



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

Purge Procedures

5/12/2025

Purge Procedures

The size of IDB is one of the most significant factors that affect Interaction Concentrator performance. To maintain a manageable IDB, you must implement a suitable purging strategy.

The Interaction Concentrator purge stored procedures provide the following functionality:

- Safely purge voice, multimedia, and outbound interaction data from IDB. Also safely purge agent login sessions and attached data. All logically related information is deleted at the same time.
- Do not affect ICON performance while operating.
- Accept input parameters to limit the range of deleted data.

You can execute only one instance of a purge procedure at any one time. The purge procedures can run in the background while ICON is writing data to the IDB.

Interaction Concentrator 8.1 provides three sets of purge procedures to purge data safely from IDB:

- **gsysPurge81**—Interaction Concentrator v8.1.x and higher provides the gsysPurge81 stored procedure. (Interaction Concentrator release 8.0.000.32 and earlier includes the prior version of the purge procedure, gsysPurge80, which does not purge Outbound Contact data.)
 - Purges data from IDB voice, multimedia, attached data, agent login sessions, and outbound data tables.
 - Use the gsysPurge81 purge procedure if one of the following applies:
 - You are a new ICON 8.1 customer.
 - You need to purge large amounts of data.
 - You are concerned about performance.
- **purgePartitions811**—If you are running Oracle 11g or higher and need to purge large amounts of data efficiently, you can create a partitioned IDB that can be purged by truncating partitions.
- **gsysPurgeIR, gsysPurgeUDH, gsysPurgeLS, and gsysPurgeOS**

Important

These separate purge procedures were discontinued in release 8.1.503.03. If you are using Interaction Concentrator 8.1.503.03 or higher, use gsysPurge81 or (in Oracle environments) purgePartitions811.

The separate stored procedures purge the following data from IDB:

- **gsysPurgeIR**—Purges interaction records (IRs).
- **gsysPurgeUDH**—Purges user data history (UDH) data.
- **gsysPurgeLS**—Purges Agent login and session (ALS) data.
- **gsysPurgeOS**—Purges Outbound Contact data.

Tables Purged by the Purge Stored Procedures

Not all IDB tables are purged. For efficiency, purging is reserved for tables liable to accumulate the greatest amount of data, while avoiding those that tend to have long-lived data with low accumulation rates. The following list of tables applies to all of the purge stored procedures:

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

Retention Period

All the purge procedures accept an input parameter, *number_of_days* or *count_days*, which specifies the retention period—the number of days, including the current date, for which data will be left in the database after the purge. Data for days preceding the retention period is eligible for purging. The input parameter must be a positive integer (minimum value = 1).

delete_all_flag Parameter

The gsysPurge81 purge procedure accepts a second input parameter, *delete_all_flag*, which further defines which data to purge in IDB.

Scheduling Considerations

Consider the extraction, transformation, and loading (ETL) cycle of your downstream reporting application, to ensure that you do not delete data before your downstream reporting application has successfully extracted the data.

In general, schedule the purge procedures to run after your downstream reporting application has completed its regular ETL cycle. Keep an appropriately large sliding window of data, and delete only the oldest ETL cycle's worth of data each time you run the purge procedures.

Your downstream reporting application may provide functionality that spreads the extraction of a backlog of data across multiple ETL cycles (for example, the **limit-extract-data** configuration option in Genesys Info Mart 7.x). If you are exercising this functionality, be aware that, at the end of a particular ETL cycle, your downstream reporting application may not have extracted all the data that was available at the time of extraction.

Important

With Genesys Info Mart as your downstream reporting application, it is possible to run the purge procedures in parallel with the Genesys Info Mart Job_AggregateGIM and Job_MaintainGIM jobs.

If your downstream reporting application does not successfully complete an ETL job cycle, ensure that you stop invoking the purge procedures until the problem is fixed and the ETL job cycle starts running successfully again. Otherwise, you may delete IDB source data before the downstream reporting application has extracted it.

If ICON experiences an issue that causes it to stop writing data to IDB, ensure that you suspend your purge procedure until ICON is running correctly. If you continue to run the purge procedure while there is a backlog of data ready to be written to IDB, depending on your settings, data may be purged before the downstream reporting application has a chance to run its ETL cycle and capture that data.

Using the gsysPurge81 Stored Procedure

The gsysPurge81 procedure purges voice, multimedia, outbound, attached data, and agent login sessions from IDB. Genesys recommends that you use this procedure as your primary purge procedure if you are purging large amounts of data or if you are concerned about ICON performance. The gsysPurge81 procedure uses two user-specified input parameters, *number_of_days* and *delete_all_flag*, to purge IDB data.

