-10" />
                  <xsd:enumeration value="cust-data-11" />
                  <xsd:enumeration value="cust-data-12" />
                  <xsd:enumeration value="cust-data-13" />
                  <xsd:enumeration value="cust-data-14" />
                  <xsd:enumeration value="cust-data-15" />
                  <xsd:enumeration value="cust-data-16" />
                  <xsd:enumeration value="cust-data-17" />
                  <xsd:enumeration value="cust-data-18" />
                  <xsd:enumeration value="cust-data-19" />
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="AdataHistoryRecord">
          <xsd:attribute name="name" type="xsd:string" />
          <xsd:attribute name="source" type="icon:AdataSource" />
          <xsd:attribute name="history" type="icon:AdataHistoryType" />
</xsd:complexType>
<xsd:complexType name="AdataCallRecord">
          <xsd:attribute name="name" type="xsd:string" />
<xsd:attribute name="source" type="icon:AdataSource" />
          <xsd:attribute name="history" type="icon:AdataCallHistoryType" />
          <xsd:attribute name="field" type="icon:AdataCallFieldType" />
</xsd:complexType>
<xsd:complexType name="AdataCallCustRecord">
          <xsd:attribute name="name" type="xsd:string" />
          <xsd:attribute name="source" type="icon:AdataSource" />
          <xsd:attribute name="history" type="icon:AdataCallHistoryType" />
          <xsd:attribute name="field" type="icon:AdataCallCustFieldType" />
</xsd:complexType>
<xsd:element name="adata_spec">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="public" type="icon:AdataHistoryRecord" />
                  <xsd:element name="secure" type="icon:AdataHistoryRecord" />
                  <xsd:element name="call" type="icon:AdataCallRecord" />
                  <xsd:element name="call-cust" type="icon:AdataCallCustRecord" />
                  <xsd:element name="call-cust1" type="icon:AdataCallCustRecord" />
                  <xsd:element name="call-cust2" type="icon:AdataCallCustRecord" />
  </xsd:sequence>
  </xsd:complexType>
  </xsd:element>
  </xsd:schema>
```

Basic Sample

Sample Basic Attached Data Specification

The following is an example of a basic `adata_spec.xml` file for voice and multimedia interactions. A similar sample file, `ccon_adata_spec_example.xml`, is available in your ICON installation folder after you install the application.

Attached data that you specify in the `<public>` and `<secure>` storage type sections of the specification file applies to all interactions. Attached data that you specify in the `<call>` and `<call-cust>` sections applies to voice calls only.

This sample includes examples of keys in a nested key-value list (in this case, attached data from Universal Routing Server [URS]). For more information about specifying URS attached data, see Universal Routing Server Attached Data on the Storing Attached Data page.

```
<?xml version="1.0" encoding="utf-8" ?>
<adata_spec>
<public>
        <key name = "u_key1" source="userdata"    history ="first"/>
<key name="u_key2" source="userdata" history="all" />
<key name="r_key1" source="reasons" history="last" />
<key name="RTargetAgentSelected" source="userdata" history="first" />
<key name="RTargetUsed/RTargetName" source="userdata" history="first" />
<key name="RTargetUsed/RTargetType" source="userdata" history="first" />
</public>
<secure>
        <key name="RTenant" source="userdata" history="first" />
</secure>
<call>
        <key name = "customer-segment" source="userdata" history ="first" field="customer-
segment"/>
<key name = "svc_class_cd" source="userdata" history ="first" field="service-type"/>
<key name = "CCTP_CALLTYPE" source="userdata" history ="first" field="service-subtype"/>
<key name = "cid" source="userdata" history ="first" field="customer-id"/>
<key name = "transact_tn_final" source="userdata" history ="first" field="transaction-id"/>
</call>
<call-cust>
        <key name = "STATE" source="userdata"  history ="last" field="cust-data-1"/>
</call-cust>
<call-cust1>
        <key name = "PegTD" source="userdata"  history ="last" field="cust-data-2"/>
</call-cust1>
<call-cust2>
        <key name = "vrapp_ctl_lang" source="userdata"  history ="last" field="cust-data-3"/>
</call-cust2>
</adata_spec>
```

> **Important**
>
> ICON ignores duplicate keys. Only the first occurrence of a key name will be used to update the applicable database table.

In the following example, only the cust_data_2 field in the G_CALL_USERDATA_CUST table is populated by the value corresponding to key name = customer-segment (if it is present in userdata). The field cust_data_2 in table G_CALL_USERDATA_CUST1 is not updated.

