Interaction Concentrator 8.0

Deployment Guide

About Genesys

Alcatel-Lucent's Genesys solutions feature leading software that manages customer interactions over phone, Web, and mobile devices. The Genesys software suite handles customer conversations across multiple channels and resources—self-service, assisted-service, and proactive outreach—fulfilling customer requests and optimizing customer care goals while efficiently using resources. Genesys software directs more than 100 million customer interactions every day for 4000 companies and government agencies in 80 countries. These companies and agencies leverage their entire organization, from the contact center to the back office, while dynamically engaging their customers. Go to www.genesyslab.com for more information.

Each product has its own documentation for online viewing at the Genesys Technical Support website or on the Documentation Library DVD, which is available from Genesys upon request. For more information, contact your sales representative.

Notice

Although reasonable effort is made to ensure that the information in this document is complete and accurate at the time of release, Genesys Telecommunications Laboratories, Inc., cannot assume responsibility for any existing errors. Changes and/or corrections to the information contained in this document may be incorporated in future versions.

Your Responsibility for Your System's Security

You are responsible for the security of your system. Product administration to prevent unauthorized use is your responsibility. Your system administrator should read all documents provided with this product to fully understand the features available that reduce your risk of incurring charges for unlicensed use of Genesys products.

Trademarks

Genesys, the Genesys logo, and T-Server are registered trademarks of Genesys Telecommunications Laboratories, Inc. All other trademarks and trade names referred to in this document are the property of other companies. The Crystal monospace font is used by permission of Software Renovation Corporation, www.SoftwareRenovation.com.

Technical Support from VARs

If you have purchased support from a value-added reseller (VAR), please contact the VAR for technical support.

Technical Support from Genesys

If you have purchased support directly from Genesys, please contact Genesys Technical Support at the regional numbers provided on page 11. For complete contact information and procedures, refer to the ***Genesys Technical Support Guide***.

Ordering and Licensing Information

Complete information on ordering and licensing Genesys products can be found in the ***Genesys Licensing Guide.***

Released by

Genesys Telecommunications Laboratories, Inc. www.genesyslab.com

Document Version: 80icon_dep_06-2011_v8.0.002.00

Table of Contents

List of Procedures

Preface

Welcome to the *Interaction Concentrator 8.0 Deployment Guide.* This document provides:

- A high-level overview of Interaction Concentrator 8.0, including architectural graphics
- Deployment planning materials, including typical deployment scenarios
- Deployment procedures, including troubleshooting tips
- Metrics definitions, illustrated by basic call flows
- An introduction to the Interaction Database (IDB) schema, including sizing recommendations

This document is valid only for the 8.0 release(s) of this product.

Note: For versions of this document created for other releases of this product, visit the Genesys Technical Support website, or request the Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

This preface contains the following sections:

For information about related resources and about the conventions that are used in this document, see the supplementary material starting on page 175.

About Interaction Concentrator

Interaction Concentrator collects and stores detailed data about the interactions and resources in customer interaction networks that use Genesys Framework (contact center, enterprise-wide, or multi-enterprise telephony and computer

networks). Downstream reporting systems can access Interaction Concentrator data in near real time.

Intended Audience

This document is primarily intended for system integrators. It has been written with the assumption that you have a basic understanding of:

- Computer-telephony integration (CTI) concepts, processes, terminology, and applications
- Network design and operation
- Database design and operation
- Your own network configurations

You should also be familiar with:

- Genesys Framework architecture and functions
- Genesys products deployed in your contact center
- Your real-time and historical reporting objectives

Making Comments on This Document

If you especially like or dislike anything about this document, feel free to e-mail your comments to Techpubs.webadmin@genesyslab.com.

You can comment on what you regard as specific errors or omissions, and on the accuracy, organization, subject matter, or completeness of this document. Please limit your comments to the scope of this document only and to the way in which the information is presented. Contact your Genesys Account Representative or Genesys Technical Support if you have suggestions about the product itself.

When you send us comments, you grant Genesys a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

Contacting Genesys Technical Support

If you have purchased support directly from Genesys, contact Genesys Technical Support at the following regional numbers:

Region	Telephone	E-Mail
North America and Latin America	+888-369-5555 (toll-free) +506-674-6767	support@genesyslab.com
Europe, Middle East, and Africa	+44-(0)-1276-45-7002	support@genesyslab.co.uk
Asia Pacific	+61-7-3368-6868	support@genesyslab.com.au
Malaysia	1-800-814-472 (toll-free) +61-7-3368-6868	support@genesyslab.com.au
India	000-800-100-7136 (toll-free) +61-7-3368-6868 (International)	support@genesyslab.com.au
Japan	+81-3-6361-8950	support@genesyslab.co.jp
Before contacting technical support, refer to the *Genesys Technical Support Guide* for complete contact information and procedures.		

Documentation Change History

This section lists topics that are new or have changed significantly since the first release of this document.

New in Document Version 8.0.002.00

- The references to `GSYSPurge76` have been updated to `GSYSPurge80`. Also, the description of this procedure includes the fact that it now purges Outbound Contact data as well as voice and multimedia interactions, attached data, and agent login session data. For details, see "Purge" on page 17.
- A note has been added indicating that you must restart Interaction Concentrator after a backup instance is configured of any application, such as Outbound Contact Server (OCS) or T-Server, for which Interaction Concentrator has a connection configured on the `Connections` tab has been added (page 84).

- The procedure for having Interaction Concentrator recognize Interaction Servers configured with the `Interaction Server` application type rather the `T-Server` application type has been expanded for clarity. For the complete procedure, see "Connections" on page 63.
- A description of the new `partition_type` option has been added to page 111.
- Descriptions of the new `use-dss-monitor` (page 118) and `dss-no-data-tout` (page 118) options have been added.
- The descriptions of the `gls-max-duration` (page 144) and `gls-max-inactivity` (page 144) options have been updated.
- The "Sample Specification for Multimedia Attached Data" on page 167 has been updated.

Chapter

1 Product Overview

This chapter describes the basic Interaction Concentrator architecture, the components of Interaction Concentrator and their functions, various deployment scenarios, and the functional roles that single or multiple Interaction Concentrator (ICON) instances play in a contact center. It also provides a high-level overview of Interaction Concentrator functionality, including features and functionality that are new in release 8.0.

This chapter contains the following sections:

Basic Architecture

Interaction Concentrator is a Genesys product that collects and stores detailed data from various sources in a contact center that is empowered with Genesys software. Downstream reporting systems can access Interaction Concentrator data in near real time.

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.

Figure 1 depicts the basic ICON architecture, omitting most of the Framework components for the sake of simplicity.

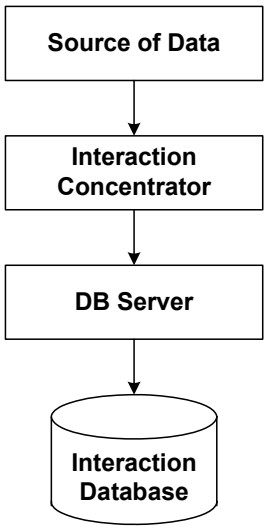

Figure 1: Basic Interaction Concentrator Architecture

Components and Functions

Interaction Concentrator consists of the following elements:

- ICON server
- Interaction Database (IDB) (see page 17)

The following subsections describe each of these in turn.

ICON Server

The ICON server:

- Performs preprocessing of events received from Configuration Server, T-Server, Interaction Server, and Outbound Contact Server (OCS), according to the role configured for the ICON instance.

  Processing 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 in-memory queue configuration options, refer to the option descriptions beginning on page 105.
- Prepares the data that will be stored in IDB.
- Writes the prepared data from the in-memory queue to the persistent queue and persistent caches. For more information, see "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 Chapter 6 on page 93.

Persistent Queue and Persistent Caches

ICON 8.0 uses the following temporary storage files:

- Persistent Queue (.pq File)
- Persistent Cache for Configuration Data, page 15
- Persistent Cache for Agent Login Session Data, page 16

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.

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 about ICON roles, see "ICON Roles" on page 26.

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 about the persistent queue configuration options, see the options starting on page 112.

Note: The size of the persistent queue is not formally limited by ICON, but the operating system may impose some limitations.

Persistent Cache for Configuration 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 8.0 User's Guide*.

For recommendations about best practices regarding synchronization, see the chapter about resynchronization in the *Interaction Concentrator 8.0 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 8.0 User's Guide*.

A configuration option, `agent-pstorage-name` (see page 112), enables you to specify the name of this persistent cache. The default file name is `apstorage.db`.

ICON Server Interfaces

The ICON server interfaces with:

- Solution Control (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—see "Sources of Data" on page 18.)
- Message Server, to log messages to the Central Logger.

Note: Interaction Concentrator does not support the use of the Transport Layer Security (TLS) protocol to secure data exchange between the components with which the ICON server interfaces.

Interaction Database

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 8.0 User's Guide*. For a complete table structure and descriptions of all IDB tables and fields, see the *Interaction Concentrator 8.0 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 totally internal to Interaction Concentrator functioning. Therefore, detailed information about them is not relevant for end users.

Starting with Interaction Concentrator release 8.0, most stored procedure names start with a schema-specific prefix, so that they constitute a schema-specific package. Each ICON 8.x release works only with the stored procedures package for the associated schema version. 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.

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

Merge

- `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

- `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`—An alternative procedure that safely purges voice and multimedia interactions; attached data; agent login session data; and Outbound Contact data from IDB.

> **Note:** Interaction Concentrator release 8.0.000.30 and earlier includes the prior version of the purge procedure, `gsysPurge76`, which does not purge Outbound Contact data.

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.

For more information about these stored procedures, refer to the chapter about special stored procedures in the *Interaction Concentrator 8.0 User's Guide.*

Sources of Data

Table 1 summarizes the sources from which ICON collects data.

Table 1: ICON Sources of Data

Type of Data	Source
Configuration data for your contact center resources	Configuration Server
Detailed CTI-related data about call activity in your contact center	T-Server
Detailed CTI-related data about Voice over IP (VOIP) and SIP chat interaction activity in your contact center	SIP Server
Detailed data about Genesys eServices/Multimedia (including e-mail and non-SIP chat) and 3rd Party Media (formerly known as Open Media) interaction activity in your contact center	Interaction Server
Media types for 3rd Party (custom-defined) Media	Interaction Server
Detailed data about virtual queue usage in interaction processing	• For voice calls: T-Server (from Universal Routing Server [URS]) • For multimedia interactions: Interaction Server (from URS)
Data specific to outbound calls and campaigns	OCS

Terminology Note: SIP Server

SIP Server is a type of Genesys T-Server. In the remainder of this document and in the rest of the Interaction Concentrator documentation suite, all references to *T-Server* include SIP Server.

Deployment Scenarios

The Interaction Concentrator architecture is flexible enough to store reporting data for a contact center environment of practically any size. This section provides general descriptions of the following basic deployment scenarios:

- Single-Site Deployment, page 20
- Multi-Site Deployment, page 21
- Network Deployment, page 24

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.

For additional considerations that may affect your choice of deployment, see “Recommended Role Assignment” on page 27.

Note: 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 rules governing role assignments, see “Rules and Restrictions” on page 27

Database Sizing Tool

Genesys provides an interactive tool to help you estimate the required size of IDB, based on details you provide about projected agent activity, outbound activity, ICON server and database settings, and user data. This tool, the *Interaction Concentrator 8.0 Database Size Estimator,* is a Microsoft Excel spreadsheet that is available from the Genesys Technical Support website. See also the Interaction Concentrator chapter in the *Genesys 8 Hardware Sizing Guide*.

Diagram Conventions

To simplify the deployment diagrams in this section:

- 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 Figure 1 on page 14).

- 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 the chapter about integrating with Outbound Contact in the *Interaction Concentrator 8.0 User's Guide.*
- Interaction Server is not shown for deployments that include eServices/Multimedia. Notes in the text indicate the deployment scenarios that are suitable for multimedia environments (see "One ICON and One IDB" on page 20 and "One ICON and One IDB per Site" on page 21). In these environments, the Interaction Server occupies the same position in the architecture as a T-Server.

Single-Site Deployment

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, as shown in Figure 2 on page 21.

Deployments with multiple instances of ICON and multiple instances of IDB are straightforward extensions of this model.

Note: Figure 2 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.

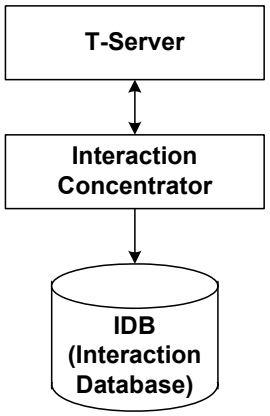

Figure 2: Single-Site Deployment: A Single ICON and Single IDB Instance

Multi-Site Deployment

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 (see page 21)
- A single ICON instance and a single, centralized IDB for the entire contact center (see page 22)
- Multiple ICON instances and a single, centralized IDB for the entire contact center (see page 23)

The following subsections describe each of these deployments in turn.

See also "Network Deployment" on page 24.

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 other with CTI-related data from the T-Server that serves that site (see Figure 3 on page 22).

Note: Figure 3 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.

