



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

The Merge Stored Procedure

4/10/2025

Contents

- 1 The Merge Stored Procedure
 - 1.1 Scheduling
 - 1.2 gsysIRMerge and gsysIRMerge2
 - 1.3 Setting Up the Merge Procedure
 - 1.4 gsysIRMerge2 Parameters
 - 1.5 Executing the Merge Procedure

The Merge Stored Procedure

Important

- The merge stored procedure described on this page is necessary only if you are running Genesys Info Mart 7.6 or earlier or you are running Interaction Concentrator *without* Genesys Info Mart. Genesys Info Mart provides a merge procedure that supplants the one documented on this page.
- This merge procedure is not supported on PostgreSQL RDBMSs.

The merge stored procedure merges voice interaction records in the Interaction Database (IDB) in order to finalize data processing of closed single-site and multi-site interactions. The procedure merges voice interaction records that are stored within a single IDB (intra-IDB merge). The procedure does not merge records that are distributed across more than one IDB (inter-IDB merge or multi-IDB merge).

The merge procedure can run in the background while instances of the ICON process are actively writing data to IDB.

Scheduling

It is the responsibility of the customer (usually the Database Administrator) to run the merge procedure as required. Genesys recommends that you schedule the merge stored procedure to run on a regular basis. If you do not have this stored procedure scheduled to run regularly (not less than every 15 minutes), ensure that you run it before any data is extracted from IDB.

Important

If you are using Genesys Info Mart 7.6, run this procedure every five minutes. Genesys Info Mart 8.x does not use this stored procedure. Instead, it merges interactions automatically as a part of the scheduled ETL process.

If you run the merge 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.

This section contains the following information about the merge procedure:

- How the [gsysIRMerge](#) and [gsysIRMerge2](#) stored procedures function
- [Setting Up the Merge Procedure](#)
- [gsysIRMerge2 Parameters](#)

- Executing the Merge Procedure

gsysIRMerge and gsysIRMerge2

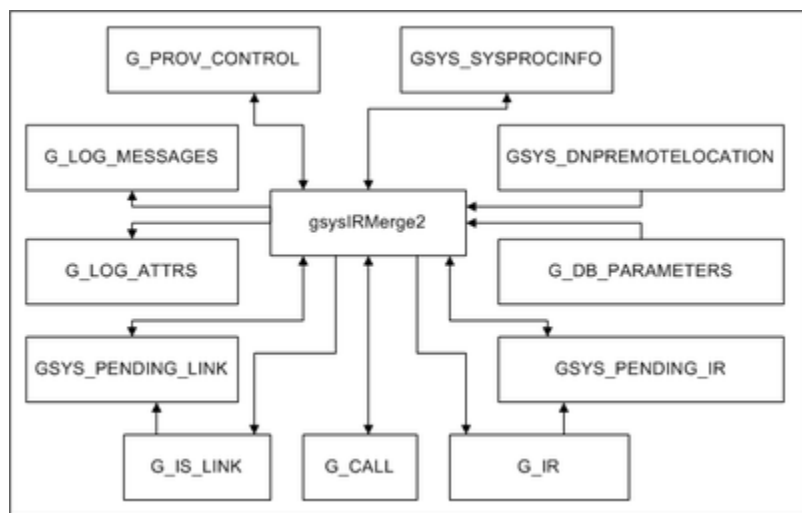
For backward compatibility and uncommon environments, Interaction Concentrator retains support for two stored procedures for voice data merge:

- gsysIRMerge—A wrapper for gsysIRMerge2.
- gsysIRMerge2—The stored procedure that actually performs the merge.

gsysIRMerge was originally provided to ensure backward compatibility with Interaction Concentrator 7.2. Calls to earlier versions of gsysIRMerge and gsysIRMerge2 are compatible with calls to gsysIRMerge and gsysIRMerge2 in Interaction Concentrator release 8.x.

Tables Used by gsysIRMerge2

The following figure shows the tables involved in the merge procedure.



The following subsections describe each of these tables, and how they are used in the merge procedure.

G_LOG_MESSAGES and G_LOG_ATTRS

- G_LOG_MESSAGES—Contains information about the start (MESSAGE_ID = 5020) and end (MESSAGE_ID = 5030) of the merge procedure.
- G_LOG_ATTRS—Contains the error descriptions (code and message), and detailed information about the processed data.

You can read information from these two tables by using Genesys Solution Control Interface (SCI), which is the graphical user interface (GUI) component of the Genesys Management Layer.