```
<call-cust>
<key name = "customer-segment" source="userdata" history ="first" field="cust-data-2"/>
<key name = "STATE" source="userdata" history ="last" field="cust-data-1"/>
</call-cust>

<call-cust1>
<key name = "customer-segment" source="userdata" history ="first" field="cust-data-1"/>
<key name = "PegTD" source="userdata" history ="last" field="cust-data-2"/>
</call-cust1>
```

Multimedia Sample

Sample Specification for Multimedia Attached Data

The following is an example of an adata_spec.xml file that has been modified to capture multimedia-specific attached data. A similar sample file, ccon_adata_spec_mcr_example.xml, is available in your ICON installation folder after you install the application.

With this attached data specification, values for the following predefined and customer-defined keys and attributes are stored in IDB in the GM_F_USERDATA or the GM_L_USERDATA table.

The GM_F_USERDATA table

The following user data key names and event attributes are always stored, whether or not they are defined explicitly in the XML file.

```
<mcr-f>
<key name = "FromPersonal" source="userdata"  history ="first" field="mcr-from-name"/>
<key name = "FromAddress" source="userdata"  history ="first" field="mcr-from-address"/>
<key name = "Subject" source="userdata"  history ="first" field="mcr-subject"/>
<key name = "Origination_Source" source="userdata"  history ="first" field="mcr-origin-
source"/>
<key name = "attr_itx_subtype" source="userdata" history="first"  field="mcr-sub-type”/>
<key name = "attr_itx_received_at" source="userdata" history="first"  field="mcr-received-
at”/>
</ mcr-f>
```

You can use the XML to define four additional keys that will be stored in the GM_F_USERDATA table. For example, you might define keys similar to the following ones:

```
<key name = "Custom Key Name - 1" source="userdata"  history ="first" field="mcr-reserved-1"/>
<key name = "Custom Key Name - 4" source="userdata"  history ="first" field="mcr-reserved-4"/>
```

The GM_L_USERDATA table

The following user data key names are always stored, whether or not they are defined explicitly in the

XML file.

```
<key name = "SuggestedResponseID"         source="userdata"  history ="last" field="mcr-
suggested-response"/>
<key name = "AutoResponseID"                 source="userdata"  history ="last" field="mcr-
auto-response"/>
<key name = "AutoACKID"                 source="userdata"  history ="last" field="mcr-auto-
ack"/>
<key name = "Stop_Reason"                 source="userdata"  history ="last" field="mcr-stop-
reason"/>
<key name = "ContactId"                 source="userdata"  history ="last" field="mcr-ucs-
contact-id"/>
```

Predefined event attribute: `attr_reason_system_name`

Customer-defined keys: `SuggestedResponseID, AutoResponseID, AutoACKID`

For information about the IDB fields in which the values will be stored, see IDB Fields.

Sample adata_spec.xml File

In the `GM_F_USERDATA` section of the sample `adata_spec.xml` file that follows, the `FromPersonal`, `Subject`, and `Origination_Source` key names are defined, though they would appear in the table in any case. The `CalledBack` key name is user-defined and would not appear without the explicit definition. The event attributes are also pre-defined. In the `GM_L_USERDATA` table section, all of the key names are ones that are also predefined.