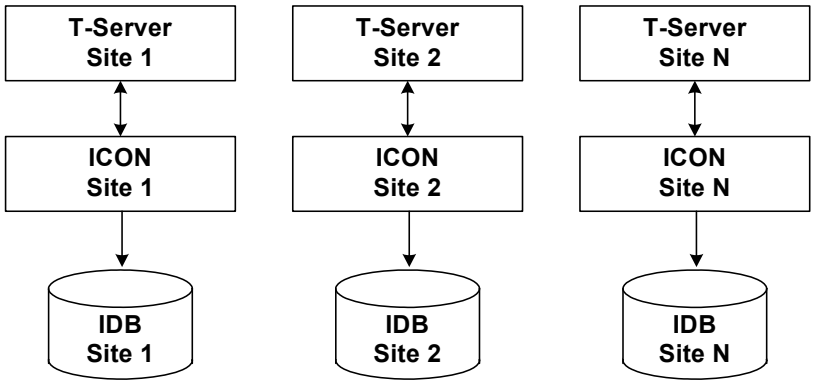

Figure 3: Multi-Site Deployment: Independent IDB Instances

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 (see Figure 4).

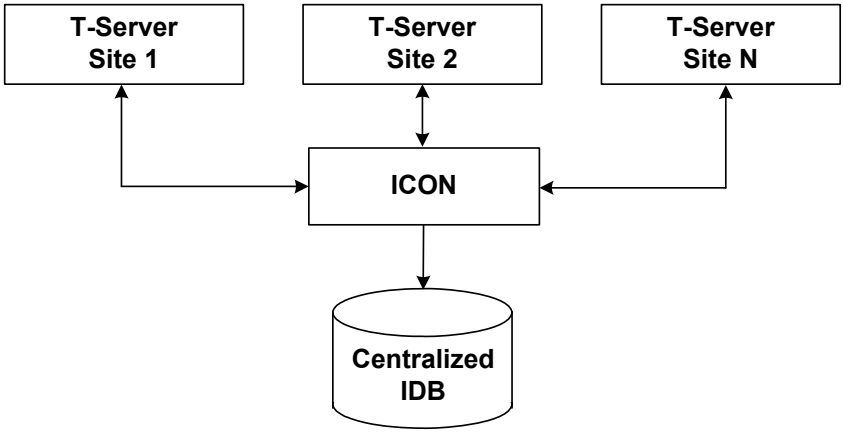

Figure 4: Multi-Site Deployment: A Single ICON and a Centralized IDB Instance

This scenario helps you avoid the need to merge data from different databases. However, be aware of the following:

- 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 8.0 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, call details from all contact center sites come into the same database through separate ICONs (see Figure 5).

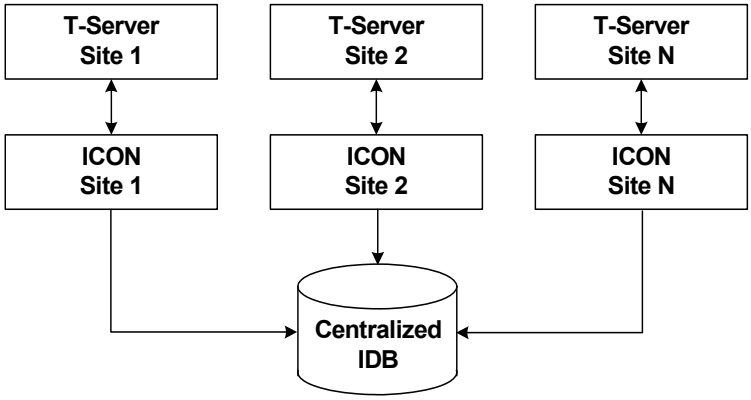

Figure 5: Multi-Site Deployment: Multiple ICONs and a Centralized IDB Instance

Like the scenario of one ICON and one IDB for the contact center (see Figure 4 on page 22), 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 8.0 User's Guide*).

Network Deployment

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.

ICON 8.0 supports the following Network 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 (see page 25)

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 (see Figure 6).

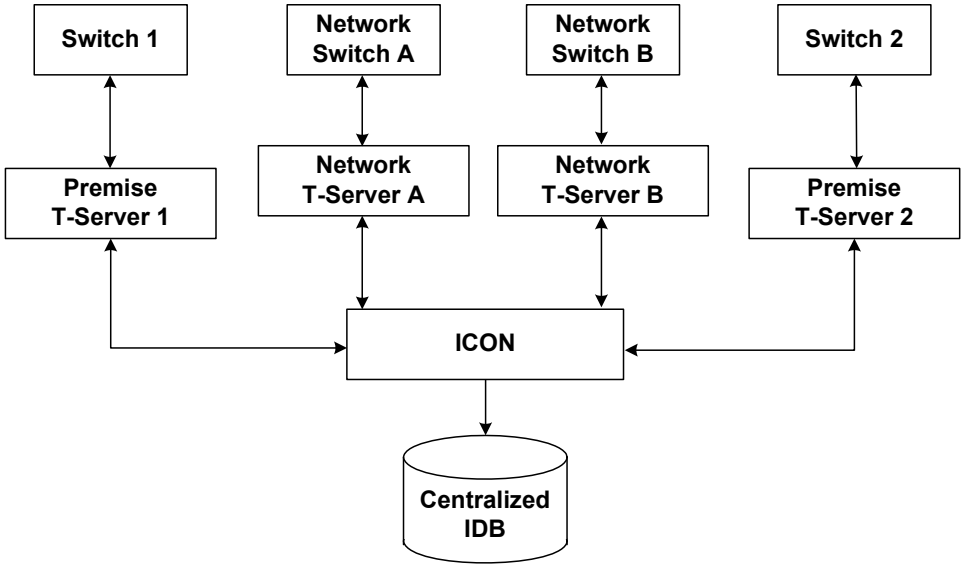

Figure 6: Network Deployment: A Single Network T-Server per Network Switch

Note: Figure 6 does not show the interconnections among T-Server objects that are required for a Network T-Server deployment.

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

If the network switch and multiple Network T-Servers have been configured to work in load-balancing mode, ICON 8.0 supports deployments in which a single ICON instance connects to multiple Network T-Servers that serve the same network switch (see Figure 7).

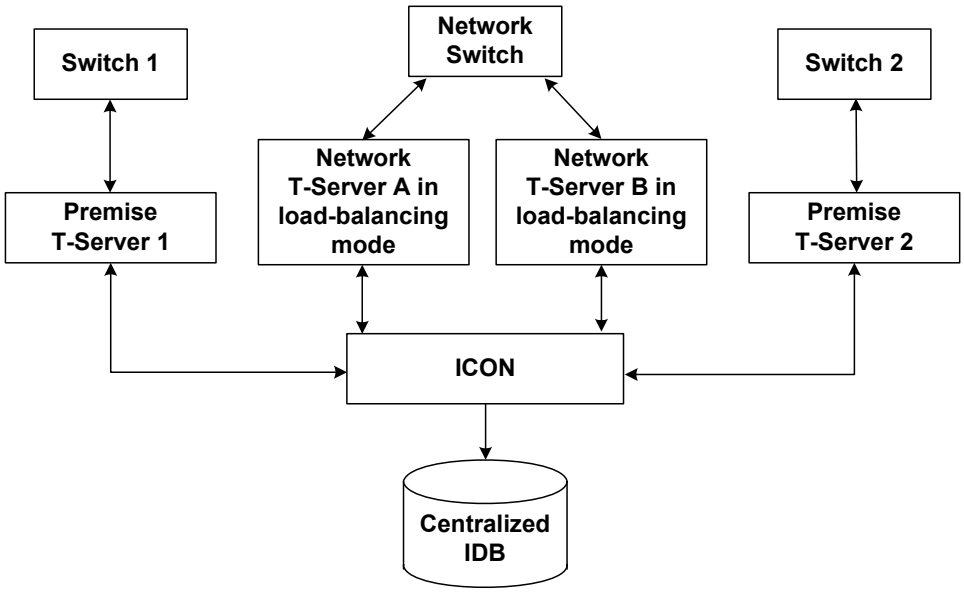

Figure 7: Network Deployment: Multiple Network T-Servers per Network Switch (Load-Balancing Configuration)

Note: Figure 7 does not show the interconnections among T-Server objects that are required for a Network T-Server deployment.

In the configuration in which multiple Network T-Servers serve the same network 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, `load-balancing-on-ntwk-switch`, must be set on the `Switch` object in the Configuration Layer in order for ICON to identify whether the network switch and related Network T-Servers are operating in load-balancing mode. For more information about this option, see page 145.

Note: The deployment with Network T-Servers in load-balancing mode is the only configuration 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.

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).

  Note: 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 on page 104). 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.

In the example shown in Figure 8 on page 27, 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.

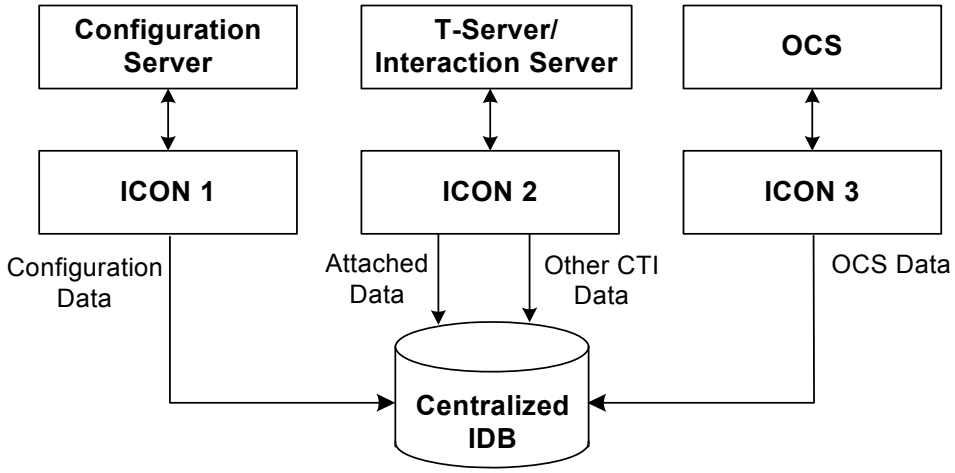

Figure 8: Example of Role Division Among ICON Instances

Recommended Role Assignment

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.

> **Note:** 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".

For the values that enable role assignments, see the description of the `role` configuration option on page 104.

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.

Supported Features and Functionality

In addition to new functionality described in "New in This Release" on page 30, and depending on the role configured for the Interaction Concentrator instance, Interaction Concentrator provides the following features and functionality to support reporting about contact center activities:

- Captures and stores information about the current contact center configuration (objects and associations), and preserves information about deleted configuration objects and terminated associations.
- Captures and stores detailed information about active and completed voice interactions, including switch, DN, time, and routing information about calls and parties. Interaction Concentrator uses a globally unique call identifier.
- Captures and stores detailed information about multimedia interactions (e-mail, non-SIP chat, and custom-designed media such as fax and web forms). Interaction Concentrator can support a large number of concurrently active multimedia interactions, and also provides the ability to identify if a chat session has the focus. For more information, see the chapter about integrating with Genesys eServices/Multimedia and 3rd Party Media in the *Interaction Concentrator 8.0 User's Guide.*
- Captures and stores detailed information about SIP chat interactions.
- Captures and stores detailed information about 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.

  For voice interactions, functionality includes 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.

  For more information, see the chapter about agent states and login sessions in the *Interaction Concentrator 8.0 User's Guide.*

- Supports custom agent states, for agents handling voice interactions. For more information, see the chapter about custom states and common data in the *Interaction Concentrator 8.0 User's Guide.*
- For all types of interactions, captures and stores 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. For more information, see the chapter about monitoring virtual queues in the *Interaction Concentrator 8.0 User's Guide.*
- For voice interactions, captures and stores detailed information about virtual routing point (VRP) usage in call processing.
- Captures and stores detailed information about interactions that are generated in a network-based contact solution.
- Captures and stores detailed information about interactions that are generated in a network call parking environment.
- Stores attached data and captures the history of attached data changes for voice interactions as well as eServices/Multimedia (e-mail and chat) and 3rd Party Media interactions. For more information, see the chapter about attached data in the *Interaction Concentrator 8.0 User's Guide.*
- Supports customized attached data processing for voice calls. For more information, see the chapter about attached data in the *Interaction Concentrator 8.0 User's Guide.*
- Captures and stores detailed information about outbound campaigns, including:
  - History of campaign processing
  - History of chain processing
  - Precalculated metrics provided by OCS

  For more information, see the chapter about integrating with Outbound Contact in the *Interaction Concentrator 8.0 User's Guide.*
- Provides a configurable filtering mechanism for certain types of data, to enable the optimization of database size and performance. For more information, see the chapter about filtering IDB data in the *Interaction Concentrator 8.0 User's Guide.*
- Provides the ability to resynchronize the configuration data in IDB with Configuration Database on demand. For more information, see the chapter about resynchronizing configuration changes in the *Interaction Concentrator 8.0 User's Guide.*
- 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.

