



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 User's Guide

Processing Attached Data

12/17/2025

# Processing Attached Data

This page describes how Interaction Concentrator (ICON) processes user data that is attached to voice calls, eServices, and 3rd Party Media interactions. It contains the following sections:

- [Attached Data Specification File](#)
- [Attached Data Processing for Voice Calls](#)
- [Customized Attached Data Processing](#)
- [Attached Data Processing for Multimedia](#)

The processing and storage of attached data is resource intensive and expensive. Genesys recommends that you carefully consider your reporting and troubleshooting requirements in order to limit the amount of attached data that you configure Interaction Concentrator to capture.

For information about ICON configuration and other Configuration Layer settings that make attached data available in Interaction Database (IDB), see [Configuring for Attached Data](#) in the *Interaction Concentrator Deployment Guide*.

## Attached Data Specification File

The attached data specification is an XML file stored in the installation directory that you specify when you install the Interaction Concentrator application.

The attached data specification file (by default, named **ccon\_adata\_spec.xml**) maps the key-value pairs (KVPs) in reporting event attributes to IDB tables and fields.

For information about creating an attached data specification for ICON to use, about the XML schema definition, and for sample attached data specification files, see [Attached Data Specification File](#) in the *Interaction Concentrator Deployment Guide*.

### Important

In releases prior to 8.1.5, you must restart ICON after you make changes to the attached data specification file for the changes to take effect.

When ICON reads the attached data specification file, it creates a dictionary, which then becomes active. All records for newly-created interactions are associated with this active dictionary and all data processing for this interaction is implemented in accordance with it.

When you update the adata-spec-name option with the name of a new attached data specification file, ICON reads the new file and creates a new active dictionary. All new interactions are associated with this new active dictionary. Data processing of old interactions (those created before the new

dictionary became active) is continued based on old dictionary. The new dictionary is used only for new interactions.

After you edit the old attached data specification file, ensure that ICON reloads it by changing the file name and updating the value of the **adata-spec-name** option accordingly.

After reloading the attached data specification file, ICON compares the active (older) dictionary with newly-loaded one. If these two dictionaries are identical (that is, they contain the same key names and definitions) ICON continues to use the currently active dictionary and does not create a new one.

The result of the reloading procedure is then written in a message in the Interaction Concentrator log file.

### Example:

1. ICON currently uses an attached data specification file named **ccon\_adata\_spec.xml**.
2. After you modify this file, you save it as **ccon\_adata\_spec2.xml**.
3. To enforce the attached data specification file reload procedure, you change the value of the **adata-spec-name** option from **ccon\_adata\_spec.xml** to **ccon\_adata\_spec2.xml**.
4. As a result, ICON rereads the attached data specification from the file named **ccon\_adata\_spec2.xml**.
5. Noting differences between **ccon\_adata\_spec.xml** and **ccon\_adata\_spec2.xml**, ICON creates a new active dictionary that is used to process all new interactions. Interactions that ICON started processing based on the old dictionary continue to be processed using the old dictionary.
6. ICON writes a message in the Interaction Concentrator log recording the result of the attached data specification file reload.

## The Number of Dictionaries ICON Simultaneously Maintains in Active Memory

ICON keeps an old dictionary until all the interactions using it are completed. However, by default the maximum number of dictionaries ICON keeps in active memory is twelve. ICON cannot process additional dictionaries once it reaches this limit. Therefore, Genesys recommends that you do not change the attached data specification more than ten times during the usual lifetime of the longest-living interaction type. If you need additional dictionaries in memory, contact Genesys Customer Care for assistance.

## Limitations

- Consultation calls and call merging:

During a long call that involves a consultation, the main call and the consultation call might have different dictionaries. For example, the consultation call might be created after the attached data specification file was changed. As a result, some key names from the consultation call might not be transferred to the main call because of the differences in their dictionaries.

When the merge is completed, only one call is still alive (usually the main call). Data processing for that call continues according to the dictionary assigned for that call.

- Number of simultaneously loaded active dictionaries in ICON memory:

By default, the number of dictionaries ICON keeps in memory at once is twelve. If you need to increase the maximum number of active dictionaries loaded in memory, contact Genesys Customer Care for assistance.

- High availability:

In high availability (HA) environments, two ICON instances work with the same T-Server/SIP Server/Interaction Server. It is possible that the one of the two ICON instances might load an update to the attached data specification file slightly before the other.

Because there is no synchronization mechanism between the HA ICON pair, ICON-1 might finish processing the updated attached data specification file at one time (T-0) and ICON-2 later, at another time, T-1.

All interactions created on ICON-1 after T-0 use dictionary-2, but until T-1 ICON-2 still continues to use dictionary-1. This might produce some discrepancy in IDB tables related to user data. Only after T-1 do both ICON instances use dictionary-2 and unquestionably store the same data.

## Attached Data Processing for Voice Calls

This section briefly describes how Interaction Concentrator (ICON) processes user data that is attached to voice calls in events from T-Server.