Important

The purge procedure has been renamed to GSYS_Purge81_Common; however, the wrapper name, GSYS_Purge81, remains the same as in previous releases, so you do not need to change scripts as a result of this update.

number_of_days Input Parameter

Use the *number_of_days* parameter to calculate the retention period for data. Enter the number of days, including the current one, for which data should be retained. For example, if you run this stored procedure with *number_of_days* = 1, and *delete_all_flag* = 1 (delete all records), then *all* voice, multimedia, open media, attached data, and agent login session interaction data older than one day will be purged from IDB. The valid values for this parameter are positive integers greater than, or equal to, 1.

delete_all_flag Input Parameter

Use the *delete_all_flag* input parameter to define which data to purge in IDB. The only valid values for

this parameter are 0 and 1, where:

- 0—ICON deletes only terminated records older than *number_of_days* day(s). Non-terminated records—records that ICON may update in the future—are retained in IDB.
- 1—ICON deletes all records (voice and multimedia) older than *number_of_days* day(s).

Important

The purge procedure deletes from the GO_CAMPAIGN and GO_CHAIN tables all records older than the number of days specified in the *number_of_days* parameter and does not take account of the value of the *delete_all_flag* parameter.

gsysPurge81 Purge Optimization

The gsysPurge81 stored procedure differs from the gsysPurgeIR procedure in that it does not consider merged interactions when purging data. Instead, gsysPurge81 collects information about existing records in IDB, and then uses this data to optimize its performance in subsequent purges. This data may take considerable time to acquire the first time the purge procedure is executed.

This purge procedure also provides the option to purge only terminated records using the *delete_all_flag* parameter. This will affect the time it takes to perform the purge operation, because the purge procedure first conditionally deletes terminated records in IDB and then rechecks all records not deleted on the first pass in case previously non-terminated records are now terminated.

Purge Lock Mechanisms

The lock mechanisms available depend on the release of Interaction Concentrator you are running.

Functionality in Release 8.1.503 and Higher

There are two purge locking options:

- **Use of the the PSTATE record** (supported only on Oracle and DB2).
- Native RDBMS lock mechanisms (supported only on Oracle, PostgreSQL, and Microsoft SQL).

Native Lock Mechanisms

RDBMS native lock mechanisms are also used to prevent more than one instance of the purge stored procedure from running simultaneously within a single data schema. Using the native lock mechanisms avoids the need to manually unlock the purge procedure if the process is halted by an outside command. Use the appropriate one of the following RDBMS-specific lock mechanisms:

- Oracle: DBMS_LOCK package (DBMS_LOCK.REQUEST / DBMS_LOCK.RELEASE)
- PostgreSQL: Advisory Locks (PG_TRY_ADVISORY_LOCK / PG_ADVISORY_UNLOCK)
- Microsoft SQL: Application Locks (SP_GETAPPLLOCK / SP_RELEASEAPPLLOCK)

Important

These native locks are automatically released when a session terminates.

Running the Purge Lock Mechanisms

On Microsoft SQL, PostgreSQL, and DB2 RDBMSs, you have only one option for purge locking. The available mechanism runs automatically when you start the purge procedure.

Special Considerations for Oracle

If you are using an Oracle RDBMS, you can choose to run whichever lock mechanism you prefer.

- By default, Oracle uses the PSTATE lock mechanism. If you choose to use this form of locking, you do not need to perform any extra steps when calling the purge procedure.
- If you choose to use the native lock mechanism, perform the following steps:

1. Make sure that you have the Oracle DBMS_LOCK package installed.

Important

Do not initialize IDB until after you install the Oracle DBMS_LOCK package.

2. Make sure that you have granted the EXECUTE privilege to the users who will be executing the purge stored procedure. For example, execute the following command:
 - GRANT EXECUTE ON DBMS_LOCK TO USER1;
3. Modify GSYPurge81 by changing the GSYPURGE81PState() call to GSYPURGE81NativeLock(). The example below shows how the command should appear:

```
CREATE OR REPLACE PROCEDURE GSYPURGE81 (
    NUM_OF_DAYS$      INTEGER,
    DELETE_ALL_FLAG$  INTEGER)
AS
BEGIN
    GSYPURGE81NativeLock (
        NUM_OF_DAYS$,
        DELETE_ALL_FLAG$);
END;
```

Functionality in Releases Prior to 8.1.503

Releases prior to 8.1.503.xx use only the PSTATE lock mechanism.

The PSTATE Lock Mechanism

When the purge procedure starts, it tries to insert a special record, PSTATE, in the G_PURGE_STATE

table.

- If the insert is successful, no other purge instances are running, and the purge procedure continues. The purge procedure has switched to *lock mode*.
- If the insert fails, the purge procedure halts because another purge process is already active.