```
<?xml version="1.0" encoding="utf-8" ?>
<adata_spec>
<mcr-f>

<key name = "FromPersonal" source="userdata"  history ="first" field="mcr-from-name"/>
<key name = "FromAddress" source="userdata"  history ="first" field="mcr-from-address"/>
<key name = "Subject" source="userdata"  history ="first" field="mcr-subject"/>
<key name = "Origination_Source" source="userdata"  history ="first" field="mcr-origin-
source"/>
<key name = "attr_itx_subtype" source="userdata" history="first"  field="mcr-sub-type”/>
<key name = "attr_itx_received_at" source="userdata" history="first"  field="mcr-received-
at”/>
</ mcr-f>
<mcr-l>

<key name = "SuggestedResponseID" source="userdata" history ="last" field=" mcr-suggested-
response "/>
<key name = "AutoResponseID" source="userdata" history ="last" field=" mcr-auto-response "/>
<key name = "AutoAckID" source="userdata" history ="last" field=" mcr-auto-ack "/>
<key name = " Stop_Reason " source="userdata" history ="last" field=" mcr-stop-reason "/>
<key name = "ContactID" source="userdata" history ="last" field=" mcr-ucs-contact-id "/>
</ mcr-l>
</adata_spec>
```

Custom Sample

Sample Specification for Customized Attached Data

The following is an example of an `adata_spec.xml` file that has been modified for customized attached data processing.

You must create a custom dispatcher stored procedure to process the custom attached data. For an example of the script to create a custom dispatcher stored procedure, see Sample Script for Custom Attached Data.

```
<?xml version="1.0" encoding="utf-8" ?>
<adata_spec>
        <cust-disp-group-1>
                <key name = "name1_1" source="userdata" history ="first" field="cust-
int-1"></key>
<key name = "name2_1" source="userdata" history ="last" field="cust-int-2"></key>
...
<key name = "name17_1" source="userdata" history ="last" field="cust-int-17"></key>
<key name = "name18_1" source="userdata" history ="last" field="cust-str-1"></key>
<key name = "name19-1" source="userdata" history ="last" field="cust-str-2"></key>
...
<key name = "name34_1" source="userdata" history ="last" field="cust-str-17"></key>
        </cust-disp-group-1>
        <cust-disp-group-2>
                <key name = "name1_2" source="userdata" history ="first" field="cust-
int-1"></key>
<key name = "name2_2" source="userdata" history ="last" field="cust-int-2"></key>
...
<key name = "name17_2" source="userdata" history ="last" field="cust-int-17"></key>
<key name = "name18_2" source="userdata" history ="last" field="cust-str-1"></key>
<key name = "name19_2" source="userdata" history ="last" field="cust-str-2"></key>
...
<key name = "name34_2" source="userdata" history ="last" field="cust-str-17"></key>
        </cust-disp-group-2>
...
        <cust-disp-group-16>
                <key name = "name1_3" source="userdata" history ="first" field="cust-
int-1"></key>
<key name = "name2_3" source="userdata" history ="last" field="cust-int-2"></key>
...
<key name = "name17_3" source="userdata" history ="last" field="cust-int-17"></key>
<key name = "name18_3" source="userdata" history ="last" field="cust-str-1"></key>
<key name = "name19_3" source="userdata" history ="last" field="cust-str-2"></key>
...
<key name = "name34_3" source="userdata" history ="last" field="cust-str-17"></key>
        </cust-disp-group-16>
</adata_spec>
```

Sample Script for Custom Attached Data

This page presents a sample SQL script to create a custom dispatcher stored procedure—gudCustDISP1 or gudCustDISP2—and a custom attached data storage table in your Interaction Database (IDB) schema.

The sample script in this appendix is for Microsoft SQL Server (MSSQL). After you install the Interaction Concentrator (ICON) application, sample scripts for each supported RDBMS type–SampleProc_<db_type>.sql–are available in the scripts subfolder in the directory to which you installed the application.

> **Tip**
> Carefully verify the syntax and operation of your modified gudCustDISP1 or gudCustDISP2 stored procedure. Any types of errors or RDBMS violations that the custom dispatcher stored procedure produces can affect ICON processing of all other attached data for voice calls and multimedia interactions.

The following MSSQL sample script illustrates how you can create a custom attached data storage table (G_SAMPLE_CUST_ADATA) and modify the gudCustDISP1 or gudCustDISP2 stored procedure in the CoreProcedures_<db_type>.sql script. The modified stored procedure stores arguments in the G_SAMPLE_CUST_ADATA table.

