



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

Interaction Concentrator Deployment Guide

Special Configuration Requirements

12/20/2025

Special Configuration Requirements

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

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

Voice

Storing Voice Data

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

Connections:

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

Configuring for Voice Data

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

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

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

Filtering Data

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

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

certain ICON filtering options can effectively disable this functionality.

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

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

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

Multimedia

Storing Multimedia Data

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

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

[+] ICON Application

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

Connections

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

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

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

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

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

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

Important

Starting in release 8.1.2, if host or port information changes for any T-Server listed on the Connections tab, Interaction Concentrator dynamically reconnects using the new connection parameters.

While ICON disconnects from the prior host/port and connects to the new one, there might be a brief gap in data received from the T-Server.

In releases 8.1.0 and 8.1.1, you must restart ICON for updates to listed T-Servers to take effect.

[+] Other Configuration for Multimedia

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

[+] Configuring for 3rd Party Media

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

Attached Data

Storing Attached Data

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

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

1. Specify the attached data key configuration file, which maps the key-value pairs (KVPs) in reporting event attributes to IDB tables and fields. For more information, see the [Attached Data Specification File](#).
2. Specify the attached data configuration settings in the Genesys Configuration Layer.
 - For more information about attached data in Interaction Concentrator, see the chapter about processing attached data in the [Interaction Concentrator 8.1 User's Guide](#).
 - For information about configuring Interaction Concentrator to store user data from EventUserEvents that are distributed by T-Server or Interaction Server from other client applications (for example, from an agent desktop application), see the Storing Agent State and Login Session Data tab on this [Special Configuration Requirements](#) page.

[+] ICON Application

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

ICON Role Configuration Option

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

Attached Data Specification File

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

Attached Data Configuration Options

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

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

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

Custom Dispatcher Configuration Options

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

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

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

[+] Overview: Attached Data Specification File

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

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

Important

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

For sample attached data specifications, see:

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

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

This section provides information about the following topics:

[+] Parser Limitations

The ICON XML parser imposes the following limitations:

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

[+] Attribute Values

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

History Types

The following values can be used as history types:

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

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

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

Data Source Types

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

Attribute Name	TEvent Attribute Name
reasons	AttributeReasons
extensions	AttributeExtensions
userdata	AttributeUserData

[+] IDB Fields

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

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

Predefined IDB Columns—Voice

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

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

Predefined IDB Columns—Multimedia

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

Important

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

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

Predefined IDB Columns—Custom Fields

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

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

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

[+] Universal Routing Server Attached Data

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

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

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

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

Important

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

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

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

Virtual Queues

Storing Virtual Queue Data and Extended Route Results

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

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

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

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

[+] Universal Routing Server

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

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

- `report_reasons`
- `report_targets`

For more information about these URS configuration options, see the [Universal Routing Reference Manual](#).

[+] ICON Application

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

Connections

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

`vq-write-mode`

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

`extended-route-result`

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

on the Options tab of the ICON Application object.

Important

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

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

[+] Virtual Queue DN

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

monitor

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

[+] Switch

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

support-dn-type-5

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

Agent States and Logins

Storing Agent State and Login Session Data

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

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

Important

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

[+] ICON Application Configuration

ICON Role

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

Other Options

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

[+] Configure ICON to Use Custom States

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

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

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

Agent Desktop Application Configuration

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

[+] Start Recording a Custom State

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

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

Example

"AfterCallWork", "+"

[+] Send Custom State Data

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

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

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

Example

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

In this example, ICON will store the following values:

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

[+] Stop Recording a Custom State

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

Key = "<StateKeyName>", Value = "-"

Example

"AfterCallWork", "-"

[+] Use Multiple Custom States at Once

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

Example

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

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

Outbound Contact

Storing Outbound Contact Data

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

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

Outbound Contact Configuration

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

[+] Field Object

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

- `icon_attribute`
- `send_attribute`

Important

Interaction Concentrator reads Field object configuration information only at startup. No real-time configuration changes to Field objects are recognized. To accept changes to Field configuration, restart Interaction Concentrator.

`icon_attribute`

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

To configure this option:

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

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

`send_attribute`

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

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

Field	Key Name
<code>contact_info</code>	<code>GSW_PHONE</code>
<code>chain_id</code>	<code>GSW_CHAIN_ID</code>

Field	Key Name
attempt	GSW_ATTEMPTS
call_result	GSW_CALL_RESULT

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

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

[+] Campaign Group Object

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

[+] Outbound Contact Server

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

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

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

ICON Application

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

[+] Connections

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

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

[+] ICON Role Configuration Option

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

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

[+] OCS-Specific ICON Options

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

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

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

Multi-Tenant Considerations

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

[+] Multi-Tenant Example

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

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

License Reporting Manager

Storing License Reporting Data

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

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

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

[+] Configuring ICON to store LRM data

To configure ICON, perform the following steps:

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

[+] Configuration Notes

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

Important

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

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

Partitioned IDB

Deploying a Partitioned Oracle IDB

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

Important

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

[+] Overview

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

Warning

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

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

[+] Creating Your Database

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

- CoreSchemaPart_ora.sql
- CoreProcedures_ora.sql

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

- PurgePart_ora.sql

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

[+] About Partitioning

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

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

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

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

[+] Purging

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

Instructions for how to run the `purgePartitions811` procedure, how to schedule it, and all other operational considerations are documented in the “Purging by Truncating Partitions” section the “Using Special Stored Procedures” chapter in the [Interaction Concentrator 8.1 User’s Guide](#).

[+] Partitioned Tables

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

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

High Availability

Configuring for High Availability

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

In an HA deployment, observe the following rules:

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

For more information about configuring applications and connections in Configuration Manager, see the [Framework Deployment Guide](#).