After the purge procedure has successfully completed, the special PSTATE record is deleted from the G_PURGE_STATE table. Purging is now set to *unlocked mode*.

If the purge procedure stops unexpectedly, or if purging is halted by an external command, you must manually delete the PSTATE record from the G_PURGE_STATE table before you can run another purge instance.

gsysPurge81 Logging and Error Handling

The gsysPurge81 purge procedure provides logging information about the state and the results of the purge procedure, including the number of records that are flagged for deletion in a partition and the actual number of purged records in a partition.

The gsysPurge81 purge procedure stores information about the purge process in two IDB tables:

- G_LOG_MESSAGES— Stores log messages in the MESSAGETEXT and APPNAME fields.
- G_LOG_ATTRS—Stores parameters of log messages as attribute pairs in the ATTR_NAME and ATTR_VALUE fields.

G_LOG_MESSAGES

The MESSAGETEXT field in G_LOG_MESSAGES contains information about the action performed by the gsysPurge81 procedure, including:

- The start of the purge procedure (MESSAGE_ID=25922)
- The end of the purge procedure (MESSAGE_ID=25927)
- The start of the next purge procedure action (MESSAGE_ID=25920)
- The end of the purge procedure action (MESSAGE_ID=25930)
- Other purge instance in progress (MESSAGE_ID = 25925)
- Input parameters are incorrect (MESSAGE_ID=25925)

The APPNAME field in G_LOG_MESSAGES contains information about the name of the stored procedure that created the record in G_LOG_MESSAGES. It has the format: ICON DB:gsysPurge81

G_LOG_ATTRS

The G_LOG_ATTRS table stores parameters of log messages, generated by the gsysPurge81 procedure, as attribute pairs in the ATTR_NAME and ATTR_VALUE fields of the G_LOG_ATTRS table. The primary key (ID) of the G_LOG_MESSAGES table is used to identify related records in the G_LOG_ATTRS table, where LRID is a foreign key. For example, a record in G_LOG_MESSAGES where ID = *id_value* relates to the corresponding records in G_LOG_ATTR that have the same LRID value: LRID = *id_value*.

The table below shows all possible logging messages (MESSAGES_TEXT) and attributes (ATTR_NAME,ATTR_VALUE) that the gsysPurge81 purge procedure (APPNAME) generates and stores in the G_LOG_MESSAGES and G_LOG_ATTRS tables.

Important

You can read information from the G_LOG_MESSAGES and G_LOG_ATTRS tables in their entirety by using Genesys Solution Control Interface (SCI).

Messages and Attributes Generated by gsysPurge81

G_LOG_MESSAGES Table	G_LOG_ATTRS Table	
MESSAGETEXT	ATTR_NAME	ATTR_VALUE
Purge procedure <i>purge_procedure_version</i> started	MAX_PARTITION_TO_PURGE	<i>yyyymmdd</i>
	NUMBER_OF_DAYS_LEFT	<i>positive_integer</i>
	DELETE_ALL	<i>0 or 1</i>
Purge initialization completed	PartitionID	<i>0</i>
	Record_Count	<i>0</i>
GSYS_MARK_PARTITION mark partition started...	PARTITION_TO_MARK From	<i>yyyymmdd</i>
	PARTITION_TO_MARK To	<i>yyyymmdd</i>
purging in progress...	MAX_PARTITION_TO_PURGE	<i>yyyymmdd</i>
	PARTITION_COUNT	<i>number_of_terminated_and_non-terminated_records_in_partition</i>
	DELETE_ALL	<i>0 or 1</i>
GSYS_PURGE_PARTITION purge started...	PartitionID	<i>yyyymmdd</i>
	Record_Count	<i>0</i>
IDB:Purge-table: <i>table_name</i> initiated...	PartitionID	<i>yyyymmdd</i>
	Record_Count	<i>number_of_terminated_and_non-terminated_records_in_table_that_belongs_to_partition</i>
IDB:Purge-table: <i>table_name</i> completed...	PartitionID	<i>yyyymmdd</i>
	Record_Count	<i>actual_number_of_deleted_records</i>
GSYS_PURGE_PARTITION purge completed	PartitionID	<i>yyyymmdd</i>
GSYS_Purge81 purge completed	MAX_PARTITION_TO_PURGE	<i>yyyymmdd</i>
	NUMBER_OF_DAYS_LEFT	<i>positive_non-zero_integer</i>
	DELETE_ALL	<i>0 or 1</i>

Consider the following examples:

Example 1

The G_LOG_ATTRS table contains two attribute pairs, (PartitionID/20080627) and (Record_Count/

2673), corresponding to the log message, GSYS_PURGE_PARTITION purge started... in the G_LOG_MESSAGES table.