G_PROV_CONTROL

From the G_PROV_CONTROL table, the merge procedure reads information about the ICON instances that are writing data to the database, and the corresponding number of the latest transaction. The procedure executes the following query against the G_PROV_CONTROL table:

```
SELECT PRIMARYID SYS_ID          /* ICON instance identity*/
       SEQCURRENT      /* number of latest transaction*/
FROM G_PROV_CONTROL
   WHERE PROVIDERTAG = 1 AND SEQCURRENT IS NOT NULL
```

In addition, the merge procedure reads from this table the number of its own latest transaction, using the following query:

```
SELECT SEQCOUNTER FROM G_PROV_CONTROL
   WHERE DOMAINID = 0
   AND PRIMARYID = 99
   AND PROVIDERTAG = 0
```

After this, the merge procedure starts data processing. After data processing is complete, the merge procedure updates the number of its own transaction to a new value.

GSYS_SYSPROCINFO

From the GSYS_SYSPROCINFO table, the merge procedure reads information about the number of the latest ICON transaction that was processed, using the following queries for each ICON instance:

```
SELECT SEQPROCESSED FROM GSYS_SYSPROCINFO
   WHERE PROVIDERTAG = 1 AND PROCNAME = 'gsysirmerge'
```

After data processing is complete, the merge procedure updates the number of the transaction to the processed one.

GSYS_DNPREMOTELOCATION and G_DB_PARAMETERS

The data from the GSYS_DNPREMOTELOCATION and G_DB_PARAMETERS tables is used to process stuck inter-server (IS) links (that is, links for which there is no information about a corresponding IS-Link at another site). The normal situation for the merge procedure is to merge interactions when IDB contains complete information about all parts of the interaction. However, for interactions that cross multiple sites, if all the sites are not monitored by ICON instances writing to the same IDB, the IDB will never have enough information to conclude that the interaction has ended and to identify the constituent parts of the interaction. The IS-Link is stuck.

GSYS_DNPREMOTELOCATION

From the GSYS_DNPREMOTELOCATION table, the merge procedure reads the names of the switches that are not monitored by any ICON instance that writes to this particular instance of IDB. From the point of view of this database, unmonitored switches are considered to be remote locations.

Records in the GSYS_DNPREMOTELOCATION table indicate to the merge procedure that IS-Links that point to remote locations are unpaired within this instance of IDB. Therefore, the merge procedure does not expect any further information about the interaction on the other end of the IS-Link. In this way, data from the GSYS_DNPREMOTELOCATION table enables the merge procedure to minimize the amount of time required in order to process stuck IS-Links, because the procedure will not wait for the IS-Link timeout to expire. (For more information about the IS-Link timeout, see [G_DB_PARAMETERS](#). See also [stuckthreshold](#).)

G_DB_PARAMETERS

From the G_DB_PARAMETERS table, the merge procedure reads the time interval, in seconds, at which information about the IS-Link at another site is expected to be reported, using the following query:

```
SELECT VAL FROM G_DB_PARAMETERS
WHERE          SECT = 'merge' AND
              OPT = 'stuckthreshold'
```

The merge procedure also uses other parameters from the G_DB_PARAMETERS table. For more information, see [IS-Link Timeout and Other Parameters](#).

GSYS_PENDING_IR and GSYS_PENDING_LINK

The GSYS_PENDING_IR and GSYS_PENDING_LINK tables track the merge state of the interaction records and IS-Links while they are being processed. The merge procedure populates the GSYS_PENDING_IR and GSYS_PENDING_LINK tables with temporary data.

G_IS_LINK, G_IR, and G_CALL

The merge procedure uses the G_IS_LINK table to determine multi-site interactions. A selected multi-site interaction is considered to be completed when one of the following occurs:

- All IS-Links have a corresponding IS-Link from another site.
- An IS-Link has been opened to an unmonitored switch.
- The timeout for an IS-Link to arrive from another site has expired, and all corresponding IRs are closed.

When a multi-site interaction is completed, the procedure makes the following updates:

```
UPDATE G_IS_LINK SET MERGESTATE = 3
WHERE LINKID in (ALL IS LINK IN INTERACTION)

UPDATE G_IR SET MERGESTATE = 3
      ROOTIRID = (FIRST IR IN INTERACTION)
      GSYS_MSEQ = (the procedure's current TRANSACTION ID)
      GSYS_MSEQ_TS = (time of TRANSACTION)
WHERE ROOTIRID in (ALL ROOT IRs IN INTERACTION)

UPDATE G_CALL
      SET ROOTIRID = (FIRST IR IN INTERACTION)
WHERE ROOTIRID in (ALL ROOT IRs IN INTERACTION)
```

For a single-site interaction, if all corresponding IRs are closed, the interaction is completed, and the procedure makes the following update:

```
UPDATE G_IR SET MERGESTATE = 3
      GSYS_MSEQ = (the procedure's current TRANSACTION ID)
      GSYS_MSEQ_TS = (time of TRANSACTION)
WHERE IRID in (IRs IN INTERACTION)
```

Setting Up the Merge Procedure

To set up the merge procedure, ensure that the following information in IDB is correct for your purposes:

Unmonitored Switches

The `GSYS_DNPREMOTELOCATION` table contains the names of the switches that are not monitored by any ICON instance that writes to this IDB. If necessary, manually update the `GSYS_DNPREMOTELOCATION` table, using a standard `INSERT` statement. The `REMOTELOCATION` column in the `GSYS_DNPREMOTELOCATION` table stores the names of unmonitored switches.