### Important

For information about attached user data that is sent in User Events, see [Processing Attached Data](#).

## T-Server Interactions

When processing data from T-Server, ICON checks all TEvents for changes to attached data. When attached data changes for a particular interaction, ICON analyzes the change and stores the data in IDB, according to either its application configuration or the attached data specification.

### Important

A key-value pair is identified by its unique combination of the key name and the datatype specified for the value. When the datatype for the value changes, ICON creates new key-value pair and marks the key with the previous data type as deleted.

in the G\_USERDATA\_HISTORY table.

## Call-Specific Data

Call-specific data is attached data that is associated only with a call. This data is stored in IDB when the call is cleared after a specified timeout. Use the call-deletion-timeout Switch-level configuration option to configure the timeout interval.

## Historical Data

Historical data is associated with the attached data. Historical data can have the following associations:

- **Party**—When AttributeCtrlParty represents a party in EventCallDataChanged. The party with which the historical data is associated is the last party that updated the user data, even if that party had been terminated from the call before the call ended (see [Post-Routing User Data Processing](#)).
- **Endpoint**—When AttributeCtrlParty is specified in EventCallDataChanged.
- **Agent**—When an agent is associated with a specific device at the moment when the data is modified.

The stored change type reflects the change type for records with a history type of all. For more information about the history types, see [Configuring for Attached Data](#) in the *Interaction Concentrator Deployment Guide*.

The following table defines the values for types of attached data changes.

Type of Change	Numeric Equivalent	Description
created	1	Call was created with the specified key.
added	2	Key was added into the existing attached data.
updated	3	Existing key was updated.
deleted	4	Existing key was deleted.
terminated	5	Call was terminated with an existing key.

## Post-Routing User Data Processing

Starting with Interaction Concentrator release 8.0, ICON supports the scenario in which a party (strategy or agent) updates user data after the call has left the party. The following are examples of this scenario:

### Use Case Examples

- A call reaches a Routing Point, and a strategy is loaded on the Routing Point. The strategy routes the call to an agent. After the call leaves the Routing Point, the strategy updates the user data.

- Agent 1 is handling a call, initiates a consultation call to Agent 2, and then transfers the call to Agent 2. After the call has been transferred, Agent 1 attaches or updates user data.

Provided that the user data is updated within the call deletion timeout, ICON reports the PartyID of the last party on the device that updates user data, even if that party is no longer part of the call.

### Implementation

To support this functionality, ICON stores in memory a list of every device that participated in a call and that is a potential source of user data. For each device (EndpointID), ICON stores the PartyID of the last party that was created on the device. The history is updated every time a party is deleted from the call. The history is temporary; it is stored in memory until the call itself is deleted and the user data is written to IDB in the usual way (for example, in the G\_USERDATA\_HISTORY table). In the IDB record, the EndpointID and PartyID are the most recent device and party associated with a user data update.

### Limitations

Note the following limitations for this feature:

- Interaction Concentrator reporting of the party that attached or updated user data is reliable except in the scenario in which multiple parties were created on the same device in the same call, and the endpoint did not send the request(s) for attached data before the next party was created.

For example, a call reaches a Routing Point, and a strategy is loaded on the Routing Point. In a first pass, the strategy attaches user data and returns the call to the Routing Point; in a second pass, the strategy routes the call to an agent; then the Routing Point sends the user data update. ICON will associate the user data update with the second pass through the strategy (the last party on the device).

- The history of recent parties is stored in memory, and it will be lost in the event of a shutdown or restart of ICON before the UserData record has been written to IDB.

## Database Schema Extensions for Voice Attached Data

IDB contains two types of tables for UserData collection:

- So-called flat tables are used when data that corresponds to a particular key name is stored into a particular field in a table.
- Generic, or historical, tables are used when each attached data value is stored in a separate row with the corresponding context.

The following table lists IDB tables that are used to store user data that is attached to voice calls. The attached data storage tables store user data about each call in a separate record (one record per call).

### Attached Data Storage Tables for Voice

Table Name	Type of Data	Description
G_CALL_USERDATA	Call	Designed for predefined UserData that is collected during

Table Name	Type of Data	Description
		the entire duration of a call.
G_CALL_USERDATA_CUST	Call	Designed for custom UserData that is collected during the entire duration of a call.
G_CALL_USERDATA_CUST1	Call	Designed for custom UserData that is collected during the entire duration of a call.
G_CALL_USERDATA_CUST2	Call	Designed for custom UserData that is collected during the entire duration of a call.
G_USERDATA_HISTORY	Historical	Designed to store historical changes to UserData.
G_SECURE_USERDATA_HISTORY	Historical	Designed to store historical changes to UserData. Permissions for this table must be set at your particular site.

### Attached Data Security

By providing a number of separate tables in which to store attached data, Interaction Concentrator provides a mechanism to secure sensitive user data. The Database Administrator (DBA) can assign user privileges in order to restrict access to the secure user data history table. Similarly, the DBA can restrict access to one or more of the flat tables.