In this example:

- The first attribute pair, (PartitionID/20080627), provides the ID of the partition to be purged.
- The second set of attributes, (Record_Count/2673), returns the total number of records found in the partition. In this case, 2673 records were found in partition 20080627. The actual number of records purged depends on the value of *delete_all_flag*.

Example 2

The following is a sample SQL statement which you might use to extract information from G_LOG_MESSAGES and G_LOG_ATTRS tables:

```
SELECT  
G_LOG_MESSAGES.id,G_LOG_MESSAGES.TIMEWRITTEN,G_LOG_MESSAGES.MESSAGETEXT,G_LOG_ATTRS.attr_name,G  
FROM G_LOG_MESSAGES,G_LOG_ATTRS  
WHERE G_LOG_MESSAGES.ID=G_LOG_ATTRS.lrid and G_LOG_MESSAGES.id>0 order by  
G_LOG_MESSAGES.id;
```

Scheduling gsysPurge81

It is the responsibility of the customer (usually the Database Administrator) to run the gsysPurge81 procedure as required. Genesys recommends that you run the purge procedure at a time when contact center activity is low.

If you run the purge procedure occasionally, on an ad hoc basis, execution can take significantly longer than if you run it regularly. Also, in this case, ICON performance can be adversely affected while the procedure is executing.

There are no specific restrictions about the order in which you must run the gsysPurge81 purge procedures.

Setting up gsysPurge81

The G_DB_PARAMETERS table stores parameters that the purge procedures use to control their operation. To set up the gsysPurge81 procedure, ensure that the parameter settings in IDB are suitable for your deployment.

The table below lists the parameters and their default values.

gsysPurge81 Parameters in the G_DB_PARAMETERS Table

G_DB_PARAMETERS Table		Description
OPT Column	VAL Column	
rowsperttransaction	0	gsysPurge81 purges all records in a single table with the same partition ID in one transaction. This is the default value.

G_DB_PARAMETERS Table		Description
		Note: Values from 1 to 100,000 are considered invalid and are treated the same as a value of 0.
	>100000	Specifies, in number of records, the maximum size of one transaction.
	no value	If no value has been stored in the G_DB_PARAMETERS table, gsysPurge81 sets a default transaction size of 200000 when it is executed, and stores the corresponding record in the G_DB_PARAMETERS table.
Note: All entries have column SETID = 0, and column SECT = 'gsyspurge81'.		

Updating the G_DB_PARAMETERS Table

If necessary, update the G_DB_PARAMETERS table. Interaction Concentrator provides a stored procedure, svcUpdateDBParameters, to perform this function. The stored procedure requires you to specify values for SECT (always 'gsyspurge81'), OPT (always 'rowspertransaction'), and VAL.

Executing gsysPurge81

Genesys recommends that you execute the gsysPurge81 procedure on a daily basis, in order to reduce the amount of time that is required for data processing and for the delivery of correct data to other applications—for example, downstream reporting systems. For additional scheduling considerations, see [Scheduling gsysPurge81](#).

Calling gsysPurge81

You must supply the *number_of_days* and *delete_all_flag* input parameters when you call the gsysPurge81 purge procedure. To execute the gsysPurge81 stored procedure, use the statement that corresponds to your RDBMS:

On Microsoft SQL

```
EXEC gsysPurge81 number_of_days, delete_all_flag
```

On Oracle

```
EXEC gsysPurge81 (number_of_days, delete_all_flag);  
commit
```

On DB2

```
CALL gsysPurge81 (number_of_days, delete_all_flag);
```

On PostgreSQL

```
SELECT gsysPurge81 (number_of_days, delete_all_flag);  
commit
```

Examples The following examples illustrate the syntax required to purge terminated multimedia interaction records older than 30 days:

Microsoft SQL

```
EXEC gsysPurge81 30,0
```

Oracle

```
EXEC gsysPurge81 (30,0);
```

commit

DB2

```
CALL gsysPurge81 (30,0);
```

PostgreSQL

```
SELECT gsysPurge81 (30,0);
```

```
commit;
```

Important

In these examples, gsysPurge81 retains any non-terminated records older than 30 days.

Purging by Truncating Partitions

Environments with large amounts of data to maintain can choose to partition their database and then purge efficiently by truncating entire partitions using the purgePartitions811 stored procedure. During this purge, all records in the specified partitions—both terminated and non-terminated—are truncated unconditionally.

If you need to purge only non-terminated records, use the gsysPurge81 purge procedure instead.

Important

This purge partitions functionality is supported only for Oracle 11g and higher.