```
Sample Custom Dispatcher
/*==============================================================*/
/* Table: G_SAMPLE_CUST_ADATA                                   */
/*==============================================================*/
create table G_SAMPLE_CUST_ADATA (
   ID                   numeric(16)          identity,
   CALLID               varchar(50)          not null,
   CALL_TS              int                  not null,
   SWITCH_ID            int                  not null,
   TENANT_ID            int                  not null,
   C_INT_1              int                  null,
   C_INT_2              int                  null,
   ...
   C_INT_34             int                  null,
   C_STR_1              varchar(10)          null,
   C_STR_2              varchar(10)          null,
   ...
   C_STR_34             varchar(10)          null,
   GSYS_DOMAIN          int                  null,
   GSYS_PARTITION       int                  null,
   GSYS_SYS_ID          int                  null,
   GSYS_SEQ             bigint               null,
   GSYS_USEQ            bigint               null,
   GSYS_TS              datetime             null,
   GSYS_TC              int                  null,
   GSYS_EXT_VCH1        varchar(255)         null,
   GSYS_EXT_VCH2        varchar(255)         null,
   GSYS_EXT_INT1        int                  null,
   GSYS_EXT_INT2        int                  null,
```

```
   constraint PK_G_SAMPLE_CUST_ADATA primary key  (ID)
)
go

/*==============================================================*/
/* Index: IDX_G_SAMPLE_CDATA_TS                                 */
/*==============================================================*/
create   index IDX_G_SAMPLE_CDATA_TS on G_SAMPLE_CUST_ADATA (
CALL_TS ASC
)
go

-- ============================================================ --
-- Name:   -- gudCustDISP1 --
-- Group:  User data related procedures
-- Brief:  -- gudCustDISP DISPATCH --
-- ============================================================ --
DROP PROCEDURE gudCustDISP1
go

CREATE PROCEDURE gudCustDISP1
        @GROUPID                INTEGER,
        @CALLID                 VARCHAR(64),
        @P_CALL_TS              INTEGER,
        @SWITCHID               INTEGER,
        @TENANTID               INTEGER,
        @TS_S                   INTEGER,
        @TS_MS                  INTEGER,
        @P_STR_1                VARCHAR(255),
        @P_STR_2                VARCHAR(255),
        ...
        @P_STR_17               VARCHAR(255),
        @P_INT_1                INTEGER,
        @P_INT_2                INTEGER,
        ...
        @P_INT_17               INTEGER,
        @P_STR_RES1             VARCHAR(255),
        @P_STR_RES2             VARCHAR(255),
        @P_STR_RES3             VARCHAR(64),
        @P_STR_RES4             VARCHAR(255),
        @P_INT_RES1             INTEGER,
        ...
        @P_INT_RES7             INTEGER,
        @DOMAIN                 INTEGER,
        @PARTITION              INTEGER,
        @SYS_ID                 INTEGER,
        @SYS_SEQN               NUMERIC(20,0),
        @SYS_TS                 DATETIME
AS
BEGIN

    -- Insert first portion in the long table --
    IF (@GROUPID = 1)
    BEGIN
        INSERT INTO G_SAMPLE_CUST_ADATA(
                CALLID,
                CALL_TS,
                SWITCH_ID,
                TENANT_ID,
                C_INT_1,
                C_INT_2,
                ...
                C_INT_17,
```

```
                C_STR_1,
                C_STR_2,
                ...
                C_STR_17,
                GSYS_DOMAIN,
                GSYS_PARTITION,
                GSYS_SYS_ID,
                GSYS_SEQ,
                GSYS_USEQ,
                GSYS_TS)
        VALUES (
                @CALLID,
                @P_CALL_TS,
                @SWITCHID,
                @TENANTID,
                @P_INT_1,
                @P_INT_2,
                ...
                @P_INT_17,
                @P_STR_1,
                @P_STR_2,
                ...
                @P_STR_17,
                @DOMAIN,
                @PARTITION,
                @SYS_ID,
                @SYS_SEQN,
                0,
                @SYS_TS)