Example

For example, suppose that you have two switches, each of which is monitored by a separate ICON that writes to its own IDB:

- ICON1 monitors switch `SITE1_sw1` and writes to IDB1.
- ICON2 monitors switch `SITE2_sw2` and writes to IDB2.

For optimal performance of the merge stored procedure, add the following records to the respective IDBs:

- In IDB1, set `GSYS_DNPREMOTELOCATION.REMOTELOCATION='SITE2_sw2'`
- In IDB2, set `GSYS_DNPREMOTELOCATION.REMOTELOCATION='SITE1_sw1'`

IS-Link Timeout and Other Parameters

The `G_DB_PARAMETERS` table stores parameters that the merge procedure uses to control its operation. The table below lists the parameters and their default values that the merge procedure uses.

Merge Parameters in the `G_DB_PARAMETERS` Table

OPT Column		VAL Column		Description
Oracle	Microsoft SQL	DB2		
loopresolver	0	0	0	Specifies whether the merge procedure should process the loopresolver parameter, which handles calls that form a call topology having loops. Valid values: 1—Handle calls

OPT Column	VAL Column			Description
				<p>that form a call topology with loops.</p> <p>0—Do not handle calls that form a call topology with loops. Setting VAL to 0 is equivalent to not configuring the parameter at all.</p>
nodnrl	0	0	0	<p>Specifies whether the merge procedure disregards remote locations. Valid values:</p> <p>1—Disregard REMOTELOCATION.</p> <p>0—Do not disregard REMOTELOCATION.</p>
stuckthreshold	28860	28860	28860	<p>Specifies the time interval, in seconds, at which IS-Link information is expected from another site. After this time period expires, the IS-Link is considered to be stuck.</p>
brokenthreshold	14460	14460	14460	<p>Specifies the time interval, in seconds, for which the merge procedure will wait for IS-Link information for a failed Inter Server Call Control (ISCC) transfer to another site. In other words, this parameter specifies the stuck link timeout for broken links.</p>
step	75	75	25	<p>Specifies the number of transactions that the merge</p>

OPT Column	VAL Column			Description
				<p>procedure selects at the same time (when it is reading in new IRs and new links).</p> <p>A low value limits exposure to locks on core tables. However, a very low value can result in too many iterations. For additional information about large-scale deployments using Microsoft SQL, see Note for Large-Scale Deployments Using Microsoft SQL.</p>
limit	1000	2000	1000	<p>Specifies the number of root interactions that the merge procedure considers for closure at the same time (when it updates the G_IS_LINK, G_IR, and G_CALL tables).</p> <p>This setting effectively limits the number of IRs per MSEQ. A very low value can result in too many iterations. For additional information about large-scale deployments using Microsoft SQL, see Note for Large-Scale Deployments Using Microsoft SQL.</p>
limit2	10000	10000	10000	<p>Specifies the number of new links and interaction records that the merge procedure reads before it begins the closure phase (when it updates the G_IS_LINK, G_IR, and G_CALL tables).</p>

OPT Column	VAL Column			Description
				A very high value can result in suboptimal performance.
Note: All entries have column SETID = 0 and column SECT = 'merge'.				

Note for Large-Scale Deployments Using Microsoft SQL

With Microsoft SQL, the default values of the **step** and **limit** parameters are not optimal for IDBs with large amounts of data (in the order of millions of interactions). For better performance, Genesys recommends the following as a first step:

- Increase the value of the step parameter to 200.
- Increase the value of the limit parameter to 3000.

However, you will likely need to experiment to find the optimal values for your large-scale deployment.

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 'merge'), OPT (see the OPT Column in the [Merge Parameters in the G_DB_PARAMETERS table](#)), and VAL.

For example, to write the IS-Link timeout to the database, execute the following statement (the exact syntax depends on the RDBMS):

```
EXEC svcUpdateDBParameters
0,
'merge',
'stuckthreshold',
'<TIMEOUT>'
```

The brokenthreshold Parameter

The brokenthreshold parameter is used to streamline the processing of calls when an inter-site transfer fails. For example, consider the scenario in which ICON1 monitors Site1 and ICON2 monitors Site2, and an agent on Site1 attempts to transfer a call to Site2, but the attempted ISCC transfer does not reach Site2 or the agent on Site2 does not answer. In this situation:

ICON1 will receive information from T-Server that the link has failed, and the IS-Link information that ICON1 stores in IDB will be marked as failed (G_IS_LINK.STATE=3).