  For more information, see the chapter about implementing HA in the *Interaction Concentrator 8.0 User's Guide.*

- 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. ICON receives information from T-Server regarding the relationship between a given interaction and an interaction at a different site. As a result, complete data is available for reporting across sites. Interaction Concentrator provides a stored procedure to merge the interaction records for multi-site interactions. For more information about the merge procedure, see the chapter about special stored procedures in the *Interaction Concentrator 8.0 User's Guide.*
- Supports multibyte character encoding.
- Stores time information in two formats:
  - Greenwich Mean Time (GMT)—As a `datetime` data type.
  - Coordinated Universal Time (UTC) seconds—As an `integer` data type.

  ICON 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. For more information about the purge procedures, see the chapter about special stored procedures in the *Interaction Concentrator 8.0 User's Guide.*

New in This Release

The 8.0 release of Interaction Concentrator provides the following additional or changed functionality:

- Supports the configuration in which one ICON instance is able to create and maintain connections to multiple Network T-Servers working with the same switch in load-balancing mode. For more information, see "Multiple Network T-Servers per Network Switch (Load-Balancing Configuration)" on page 25.

  A new configuration option on the `Switch` object, `load-balancing-on-ntwk-switch`, enables ICON to identify whether the Network T-Servers are operating in load-balancing mode. For more information about this option, see page 145.

  A new, standard-level log message identifies when the ICON application detects an additional connection to a Network T-Server in load-balancing mode.
- Reliably indicates whether the endpoint associated with a party is an IVR device. ICON uses a formerly reserved field in the `G_PARTY_HISTORY` table, `GSYS_EXT_INT1`, to store information that identifies when the party is

associated with an IVR device—a value of `1` in this field, in the first historical record related to the party, indicates that the associated endpoint is an IVR.

ICON determines that the party is associated with an IVR device based on the configuration of the DN or IVR port. For more information, see the description of the `ivr` option, in the `gts` section of the `DN` configuration object, on page 148.

For more information about the `GSYS_EXT_INT1` field in the `G_PARTY_HISTORY` table, see the *Interaction Concentrator 8.0 Physical Data Model* document for your particular RDBMS.

- Properly handles user data that is updated by a routing strategy or agent after the party's association with the interaction has been terminated (for example, the call was transferred). For more information, see the section about user data postprocessing in the *Interaction Concentrator 8.0 User's Guide.*
- Identifies whether the UserEvent that caused a record to be written to a custom states table came from a device at a time when that device was participating in an active call. ICON uses a formerly reserved field, `GSYS_EXT_INT1`, as a flag—a value of `1` in this field in the `G_CUSTOM_DATA_P` or `G_CUSTOM_DATA_S` tables indicates that the UserEvent was generated while the device was participating in an active call.
- If a virtual queue is involved in routing an interaction, stores the database identifier assigned to the virtual queue by Configuration Server (the DBID of the virtual queue) in the `G_ROUTE_RESULT` table in IDB, if this data is provided by URS.

  For more information, see the section about the virtual queue DBID in the chapter about monitoring virtual queues and routing points in the *Interaction Concentrator 8.0 User's Guide*.
- Enables downstream reporting applications to identify when data was not available and to evaluate the reliability of available data provided by T-Server, Interaction Server, Outbound Contact Server, and Configuration Server.

  This functionality:
  - Provides a mechanism for downstream reporting applications, such as Genesys Info Mart, to support HA for all types of data, including multimedia and Outbound Contact data, when one of a pair of ICON instances fails.
  - Enhances support for HA by enabling downstream reporting applications to detect in advance when to switch over to another IDB for data extraction.
  - Improves data integrity and ICON functioning by enabling downstream users to identify the periods when configuration data was not captured or stored and, therefore, to take the necessary action to resynchronize configuration data in a timely manner.

To enable this functionality, new tables in IDB, one for each type of data provider (as specified by the ICON role), store detailed data about connections, timestamps, and events from the specified data sources. For more information about the new tables, see the *Interaction Concentrator 8.0 Physical Data Model* document for your particular RDBMS.

For more information about using this new functionality, see the section about determining data availability and reliability in the *Interaction Concentrator 8.0 User's Guide.*

- Provides sufficient information at the call and party level for downstream reporting applications to determine which party released the call.

  To support this functionality, ICON uses the `GSYS_EXT_VCH1` and `GSYS_EXT_VCH2` fields in the `G_CALL_STAT` table and the `GSYS_EXT_INT1` field in the `G_PARTY_STAT` table to store information, if it is reported by T-Server. A new configuration option, `store-releasing-party` (see page 115), enables this functionality.

  This feature requires T-Server release 8.0 or higher. Interaction Concentrator 8.0 supports this feature for the Alcatel A4400/OXE switch. Interaction Concentrator 8.0.000.35 and higher also supports this feature for Avaya switches; for Avaya, this functionality requires T-Server for Avaya Communication Manager release 8.0.101.05 or higher.

  For more information about using the new functionality, see the section about identifying who released the call in the *Interaction Concentrator 8.0 User's Guide.* See also the information about the `G_CALL_STAT` table and the `G_PARTY_STAT` table in the *Interaction Concentrator 8.0 Physical Data Model* for your particular RDBMS.

- Improves merge functionality.
  - Implements database locking in a way that reduces the potential for problems to occur during execution of the merge procedure.
  - Introduces a new stored procedure, `gsysIRMergeReset`, that simplifies the steps to reset the merge procedure if required.

- Streamlines future migration by packaging the stored procedures that support each 8.x release in a schema-specific set. A particular ICON release works only with the corresponding stored procedures package. Multiple sets of packages can exist in the same IDB, and an earlier set of stored procedures can work with a later version of the IDB schema. For more information, see the Interaction Concentrator section of the *Genesys Migration Guide.*

Operating Systems

- Introduces support for the following operating systems:
  - IBM AIX 64-bit version 6.1
  - Red Hat Enterprise Linux AS Edition version 5.0, for 32-bit Intel platform
  - HP-UX/PA 64-bit version 11i v3
  - Microsoft Windows Server 2008 on 32-bit and 64-bit platforms

- Discontinues support for the following operating systems:
  - Microsoft Windows Server version Win2000 on 32-bit Intel processors
  - Solaris/SPARC 32-/64-bit version 7
  - HP-UX/PA 32-/64-bit version 11.00

Virtual Environments

- Introduces support for the following virtual environments:
  - IBM PowerVM LPAR
  - Sun Solaris Containers

RDBMSs

- Introduces support for the following relational database management systems (RDBMSs):
  - Oracle 11g
  - Microsoft SQL Server 2008
  - IBM DB2 9.5
- Discontinues support for Microsoft SQL Server version 2000.

Chapter

2 Deployment Planning

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.

This chapter contains the following sections:

Compatibility

This section lists the various Genesys components with which Interaction Concentrator release 8.0 is compatible.

For information about supported operating systems and relational database management systems (RDBMSs), see the *Genesys Supported Operating Environment Reference Manual* document. Note that Oracle versions 8.1 and earlier are not supported.

Table 2 lists the Genesys product components with which Interaction Concentrator operates. Refer to the *Interaction Concentrator 8.0 Release Notes* for any updates to the release requirements for the various components.

Table 2: Interaction Concentrator Compatibility

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, if you are running Interaction Concentrator release 8.0.000.35 or higher, is also supported on Avaya switches (requires Avaya Communication Manager 8.0.101.05 or higher).
Multimedia	• eServices/Multimedia 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.
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.
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

Interaction Concentrator has important specific requirements in the following areas:

- Hosting (see page 37)
- Genesys Framework (see page 38)
- Outbound Contact (see page 39)
- Universal Routing (see page 39)
- Multimedia (see page 40)
- Interaction Database (see page 40)

The following subsections describe each of these areas in turn. 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).
- You can use the same host computer for ICON and T-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.

You will need a Genesys Management Framework product CD in order to install components of the Configuration Layer.

For information about, and deployment instructions for, these Framework components, see the *Framework Deployment Guide.*

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.

You will need a Genesys Management Framework product CD in order to install the components of the Management Layer.

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" on page 137 and "DN Configuration Options" on page 147.

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.

Note: 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.

You will need a Genesys Media product CD in order to install your T-Server.

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.

Note: 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 latest *Genesys Migration Guide.* For recommendations on how to enable outbound reporting in Interaction Concentrator, refer to the chapter about integrating with Outbound Contact in the *Interaction Concentrator 8.0 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.

Note: 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 storing, in IDB, extended routing results from virtual queues 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 Deployment Guide.* For Universal Routing migration instructions, see the latest *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 8.0 User's Guide.*

Multimedia

If you intend to collect interaction, agent state, and agent login session data for eServices/Multimedia (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 that the `Switch` object that 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.

Note: 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" on page 63.

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 (Multimedia) Deployment Guide*. For recommendations on how to enable multimedia reporting in Interaction Concentrator, refer to the chapter about integrating with Genesys eServices/Multimedia and 3rd Party Media in the *Interaction Concentrator 8.0 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 (since ICON release 7.6). Oracle versions 8.1 and earlier are also not supported. For a list of Genesys-supported RDBMSs, see

the *Genesys Supported Operating Environment Reference Manual*. For the changes in RDBMS support introduced with Interaction Concentrator 8.0, see the "New in This Release" section, page 33.

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 8.0 User's Guide*.
- Estimate IDB size, using the *Interaction Concentrator 8.0 Database Size Estimator*. (For more information, see "Sizing the Interaction Database" on page 41.)

General Considerations

As described in Chapter 1 on page 13, 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" on page 19, and answer the following questions:

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

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

Sizing the Interaction Database

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 8.0 Database Size Estimator*, is a Microsoft Excel spreadsheet that is available on the Genesys Technical Support website.

For more information about database sizing and deployment guidelines, see the Interaction Concentrator section in the *Genesys 8 Hardware Sizing Guide*.

Chapter

3 Configuration and Installation

This chapter describes the recommended deployment sequence for Interaction Concentrator. It also describes how to deploy each component in your Genesys environment.

This chapter contains the following sections:

Note: The information in this chapter 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.

Recommended Deployment Order

Before you deploy Interaction Concentrator, review the information in Chapter 2 on page 35, 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 8.0 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 "Switch Configuration Options" on page 137 and "DN Configuration Options" on page 147.

3. DB Server

   If you decide to use a DB Server that serves another application for Interaction Concentrator storage purposes, 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 8.0 DB Server User's Guide.* (See also Procedure: Deploying DB Server, on page 45 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.

   You will need a Management Framework 8.0 product CD in order to install DB Server.

4. Interaction Concentrator

   Configure and install Interaction Concentrator, as described in Procedure: Deploying Interaction Concentrator, on page 46.

   In an environment with multiple ICON instances, repeat the steps in Procedure: Deploying Interaction Concentrator, on page 46 for each ICON instance, making the necessary adjustments when you configure ICON connections and configuration options.

   You will need an Interaction Concentrator 8.0 product CD in order to install Interaction Concentrator.

5. Interaction Database

   Install a database for ICON data on one of the supported RDBMSs and initialize IDB, as described in "Creating IDB" on page 54.

   In an environment with multiple IDB instances, repeat the steps in "Creating IDB" on page 54 for each IDB instance.

6. Database Access Point

   Configure a `DAP Application` object that specifies IDB connection parameters, as described in Procedure: Configuring a DAP, on page 59.

In an environment with multiple ICON instances, each of which stores data to its own IDB instance, repeat the steps in Procedure: Configuring a DAP, on page 59 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.

Ensure that the `role` options that you specify for the DAP are consistent with the `role` options specified for the ICON instance that it serves.

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.

Procedure: Deploying DB Server

Purpose: To configure and install as many instances of DB Server as you need.

Prerequisites

- Configure `Host` objects for Interaction Concentrator components. See "Hosting" on page 37.

Start of 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 8.0 DB Server User's Guide.*

End of procedure

Next Steps

- Procedure: Deploying Interaction Concentrator.

Deploying Interaction Concentrator

Deploy as many instances of Interaction Concentrator as you need. The following summary procedure consists of major steps, each of which is described as a separate procedure in this section.

Procedure: Deploying Interaction Concentrator

Purpose: To configure and install as many instances of Interaction Concentrator as needed on either a UNIX or Windows operating system.

Prerequisites

- Procedure: Deploying DB Server

Start of procedure

1. Import the application template for Interaction Concentrator (see page 47).
2. Configure an `Application` object for Interaction Concentrator (see page 47).
3. Install Interaction Concentrator on its host (see page 51).

End of procedure

Next Steps

- Procedure: Importing the application template

Note: Interaction Concentrator uses the `Call Concentrator` application type for its `Application` object in the Configuration Layer.

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.

Importing 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.

Procedure:
Importing the application template

Purpose: To create an application template that you can use to create as many `Application` objects of the same type as you need.

Start of procedure

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 8.0 product CD, and open the `TEMPLATES` folder.
6. Select the template file for Interaction Concentrator; it is called `Interaction_Concentrator_800.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.

End of procedure

Next Steps

- Procedure: Configuring an ICON application object

Configuring 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.

Procedure:
Configuring an ICON application object

Prerequisites

- Procedure: Importing the application template

Start of procedure

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.

General

5. In the `Properties` dialog box, click the `General` tab, and then enter a name for this application. The application template provides information about the application type and version.