    END

    -- Update record and specify more fields --
    ELSE
    IF (@GROUPID = 2)
    BEGIN
        UPDATE G_SAMPLE_CUST_ADATA SET
                C_INT_18                = @P_INT_1,
                ...
                C_INT_34                = @P_INT_17,
                C_STR_18                = @P_STR_1,
                C_STR_19                = @P_STR_2,
                ...
                C_STR_34                = @P_STR_17,
                GSYS_USEQ               = @SYS_SEQN
        WHERE CALLID = @CALLID AND CALL_TS = @P_CALL_TS


    END
END
go
```

Appendix: Migration Procedures

> **Important**
> This topic contains migration information only for 8.1.4 releases. For instruction on how to migrate to earlier releases of Interaction Concentrator, see the Interaction Concentrator chapters in the Genesys Migration Guide.

Migration Preliminaries

Complete the following procedures before starting your migration of Interaction Concentrator:

1. Migrate Management Framework, as applicable for your deployment.
2. Upgrade other prerequisite Genesys components (for example, T-Server, Interaction Server, or Universal Routing Server), as applicable for your deployment.
3. Update the contact center configuration (for example, Place Groups, Agent Groups, and DNs).

Migration Procedures

Main Steps

1. Migrate your ICON Server.
2. Check whether you need to update your IDB schema.
3. If necessary, migrate your IDB schema.

In a High Availability Environment

Genesys recommends that you run both ICONs in each ICON pair simultaneously for at least one day before the migration. At the minimum, the time the ICONs run simultaneously must exceed the time set in the Genesys Info Mart **max-call-duration** configuration option.

The recommended upgrade plan is the following:

- Stop one member of an ICON pair, migrate that ICON instance, and then restart it. Make sure it starts successfully. If not, roll back the migration (see Rolling Back the Migration).
- If the ICON migration was successful, wait until the longest active interaction in the environment finishes (usually 24 hours is enough). Then—assuming the migrated ICON has been operating normally—stop the other member of the HA pair of ICONs and perform the same migration procedure, again ensuring that the migration is successful.

In an Environment with Genesys Info Mart

> **Important**
> Your migration should be done in a working environment with an available ICON database and the associated ICON DBServer.

If Genesys Info Mart is running in your environment, stop all Info Mart extraction jobs before you start your ICON migration. Restart Job_ExtractICON only after you have successfully completed the migration of the ICON instances from which Info Mart is extracting data.

If you have multiple ICON applications, repeat the migration procedure for each ICON instance (including each ICON in any HA pair).

Migrating the ICON Server

Follow these steps to migrate the server component of Interaction Concentrator to the latest release:

1. Find a timeframe for this migration during which ICON has minimal activity.
2. Stop ICON, either using Genesys Solution Control Interface (SCI), Genesys Administrator/Genesys Administrator Extension, or manually. This may take several minutes. Make sure not to restart ICON until after the database update is complete.
3. Back up the Interaction database (IDB).
4. If you have customized the gudCustDISP1 or gudCustDISP2 dispatcher stored procedures and saved them with the default (Genesys-provided) names, back up the procedures and restore them after the upgrade.
5. Back up the entire ICON directory in case you need to roll back the migration.
   - Back up your customer-related files, which have the following suffixes:
     - ***.db**
     - ***.pq**
     - ***.xml**—You can locate this file, which by default is named **ccon_adata_spec.xml**, in the directory where the current version of Interaction Concentrator is installed.
   - Save the scripts that drop outdated stored procedures. To do so, save the **drop_<version>_<db>.sql** file to be used after successful migration completion to drop the old stored procedures. Keep the old stored procedures for approximately one week. Then you can drop them, if you prefer, but there is no need to.
6. Run the setup file (**install.sh** or **setup.exe**) provided in your installation package.

> **Important**
>
> These instructions assume you are installing the new version of ICON in the same location as the old one. If you install the upgraded version into a different directory from the existing version, verify that you are launching the correct ICON version when you restart. For example, you might need to adjust the Windows Service settings. For instructions on how to configure ICON start up, see Starting and Stopping.