To use the purgePartitions811 stored procedure, you must:

1. Set up a new IDB using the appropriate SQL scripts. For instructions, see the [Configuring a Partitioned Oracle IDB](#) in the *Interaction Concentrator Deployment Guide*.
2. Set a value of 1 or 2 for the partition-type configuration option. Note that Genesys Info Mart requires that you set the value to 2.

How purgePartitions811 Works

The number of partitions is set to 14 when you create your IDB.

Data for each individual day, as defined by UTC timestamps stored in the GSYS_TS field, is stored together in the same partition.

Depending on the value of the ICON partition-type option, the media type, and the specific table, the value stored in the GSYS_TS field might be the same as the created_ts field, might contain the timestamp when the interaction was submitted, or might contain the timestamp when the interaction was terminated.

The purge procedure does not require that a partition contain a certain amount of data before moving on to write into the next one.

ICON writes each day's data sequentially into the partitions starting with Partition 1 on Day 1 and continuing through Partition 14 on Day 14. When Partition 14 is filled, ICON returns to Partition 1 to record Day 15 data.

Important

If a partition is not purged before its next turn in the writing cycle, it could contain data from multiple days. For example, if on Day 15, Partition 1 has not yet been purged, Partition 1 will then contain data from both Day 1 and Day 15. In this scenario, when Partition 1 is purged, data from both Day 1 and Day 15 is truncated completely.

In general, ICON writes data from the current day into the current partition as it is gathered. However, if there happens to be a backlog of data for some reason, ICON uses the UTC timestamp values to write data to the appropriate partitions.

For example, let us assume that today is Day 3 and ICON is writing to Partition 3. However, if there is a backlog in writing data to IDB, and data from yesterday (Day 2) arrives, ICON writes the Day 2 data in Partition 2.

When you run the purge procedure, you specify the number of partitions to keep in the *number_of_days* parameter. Partitions are eligible to be purged based on where they are in the writing cycle relative to the partition into which ICON is currently writing.

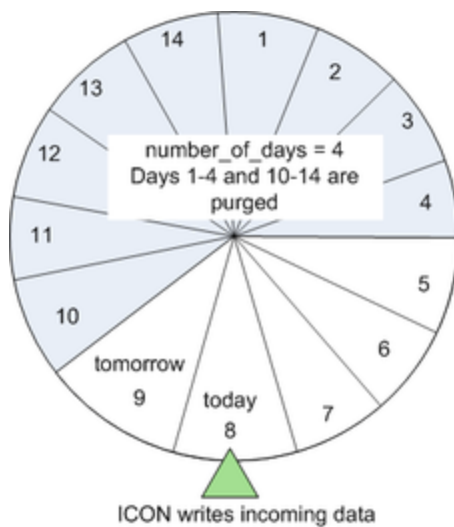
How to Configure the purgePartitions811 Procedure

When you run the purge procedure, you will specify the number of *completed* or *whole* days you want to keep. The purge procedure only considers whole days (as defined by UTC time) when calculating which days to keep and which to purge. ICON retains *number_of_days*+1 days/partitions. It keeps *number_of_days* partitions—including the one for the current day—and one future or *tomorrow* partition. The *tomorrow* partition provides a safeguard in case the purge procedure execution should overlap from one UTC day to the next. Without it, if the purge execution continued past the day boundary, data would start being written into a day that was being purged, with possible loss of data.

Example

There are 14 partitions, each of which contains the data for a day. The *number_of_days* parameter is set to 4. Today's partition is Number 8. When you run the purge procedure, it keeps four partitions, including today: Partitions 5-8. It also keeps Partition 9. Partitions 1-4 and 10-14 are truncated/purged.

If the tomorrow partition, Partition 9, was truncated, and the purge process started on Day 8 and went past the day boundary to Day 9, then it would still be running while data started being written to Partition 9. If Partition 9 were then truncated, you might lose data.



The following graphic shows why 13 is an invalid value for the *number_of_days* parameter. All days are retained, no data is purged, and an error message is logged in the G_LOG_MESSAGES table.



Executing the purgePartitions811 Procedure

To execute the purge stored procedure, use the following statement, entering a *number_of_days* value of 12 or less:

```
EXEC purgePartitions811 (number_of_days);  
commit;
```

Scheduling the purgePartitions811 Procedure

Genesys recommends the following guidelines for scheduling the purge procedure:

- Schedule execution of the stored procedure daily so as to consistently leave the number of partitions set in the *number_of_days* parameter in IDB.

- Refer to the documentation for your downstream reporting application for additional considerations and recommendations. For example, the Genesys Info Mart documentation recommends that you retain IDB data for at least 7 days to ensure that extract, transform, and load (ETL) processing, which might be delayed by network problems, ETL outage, or other issues, has time to complete before the data is purged.