> **Note:** Interaction Concentrator uses the `Call Concentrator` application type for its `Application` object in the Configuration Layer.

Server Info

6. 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.

> **Note:** 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.

Start Info

7. 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 page 85.

> **Note:** These properties are updated automatically during the installation procedure.

Options

8. 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, see Chapter 4 on page 61.

   For information about all the ICON configuration options, see "ICON Configuration Options" on page 93.

   Also:

   - Configure both ICON-specific log options and common log options in the log-related configuration sections. For option descriptions, see "log Section" on page 134 and the *Framework 8.0 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.
   - Configure an HTTP listener by creating a `listeners` section and specifying the appropriate option. For option descriptions, see "listeners Section" on page 132.
   - 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 option description on page 104. For more information about role assignments and restrictions for the `cfg` role, see "Recommended Role Assignment" on page 27.

Tenants

9. 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.

Notes: 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.

Connections

10. Click the `Connections` tab, and then add the following connections:

> **Note:** 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 8.0 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 Network Switch (Load-Balancing Configuration)" on page 25.
- 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 page 63).

> **Note:** 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.

Note: 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 8.0 Configuration Manager Help*.

- After you configure one or more `DAP Application` objects (see page 59), add to the ICON `Connections` tab any `DAP Application` objects through which this ICON instance will access IDBs.

11. Click `OK` to save your changes and close the `Properties` dialog box.

End of procedure

Next Steps

- Manually install the Interaction Concentrator application on your specific operating system. Do one of the following:
  - Procedure: Installing the ICON application on UNIX, on page 51
  - Procedure: Installing the ICON application on Windows, on page 52

Installing the ICON Application

After you use Configuration Manager to create an `Application` object for Interaction Concentrator, install Interaction Concentrator. The following procedures provide instructions for installing the application on UNIX and Windows operating systems, respectively.

Procedure: Installing the ICON application on UNIX

Purpose: To install the ICON application by using the shell script provided on the Interaction Concentrator product CD.

Prerequisites

- Procedure: Configuring an ICON application object, on page 47

Start of procedure

1. On the Interaction Concentrator 8.0 product CD, in the appropriate `icon/<operating_system>/` directory, locate the `install.sh` shell script.
2. Run this script from the command line by entering the following command:

   ```
   ./ install.sh
   ```

3. When prompted, specify the host name of the computer on which you want to install ICON.
4. When prompted, specify:
   a. The host name of the computer on which Configuration Server is running.
   b. The port client that applications use to connect to Configuration Server.
   c. The user name that is used to log in to the Configuration Layer.
   d. The password that is used to log in to the Configuration Layer.
5. The installation displays the list of `Application` objects of `Call Concentrator` type that are configured for this host. Type the number corresponding to the ICON `Application` that you want to install.
6. Specify the destination directory into which you want to install ICON.
7. If prompted for the version of the product to install (32- or 64-bit), select the appropriate version for your environment.

As soon as the installation process is complete, a message appears, announcing that installation was successful. The installation process creates a directory with the name that you specified in Step 6, and it places ICON in this directory.

End of procedure

Next Steps

- Use the database scripts to initialize IDB. See “Creating IDB” on page 54.

Procedure: Installing the ICON application on Windows

Purpose: To install the ICON application by using the Genesys installation wizard provided on the Interaction Concentrator product CD.

Start of procedure

1. On the Interaction Concentrator 8.0 product CD, open the `icon\Windows\` directory.
2. Locate and double-click `setup.exe` to start the Genesys Installation Wizard.
3. On the `Welcome` page, click the `About` button to review the `read_me` file for this installation package. The file also contains a link to the Release Notes file for Interaction Concentrator.
4. Click `Next` to proceed with the installation.

> **Note:** Click `Next` at the end of each step to proceed to the next page in the wizard.

5. On the `Connection Parameters to the Genesys Configuration Server` page, specify the following login parameters:
   - Host and port of Configuration Server
   - User name and password used to log in to the Configuration Layer
6. The `Select Application` page displays all applications of the `Call Concentrator` type in the Configuration Database. When you select one application from the list, the wizard displays some of the parameters that are configured for that application (in particular, the application type, host, working directory, command line, and command-line arguments).

   Select the application that you want to install.
7. The `Choose Destination Location` page displays the destination directory, as specified in the `Working Directory` property of the ICON `Application` object (see Step 7 on page 48). If the path that is configured as `Working Directory` is invalid, the wizard generates the following path to the destination directory:

   `C:\Program Files\GCTI\Interaction Concentrator\<ICON Application Name>\`

   If necessary, click one of the following:
   - `Browse`—To select another destination folder. The wizard updates the `Application` object's `Working Directory` property in the Configuration Database.
   - `Default`—To reinstate the path that is specified in the `Application` object's `Working Directory` property.
8. On the `Ready to Install` information page, click one of the following:
   - `Back`—To update any installation information.
   - `Install`—To proceed with installation. The `Installation Status` window appears, showing the installation progress.

9. On the `Installation Complete` page, click `Finish`.

   As a result of the installation, the wizard adds `Interaction Concentrator` icons to the following:

   - Windows `Start` menu, under `Programs > Genesys Solutions`.
   - Windows `Add or Remove Programs` dialog box, as a Genesys server.
   - Windows `Services` list, as a Genesys service with the `Automatic` startup type.

   The wizard also places the following files into the folder to which you installed Interaction Concentrator:

   - A template file for the ICON attached data specification. (For more information about the attached data specification file, see "Attached Data Specification File" on page 65.)
   - An example of the ICON attached data specification.
   - Scripts for IDB, for every supported RDBMS type.

End of procedure

Next Steps

- Use the database scripts to initialize IDB. See "Creating 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.

Note: The user account that is created for IDB must have permissions to create database objects such as tables, stored procedures, and sequences.

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. Table 3 describes the purpose of these scripts. In the script names in Table 3, `<db_type>` is a placeholder for the specific RDBMS type (`db2`, `mssql`, or `ora` [for Oracle]).

Note: If you are migrating from an existing IDB, do not simply apply all the scripts in Table 3. To avoid damaging or erasing existing data, follow the migration procedures that are described in the Interaction Concentrator chapters in the *Genesys Migration Guide*.

Table 3: IDB Scripts

Script Name	Description
CoreSchema_<db_type>.sql	(For initial installation only) Creates the core IDB tables and indexes.
<#>_UpgradeSchema_<db_type>.sql	(For migration only) Upgrades the IDB schema. The scripts you must execute depend on the release from which you are upgrading. For more information, see the instructions in the Interaction Concentrator chapters in the *Genesys Migration Guide.*
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` stored procedure.
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.

Initializing the Database

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

- Procedure: Initializing IDB on DB2, on page 56
- Procedure: Initializing IDB on Microsoft SQL, on page 57
- Procedure: Initializing IDB on Oracle, on page 58

Notes: 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, script `CoreSchema_<db_type>.sql` will return 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 "Storing Attached Data" on page 64. For an example of a script to create a custom dispatcher stored procedure and custom storage table, see Appendix B on page 171.

Procedure: Initializing IDB on DB2

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

Start of procedure

1. Go to the directory to which you installed ICON.
2. Go to the `scripts\db2` subdirectory.
3. Execute the following scripts in the order shown:
   - `CoreSchema_db2.sql`
   - `CoreProcedures_db2.sql`
   - `Purge2_db2.sql`—This script is optional. Execute this script if you want to use the `gsysPurge80` stored procedure in your deployment.
   - `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.

   To execute the scripts:

   a. Insert the following command line at the beginning of each script, providing appropriate values for the placeholders:

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

b. Use the following command line to load each initialization script:

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

End of procedure

Next Steps

- Procedure: Configuring a DAP, on page 59

Procedure: Initializing IDB on Microsoft SQL

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

Start of procedure

1. Go to the directory to which you installed ICON.
2. Go to the `scripts\mssql` subdirectory.
3. Execute the following scripts in the order shown:
   - `CoreSchema_mssql.sql`
   - `CoreProcedures_mssql.sql`
   - `Purge2_mssql.sql`—This script is optional. Execute this script if you want to use the `gsysPurge80` stored procedure in your deployment.
   - `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.

   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`

End of procedure

Next Steps

- Procedure: Configuring a DAP, on page 59

Procedure: Initializing IDB on Oracle

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

Start of procedure

1. Go to the directory to which you installed ICON.
2. Go to the `scripts\oracle` subdirectory.
3. Execute the following scripts in the order shown:
   - `CoreSchema_ora.sql`
   - `CoreProcedures_ora.sql`
   - `Purge2_ora.sql`—This script is optional. Execute this script if you want to use the `gsysPurge80` stored procedure in your deployment.
   - `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.

   To execute the scripts, log in to the `sqlplus` command processor, and type the following at the command prompt:

   `@ <script_name>`

End of procedure

Next Steps

- Procedure: Configuring a DAP, on page 59

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 8.0 Deployment Guide.*
- *Framework 8.0 DB Server User’s Guide.*

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

Procedure: Configuring a DAP

Purpose: To configure a `DAP Application` object for IDB.

Start of 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 Step 3 on page 44 and "Deploying DB Server" on page 45).

   Note: Do not select the `JDBC Connection` check box, because it does not apply to database connections through DB Server.

3. 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 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, 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:
  - `upper`—Field names are converted into uppercase.
  - `lower`—Field names are converted into lowercase.
  - `any`—Field names are not converted.

  This option 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.

Warning! For the `Case Conversion` option, use the default value (`any`), unless Genesys Technical Support directs you to do otherwise.

Note: When configuring a `DAP Application` object for IDB, do not configure any properties on the `JDBC Info` tab.

4. 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. On the `Options` tab:
   a. Create a section named `callconcentrator`.
   b. Within the `callconcentrator` section, create a configuration option named `role`.
   c. Set the option value to indicate the types of data that will be stored through this DAP. For more information, see the option description on page 151.

      For optimal performance, Genesys recommends the following sets of values for a given database access point:
      - `gcc,gud,gls`
      - `cfg`
      - `gos`

End of procedure

Next Steps

- 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.

Chapter

4 Special Configuration Requirements

This chapter 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).

This chapter contains the following sections:

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.

ICON Application

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 will process 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. For more information about this option, see page 104.

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`. For more information about this option, see page 115.

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" on page 125, 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 will 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 of 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 8.0 User's Guide.*

Note: You must restart Interaction Concentrator after a backup instance is configured of any application, such as OCS or T-Server, for which Interaction Concentrator 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.

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 8.0 User's Guide*.

ICON Application

To enable ICON to receive eServices/Multimedia 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.

To have ICON 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.

Configuring 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. For more information about this option, see page 104.

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 in the callconcentrator section of the ICON Application object. For more information about this option, see page 108.

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 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 "Attached Data Specification File" on page 65.
2. Specify the attached data configuration settings in the Genesys Configuration Layer. For more information, see "ICON Application".

For more information about attached data in Interaction Concentrator, see the chapter about processing attached data in the *Interaction Concentrator 8.0 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 "Storing Agent State and Login Session Data" on page 74.

ICON Application

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

ICON Role

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 on page 104.

Attached Data Specification File

The following ICON configuration option enables you to point ICON to a different attached data specification file:

- `adata-spec-name`

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 attached data configuration options starting on page 98. 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 custom dispatcher configuration options starting on page 101. Select the appropriate values for your environment, and make related configuration changes on the `Options` tab of the ICON `Application` object.

Attached Data Specification File

The attached data specification is an XML file stored in the installation directory that you specify when you install the Interaction Concentrator application (see “Deploying Interaction Concentrator” on page 46).

If you require ICON to store attached data in IDB, create an attached data specification for ICON to use. For the XML schema definition for your attached data specification, see “Schema Definition” on page 163.

For sample attached data specifications, see:

- Sample Basic Attached Data Specification, page 165
- Sample Specification for Multimedia Attached Data, page 167
- Sample Specification for Customized Attached Data, page 169

This section provides information about the following topics:

- Parser Limitations, page 66

Formatting Note

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.

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 will be used to update the specified field in the database table (see Note and Example on page 167).

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

Table 4 shows the database table in which each attribute is stored.

Table 4: Attached Data Storage Types

Attribute Name	IDB Table Name
public	G_USERDATA_HISTORY
secure	G_SECURE_USERDATA_HISTORY

Table 4: Attached Data Storage Types (Continued)

Attribute Name	IDB Table Name
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
cust-disp-group-n[a]	Customer-defined, as specified in the custom dispatcher[a]

a. Example of an attribute group name you might use with a custom dispatcher.

Data Source Types

Table 5 shows the TEvent attribute from which each attribute is derived.

Table 5: Attached Data Attribute Source Types

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 (see Table 6)
- Multimedia-specific attached data (see Table 7 on page 68)
- Custom attached data (see Table 8 on page 69)

Predefined IDB Columns—Voice

For voice calls, Table 6 shows the predefined IDB field in which each attribute is stored in the `G_CALL_USERDATA` table.

Table 6: Predefined Attached Data Fields for Voice Calls

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/Multimedia and 3rd Party Media interactions, Table 7 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 Table 7 can be used for customer-defined keys.