7. Restore the backup customer-related files saved in Step 5.
   - Replace the files installed in the folder where you deployed ICON 8.1.x during migration with your saved files.
8. For environments running any Genesys Info Mart release prior to 8.5.007.xx, run the **update_idb_for_gim.sql** or **update_idb_for_gim_mm.sql** script, as appropriate, every time you migrate to a new release of Interaction Concentrator.

Checking Your IDB Schema Version

Check if an ICON database update is necessary by comparing database schema versions.

1. The database schema version of the new installation can be identified from the name of the following file: **drop_<*VERSION*>_<*db_type*>.sql**
   where
   - <*VERSION*> stands for the new database version. The version has the format *N.N.NNN.NN*.
   - <*db_type*> stands for the short name of the database type.

   For example, the filename might be: **drop_8.1.512.xx_postgre.sql**.
2. The currently-installed RDBMS version can be identified by either the name of the **drop_<*VERSION*>_<*db_type*>.sql** file in the previous installation of ICON or using the following database query:
   `SELECT VAL FROM G_DB_PARAMETERS where SECT = 'schema' and OPT = 'version'`

> **Important**
>
> If both the current and new ICON schema versions are the same, you do not need to update your IDB. The version of the database schema does not necessarily match the release number of the component. For example, for Interaction Concentrator release 8.1.509.09, the database schema version is 8.1.508.05.

Updating IDB

The update includes an upgrade of the stored procedures and it might include a change to the

database schema.

To perform the schema upgrade, apply all the upgrade scripts sequentially in the order of increasing versions starting from the lowest numbered upgrade script version above the current (old) ICON database schema version through the highest version found in the directory.

The Interaction Concentrator installation package may contains one or more upgrade schema scripts in the following format: **Upgrade_N.N.NNN.NN_<db_type>.sql**
where:

- *N.N.NNN.NN* indicates the schema version to which this script upgrades the database schema.
- *<db_type>* defines the type of database (`ora`, `mssql`, `db2`, or `postgre`).

Examples

Example A

If the old IDB version is 8.1.100.28 on ORACLE and the new ICON installation provides the following upgrade scripts:

- **Upgrade_8.1.100.27_ora.sql**
- **Upgrade_8.1.100.29_ora.sql**
- **Upgrade_8.1.100.32_ora.sql**

Then run the following scripts in the specified order:

- **Upgrade_8.1.100.29_ora.sql**
- **Upgrade_8.1.100.32_ora.sql**

Example B

If the old IDB version is 8.1.100.32 on ORACLE and the new ICON installation provides the following upgrade scripts:

- **Upgrade_8.1.100.27_ora.sql**
- **Upgrade_8.1.100.29_ora.sql**
- **Upgrade_8.1.100.32_ora.sql**

There is no need to apply any of the upgrade scripts.

Example C

If the old IDB version is 8.1.100.35 on ORACLE and the new ICON installation provides the following upgrade scripts:

- **Upgrade_8.1.100.27_ora.sql**
- **Upgrade_8.1.100.29_ora.sql**
- **Upgrade_8.1.100.32_ora.sql**

There is no need to apply any of the upgrade scripts.

Example D

If the old IDB version is 8.1.100.27 on ORACLE and the new ICON installation provides the following upgrade scripts:

- **Upgrade_8.1.100.27_ora.sql**
- **Upgrade_8.1.100.29_ora.sql**
- **Upgrade_8.1.100.32_ora.sql**

Then run the following scripts in the specified order:

- **Upgrade_8.1.100.29_ora.sql**
- **Upgrade_8.1.100.32_ora.sql**

Upgrading the Stored Procedures

1. To create a new set of stored procedures in IDB to support your new ICON installation, run **CoreProcedures_<*db_type*>.sql**.
2. If you are running Oracle 11 or higher and plan to purge by truncating partitions, you must create a new partitioned IDB. See Configuring a Partitioned Oracle IDB in the *Interaction Concentrator Deployment Guide* for deployment considerations and instructions.

   > **Important**
   > If you have an existing partitioned IDB, you cannot migrate that IDB to a non-partitioned database structure.

3. Upgrade your purge procedure. To do so, execute the appropriate one of the following scripts:
   - **Purge2_<*database*>.sql**
   - **PurgePart_ora.sql**—If you have a partitioned IDB.
   - **Purge2_PartitionType0_ora.sql**—If you have a non-partitioned Oracle IDB with the ICON **partition-type** configuration option set to `0`. Available in release 8.1.505.05 and higher.
