



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

Extracting Data in an HA Deployment

12/21/2025

Contents

- 1 Extracting Data in an HA Deployment
 - 1.1 Extracting HA Data
 - 1.2 Interrupted Data Flow with Calls Scenario
 - 1.3 Extracting Multimedia Data
 - 1.4 Extracting Virtual Queue-Specific Data
 - 1.5 Extracting Configuration Data
 - 1.6 Extracting Agent-Specific Data

Extracting Data in an HA Deployment

Through the use of data source session control tables, Interaction Concentrator provides information about the availability and reliability of data in Interaction Database (IDB). This page describes how downstream reporting applications can use that information to optimize their extraction, loading, and transformation (ETL) processes to extract the most reliable data. This chapter also describes additional considerations for specific types of high availability (HA) data.

This page is intended for users who develop their own ETL engine or who use Genesys Info Mart 8.0 or later to create reports that are based on data extracted from IDB.

This page contains the following sections:

- [Extracting HA Data](#)
- [Extracting Multimedia Data](#)
- [Extracting Virtual Queue-Specific Data](#)
- [Extracting Configuration Data](#)
- [Extracting Agent-Specific Data](#)

Extracting HA Data

Downstream reporting applications can leverage information about data availability in order to optimize ETL processes for extracting data from an HA pair of IDBs. Before it extracts data from a particular IDB for a particular time period, the downstream reporting application can use information about gaps in the data flow to determine the reliability of data in that IDB. Adjusted ETL processes can then avoid reading the unreliable data from one IDB, and switch over to extracting data from the other IDB for that particular time period.

For information about identifying gaps in the data flow from a particular Interaction Concentrator (ICON) instance to a particular IDB, see [Determining Data Availability and Reliability](#).

For information about the data source session control tables that support this functionality, [Data Source Session Control Tables](#).

Interrupted Data Flow with Calls Scenario

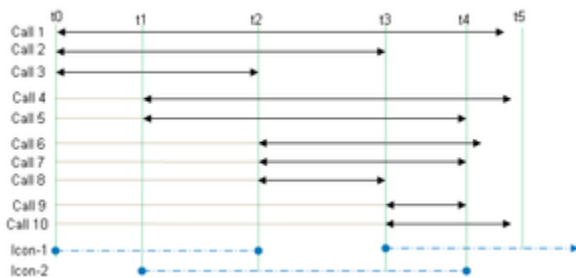
This section uses a sample scenario to illustrate two methods of optimizing data extraction:

- [Extracting HA Data: Approach 1](#)
- [Extracting HA Data: Approach 2](#)

Consider the following scenario:

- One data source (T-Server) and one provider (gcc provider).
- Connection and call events occur at the following times:
 - t0—ICON-1 starts. Calls 1, 2, and 3 start.
 - t1—ICON-2 starts. Calls 4 and 5 start.
 - t2—ICON-1 disconnects/terminates unexpectedly. Calls 6, 7, and 8 start.
 - t3—ICON-1 reconnects/restarts. Calls 9 and 10 start.
 - t4—ICON-2 disconnects/terminates unexpectedly.
 - t5—No new events. Calls continue.
- All timestamps use T-Server time.
- Reporting data is required for the interval t0–t5.

The figure below, Calls in Relation to Interrupted Data Flow, illustrates the relationship between the calls and the interrupted data flow in this scenario.



Calls in Relation to Interrupted Data Flow

The following tables show the values of selected fields in the records that the ICON instances create in the G_DSS_GCC_PROVIDER table in IDB-1 and IDB-2.

- For an explanation of the column headers, see the [legend](#) in the last row of the table.
- For more information about the G_DSS_GCC_PROVIDER table fields, see the Interaction Concentrator Physical Data Model for your RDBMS.