If ICON Data Writing is Partially or Completely Interrupted

If for some reason, ICON is stopped or not writing data from some sources, this gap in writing data does not stop the purge procedure. It continues each day to advance the partition it considers to be the current one.

The purge procedure does not take into account any aspect of the data itself. For example, if records for a non-terminated interaction are located in a partition that is scheduled for purging, that partition will be truncated, regardless of the state of the data.

Important

For this reason, do not schedule automatic daily purging unless you also schedule an automated task to check that ICON is correctly writing all required data to IDB. For example, you might check DSS table information to make sure all source streams are current and that ICON has not fallen behind.

If, for some reason, purging is suspended so long that the partition into which ICON is now writing data already contains data from the previous writing cycle, the existing data is not lost.

However, it means that obsolete data is retained because that partition is not eligible to be purged until the value set in the *number_of_days* parameter has elapsed. This is also why the *number_of_days* parameter value must be smaller than the number of partitions, or else data is never purged.

Logging and Error Handling

The G_LOG_MESSAGES table logs the following information about the state of a purge procedure operation:

- Started—The primary procedure has started.
- ERROR—A database error has occurred. The G_LOG_ATTRS table provides a description of the error.
- Completed [OK|ERROR|Locked]—The primary procedure has ended or stopped. Completed: Locked means that the primary procedure was terminated because a prior instance of the procedure was still running.

The G_LOG_ATTRS table contains the following information about a purge procedure operation:

- Error descriptions (code and message).
- The value of the *number_of_days* input parameter.
- Information about the processed data, including the number of partitions purged from the each table subject to purging by truncating of partitions.

You can read information from the G_LOG_MESSAGES and G_LOG_ATTRS tables by using Genesys Solution Control Interface (SCI).

Using Separate Purge Procedures

This section explains how to use the separate gsysPurgeIR, gsysPurgeUDH, gsysPurgeLS, and gsysPurgeOS procedures to purge voice interactions, user data history, agent login session, and Outbound Contact data, respectively, from IDB.

Important

These separate purge procedures were discontinued in release 8.1.503.03. If you are using Interaction Concentrator 8.1.503.03 or higher, use gsysPurge81 or (in Oracle environments) purgePartitions811.

Purging Outbound Sessions Data

You can use either gsysPurgeOS or gsysPurge81 to purge outbound-session (OS) data. If you choose to use gsysPurge81, review the information in [Using the gsysPurge81 Stored Procedure](#). The present section describes gsysPurgeOS.

The gsysPurgeOS procedure calls the gsysPurge_GOS stored procedure, which deletes data related to outbound sessions.

Input Parameter

The gsysPurgeOS input parameter, *count_days*, is used to calculate the retention period for outbound-session data. The purge procedure deletes all session-related data for closed outbound campaign sessions that have a termination date earlier than the current date minus *count_days*. The current date starts at midnight, and the time segment of the current date is ignored.

gsysPurgeOS deletes outbound session-related data for closed campaign sessions only. If the parent campaign session is not closed, all corresponding chains are not deleted.

Tables Purged by the OS Purge Procedure

GO_CAMPAIGN	GO_CAMPPROP_HIST	GO_METRICS	GO_CHAIN
GO_CAMPAIGNHISTORY	GOX_CHAIN_CALL	GO_RECORD	GO_CHAINREC_HIST
GO_CUSTOM_FIELDS	GO_SECURE_FIELDS	GO_SEC_FIELDHIST	GO_FIELDHIST

After the OS purge procedure is completed, there are no references to non-existent data in the tables, except for possible references to non-existent interaction records.

Purging Voice and Logically Related Interaction Data

Genesys recommends using the `gsysPurge81` purge procedure if you are purging large amounts of data, or if you are concerned about ICON's performance. See [Using the `gsysPurge81` Stored Procedure](#).

Purging Voice Interaction Data

The `gsysPurgeIR` procedure, which is used to purge IRs, calls the following stored procedures:

- `gsysPurge_GCC`—Deletes IR data.
- `gsysPurge_GUD`—Deletes user data.

The `gsysPurgeIR` procedure deletes only merged IRs.

Input Parameter

The `gsysPurgeIR` input parameter, *count_days*, is used to calculate the retention period for IR-related data. The purge procedure deletes all IR-related data for merged IRs that have a merge date earlier than the current date minus *count_days*. The current date starts at midnight, and the time segment of the current date is ignored.

For example, if the date and time at which the stored procedure is invoked is 05/05/2007 13:15 and the input parameter is 1, all IRs that have a merge date earlier than 05/04/2007 00:00 will be deleted.