4. Verify the purge parameters. Verify that the GSYSPurge81 procedure has been set up appropriately for your deployment. To do this:
   a. Retrieve the **transaction_size** value by selecting
   ```
   SELECT Val
   from
   G_DB_PARAMETERS
   where
   SECT = 'GSYSPurge81'
   and
   OPT = 'rowspertransaction'
   ```
   b. Make sure `Val` has the required value (the default value is `200,000` records). If necessary, you can make changes manually or you can use an optional Interaction Concentrator stored procedure, svcUpdateDBParameters, provided in the **Wrapper_for_<*version*>_<*db_type*>.sql** script.

5. Upgrade the wrappers. The ICON scripts directory contains wrappers for the stored procedures. For each RDBMS type, there is a file that matches pattern **Wrapper_for_*_<*db_type*>.sql**. For example, for Microsoft SQL, the wrapper name is **Wrapper_for_<*version*>_mssql.sql**.
   - Execute the wrapper for your RDBMS.

Re-Starting ICON and Verifying the Migration

1. Start the upgraded ICON. Genesys recommends that you use Solution Control Interface (SCI) or Genesys Administrator/Genesys Administrator Extension to start ICON.
2. Wait while ICON completes startup operations. The startup time strictly depends on the size of your configuration environment and may take more than 30 minutes.
3. Check the log for errors to verify that ICON is running properly. The criteria for a successful migration is that ICON starts and no database problem is detected.
   ICON is started when all three of the following trace messages below have been printed in the log:
   - Trace log message 09-25004: `Database queue [ID]: persistent queue transaction [number1] is processed, committed and removed. [number2] records are written.`
   - Trace log message 09-25016: `Persistent Queue [ID]: transaction [number1] is committed. [number2] records written into the queue.`
   - Trace log message 00-04541: `Message [message type] received from [socket number] ([application type] [application name]).`

   If ICON is unable to write interaction data in IDB, then see Rolling Back the Migration for instruction on reversing the migration and restoring your previous version of ICON.

Follow-Up Steps to Complete Your Migration

1. If ICON started successfully, you can start the Genesys Info Mart Job_ExtractICON.
2. (Optional) After a week—to allow time to be certain that you will not need to roll back your upgrade—you can delete all old stored procedures for previous Interaction Concentrator releases. To do so, execute the **drop_<*previous_schema_version*>_<*db_type*>.sql** script you saved in Step 5 in "Migrating the ICON Server" (above).

Rolling Back the Migration

If the upgrade failed and ICON did not start successfully, perform the following steps:

1. Stop Interaction Concentrator.
2. Restore the IDB database from your backup.
3. Restore entire the Interaction Concentrator directory, including the **icon.exe** file, as well as the ***.pq**

and ***.db** files.

4. Start the restored version of ICON.
5. If ICON starts successfully, you can then start the Genesys Info Mart extraction job.

Important Migration Notes

- You can migrate Interaction Concentrator from any version of release 7.5 or 7.6 directly to release 8.x.
- To migrate Interaction Concentrator 7.2 to 8.x, first, migrate to release 7.5 or 7.6 and then migrate to release 8.x.
- ICON processing must be suspended while the Interaction Concentrator migration is occurring. The content of the persistent queue file (**icon_*.pq**) is lost as a result of the upgrade. To minimize the loss of data, perform the upgrade when the contact center load is minimal.
- Stopping ICON does not stop the execution of any stored procedures that may be running or scheduled to run during the upgrade. In particular, if the merge or purge stored procedures are executing when you run the database scripts, these stored procedures are not successfully dropped and then re-created. You are not notified if you did not upgrade these service procedures correctly.
- You cannot migrate from a non-partitioned IDB to a partitioned IDB, or vice versa. If you want to start or stop using partitions, you must create a new IDB. (Partitioning is available on Oracle RDBMSs only.) For detailed information on this topic, see Purging by Truncating Partitions in the *Interaction Concentrator User's Guide*.

Document Change History

This section lists content that is new or that has changed significantly since the first release of this document. The most recent changes appear first.

> **Important**
> There was no 8.1.3 release of Interaction Concentrator.

8.1.401.00