Table 7: Predefined Attached Data Fields for Multimedia Interactions

Predefined Key Name	Attached Data Specification Attribute		IDB Field
	Key Name	Field	
GM_F_USERDATA Table			
FromPersonal	*MyKeyName* (customer-defined)	mcr-from-name[a]	G_FROM_NAME
		mcr-called-back	G_CALLED_BACK
Subject		mcr-subject[a]	G_SUBJECT
Origination_Source		mcr-origin-source[a]	G_ORIGIN_SOURCE
FromAddress		mcr-from-address[a]	G_FROM_ADDRESS
		mcr-reserved-1 through mcr-reserved-4	G_RESERVED1 through G_RESERVED4

Table 7: Predefined Attached Data Fields for Multimedia Interactions (Continued)

Predefined Key Name	Attached Data Specification Attribute		IDB Field
	Key Name	Field	
GM_L_USERDATA Table			
	MyKeyName (customer-defined)	mcr-suggested-response	G_S_RESPONSE
		mcr-auto-response	G_A_RESPONSE
		mcr-auto-ack	G_A_ACK
ContactId		mcr-ucs-contact-id[a]	G_UCS_CONTACT_ID

a. If this field attribute 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 IDB Columns—Custom Fields

Table 8 shows the IDB fields that ICON creates 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.

Table 8: Predefined Attached Data Fields for Custom Keys

Attribute Name	G_CALL_USERDATA_CUST* Field Name
cust-data-1	CUST_DATA_1
cust-data-2	CUST_DATA_2
cust-data-3	CUST_DATA_3
cust-data-4	CUST_DATA_4
cust-data-5	CUST_DATA_5
cust-data-6	CUST_DATA_6
cust-data-7	CUST_DATA_7
cust-data-8	CUST_DATA_8
cust-data-9	CUST_DATA_9
cust-data-10	CUST_DATA_10
cust-data-11	CUST_DATA_11

Table 8: Predefined Attached Data Fields for Custom Keys (Continued)

Attribute Name	G_CALL_USERDATA_CUST* Field Name
cust-data-12	CUST_DATA_12
cust-data-13	CUST_DATA_13
cust-data-14	CUST_DATA_14
cust-data-15	CUST_DATA_15
cust-data-16	CUST_DATA_16
cust-data-17	CUST_DATA_17
cust-data-18	CUST_DATA_18
cust-data-19	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 (see page 100). If you require some or all of the following keys to be stored, explicitly define the respective keys in your attached data specification.

source="userdata"

CBR-Interaction_cost
CBR-IT-path_DBIDs
CBR-actual_volume
CBR-contract_DBIDs
RStrategyDBID
ServiceType
ServiceObjective
RVQID
RTargetObjSelDBID
RTargetRequested
RTargetAgentGroup
RTargetRuleSelected[a]
RTargetAgentSelected[a]
RTargetPlaceSelected[a]
RStrategyName[a]
RRequestedSkillCombination[a]
RTenant[a]
RTargetUsed/RTargetName
RTargetUsed/RTargetType
RTargetAgSelDBID
CustomerSegment
RTargetPlSelDBID
RRequestedSkills
RTargetPlaceGroup

RTargetObjectSelected[a]
RTargetTypeSelected[a]
RTargetCampaignGroup
RouterData70

source="reasons"

RTR
CBR-Interaction_cost
CBR-IT-path_DBIDs
CBR-actual_volume
CBR-contract_DBIDs
RStrategyDBID
ServiceType
ServiceObjective
RVQID
RTargetObjSelDBID
RTargetRuleSelected[a]
RTargetObjectSelected[a]
RTargetTypeSelected[a]
RTargetAgentSelected[a]
RTargetPlaceSelected[a]
RStrategyName[a]
RRequestedSkillCombination[a]
RTenant[a]
RTargetUsed/RTargetName
RTargetUsed/RTargetType
RTargetAgSelDBID
CustomerSegment
RTargetPlSelDBID
RRequestedSkills
RTargetPlaceGroup
RTargetCampaignGroup
RouterData70

source="extensions"

Reasons/RTR
Reasons/ServiceType
Reasons/ServiceObjective
Reasons/RVQID
Reasons/RTargetObjSelDBID
Reasons/RStrategyDBID
Reasons/RTargetPlaceGroup
Reasons/RTargetCampaignGroup
Reasons/RouterData70
ReportingEventSequenceNumber
Reasons
RTargetUsed

Reasons/RTargetRuleSelected[a]	RTargetUsed/RTargetName
Reasons/RTargetObjectSelected[a]	RTargetUsed/RTargetType
Reasons/RTargetTypeSelected[a]	Reasons/RTargetUsed
Reasons/RTargetAgentSelected[a]	Reasons/RTargetUsed/RTargetNam e
Reasons/RTargetPlaceSelected[a]	Reasons/RTargetUsed/RTargetType
Reasons/RStrategyName[a]	Reasons/CBR-IT-path_DBIDs
Reasons/RRequestedSkillCombination[a]	Reasons/CBR-Interaction_cost
Reasons/RTenant[a]	Reasons/CBR-actual_volume
Reasons/RTargetAgSelDBID	Reasons/CBR-contract_DBIDs
Reasons/CustomerSegment	Reasons/RTR
Reasons/RTargetPlSelDBID	Reasons/RTargetAgentGroup
Reasons/RRequestedSkills	Reasons/RTargetRequested

a. As a result of separate ICON processing, the value of this key 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.

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 “Sample Basic Attached Data Specification” on page 165.

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 8.0 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 8.0 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 8.0 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 8.0 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.

Note: To store extended route results in IDB, ICON requires URS release 7.6 and Interaction Server release 7.6.000.18 (or higher).

For more information about these options, see page 119.

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 page 148.

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 page 146.

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

ICON Application

Connections 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.

Configuring for Agent Data

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 on page 104.

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` (page 95) and `gls-active-reason-codes` (page 97). Review the options between page 95 and 97, and set appropriate values for your deployment.

Using Custom States

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

- Set the appropriate ICON application configuration options:
  - `AgentRecordUserTypes`
  - `AgentUserFields`
  - `EventData`
  - `GlobalData`
  - `store-event-data`

  The `AgentRecordUserTypes` option defines the custom agent states.

  For more information, see the descriptions of the options starting on page 120.
- 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` (see "Agent Desktop Application Configuration").

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`.

Starting Recording of 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", "+"
```

Sending 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`

Stopping Recording of 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", "-"
```

Using Multiple Custom States Simultaneously

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` (see “Example”).

Example

```
"AfterCallWork", "+"
"Break", "+"
"Comment", "207, This is data about the call"
"Comment", "208, This is data about the break"
"Break", "-"
"AfterCallWork", "-"
```

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

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 Server in the *Interaction Concentrator 8.0 User's Guide*.

Outbound Contact

Special configuration 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

There are two field-level configuration options that control whether ICON will receive and store field values:

- `icon_attribute`
- `send_attribute` (see page 78)

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. Go to the Configuration Manager main window.
2. Open the `Properties` dialog box for a particular `Field` configuration object.
3. Click the `Annex` tab.
4. Create a new section named `default`, if it does not already exist.
5. Within this section, create a new option named `icon_attribute`.
6. 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 Table 9. The table also shows the default key name for the attached data key-value pair.

Table 9: Mandatory Fields Attached by Default

Field	Key Name
contact_info	GSW_PHONE
chain_id	GSW_CHAIN_ID
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 ICON receives no such data.

For more information, see the section about field-level options in the Outbound Contact configuration options chapter in the *Outbound Contact 8.0 Deployment Guide*. See also the section in the *Outbound Contact 8.0 Reference Manual* about attaching record information to desktop and OCS user events.

Note: ICON reads `Field` object configurations only at startup. ICON does not recognize configuration changes that are made to `Field` objects during runtime. If you change `Field` object configuration settings, you must restart ICON in order for ICON to accept the changes.

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.

Note: You must restart Interaction Concentrator after a backup instance is configured of any application, such as OCS or T-Server, for which Interaction Concentrator 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.

Outbound-Related Options

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 on page 104.

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. For more information, see the description of the role option in the *Interaction Concentrator 8.0 Deployment Guide*.

OCS-Specific 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 starting on page 110. Select the appropriate values for your environment, and make related 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.

Example

For example, you might create an outbound `Calling List` object under a tenant called Outbound, and the calling list uses 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. For more information, see "icon_attribute" on page 77.
   - Configure the `send_attribute` option for all the fields that you want ICON to store. For more information, see "send_attribute" on page 78.
2. Add both the Environment tenant and the Outbound tenant on the `Tenants` tab of the ICON `Application`.

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 8.0 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 sections in this chapter.

- 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 8.0 Deployment Guide.*

Chapter

5 Starting and Stopping Interaction Concentrator

This chapter describes the prerequisites for Interaction Concentrator startup, and it provides instructions for starting and stopping Interaction Concentrator (ICON). It contains the following sections:

Overview

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.

Note: For information about using the Management Layer, startup files, and Services Manager for startup, see the *Framework Deployment Guide*.

Before You Begin

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.

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.

Note: 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 Procedure: Configuring an ICON application object, Step 10 on page 50.

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 “ICON Configuration Options” on page 93.

Command-Line Parameters

The following startup command-line parameters are supported by Interaction Concentrator:

`-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.
`-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 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 ICON

This section provides manual startup instructions for ICON server. You can start ICON in any of the following ways:

- From SCI (see page 86).
- Manually on UNIX (see page 87).
- Manually on Windows (see page 88).
- As a Windows Service on Windows (see page 88).

Starting ICON with Solution Control Interface

Complete the following procedure to start ICON with Solution Control Interface (SCI).

Procedure: Starting ICON with SCI

Prerequisites

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.)

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 8.1 Deployment Guide*
- *eServices Deployment Guide*

Start of procedure

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.

End of procedure

Next Steps

- You have completed all the steps necessary to start ICON using SCI.

Starting ICON Manually

Complete the following procedure to start ICON manually on UNIX.

Procedure: Starting ICON manually on UNIX

Start of procedure

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.

Note: 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"
```

End of procedure

Next Steps

- 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.

Procedure: Starting ICON on Windows

Purpose: To start ICON from the `Start > Programs` menu, or from the console window.

Start of procedure

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.

Note: 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"
```

End of procedure

Next Steps

- 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.

Procedure: Starting ICON as a Windows service

Start of 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.)

Note: 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.

End of procedure

Next Steps

- You have completed all the steps necessary to start ICON as a Windows service.

Stopping ICON

You can stop ICON in any of the following ways:

- From SCI (see page 90). (This is the recommended method.)
- Manually on UNIX (see page 90).
- Manually on Windows (see page 91).
- As a Windows Service on Windows (see page 92).

Note: To prevent ICON from self-starting, make sure that you clear the `autorestart` property in the ICON `Application` object in Configuration Manager.

Stopping ICON with Solution Control Interface

If you are using LCA and SCS, complete the following procedure to stop ICON with SCI.

Procedure: Stopping ICON using SCI

Start of procedure

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.

End of procedure

Next Steps

- You have completed all the steps to stop ICON using SCI.

Stopping ICON on UNIX

Stop ICON on UNIX by using one of the following procedures.

Procedure: Stopping ICON on UNIX from the command line

Start of procedure

- On the command line, enter the following:

  `kill -SIGTERM <processid>`

  Where `<processid>` is the application's UNIX process ID.

End of procedure

Next Steps

- You have completed all the steps to stop ICON from the command line.

Procedure: Stopping ICON on UNIX from the console window

Start of procedure

- From the active console window, press `CTRL+C`.

End of procedure

Next Steps

- You have completed all the steps to stop ICON from the console window.

Note: If you are using LCA and SCS, you can also use SCI to stop ICON (see "Stopping ICON with Solution Control Interface" on page 90).

Stopping ICON on Windows

If ICON is running as an application—not as a Windows Service—stop it using the following procedure.

Procedure: Stopping ICON on Windows from the console window

Start of procedure

- From the application's console window, press `CTRL+C`.

End of procedure

Next Steps

- You have completed all the steps to stop ICON from the console window.

Note: If you are running ICON as a Windows Service, you should stop it only from the Services Control Manager (see "Stopping ICON as a Windows Service" below).

Stopping ICON as a Windows Service

To stop Interaction Concentrator running as a Windows Service, use the following procedure.

Procedure: Stopping ICON running as a Windows service

Start of 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`.

End of procedure

Next Steps

- You have completed all the steps to stop ICON running as a Windows Service.

Chapter

6 Configuration Options

This chapter describes the configuration options that you can set for effective operation of your Interaction Concentrator (ICON) `Application` object. It contains the following sections, one for each `Application` object for which you must set options:

Note: In addition to the configuration options described in this chapter, Interaction Concentrator supports the *common log options* that are described in the *Framework 8.0 Configuration Options Reference Manual.*

Interaction Concentrator also connects to Message Server, but no ICON-specific configuration settings are required. For information about Message Server configuration options, see the *Framework 8.0 Configuration Options Reference Manual*.

ICON Configuration Options

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

Available Options

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` (see page 94)
- `custom-states` (see page 120)
- `dbw-error-reactions` (see page 123)
- `filter-data` (see page 125)
- `listeners` (see page 132)
- user-defined section identifying the HTTP connection (see page 133)
- `log` (see page 134)

callconcentrator Section

You must name this section `callconcentrator` in the configuration. The options in this section configure the main functionality of Interaction Concentrator.