Tables Purged by the `gsysPurgeIR` Purge Procedure

G_IR	G_IR_HISTORY	G_CALL	G_CALL_HISTORY
G_PARTY	G_PARTY_HISTORY	G_IS_LINK	G_IS_LINK_HISTORY
G_ROUTE_RESULT	G_ROUTE_RES_VQ_HIST	G_VIRTUAL_QUEUE	G_CALL_STAT
G_PARTY_STAT	G_CALL_USERDATA	G_CALL_USERDATA_CUST	G_CALL_USERDATA_CUST1
G_CALL_USERDATA_CUST2			

After the IR purge procedure is completed, there are no references to non-existent data in the tables. For example, the `G_PARTY_STAT` table does not contain references to a party that has been purged from the `G_PARTY` table.

Purging User Data History

Genesys recommends using the `gsysPurge81` purge procedure if you are purging large amounts of data, or if you are concerned about ICON's performance. See [Using the `gsysPurge81` Stored Procedure](#).

The `gsysPurgeUDH` procedure is used to purge User Data History (UDH) records. The procedure deletes records that are related to merged IRs only.

Input Parameter

The gsysPurgeUDH input parameter, *count_days*, is used to calculate the retention period for UDH data. The purge procedure deletes all UDH records that satisfy one of the following conditions:

The merge date of the corresponding IR is earlier than the current date minus *count_days*. No corresponding IR exists, and the date the history record was added is earlier than the current date minus *count_days*.

The current date starts at midnight, and the time segment of the current date is ignored.

Tables Purged by the UDH Purge Procedure

The following tables contain UDH data. Depending on the roles and the attached data specification configured for the ICON application(s) writing to IDB, the tables may not be populated.

- G_SECURE_USERDATA_HISTORY
- G_USERDATA_HISTORY

After the UDH purge procedure is completed, there are no references to non-existent data in the tables.

Purging Agent Login Sessions

Genesys recommends using the gsysPurge81 purge procedure if you are purging large amounts of data, or if you are concerned about ICON's performance. See [Using the gsysPurge81 Stored Procedure](#).

The gsysPurgeLS procedure, which is used to purge Agent Login Session (ALS) data, calls the following stored procedures:

- gsysPurge_GLS—Deletes sessions.

The gsysPurgeLS procedure deletes only closed sessions.

Input Parameter

The gsysPurgeLS input parameter, *count_days*, is used to calculate the retention period for ALS-related data. The purge procedure deletes all session-related data for closed agent login sessions that have a termination date earlier than the current date minus *count_days*. The current date starts at midnight, and the time segment of the current date is ignored.

Tables Purged by the ALS Purge Procedure

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

After the ALS purge procedure is completed, there are no references to non-existent data in the tables, except for possible references to non-existent IRs.

Logging and Error Handling

The G_LOG_MESSAGES table logs the following information about the state of a purge procedure operation:

- Started—The primary procedure has started.
- ERROR—A database error has occurred. The G_LOG_ATTRS table provides a description of the error.
- Completed [OK|ERROR|Locked]—The primary procedure has ended or stopped. Completed: Locked means that the primary procedure was terminated because a prior instance of the procedure was still running.

The G_LOG_ATTRS table contains the following information about a purge procedure operation:

- Error descriptions (code and message).
- The value of the *count_days* input parameter.
- Information about the processed data, including the number of records purged from the main table (for each primary purge procedure) and the number of records purged from all tables.

You can read information from the G_LOG_MESSAGES and G_LOG_ATTRS tables by using Genesys Solution Control Interface (SCI).

Auto-Recovery

The purge procedures use a number of temporary tables and also update additional indexes during execution. If an error occurs during execution of a purge procedure, the procedure is terminated, but no special action is required to reset the procedure. The next time the procedure is started, the entry point is calculated from minimum merge sequence or timestamp values in the applicable IDB tables.

Timeout Interval

A maximum transaction time parameter sets a timeout interval for the procedure lock. If a purge procedure terminates because of an unhandled error, the lock will be overridden if the next instance of the procedure starts after the timeout interval has expired. For more information about the maximum transaction time parameter, see [Setting Up the Separate Purge Procedures](#).

Scheduling the Separate Purge Procedures

It is the responsibility of the customer (usually the Database Administrator) to run the purge procedures as required. Genesys recommends that you run the purge procedures at a time when contact center activity is low. If you run the purge procedure occasionally, on an ad hoc basis, execution can take significantly longer than if you run it regularly. Also, in this case, ICON performance can be adversely affected while the procedure is executing.