New in Document Version 8.1.401.00

- The Deployment Guide was converted from PDF-only format to this online format, with the option of printing a PDF, if desired.
- Added descriptions of two new configuration options, `cfg-annex` in the Configuration functional group under the callconcentrator section and `do-not-register`, which is configured on the DN configuration object.
- Updated the name of the new migration script name format used in releases 8.1.4 and higher. For details on the migration scripts, see the Interaction Concentrator 8.1 Deployment Procedure.
- Added an item to the About Interaction Concentrator topic noting support for encrypted RDBMSs and support for having TEvent attached data hidden in log files. For details on this functionality and how to configure it, see the Genesys 8.1 Security Deployment Guide.

8.1.201.00

New in Document Version 8.1.201.00

- Added information on how to configuring IDB on PostgreSQL and specified the details of DAP configuration appropriate to PostgreSQL.
- Added notes stating new ICON support for dynamic updates to T-Server host and port changes.
- Added a section on security features and configuration requirements to the section on Deployment Planning.
- Added a recommendation to install ICON on the same host as the T-Server or Interaction Server for

which it writes data.

- Added information about the option to purge the persistent queue file upon ICON startup to the information on the Persistent Queue file.
- Added notes to indicate that the gsysIRMerge and gsysIRMerge2 stored procedures are compatible with Genesys Info Mart 7.x but are not needed by Genesys Info Mart 8.x.
- Added notes to indicate that the gsysIRMerge and gsysIRMerge2 stored procedures are not applicable to the PostgreSQL RBDMS.
- Configuration option additions, updates and corrections:
  - Added descriptions of the cseq-adjustment, gls-enforce-reason-code, pq-startup-purge , ring-divert, and gls-improve-data-for-agent configuration options.
  - Added a reference to the information about the GSYS_EXT_INT1 field in the G_PARTY_HISTORY table to the description of the ivr configuration option.
  - Added notes to the om-memory-optimization and calls-in-the-past options stating the releases in which the om-memory-optimization setting could override the calls-in-the-past setting.
  - Corrected and clarified the wording for the gls-acw-first option descriptions for the ICON Application object and the Switch Application object.
  - Corrected and amplified the description for the sst-options configuration option.
  - Added notes to advise users that the values for the role option must be set in lowercase only, not uppercase or mixed case.
- Updated the name of the Genesys eServices product by changing it from eServices/Multimedia simply to eServices, its current name. Multimedia was the former product name.
- Added notes instructing readers to restart the ICON server after any change to the attached data specification file.

8.1.101.00

New in Document Version 8.1.101.00

- Purge by Truncating Partitions
  - Described new scripts and stored procedure, purgePartitions811, for creating and using a partitioned Oracle RDBMS that is set up so as to be able to purge data by truncating partitions:
  - Added the purgePartitions811 stored procedure to the list of purge procedures.
  - Noted important considerations to take into account before implementing the purge by truncating partitions functionality.
  - Added deployment instructions for creating a partitioned Oracle IDB that can then be purged using the purgePartitions811 stored procedure.
  - Listed the new initialization scripts used to partition IDB and to enable purging by truncating partitions.

> **Important**
> In this release, partitioning is supported only on Oracle 11 or higher.

- New Configuration Options—Added descriptions of the following configuration options:
  - `cfg-auto-resync`
  - `advanced-ext-party-reconstruction`
  - `dest-busy-processing`
  - `x-server-http-trace-level`
  - `x-server-smtp-trace-level`
  - `x-server-dbw-debug-level`
  - `third-party-queue-in-divert`
- Corrected Configuration Option Values
  - Corrected the Changes Take Effect values for the `pq-backlog-alarm-threshold` and `pq-backlog-clearance-threshold` options from Immediately to After restart.
- Added Information on IDB Update Scripts for Genesys Info Mart—Added descriptions of the `update_idb_for_gim.sql` and `update_idb_for_gim_mm.sql` scripts, required for environments running Genesys Info Mart.
- New PQ File Requirement—Added a note informing users of the requirement to locate persistent queue files on a local drive.