Interaction Concentrator User's Guide

Processing User Events and Custom-Defined States

12/19/2025

Processing User Events and Custom-Defined States

Interaction Concentrator support for customer-defined states is designed to provide compatibility with Call Concentrator functionality in legacy deployments. This feature enables customers to use existing investments in customized desktop applications.

This chapter describes how Interaction Concentrator (ICON) processes user data from User Events. It contains the following sections:

- Custom States in Interaction Concentrator
- Storing Data from EventUserEvent
- Processing Data from EventCustomReporting

For information about ICON configuration and agent desktop application settings that make data about custom states and common data available in Interaction Database (IDB), see Configuring for Agent State and Login Data.

Custom States in Interaction Concentrator

The term custom states refers to customer-defined states of endpoints. If it has been configured to do so, Interaction Concentrator (ICON) supports the processing of data from the T-Server EventUserEvent, and Interaction Server EventCustomReporting, to store associations between:

- Common data and the voice call/multimedia interaction (or the call/interaction party, when applicable).
- Custom states and the voice call (or the call party, when applicable). Note that Interaction Concentrator only tracks custom states for voice calls, not for multimedia interactions.

Both an active call/interaction and the last call/interaction on the device can participate in the association (the last call/interaction being the one that ended immediately before the start of the custom state or the arrival of EventUserEvent/EventCustomReporting).

By default, ICON stores CallID and PartyID information for the sixteen most recent calls/interactions and parties associated with a device to be able to associate them with information from EventUserEvent and EventCustomReporting events. You can change the number of calls/interactions and parties stored by setting a different value in the max-party-info configuration option.

> **Important**
>
> If you remove a device from the configuration layer, all calls/parties associated with

> the device are lost. As a result, ICON cannot restore associations if an associated device was removed from the configuration before the arrival of EventUserEvent/ EventCustomReporting.

Storing Data from EventUserEvent

ICON processes data from EventUserEvent separately from call user data.

Common Data

If it has been configured to support custom states, ICON stores common data from EventUserEvent in two tables that are created by the IDB initialization script: G_CUSTOM_DATA_S and G_CUSTOM_DATA_P. You can configure unique keys to store values of customer-defined keys. Duplicate key names in the attached data are not supported.

Custom States

ICON stores data related to custom states from EventUserEvent in the G_CUSTOM_STATES table, which is created by the IDB initialization script.

ICON writes information to IDB when the custom state is finished. To avoid problems arising from stuck custom states, ICON clears all active custom states when the agent’s login session is terminated.

For more information about the custom data and custom state tables, see the *Interaction Concentrator 8.1 Physical Data Model* document for your particular RDBMS.

Processing Data from EventCustomReporting

Interaction Concentrator stores EventCustomReporting data from any multimedia application that uses the Interaction Server protocol. At this time, the main use case for this functionality is to receive focus time data, via Interaction Server, from Genesys Workspace Desktop Edition v8.5.111 and higher, and provide this data for use by downstream Reporting applications.

Using EventCustomReporting from Interaction Server for Focus Time Reporting

ICON stores data from EventCustomReporting to enable reporting on how much time a particular interaction was in focus (that is, actively being processed) on the agent desktop. ICON tries to identify the interaction from the data included in the event. If ICON identifies the interaction, it writes the user data key-value pairs (KVPs) listed in **attr_event_content** into the G_CUSTOM_DATA_S table.

To enable ICON to process EventCustomReporting events:

1. Include the gud role in the ICON configuration.
2. Set the value of the store-event-data option to `all` or `conf`. If you set the value to `conf`, ICON stores the values of the keys that are configured in the **EventData** option.
3. To configure the EventData option, configure **R_TimeInFocus** (char), **R_AgentDBID** (int), **R_PlaceDBID** (int), and **R_InteractionId** (char) as special keys. These specify that you want to store focus time data and that identify the associated Agent, Place, or Interaction respectively.

> Important
>
> If you want to *exclude* new records from the G_CUSTOM_DATA_S table and ignore focus time information, you can filter out the relevant keys using the EventData option.

Mandatory EventCustomReporting Attributes

ICON processes EventCustomReporting only if it includes the following attributes:

- **attr_event_time**
- **attr_itx_id**; alternatively, ICON can use the optional **R_InteractionId** key.
- **attr_event_content** (a non-empty list), with the mandatory **R_PlaceDBID** key.
- **ReportingEventSequenceNumber** in **attr_extension**; ICON might try to process the data without this attribute, but the recorded data will lack the sequence number.

ICON ignores EventCustomReporting events with missing mandatory attributes.

EventCustomReporting Processing Logic

1. Keys and values are taken from the **attr_event_content** attribute of the EventCustomReporting event.
2. ICON uses the value of the **attr_itx_id** attribute to identify the interaction and checks whether the interaction is still alive.
3. ICON attempts to identify the endpoint Place using **R_PlaceDBID**. ICON does not process EventCustomReporting if **R_PlaceDBID** is missing.
4. ICON attempts to identify the agent using **R_AgentDBID**. If this attribute is not specified, ICON searches for an agent last logged into the Place found in the previous step. If the agent cannot be identified, ICON writes a `0` (zero) value for the agent identification attributes.
5. ICON identifies the Party based on the Place ID. Since EventCustomReporting may arrive after a Party left the Place, ICON keeps a history of Parties for a specific Place. For the specified Place, ICON searches for the interaction mentioned in **attr_itx_id** and uses the Party assigned to this interaction in this Place.
6. ICON writes data into the G_CUSTOM_DATA_S table, depending on the values you set for the store-event-data option and (optionally) the EventData option.

The following are differences in how ICON processes EventCustomReporting data compared with EventUserEvent data:

- The **EndPointDN** field in the G_CUSTOM_DATA_S table stores the Place name rather than DN name.
- The **SWITCH** field in the G_CUSTOM_DATA_S table stores the DBID of the Interaction Server Application object.
- ICON treats **R_TimeInFocus**, **R_AgentDBID**, **R_PlaceDBID**, and **R_InteractionId** as special keys and uses them only to identify the associated focus time value, Agent, Place, or Interaction. Unless these keys are configured explicitly using the **EventData** configuration option, ICON does not write these keys and their values into the G_CUSTOM_DATA_S table.
- ICON does not track custom states via EventCustomReporting.