Note: The name of this configuration option section is derived from the type of application that is used for the Interaction Concentrator `Application` object in the Genesys Configuration Layer. Otherwise, there is no relationship between the Genesys Interaction Concentrator and Call Concentrator products.

Table 10 describes the options in the `callconcentrator` section. For ease of reference, the options are grouped by area of functionality (presented in alphabetical order) as follows:

- Agent login session metrics, page 95
- Agent login session, page 97
- Attached data, page 98
- Custom dispatcher, page 101
- Database writing, page 101
- ICON Role, page 104
- IDB, page 105
- In-memory queue, page 105
- Multimedia, page 106
- Operational memory, page 108
- Outbound metrics, page 110
- Partitioning, page 111

- Persistent queue, page 112
- Scenario recognition, page 115
- Synchronization, page 116
- Virtual Queue, page 119

Table 10: ICON Configuration Options—callconcentrator Section

Area of Functionality	Option	Description
Agent login session metrics	gls-acw-first	Specifies which interaction ICON associates with after-call work (ACW). 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 immediately precedes the *start* of the after-call work (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. This option applies to all switches that ICON is configured to monitor; but, this option's value does not override the value of the `gls-acw-first` configuration option (described on page 140) if configured within the switch's configuration object. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Agent login session metrics (continued)	gls-stats-update	Specifies whether agent metrics (such as the duration of a particular agent state) are updated in IDB (`GS_AGENT_STAT` and `GS_AGENT_STAT_WM` tables) as the agent login session progresses. By default (value = `false`), ICON stores agent metrics only after an agent login session ends. 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`. Default Value: `10` Valid Values: Any integer Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Agent login session metrics (continued)	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. **Note:** To provide event sequence numbers with multimedia events, Interaction Server release 7.6 is required. 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	Specifies whether ICON captures and stores the values of active Agent state reason codes. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Attached data	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 (see page 99). ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud` (see page 104). This option applies to voice and multimedia interactions. Default Value: `public` Valid Values: `public, secure` Changes Take Effect: After restart **Note:** For descriptions of these values, see "Storage Types" on page 66.
	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 (see page 99). ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud` (see page 104). This option applies to voice and multimedia interactions. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Attached data (continued)	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 (see page 99). ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud` (see page 104). This option applies to voice interactions only. 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` (see page 104). For more information about the specification, see "Attached Data Specification File" on page 65. Default Value: `ccon_adata_spec.xml` Valid Values: Any valid name Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Attached data (continued)	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 (see page 99). ICON processes this option only if you enable attached data storage by setting the `role` option to either `all` or `gud` (see page 104). This option applies to voice and Multimedia interactions. 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
	suppress-user-data	Specifies whether ICON instructs T-Server to propagate attached data only when the attached data changes. This optimizes processing of attached data by reducing network traffic. **Note:** This option can be set at the level of the ICON application or the switch (see the description of the `suppress-user-data` option on page 146). 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 application-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. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Custom dispatcher	gud-cust-disp	Specifies whether ICON calls a custom stored procedure to handle attached data and store the information in custom tables. **Note:** ICON starts executing the new custom dispatcher as soon as the new configuration option value is set. Processing that was begun by the old custom dispatcher of interaction information stored in the persistent queue is handled in IDB by the old custom dispatcher. 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 **Note:** For more information, refer to the section about custom dispatchers in the *Interaction Concentrator 8.0 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). Default value: `16` Valid values: `0`–`255` (`0` indicates that ICON will process no group) Changes Take Effect: After restart
Database writing	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. Default Value: `600` Valid Values: Any integer Changes Take Effect: Immediately **Examples:** • `dbw-request-tout = 30` • `dbw-request-tout = 120`

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Database writing (continued)	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 *N*th 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 *N*th transaction, ICON uses the same logic to update the value of the `SEQCOUNTER` field in the `G_PROV_CONTROL` table. Default Value: `500` Valid Values: Any integer Changes Take Effect: After restart **Examples:** • `dbw-seq-step = 100` • `dbw-seq-step = 300`

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Database writing (continued)	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 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 8.0 User's Guide.*) Default Value: `60` Valid Values: Any integer Changes Take Effect: Immediately **Examples:** • `dbw-seq-tout = 30` • `dbw-seq-tout = 120`

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
ICON Role	role	Specifies the type of data that this ICON instance processes and stores in IDB. 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 Outbound Contact, stores OCS data that pertains to outbound calls and campaigns. **Note:** 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. Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
ICON Role (continued)	role (continued)	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. **Note:** Ensure that the role that you specify for the ICON instance is consistent with the role that you specify for the DAP (see page 151). **Examples:** • `role = cfg,gcc,gud` • `role = all` • `role = gcc,gud,gls,gos` • `role = ~cfg` (The last two examples are equivalent.)
IDB	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. Default value: Empty Valid values: Any string Changes Take Effect: After restart
In-memory queue	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. 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. Default Value: `5` Valid Values: Any positive integer Changes Take Effect: Immediately

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
In-memory queue (continued)	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. Default Value: `500` Valid Values: Any positive integer Changes Take Effect: Immediately
Multimedia	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. **Note:** ICON cannot restore a correct timestamp of interaction record creation, or the information about previous parties, or the first values of user data keys. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Multimedia (continued)	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. 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 **Note:** 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.

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Multimedia (continued)	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 With the 0 setting, ICON processes neither interactions nor agent data for 3rd Party Media. 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 **Note:** 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 8.0 User's Guide.*
Operational memory	om-check-filter-flag	Specifies whether or not ICON stores strategy activity according to the value of the om-activity-report configuration option (see page 150) that is defined on 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. Default Value: 1 Valid Values: 0, 1 Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Operational memory (continued)	om-max-in-memory	Specifies the maximum number of keep-in-memory interactions that were placed in queues or interaction workbins (in units of one thousand). 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 Technical Support. 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. **Note:** An incorrect value for this option can affect ICON performance, or cause ICON to stop processing interactions.
	om-memory-optimization	Specifies whether or not memory usage will be optimized. Default Value: `false` Valid Values: `true` ICON optimizes memory usage according to the values that are set for the following options: `om-max-in-memory` (see page 109) and `om-memory-clean` (see page 149). `false` Preserves legacy behavior (prior to ICON release 7.6.1). Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Outbound metrics	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`. 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 **Note:** For more information about outbound-related metrics, refer to the chapter about integrating with Outbound Contact in the *Interaction Concentrator 8.0 User's Guide.*
	gos-write-metrics	Specifies whether ICON writes any precalculated OCS metrics to IDB. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Outbound metrics (continued)	gos-write-metrics-only	Specifies whether ICON excludes from database storage all outbound data except precalculated metrics. 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 (see page 110). `1` Provided that the `gos-write-metrics` option is also set to 1, ICON stores precalculated metrics, but not the other data that OCS provides. Changes Take Effect: After restart
Partitioning	partition-type	Dynamically specifies the content of the `gsys_partition` field in IDB tables that contain this field. Default value: `0` Valid Values: `0, 1, 2` `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. **Note:** For voice interactions, setting `partition-type=2` has the same effect as setting `partition-type=1`. Changes Take Effect: Immediately

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Persistent queue	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. 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.

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Persistent queue (continued)	pq-backlog-alarm-threshold (continued)	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 8.0 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. Default value: `0` Valid values: `0`–(2^{32}–1) (`0` indicates that no log message will be generated) Changes Take Effect: Immediately
	pq-backlog-clearance-threshold	Specifies the minimum number of records pending in the persistent queue at which ICON will generate message `25026`, if log message `25025` was previously generated (see `pq-backlog-alarm-threshold`). Default value: `0` Valid values: 0–value of `pq-backlog-alarm-threshold` (`0` indicates that no log message will be generated) Changes Take Effect: Immediately
	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 (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. Default Value: `icon.pq` Valid Values: Any valid file name Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Persistent queue (continued)	pq-dbname (continued)	**Notes:** • 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 `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. 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. Default Value: `true` Valid Values: `true, false` Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Scenario recognition	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. 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. For information about how ICON populates the values of the fields, see the section about identifying who released the call in the *Interaction Concentrator 8.0 User's Guide*. **Note:** In Interaction Concentrator release 8.0, this feature is supported for the Alcatel A4400/OXE switch and, if you are running Interaction Concentrator release 8.0.000.35 or higher, for Avaya switches (requires Avaya Communication Manager 8.0.101.05 or higher). `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
	gcti-mode-monitoring	Regulates the mode that ICON uses for multi-site scenario recognition. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Synchronization	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. **Note:** 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. 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

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Synchronization (continued)	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 will be 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. **Note:** Genesys recommends that you do not change the default value. Default value: `1000000` Valid values: `0`–(2^{32}–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 (page 145). Default Value: `1000` Valid Values: `0`–`2000` (`0` indicates that no log message will be generated) Changes Take Effect: Immediately

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
synchronization (continued)	use-dss-monitor	Valid Values: true, false Default Value: false When you set the value for this option to 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. **Note:** If you want the G_DSS_*_PROVIDER tables to be populated, you must set the value to true. If the value for this option is 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, if this option set to false, ICON does not write data to the G_DSS_*_PROVIDER tables. Changes Take Effect: After restart
	dss-no-data-tout	Valid Values: Any integer in the range of 60 to 86400 Default Value: 300 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. 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. Changes Take Effect: After restart

Table 10: ICON Configuration Options—callconcentrator Section (Continued)

Area of Functionality	Option	Description
Virtual Queue	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. 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—statuses of interactions distributed by URS 7.6 or higher—in IDB. Default Value: `0` Valid Values: `0` ICON stores route results in `G_VIRTUAL_QUEUE` and `G_ROUTE_RESULT` IDB tables as implemented in ICON release 7.5. `1` ICON stores extended routing results in `G_VIRTUAL_QUEUE` and `G_ROUTE_RESULT` IDB tables as implemented in ICON release 7.6 (see Note 2). Changes Take Effect: After restart **Note 1:** 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 8.0 User's Guide.* **Note 2:** The feature implemented in Interaction Concentrator 7.6 related to writing extended routing results into IDB (`G_ROUTE_RESULT` and `G_VIRTUAL_QUEUE` IDB tables) requires: • Universal Routing Server (URS) release 7.6 or higher. • URS configuration options `report_reasons` and `report_targets` set to `true`.

custom-states Section

You must name this section `custom-states` in the configuration.

The options in this section configure Interaction Concentrator support for the processing of custom agent states and common user data.

Table 11 describes the options in the `custom-states` section. For ease of reference, the table presents the options in alphabetical order.

Table 11: ICON Configuration Options—custom-states Section

Option	Description
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 (see page 121). For more information about ICON custom state recording, see "Using Custom States" on page 75. 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`

Table 11: ICON Configuration Options—custom-states Section (Continued)

Option	Description
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. 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>. **Note:** 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. Changes Take Effect: After restart **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. Default Value: No default value Valid Values: A comma-separated list of the data types and key names in the format <Type>,<KeyName>. **Note 1:** The limit for option specifications in Configuration Manager 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 from 1. ICON recognizes all the EventData specifications as one option, and it concatenates the content of the options in sequence. **Note 2:** 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. **Note 3:** Ensure that the key name you specify does not conflict with a key name specified in the GlobalData option (see page 122). The key names specified in the EventData and GlobalData options must be unique. Changes Take Effect: Immediately **Example:** EventData = char,CUSTOMER_NAME,int,CUSTOMER_PHONE

Table 11: ICON Configuration Options—custom-states Section (Continued)

Option	Description
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. Default Value: No default value Valid Values: A comma-separated list of the data types and key names in the format `<Type>,<KeyName>`. **Note 1:** The limit for option specifications in Configuration Manager is 255 characters. If your desired `GlobalData` option specification exceeds this limit, you can specify additional options in the format `GlobalData_X`, where `X` is any integer, starting from `1`. ICON recognizes all the `GlobalData` specifications as one option, and it concatenates the content of the options in sequence. **Note 2:** All the custom data fields in the `G_CUSTOM_DATA_P` 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 `G_CUSTOM_DATA_P` table. If the value of the key in the `UserData` KVP is `KVList`, ICON ignores the value. **Note 3:** Ensure that the key name you specify does not conflict with a key name specified in the `EventData` option (see page 121). The key names specified in the `EventData` and `GlobalData` options must be unique. Changes Take Effect: Immediately **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.

Table 11: ICON Configuration Options—custom-states Section (Continued)

Option	Description
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. 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

You must name this section `dbw-error-reactions` in the configuration.

The 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, the specified error reaction will be applied.

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

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

Preconfigured Error

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

Option Name: `uniqueness`

Option Value: `error=unique;reaction=ignore`

Table 12 describes the options in the `dbw-error-reactions` section.

Table 12: ICON Configuration Options—dbw-error-reactions Section

Option	Description
<error-name>	Defines how ICON reacts to a database error message that contains a particular text substring. Create a separate option for every database error message for which a certain reaction is required. Specify any meaningful name as the option name, making it unique within the `dbw-error-reactions` section; ICON does not process the name parameter. 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.