There are no specific restrictions about the order in which you must run the purge procedures. However, for best performance and to maintain data consistency throughout the process, Genesys recommends that you run the purge procedures sequentially, in the following order:

- gsysPurgeIR (purge interaction records)
- gsysPurgeUDH (purge user data and user data handling data)
- gsysPurgeLS (purge agent login and session data)
- gsysPurgeOS (purge Outbound Contact data)

In particular, if you are purging more than one day's worth of data, execute the purge procedures sequentially.

Note the following additional considerations and recommendations:

- If you are running the purge procedures sequentially, you may need to adjust the *count_days* input parameter for purge procedures that execute later in the sequence, because the procedures may start after midnight.

For example, if the procedure to purge IRs starts at 05/05/2007 20:00 (08:00 PM) and the *count_days* input parameter is 1, set the input parameter for all procedures that should start after 05/05/2007 23:59:59 (11:59:59 PM) to 2.

- Because the gsysPurgeIR purge procedures purge merged interactions only, consider your merge procedure schedule when scheduling the purge procedures. In particular, note that, even in a single-site deployment, you must run the merge procedure before you will be able to purge any records from IDB.

Setting Up the Separate Purge Procedures

The G_DB_PARAMETERS table stores parameters that the purge procedures use to control their operation. To set up the purge procedures, ensure that the parameter settings in IDB are suitable for your deployment. The table below lists the parameters and their default values.

Purge Parameters in the G_DB_PARAMETERS Table

Applicable Procedure	G_DB_PARAMETERS Table		Description
OPT Column	VAL Column		
gsysPurgeIR	IR_seq_step	75	Specifies the chunk size, in terms of number of ICON transactions, that are used for data lookup.
gsysPurgeUDH	UDH_time_step	300	Specifies the chunk size, in seconds, that is used to calculate the number of rows to purge in a single database transaction.
gsysPurgeLS	LS_time_step	500	Specifies the chunk size, in seconds, that is used to calculate the number of rows to purge in a single database

Applicable Procedure	G_DB_PARAMETERS Table		Description
			transaction.
gsysPurgeOS	OS_time_step	500	Specifies the chunk size, in seconds, that is used to calculate the number of rows to purge in a single database transaction.
gsysPurgeIR	IR_max_tran_time	600	Specifies the timeout interval for the procedure lock, in seconds. The maximum transaction time is the amount of time between the most recent purge transaction and the current time, after which a new instance of the procedure is allowed to execute.
Note: All entries have column SETID = 0 and column SECT = 'purge'.			

Updating the G_DB_PARAMETERS Table

If necessary, update the G_DB_PARAMETERS table. Interaction Concentrator provides a stored procedure, svcUpdatedBParameters, to perform this function. The stored procedure requires you to specify values for SECT (always 'purge'), OPT (see the OPT Column in [Purge Parameters in the G_DB_PARAMETERS table](#)), and VAL.

For example, to change the value of the procedure lock timeout for the IR purge procedure, execute the following statement (the exact syntax depends on the RDBMS):

```
EXEC svcUpdatedBParameters
0,
'purge',
'IR_max_tran_time',
'<TIMEOUT>'
```

Calling Purge Stored Procedures

You must supply the *count_days* input parameter when you call the gsysPurgeIR, gsysPurgeUDH, gsysPurgeLS, and gsysPurgeOS purge procedures.

For each supported RDBMS, there are different syntax requirements for the script that invokes a purge procedure. This section provides the following examples:

- Example for Oracle
- Example for Microsoft SQL
- Example for PostgreSQL
- Example for DB2

In these examples, the IR purge and ALS purge procedures are executed sequentially in the same session, and use different values for the retention period.

Example for Oracle

```
declare
  COUNT_DAYS$ int := 14;
begin
  gsysPurgeIR( COUNT_DAYS$ );
end;

declare
  COUNT_DAYS1$ int := 15;
begin
  gsysPurgeLS( COUNT_DAYS1$ );
end;
```

Example for Microsoft SQL

```
declare @COUNT_DAYS          int
set @COUNT_DAYS = 14
exec gsysPurgeIR @COUNT_DAYS

declare @COUNT_DAYS1          int
set @COUNT_DAYS1 = 15
exec gsysPurgeLS @COUNT_DAYS1
```

Example for PostgreSQL

```
declare COUNT_DAYS int;
declare COUNT_DAYS1 int;
begin
  COUNT_DAYS:= 14;
  perform gsysPurgeIR(COUNT_DAYS);
  COUNT_DAYS1:= 15;
  perform gsysPurgeLS (COUNT_DAYS1);
end;
```

Example for DB2

```
language SQL
begin
  declare COUNT_DAYS          int default 14;

  call gsysPurgeIR( COUNT_DAYS );

end;

language SQL
begin
  declare COUNT_DAYS1          int default 15;

  call gsysPurgeLS( COUNT_DAYS1 );

end;
```