Record	DSS_ID	ICON_STIME	DSCONN_STIME	ESCONN_ETIME	EVENT_DSTIME	EVENT_DSTIME
ICON-1: Data Flow Interruption—G_DSS_GCC_PROVIDER Table Field Values in IDB-1						
1	37	t0	t0	t2	t0	t2
2	38	t3	t3		t3	t5
ICON-2: Data Flow Interruption—G_DSS_GCC_PROVIDER Table Field Values in IDB-2						
1	3767	t1	t1	t4	t1	t4
Legend (G_DSS_GCC_PROVIDER column names):						

Record	DSS_ID	ICON_STIME	DSCONN_STIME	DSCONN_ETIME	FEVENT_DSTIME	LEVENT_DSTIME
<ul style="list-style-type: none">• DSS_ID = Data source session ID• ICON_STIME = ICON startup time• DSCONN_STIME = Start of the connection to the data source server• DSCONN_ETIME = End of the connection to the data source server• FEVENT_DSTIME = Timestamp of the first event stored on the connection (T-Server time)• LEVENT_DSTIME = Timestamp of the last event stored on the connection (T-Server time)						

Analysis

The tables above show that:

- ICON-1 has a failure timeframe from t2 to t3.
- ICON-2 was collecting data only from t1 to t4.

Alternatively, from the timestamps of the first and last saved events, the table also shows that:

- There was an uninterrupted data source session on ICON-1 from t0 to t2, and another uninterrupted data source session on ICON-1 from t3 to t5.
- There was an uninterrupted data source session on ICON-2 from t1 to t4.

Conclusions

Based on the connection information, the maximum timeframes of reliable data from each ICON are:

- [t0-t2]: ICON-1
- [t2-t4]: ICON-2
- [t4-t5]: ICON-1

Alternatively, the downstream reporting application can determine the maximum timeframes of reliable data from each ICON based on the durations of the data source sessions, and switch over from one IDB to the other at the break points.

HA Merge Results

Important

The merge procedure is not supported on PostgreSQL RDBMSs.

The two following tables compare the reliability of data from ICON-1, from ICON-2, and from both

IDBs using a simple HA merge mechanism. The first table compares data for the case where ICON-1 terminates at t2. The second table compares data for the case where ICON-1 terminates at t2 and then restarts.

Comparison of Data Reliability—ICON-1 Terminates

Call	ICON-1 (terminates at t2)		ICON-2		HA (ICON-1 + ICON-2)	
	Data	Reliable	Data	Reliable	Data	Reliable
1	No tail	No	No data	No	No tail	No
2	No tail	No	No data	No	No tail	No
3	All data	Yes	No data	No	All data	Yes
4	No tail	No	No tail	No	No tail	No
5	No tail	No	All data	Yes	All data	Yes
6	No data	No	No tail	No	No tail	No
7	No data	No	All data	Yes	All data	Yes
8	No data	No	All data	Yes	All data	Yes
9	All data	Yes	All data	Yes	All data	Yes
10	All data	Yes	No tail	No	All data	Yes

Legend

- **No tail**—ICON has not stored any data related to the end of the call.
- **No data**—There is no data about the call in IDB at all.
- **All data**—ICON stored all data about the call.

Comparison of Results—ICON-1 Terminates then Restarts

Call	ICON-1 (terminates at t2, restarts at t3)		ICON-2		HA (ICON-1 + ICON-2)	
	Data	Reliable	Data	Reliable	Data	Reliable
1	No interim	No	No data	No	No interim	No
2	No tail	No	No data	No	No tail	No
3	All data	Yes	No data	No	All data	Yes
4	No interim	No	No tail	No	No interim	No
5	No interim	No	All data	Yes	All data	Yes
6	No data	No	No tail	No	No tail	No
7	No data	No	All data	Yes	All data	Yes
8	No data	No	All data	Yes	All data	Yes
9	All data	Yes	All data	Yes	All data	Yes
10	All data	Yes	No tail	No	All data	Yes

Call	ICON-1 (terminates at t2, restarts at t3)	ICON-2	HA (ICON-1 + ICON-2)
Legend <ul style="list-style-type: none"> • No interim—ICON missed part of the call. • No tail—ICON has not stored any data related to the end of the call. • No data—There is no data about the call in IDB at all. • All data—ICON stored all data about the call. 			

Extracting HA Data Approach 1

This section describes one approach that enables the downstream reporting application to optimize the extraction and merging of data from the HA pair of IDBs. For the scenario described above ([Interrupted Data Flow with Calls Scenario](#)), the ETL:

1. Determines the timestamps of data flow interruption and the timeframes of reliable data. For the analysis applicable to this scenario example, see [Analysis](#).
2. Extracts all data from IDB-1 related to calls that were terminated during the period [t0–t2].
3. Switches over to IDB-2, and extracts all data related to calls that were terminated during the period [t2–t4].
4. Switches back to IDB-1, and extracts all data related to calls that were terminated during the period [t4–t5].