Table 12: ICON Configuration Options—dbw-error-reactions Section (Continued)

Option	Description

<error-name> (continued)	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

You must name this section `filter-data` in the configuration.

The options in this section control Interaction Concentrator output to IDB. Review the options in this section carefully and refer to the *Interaction Concentrator 8.0 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. Excluding certain types of data from IDB storage may help you save database space, and thus improve your database performance.

Table 13 describes the options in the `filter-data` section. For ease of reference, the table presents the options in alphabetical order.

Table 13: ICON Configuration Options—filter-data Section

Option	Description
acd-party-history	Specifies whether ICON should exclude, from IDB storage, party history information about distribution devices—such as ACD queues, Routing Points, virtual routing points, and External Routing Points. By default, ICON collects party history information about distribution devices and stores this information in the `G_PARTY_HISTORY` IDB table. With a setting of `1`, ICON does not store party history information in the `G_PARTY_HISTORY` table. **Notes:** • The `acd-party-history` option applies to SIP and voice interactions only. • If your deployment supports identifying the party that initiated release of the call, ensure that the value of this option is `0`. See the note on page 132 for more information. Default Value: `0` Valid Values: `1, 0` Changes Take Effect: After restart
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. With a setting of `1`, ICON does not store data in the `G_PARTY_STAT` table for distribution devices. **Notes:** • The `acd-party-metrics` option applies to SIP and voice interactions only. • If your deployment supports identifying the party that initiated release of the call, ensure that the value of this option is `0`. See the note on page 132 for more information. Default Value: `0` Valid Values: `1, 0` Changes Take Effect: After restart
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. With a setting of `1`, ICON ceases writing to this table. Default Value: `0` Valid Values: `1, 0` Changes Take Effect: After restart

Table 13: ICON Configuration Options—filter-data Section (Continued)

Option	Description
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. With a setting of `1`, ICON ceases writing to this table. Default Value: `0` Valid Values: `1`, `0` Changes Take Effect: After restart **Note:** If your deployment supports identifying the party that initiated release of the call, ensure that the value of this option is `0`. See the note on page 132 for more information.
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` With a setting of `1`, ICON collects and stores data about internal parties only (for example, interaction participants within a given switch domain). Default Value: `0` Valid Values: `1`, `0` Changes Take Effect: After restart **Note:** If your deployment supports identifying the party that initiated release of the call, ensure that the value of this option is `0`. See the note on page 132 for more information.

Table 13: ICON Configuration Options—filter-data Section (Continued)

Option	Description
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, agent state—unless certain types of data are configured to be excluded by setting one or more of the following options to `1`: • `gls-ivr` (see page 129) • `gls-no-person` (see page 130) • `gls-queue` (see page 130) • `gls-wm` (see page 131) 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` With a setting of `1` for the `gls-all` option, ICON ceases writing to these tables. Note, however, that with a `1` setting, ICON continues writing agent's ID to the `G_PARTY` table. Default Value: `0` Valid Values: `1, 0` Changes Take Effect: After restart

Table 13: ICON Configuration Options—filter-data Section (Continued)

Option	Description
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 With a setting of 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. Default Value: 0 Valid Values: 1, 0 Changes Take Effect: After restart **Note:** See page 148 for more information about setting the ivr option to configure a DN as an IVR port. For a description of how ICON identifies an IVR, see the *Interaction Concentrator 8.0 User's Guide.*
gls-metrics	Specifies whether ICON should exclude agent states from IDB storage. 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 (see page 128) • gls-ivr (see page 129) • gls-no-person (see page 130) • gls-queue (see page 130) • gls-wm (see page 131) ICON stores agent states information in the following IDB tables: • GS_AGENT_STAT • GS_AGENT_STAT_WM With a setting of 1, ICON does not store information about agent states to these tables. Default Value: 0 Valid Values: 1, 0 Changes Take Effect: After restart

Table 13: ICON Configuration Options—filter-data Section (Continued)

Option	Description
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` With a setting of `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. 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 agents' 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` With a setting of `1`, ICON ceases writing queue-related data to the first four tables (above). ICON will continue writing information to the `GX_SESSION_ENDPOINT` table about the queues where agents are logged in. Default Value: `0` Valid Values: `1`, `0` Changes Take Effect: After restart

Table 13: ICON Configuration Options—filter-data Section (Continued)

Option	Description
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 agents' work mode and changes in agents' work modes in the following IDB tables: • G_AGENT_STATE_HISTORY • G_AGENT_STATE_RC • GS_AGENT_STAT_WM With a setting of 1, ICON ignores information about work mode and work mode changes. It records a value of unknown in the (above) IDB tables. **Note:** This option does not affect ICON's ability to track after-call work. 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. With a setting of 1, ICON ceases writing data to this table. 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 With a setting of 1, ICON does not store data about the party with the role Observer to these tables. Default Value: 0 Valid Values: 1, 0 Changes Take Effect: After restart **Note:** If your deployment supports identifying the party that initiated release of the call, ensure that the value of this option is 0. See the note on page 132 for more information.

Table 13: ICON Configuration Options—filter-data Section (Continued)

Option	Description
udata-history-terminated	Specifies whether ICON should exclude, from IDB storage, information about changes in `UserData` values for certain keys. When ICON is configured in a way that it should 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` With a setting of `1`, ICON does not insert new records in these tables at call termination time. ICON does, however, continue to write information about the creation, addition, and removal of key-value pairs to these tables. Default Value: `0` Valid Values: `1, 0` Changes Take Effect: After restart

Note: To enable identification of the party that initiated release of the call, in deployments that support this feature, ensure that the value of the `external-party`, `acd-party-metrics`, `acd-party-history`, 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. Therefore, ICON will not be able to identify whether call termination was initiated on the endpoint associated with this party or on the other resource.

Similarly, ensure that the value of the `call-metrics` option is `0` (the default value). Otherwise, ICON will not store records in the `G_CALL_STAT` table.

For more information about the releasing-party feature, see the section about identifying who released the call in the *Interaction Concentrator 8.0 User's Guide*.

listeners Section

You must name this section `listeners` in the configuration.

This section references a separate configuration section that describes the HTTP listening port.

Table 14 describes the options in the `listeners` section.

Table 14: ICON Configuration Options—listeners Section

Option	Description
<http-connection>	Points to the configuration of an HTTP connection to ICON. The option name must replicate the name of another configuration section in the ICON Application object. Because ICON does not process this option value, you can use it to enter a short description for the connection; ICON will print this description to its log. 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 option that describes the parameters of an HTTP connection at a port that ICON opens for listening. **Note:** To enable access to the performance counters, configure an HTTP Listener option, and then configure a corresponding section with its options.

<http-connection> Section

You can choose any name for this section, provided that it matches the name that you specify for one of the options in the listeners section. Use the options described in Table 15 to set parameters for this connection.

Example:

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

Table 15: ICON Configuration Options—<http-connection> Section

Option	Description
port	Specifies the number of the port that ICON opens for HTTP listening. 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. **Note:** The HTTP interface is not available by default. 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. 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 8.0 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 Interaction Concentrator's log options.

For the log options described in Table 16, the meaning of the valid values is as follows:

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

Table 16 describes the options in the log section. For ease of reference, the table presents the options in alphabetical order.

Table 16: ICON Configuration Options—log Section

Option	Description
x-conn-debug-open	Specifies the verbosity with which ICON logs messages that are 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 (see page 137), with regard to network connection messages. 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 that are 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 (see page 137), with regard to incoming information messages. 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 that are 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 (see page 137), with regard to connection timer–triggering messages. 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 that are 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 (see page 137), with regard to outgoing information messages. Default Value: As specified by the `x-server-trace-level` option Valid Values: `0|1|2|3` Changes Take Effect: Immediately

Table 16: ICON Configuration Options—log Section (Continued)

Option	Description
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. 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-config-trace-level	Specifies the verbosity with which ICON logs messages that are 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 (see page 137), with regard to configuration information messages. 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 that are 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 (see page 137), with regard to data-writing operation messages. 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 `x-server-trace-level`, 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 (page 137).

Table 16: ICON Configuration Options—log Section (Continued)

Option	Description
x-server-gcti-trace-level	Specifies the verbosity with which ICON logs messages that are 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 (see page 137), with regard to CTI communications messages. Default Value: As specified by the `x-server-trace-level` option Valid Values: `0|1|2|3` 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` For more information about each function-specific option, see the descriptions starting on page 135. 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 that are related to that particular function. Default Value: `0` Valid Values: `0|1|2|3` Changes Take Effect: Immediately

Switch Configuration 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.

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.

Note: The information in this section does not apply to `Multimedia` switches.

gts Section

Table 17 describes the options in the `gts` section. For ease of reference, the table presents the options in alphabetical order.

Table 17: Switch Configuration Options—gts Section

Option	Description
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. 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). 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. **Note:** Genesys Technical Support recommends that you set the value of this option to `3` only for a particular event flow. Changes Take Effect: After restart

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
emulate-event-queued-extrp	Enables the emulation of `EventQueued` for an External Routing Point that belongs to this switch. **Note:** 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. Default Value: `1` Valid Values: `0` `EventQueued` is not emulated. `1` `EventQueued` is emulated. Changes Take Effect: Immediately **Note:** For help setting this option correctly, contact Genesys Technical Support.
emulate-event-queued-rp	Enables the emulation of `EventQueued` for a Routing Point that belongs to this switch. **Note:** 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. Default Value: `1` Valid Values: `0` `EventQueued` is not emulated. `1` `EventQueued` is emulated. Changes Take Effect: Immediately **Note:** For help setting this option correctly, contact Genesys Technical Support.
emulate-event-queued-rq	Enables the emulation of `EventQueued` for a routing queue that belongs to this switch. **Note:** 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. Default Value: `1` Valid Values: `0` `EventQueued` is not emulated. `1` `EventQueued` is emulated. Changes Take Effect: Immediately **Note:** For help setting this option correctly, contact Genesys Technical Support.

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
fix-time-stamps	Enables adjustment of timestamps when the CTI event contains an earlier timestamp than the timestamp from a previously received CTI event. 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, for this switch, which interaction ICON associates with after-call work (ACW). By default, ICON uses the setting of the `gls-acw-first` configuration option (described on page 95) as specified at the ICON application level. Setting this option to `1` enables ICON to associate after-call work with the voice interaction that immediately precedes the *start* of the after-call work (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. This option overrides an explicit setting of the `gls-acw-first` configuration option (described on page 95) at the ICON application level. **Note:** 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

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
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. 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

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
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 8.0 User's Guide*. 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. 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.) Changes Take Effect: After restart **Note:** This option is not valid for SIP-compliant switches that handle interactions other than voice interactions.

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
gls-flag-on-disconnect	Specifies how ICON handles agent states when disconnecting from, and reconnecting to, T-Server. 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 **Note 1:** Genesys recommends setting this option to `0` when the switch is monitored by T-Server 7.6. **Note 2:** 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 (see page 145) in the [`gts`] section to `0` on the switch `Annex` tab.

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
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. Default Value: `0` Valid Values: Any integer from `0` to `720` Changes Take Effect: Immediately **Note**: 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. Default Value: `0` Valid Values: Any integer from `0` to `72` Changes Take Effect: Immediately **Note**: 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.

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
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. 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. **Note:** If you set this option to `0`, make sure you also set the `gls-flag-on-disconnect` option (see page 143) to `1` in order to access available HA agent data. Changes Take Effect: After restart
load-balancing-on-ntwk-switch	Specifies whether this is a network switch, working in load-balancing mode, that is served by multiple Network T-Servers. ICON uses this option to determine whether to enable connection to more than one Network T-Server serving this switch. Default Value: `0` Valid Values: `1` Network switch in load-balancing mode. Any other integer Not a network switch in load-balancing mode. Either: • Not a network switch • A network switch, but not in load-balancing mode 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 (page 117). Default Value: `50` Valid Values: `0`–`500` (`0` indicates that no calculation will be performed) Changes Take Effect: Immediately

Table 17: Switch Configuration Options—gts Section (Continued)

Option	Description
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. 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. **Note:** For T-Server for Siemens HiPath 4000 CSTA III, set the option value to `1`. Changes Take Effect: Immediately
support-dn-type-5	Enables the processing of events that pertain to DNs of the `Virtual Queue` type that belong to this switch. 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. **Note:** This option can be set at the level of the switch or the ICON application (see `suppress-user-data` on page 100. 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 application-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. 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

DN Configuration Options

This section describes the ICON-related configuration options that 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.

Table 18 describes the options in the gts section. For ease of reference, the table presents the options in alphabetical order.

Table 18: DN Configuration Options—gts Section

Option	Description
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. **Note:** 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. 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

Table 18: DN Configuration Options—gts Section (Continued)

Option	Description
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. **Notes:** • 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 on page 129 to exclude data about agent activity associated with this IVR device from IDB storage. 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
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 (see page 146). 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

Script Configuration Options

This section describes the ICON-related configuration options that you configure on the `Annex` tab of `script` configuration objects of type `interaction queue` and `simple routing` (for a routing strategy). Interaction Concentrator processes these options.

callconcentrator Section

You must name this section `callconcentrator` in the configuration.