### Multi-Site and Distributed Environments

In a multi-site environment or a geographically distributed environment, when attached data is propagated between two T-Servers, each of these T-Servers stores the attached data in its own IDB. As a result, attached data is duplicated in IDBs across the sites.

## Customized Attached Data Processing

You can create a custom stored procedure, or custom dispatcher, to handle user data that is attached to voice calls and to store the attached data in custom tables in IDB. ICON calls the custom dispatcher when the call ends.

### Data Types

ICON can process two types of attached data values:

- String
- Integer

### Key Groups

The values of the key names, specified in the same group, are provided by the same call to the custom dispatcher stored procedure. Within the attached data configuration file, you can specify groups of keys, with each group containing a maximum of 17 string key-value pairs (KVPs) and 17 integer KVPs. You can configure the maximum number of key groups that ICON will process in the `gud-cust-disp-groups` configuration option.

For an example of the XML specification for customized attached data processing, see [Attached Data Specification File](#) in the *Interaction Concentrator Deployment Guide*.

### Custom Dispatchers

The IDB initialization scripts create two custom dispatcher stored procedures:

- `gudCustDISP1`
- `gudCustDISP2`

The `gud-cust-disp` configuration option in the ICON Application object specifies which custom dispatcher ICON calls.

While ICON is running, you can switch from one dispatcher to the other. This enables you to make changes to the custom dispatcher configuration without interrupting the processing of attached data.

#### Important

The default custom dispatchers do nothing. You must modify the scripts in order to create the custom dispatchers that store the attached data that you require. You must also create scripts that, in turn, create the required custom tables in IDB.

### Sample Scripts

In addition to the IDB initialization scripts, the Interaction Concentrator installation package contains a sample SQL script, **SampleProc\_db\_type.sql** to create a custom dispatcher stored procedure and a custom attached data storage table in your IDB schema. The sample script is partially reproduced in [Sample Script for Custom Attached Data](#) in the *Interaction Concentrator Deployment Guide*.

## Attached Data Processing for Multimedia

This section briefly describes how ICON processes user data that is attached to multimedia interactions (Genesys eServices and 3rd Party Media).



### Interaction Server Interactions

When processing data from Interaction Server, ICON checks all Multimedia Reporting Protocol events for changes to attached data. When attached data changes for a particular interaction, ICON analyzes the change and stores the data in IDB, according to its application configuration and the attached data specification.

#### Important

A key-value pair is identified by its unique combination of the key name and the datatype specified for the value. When the datatype for the value changes, ICON creates new key-value pair and marks the key with the previous data type as deleted in the G\_USERDATA\_HISTORY table.

For an example of the XML specification for multimedia attached data processing, see the [Multimedia Sample](#) in the *Interaction Concentrator Deployment Guide*.

### Multimedia Interaction-Specific Data

Multimedia interaction-specific data is attached data that is associated only with a multimedia interaction. ICON stores this data in predefined fields in separate multimedia attached data tables in IDB.

By default, Interaction Server does not automatically attach the keys that are required in order to report all the multimedia attached data that Interaction Concentrator can support. You might need to modify your routing strategies so that Interaction Server attaches data for the required keys (for example, Suggested Response Name).

### Database Schema Extensions for Multimedia Attached Data

IDB contains two tables for multimedia-specific user data collection:

- GM\_L\_USERDATA—Stores the values of attached data keys for suggested and auto responses and acknowledgements, customer IDs, and reasons for stopping processing. ICON writes information to this table when Interaction Server reports that the interaction has finished.
  - ICON captures the names of the responses and acknowledgements from the value that the applicable keys had when ICON first received the report about the interaction from Interaction Server.
- GM\_F\_USERDATA—Stores information about predefined logical keys in the attached data of multimedia interactions. Information includes:
  - Sender (“From” name and e-mail address)
  - Called back
  - Subject
  - Type and subtype
  - Origination source (webform or e-mail)

- Time the interaction was received

ICON writes information to this table when Interaction Server first sends reporting events about the interaction to ICON. Therefore, the values that are stored by ICON are the first values that are presented for the applicable keys.

## Tracking User Data Changes

The memory optimization feature enables ICON to process a large number of active multimedia interactions. In order to do so, ICON removes interactions from operational memory if they are not in the active stage of processing, and later reconstructs the interaction and attached user data for further processing.

With this feature enabled, ICON tracks changes to multimedia user data in the following ways:

- Reconstructs the user data attached to interactions that were removed from operational memory. As part of the reconstruction of an interaction, ICON also reconstructs the user data content from the EventTakenFromQueue reporting event that was received from Interaction Server. Any changes in user data are then processed.
- Processes user data for interactions in the Interaction Queue. While an interaction is in the Interaction Queue, Interaction Server sends EventPropertiesChanged reporting events to ICON for storage in IDB. Because ICON does not know whether the user data key is new or changed, ICON stores this information in IDB with an attribute indicating that it was changed.