Interaction Concentrator Deployment Guide

Interaction Concentrator Options

4/26/2025

Interaction Concentrator Options

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

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

Configuration Server recognizes the following sections for ICON Application objects:

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

callconcentrator Section

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

[+] Agent login session metrics

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

gls-acw-first

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

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

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

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

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

ICON Application Settings:

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

Switch Application Settings:

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

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

- Changes take effect: After restart

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

gls-stats-update

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

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

gls-stats-update-delta

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

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

gls-store-event-seq

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

- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- G_DND_HISTORY

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

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

[+] Agent login session

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

gls-active-reason-codes

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

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

gls-enforce-reason-code

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

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

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

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

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

[+] Attached data

adata-default-storage Option

adata-default-storage

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

This option applies to voice and multimedia interactions.

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

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

adata-extensions-history

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

This option applies to voice and multimedia interactions.

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

adata-reasons-history

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

This option applies to voice interactions only.

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

adata-spec-name

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

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

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

adata-userdata-history

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

This option applies to voice and multimedia interactions.

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

cseq-adjustment

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

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

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

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

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

suppress-user-data

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

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

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

[+] Configuration

cfg-annex Option

cfg-annex

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

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

[+] Custom dispatcher

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

gud-cust-disp

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

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

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

gud-cust-disp-groups

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

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

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

[+] Database writing

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

dbw-request-tout

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

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

Examples:

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

dbw-seq-step

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

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

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

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

Examples:

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

dbw-seq-tout

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

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

Examples:

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

[+] High Availability

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

use-dss-monitor

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

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

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

dss-no-data-tout

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

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

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

[+] HTTP Connection

http-protocol-enabled Option

http-protocol-enabled

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

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

[+] ICON Role

role Option

role

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

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

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

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

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

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

Examples of correct settings:

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

(The last two examples are equivalent.)

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

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

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

[+] IDB

db-schema-name Option

db-schema-name

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

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

[+] In-memory queue

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

acc-proc-tout

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

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

acc-queue-lifespan

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

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

acc-queue-size

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

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

[+] Multimedia

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

calls-in-the-past

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

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

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

om-force-adata

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

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

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

mcr-om-processing

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

- GX_SESSION_ENDPOINT
- G_AGENT_STATE_HISTORY
- GS_AGENT_STAT
- G_AGENT_STATE_RC
- G_CALL

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

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

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

[+] Operational memory

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

om-check-filter-flag

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

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

om-max-in-memory

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

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

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

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

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

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

om-memory-optimization

Specifies whether memory usage will be optimized.

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

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

[+] Outbound metrics

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

gos-write-duplicate-metrics

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

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

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

gos-write-metrics

Specifies whether ICON writes any precalculated OCS metrics to IDB.

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

gos-write-metrics-only

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

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

[+] Partitioning

partition-type Option

partition-type

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

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

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

[+] Persistent queue

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

agent-pstorage-name

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

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

pq-backlog-alarm-threshold

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

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

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

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

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

pq-backlog-clearance-threshold

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

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

pq-dbname

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

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

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

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

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

pq-purge-number

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

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

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

pq-startup-check

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

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

pq-startup-purge

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

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

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

[+] Scenario recognition

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

advanced-ext-party-reconstruction

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

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

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

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

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

dest-busy-processing

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

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

store-releasing-party

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

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

Important

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

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

gcti-mode-monitoring

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

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

[+] Synchronization

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

cfg-auto-resync

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

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

start-cfg-sync

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

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

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

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

sync-call-data-limit

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

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

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

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

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

tsync-threshold

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

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

[+] Virtual Queue

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

vq-write-mode

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

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

extended-route-result

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

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

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

route-res-vqid-hist-enabled

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

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

store-route-result-reliability

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

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

custom-states Section

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

AgentRecordUserTypes

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

- Configured in: ICON Application

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

Example:
AgentRecordUserTypes = 207,AfterCallWork,208,Break

AgentUserFields

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

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

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

Example:

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

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

EventData

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

- Configured in: ICON Application

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

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

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

GlobalData

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

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

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

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

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

Example:

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

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

store-event-data

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

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

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

dbw-error-reactions Section

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

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

uniqueness

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

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

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

custom-error-reactions

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

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

To configure an error reaction, perform the following steps:

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

   Where:

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

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

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

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

filter-data Section

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

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

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

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

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

acd-party-history

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

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

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

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

acd-party-metrics

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

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

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

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

call-history

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

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

call-metrics

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

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

Important

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

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

external-party

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

- G_PARTY
- G_PARTY_HISTORY
- G_PARTY_STAT

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

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

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

gls-all

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

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

ICON stores this information in the following IDB tables:

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

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

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

gls-ivr

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

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

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

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

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

gls-metrics

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

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

ICON stores agent states information in the following IDB tables:

- `GS_AGENT_STAT`
- `GS_AGENT_STAT_WM`

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

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

gls-no-person

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

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

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

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

gls-queue

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

- G_AGENT_STATE_HISTORY
- G_AGENT_STATE_RC
- GS_AGENT_STAT
- GS_AGENT_STAT_WM
- GX_SESSION_ENDPOINT

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

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

gls-wm

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

- `G_AGENT_STATE_HISTORY`
- `G_AGENT_STATE_RC`
- `GS_AGENT_STAT_WM`

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

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

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

ir-history

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

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

observer-party

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

- G_PARTY
- G_PARTY_HISTORY
- GS_PARTY_STAT

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

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

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

udata-history-terminated

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

- `G_USERDATA_HISTORY`
- `G_SECURE_USERDATA_HISTORY`

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

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

listeners Section

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

user_named_option Option

user_named_option

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

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

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

Example:

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

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

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

user_named_section Section

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

Example:

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

- `protocol=http`

port

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

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

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

protocol

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

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

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

transport

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

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

log Section

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

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

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

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

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

x-conn-debug-open

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

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

x-conn-debug-select

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

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

x-conn-debug-timers

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

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

x-conn-debug-write

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

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

x-print-attached-data

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

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

x-server-trace-level

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

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

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

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

x-server-config-trace-level

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

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

x-server-dbw-trace-level

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

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

x-server-gcti-trace-level

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

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

x-server-http-trace-level

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

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

x-server-smtp-trace-level

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

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

x-server-debug-level

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