Table 19 describes the options in the `callconcentrator` section of `script` objects of type `interaction queue` and `simple routing`.

Table 19: Script Configuration Options

Option	Description
objects of type interaction queue	
om-memory-clean	Specifies whether or not ICON immediately removes an interaction from memory when the interaction is placed in the interaction queue. **Note:** The `om-memory-optimization` option (see page 109) must be set to `true` in order for `om-memory-clean` to work. 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. **Note:** 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 (see page 109). Changes Take Effect: After restart

Table 19: Script Configuration Options (Continued)

Option	Description
objects of type simple routing	
om-activity-report	Specifies whether or not ICON will store activity data that is related to a particular strategy. If the value is set to `false`, ICON will not store in IDB any data about parties for this strategy or any user data changes that are made by this strategy. **Note:** Regardless of the value of this option, ICON will always process `EventInteractionStopped` events. Default Value: `true` Valid Values: `true, false` Changes Take Effect: After restart

DAP Configuration Option

This section describes the ICON-related 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.

callconcentrator Section

You must name this section `callconcentrator` in the configuration.

Table 20 describes the option in the `callconcentrator` section of the DAP `Application` object.

Table 20: DAP Configuration Option

Option	Description
role	Specifies the type of data that ICON is allowed to store in IDB through this DAP. Use this option when you are writing different types of ICON data to different databases. Default Value: all Valid Values: A comma-separated list including any of the following: cfg — Stores the initial configuration state, and a history of configuration changes retrieved from Configuration Server. gcc — Stores call-related and party-related information—that is, T-Server and, if applicable, Interaction Server data that pertains to calls and the parties associated with those calls. gls — Stores T-Server and, if applicable, Interaction Server data that pertains to the agent states and agent login sessions. gud — Stores T-Server and, if applicable, Interaction Server data that pertains to the attached data associated with calls. gos — In an environment with Outbound Contact, stores OCS data that pertains to outbound calls and campaigns. Any combination of the valid values can be used. Prefixing an option value with a tilde (~) excludes that type of data from database storage through this DAP, and includes all other types. For example, the value ~cfg deactivates the storage of configuration data through this DAP, and activates processing and storage of all other types of data. All types of ICON data go through the same DAP in the following cases: • No role option is specified for the DAP. • The role option is specified, and its value is explicitly set to all. • You specified only one DAP Application object on the Connections tab of the ICON Application object. **Note 1:** 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. **Note 2:** Ensure that the role that you specify for the DAP is consistent with the role that you specify for the ICON instance (see page 104). Changes Take Effect: After restart **Examples:** • role = cfg,gcc,gud • role = all • role = ~cfg

Chapter

7 Troubleshooting ICON Installation and Deployments

This chapter describes problems that you might encounter when starting or running your Interaction Concentrator (ICON) application, and how to resolve them. It contains the following sections:

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

The following are the most common startup problems:

- ICON does not connect to the Configuration Server (see "No Connection to the Configuration Server" below).
- ICON closes at startup (see "ICON Exits at Startup" on page 154).

No Connection 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, see "Command-Line Parameters" on page 85.

- 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" on page 46.

- 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" on page 46.

- 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 "Configuring a DAP" on page 58.

- 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.
- 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. For more information about the `pq-dbname` configuration option, see page 113.

  In both cases, 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 the chapter about agent states and login sessions in the *Interaction Concentrator 8.0 User's Guide.*

Runtime Problems

The following are the most common runtime problems:

- ICON does not connect to T-Server or Interaction Server (see "No Connection to T-Server or Interaction Server" on page 156).
- ICON does not receive call-related events from T-Server (see "ICON Does Not Receive Call-Related Events from T-Server" on page 156).
- ICON does not write information to the database (see "ICON Does Not Write Information to the Database" on page 157).

- ICON has lost synchronization with the Configuration Database (see "ICON Has Lost Synchronization with the Configuration Database" on page 158).
- ICON functionality is not available as configured, or ICON functioning becomes unpredictable (see "ICON Does Not Function Correctly" on page 159).

No Connection 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/Multimedia. For more information about Interaction Concentrator compatibility and interoperability with other Genesys components, see "Compatibility" on page 35.

  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. For more information, see "Starting and Stopping Interaction Concentrator" on page 83.

- 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. For more information, see "Starting and Stopping Interaction Concentrator" on page 83.
- There is no connection between the ICON `Application` object and T-Server. See "No Connection to T-Server or Interaction Server" on page 156.

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 "Configuring a DAP" on page 58.
- 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 "Configuring a DAP" on page 58, and the description of the `role` option on page 151.
- IDB has not been initialized by the Interaction Concentrator initialization scripts.

  Solution: Run the Interaction Concentrator initialization scripts. For more information, see "Creating IDB" on page 54.
- ICON was not restarted after changes were made on the ICON `Application` object's `Connections` tab.

  Solution: Stop ICON, then restart. For more information, see "Starting and Stopping Interaction Concentrator" on page 83.
- 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. For more information, see "Starting and Stopping Interaction Concentrator" on page 83.

- 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 about in-memory queue configuration options, refer to the option descriptions starting on page 105.

- 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, refer to the chapter about monitoring ICON performance in the *Interaction Concentrator 8.0 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 (see the problem about a corrupted persistent queue file on page 155).

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 the

chapter about resynchronizing configuration data in the *Interaction Concentrator 8.0 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 Procedure: Configuring an ICON application object, Step 10 on page 50.

Merge Procedure Problems

The most common problems encountered in executing the merge procedure (`gsysIRMerge` or `gsysIRMerge2`) are as follows:

- The procedure fails to complete (see "Merge Procedure Does Not Complete Successfully" below).
- The procedure does not execute (see "Merge Procedure Does Not Execute" on page 160).
- Merge procedure performance is slow or constantly aborts (see "Merge Procedure Performance Is Slow or Unstable" on page 161).

For more information about the merge procedure, see the chapter about stored procedures in the *Interaction Concentrator 8.0 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"). Then restart the merge procedure.

Merge Procedure Recovery

Starting with release 8.0, Interaction Concentrator provides a stored procedure, `gsysIRMergeReset`, to simplify the steps to reset the merge procedure to recover from a failed state. Certain error conditions could be resolved by invoking the `gsysIRMergeReset` stored procedure that resets the merge procedure.

To invoke the procedure, use an SQL statement like the following (the exactsyntax depends on the RDBMS):

```
EXEC gsysIRMergeReset
```

Note: 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 merge procedure again.

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 the chapter about stored procedures in the *Interaction Concentrator 8.0 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 Technical Support.

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: Significantly increase the values of the `LOCKLIST` and `MAXLOCKS` database configuration parameters.

  Alternatively, execute `ALTER TABLE...LOCKSIZE TABLE` statements against the `G_IR`, `G_CALL`, and `G_IS_LINK` tables.

- On a Microsoft SQL platform, in large-scale deployments, default values of certain merge procedure parameters are not optimal.

  Solution: Significantly increase the values of the `step` and `limit` parameters in the `G_DB_PARAMETERS` table. For more information, see the note for large-scale deployments using Microsoft SQL, in the chapter about stored procedures in the *Interaction Concentrator 8.0 User's Guide*.

- 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" on page 160), 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 Technical Support.

Appendix

A Attached Data Specification File

This appendix 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 appendix.

This appendix contains the following sections:

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>
```

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. For additional modifications to capture multimedia-specific attached data, see “Sample Specification for Multimedia Attached Data” on page 167.

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 page 70.

```
<?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! **Note:** 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 will be 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 will not be updated.

Example

```
<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>
```

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 will be 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 Table 7 on page 68.

Sample adata_spec.xml File

In the GM_F_USERADATA 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_USERGDATA 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>
```

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 Appendix B on page 171.

```
<?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>
```

Appendix

B Sample Script for Custom Attached Data

This appendix 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.

Note: 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
```

Related Documentation Resources

The following resources provide additional information that is relevant to this software. Consult these additional resources as necessary.

Interaction Concentrator

- The *Interaction Concentrator 8.0 User's Guide*, which will help you learn more about Interaction Concentrator functionality, including IDB architecture, models used in Interaction Concentrator, attached data processing, available stored procedures, and integration with other Genesys products.
- The *Interaction Concentrator 8.0 Physical Data Model* for your relational database management system (RDBMS) type, which will help you learn about IDB tables and their relationships.
- The *Interaction Concentrator 8.0 Database Size Estimator*, which will help you estimate the size of your IDB when you are planning your deployment. The estimator is a Microsoft Excel spreadsheet available from the Genesys Technical Support website.

Optional Supplementary Products

- The documentation set for Genesys Info Mart release 8.0, if you intend to use Interaction Concentrator as a source of data for Genesys Info Mart.
- The documentation set for Genesys Outbound Contact release 8.0, if you intend to store outbound-related data in IDB.
- The documentation set for Genesys Universal Routing release 8.0, if you intend to store information about virtual queue usage in interaction processing in IDB.
- The documentation set for Genesys eServices/Multimedia release 8.0, if you intend to store interaction-related and related data about multimedia interactions in IDB.

Genesys

- *Genesys Technical Publications Glossary,* which ships on the Genesys Documentation Library DVD and which provides a comprehensive list of the Genesys and computer-telephony integration (CTI) terminology and acronyms used in this document.
- *Genesys Migration Guide*, which ships on the Genesys Documentation Library DVD, and which provides documented migration strategies for Genesys product releases. Contact Genesys Technical Support for more information.
- Release Notes and Product Advisories for this product, which are available on the Genesys Technical Support website at http://genesyslab.com/support.

Information about supported hardware and third-party software is available on the Genesys Technical Support website in the following documents:

- *Genesys Supported Operating Environment Reference Manual*
- *Genesys Supported Media Interfaces Reference Manual*

Consult these additional resources as necessary:

- *Genesys Hardware Sizing Guide,* which contains information about recommended hardware architectures and additional information related to database size estimation for Genesys 8.x releases.
- *Genesys 8.0 Combined Log Events Help,* which describes the log events generated by every Genesys server application, including Interaction Concentrator.

For additional system-wide planning tools and information, see the release-specific listings of System Level Documents on the Genesys Technical Support website, accessible from the system level documents by release tab in the Knowledge Base Browse Documents Section.

Genesys product documentation is available on the:

- Genesys Technical Support website at http://genesyslab.com/support.
- Genesys Documentation Library DVD, which you can order by e-mail from Genesys Order Management at orderman@genesyslab.com.

Document Conventions

This document uses certain stylistic and typographical conventions—introduced here—that serve as shorthands for particular kinds of information.

Document Version Number

A version number appears at the bottom of the inside front cover of this document. Version numbers change as new information is added to this document. Here is a sample version number:

80icon_dep_12-2009_v8.0.001.00

You will need this number when you are talking with Genesys Technical Support about this product.

Screen Captures Used in This Document

Screen captures from the product graphical user interface (GUI), as used in this document, may sometimes contain minor spelling, capitalization, or grammatical errors. The text accompanying and explaining the screen captures corrects such errors *except* when such a correction would prevent you from installing, configuring, or successfully using the product. For example, if the name of an option contains a usage error, the name would be presented exactly as it appears in the product GUI; the error would not be corrected in any accompanying text.

Type Styles

Table 21 describes and illustrates the type conventions that are used in this document.

Table 21: Type Styles

Type Style	Used For	Examples
Italic	• Document titles • Emphasis • Definitions of (or first references to) unfamiliar terms • Mathematical variables Also used to indicate placeholder text within code samples or commands, in the special case where angle brackets are a required part of the syntax (see the note about angle brackets on page 178).	Please consult the *Genesys Migration Guide* for more information. Do *not* use this value for this option. A *customary and usual* practice is one that is widely accepted and used within a particular industry or profession. The formula, $x + 1 = 7$ where x stands for . . .

Table 21: Type Styles (Continued)

Type Style	Used For	Examples
Monospace font (Looks like `teletype` or `typewriter text`)	All programming identifiers and GUI elements. This convention includes: • The *names* of directories, files, folders, configuration objects, paths, scripts, dialog boxes, options, fields, text and list boxes, operational modes, all buttons (including radio buttons), check boxes, commands, tabs, CTI events, and error messages. • The values of options. • Logical arguments and command syntax. • Code samples. Also used for any text that users must manually enter during a configuration or installation procedure, or on a command line.	Select the `Show variables on screen` check box. In the `Operand` text box, enter your formula. Click `OK` to exit the `Properties` dialog box. T-Server distributes the error messages in `EventError` events. If you select `true` for the `inbound-bsns-calls` option, all established inbound calls on a local agent are considered business calls. Enter `exit` on the command line.
Square brackets ([ ])	A particular parameter or value that is optional within a logical argument, a command, or some programming syntax. That is, the presence of the parameter or value is not required to resolve the argument, command, or block of code. The user decides whether to include this optional information.	`smcp_server -host [/flags]`
Angle brackets (< >)	A placeholder for a value that the user must specify. This might be a DN or a port number specific to your enterprise. **Note:** In some cases, angle brackets are required characters in code syntax (for example, in XML schemas). In these cases, italic text is used for placeholder values.	`smcp_server -host <confighost>`

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X