Extracting HA Data Approach 2

The following is an alternative approach that enables the downstream reporting application to optimize the extraction and merging of data from the HA pair of IDBs. For the scenario described above (see [Interrupted Data Flow with Calls Scenario](#)), the ETL:

1. Determines the timestamps of data flow interruption and the timeframes of reliable data. For the analysis applicable to this scenario example, see [Analysis](#).
2. Extracts all data from IDB-1 until the timestamp of the ICON-1 break (t2).
3. In a staging area, removes all data related to non-terminated calls.
4. Switches over to IDB-2, and extracts all data for calls in IDB-2, starting with the call with the earliest timestamp that is greater than the timestamp of the last terminated call from IDB-1.

Important

This approach does not support late-arriving EventUserEvent data. EventUserEvent data that arrives after a call was terminated might be ignored, because it has a timestamp that falls between the last terminated call in IDB-1 and the next call in IDB-2.

Extracting Multimedia Data

Except for multimedia-specific attached data, data related to multimedia activity is stored in the same IDB tables as voice data. Interaction Concentrator supports HA of interaction-related data for active and completed multimedia interactions.

Extraction Procedure

To extract and merge multimedia interactions from an HA pair of IDBs with minimum overlap, follow the approach for voice calls described in [Extracting HA Data Approach 1](#):

1. Determine the timestamp of the data flow interruption (the process is described above in [Extracting HA Data](#)).
2. Extract all data from the first IDB until the break. This means that the ETL will extract all data for which the timestamp of last modification from the first ICON is earlier than the timestamp of the break.
3. Switch over to the second IDB, and extract all data for which the timestamp of last modification from the second ICON is earlier than the timestamp of the break in the second ICON's data flow.
4. Switch back to the first IDB, and so on.

For historical tables (for example, G_IR_HISTORY), in which ICON only inserts records, you can use the last modification timestamp as the marker for switching from the historical table in one IDB to the same table in the other IDB, and you can use a simple merge mechanism to reconstruct the multimedia data in the table.

However, for tables in which ICON can update as well as create records (for example, G_IR), you cannot simply add records from another IDB in a simple merge. The downstream reporting application must analyze the data to determine how to combine related records. This analysis will likely be deployment-specific.

Limitation

If some IDB data is calculated by adding a new value to a previously existing value, and if one of the records contains data that does not cover the interaction from the beginning, the data received after a merge will not be reliable. For example, 3rd Party Media statistics are processed in this cumulative manner; for a multimedia interaction scenario similar to the call scenario in [Interrupted Data Flow with Calls Scenario](#), if the ETL switches over to IDB-2 after the ICON-1 break at time t2, accumulated statistics for Call 1 and Call 2 will have incorrect values.

Extracting Virtual Queue-Specific Data

The general approach for extracting virtual queue data is the same as for other real-time interaction data. However, because virtual queue records have the same unique ID in both IDB instances (the VQID field in the G_VIRTUAL_QUEUE table), you can use the value of the CAUSE field in the

G_VIRTUAL_QUEUE table for additional data validation. For example, a virtual queue record with a value of stuck (integer value = 3) in the CAUSE field can be ignored in favor of a virtual queue record with a normal or abandoned status.

Extracting Configuration Data

The general approach for extracting configuration data is the same as for interaction data. However, because ICON uses database identifiers that are assigned by Configuration Server (DBIDs) to identify configuration objects stored in IDB, you can use IDB data for additional data validation. ICON flags the reliability of configuration data in the GSYS_EXT_INT1 field of the GC_* and GCX_* tables. When the reliability value of the same configuration record differs between two HA IDBs, extract the data from the more reliable record. For more information, see [Reliability Flag](#).

Extracting Agent-Specific Data

ICON supports high availability of agent-specific data (login sessions, agent states and reasons, and after-call work). The primary T-Server or Interaction Server creates AgentSessionIDs from the initial agent login and timestamp. It does not propagate the AgentSessionID to the backup T-Server or Interaction Server.

Instead, when the backup T-Server or Interaction Server becomes primary (after a switchover), new AgentSessionIDs are assigned to all known agent sessions. To support HA of agent data, a particular ICON instance processes only AgentSessionIDs that are received from the primary T-Server or Interaction Server.

In addition, ICON uses the mechanism of a persistent cache to verify agent login session data and to prevent storage of duplicate login sessions. For more information, see [Populating Agent Login Session Data](#).