For Site2:

If the attempted ISCC transfer does not reach Site2, ICON2 will never receive information about the attempted transfer, and ICON2 will never store any corresponding IS-Link information in IDB.

If the call reaches Site2 but the agent does not answer, ICON2 will receive information from T-Server

and will store corresponding IS-Link information in IDB, but the information might be delayed.

In both cases, by default, the merge procedure will wait up to 4 hours 1 minute for missing IS-Link information from ICON2 (instead of the default 8 hours 1 minute for other stuck links) before finalizing the merge.

To modify the value of the broken links timeout, insert a row into the G_DB_PARAMETERS table with the following attributes: SETID=0, SECT=merge, OPT=brokenthreshold, and VAL=new_timeout_value. Valid values are any positive integer. Changes take effect on the next run of the merge stored procedure.

Important

- If you set the brokenthreshold parameter to a value greater than the stuck link timeout (the stuckthreshold parameter), the stuckthreshold parameter will control the timeout for broken links as well.
- If you are migrating from an Interaction Concentrator release earlier than 8.0.000.42, the brokenthreshold parameter does not control the timeout for broken links that are already in the GSYS_PENDING_LINK table. The broken links timeout applies only to new broken links that are processed after you upgrade to release 8.0.000.42 or later.

gsysIRMerge2 Parameters

The gsysIRMerge2 stored procedure accepts five parameters: two input parameters and three output parameters. When the gsysIRMerge2 stored procedure is invoked by a call to the gsysIRMerge stored procedure, gsysIRMerge supplies the required parameters.

The initial implementation of gsysIRMerge2 required the parameters for concurrency control, to ensure that no more than one instance of the procedure was running at any given time. Subsequent changes to the merge procedure have relaxed the restrictions for database locking. All the gsysIRMerge2 parameters have been retained for backward compatibility. However, there is no meaningful difference in the way the merge procedure executes, regardless of the values you enter for the input parameters.

Important

Genesys Info Mart 8.x does not use the Interaction Concentrator merge stored procedures.

The table below lists the gsysIRMerge2 input and output parameters, as well as the values that are used for the input parameters when gsysIRMerge2 is called from gsysIRMerge.

Merge Procedure Parameters

Parameter	Data Type	Description
Input		
OVERRIDE	int	<p>Specifies the lock override setting. Valid values:</p> <p>0—Do not override lock.</p> <p>1—Override if PREVCALLER=CALLER.</p> <p>2—Always override.</p> <p>gsysIRMerge value: 0</p>
CALLER	varchar	<p>Specifies the name of the caller of the stored procedure.</p> <p>Valid values: Any string (for example, Infomart_DBID_1001, DCA6) gsysIRMerge value: singleIDBMerge</p>
Output		
RESULT	int	<p>Specifies the result of the stored procedure call.</p> <p>Valid values:</p> <p>0—Success</p> <p>2—IDB locked</p> <p>1—Other error</p> <p>gsysIRMerge value: RESULT</p>
PREVCALLER	varchar	<p>If RESULT=2, specifies the name of the previous caller.</p> <p>gsysIRMerge value: PREVCALLER</p>
PREVAGE	int	<p>If RESULT=2, specifies the number of seconds since the previous caller obtained the lock.</p> <p>gsysIRMerge value: PREVAGE</p>

The RESULT parameter provides information about the status of execution of the procedure.

Important

The gsysIRMerge stored procedure discards the output parameters (RESULT, PREVCALLER, and PREVAGE).

Executing the Merge Procedure

Genesys recommends that you execute the merge procedure periodically, 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.

You can call the merge stored procedure in two ways:

- By calling `gsysIRMerge`
- By calling `gsysIRMerge2`

The following subsections discuss each of these methods in turn.

In Interaction Concentrator 8.x, the only meaningful difference between the two methods is that calling `gsysIRMerge2` returns the result of the merge procedure, as well as other output parameters, as shown in the [Merge Procedure Parameters](#) table.

Important

The merge procedure can be executed on voice interaction data only.

Calling `gsysIRMerge`

To execute the merge procedure, use the following statement (the exact syntax depends on the RDBMS):

```
EXEC gsysIRMerge
```

The `gsysIRMerge` stored procedure then, in turn, calls `gsysIRMerge2`, using the following parameters: 1, `singleIDBMerge`, `RESULT`, `PREVCALLER`, and `PREVAGE`.

Calling `gsysIRMerge2`

Invoking the `gsysIRMerge2` stored procedure directly enables you to specify your own values for the input and resulting output parameters. You must supply the required parameters when you call the procedure. (For more information about the input and output parameters, see [gsysIRMerge2 Parameters](#).)

For each supported RDBMS, there are different syntax requirements for the script that invokes the `gsysIRMerge2` stored procedure directly. This section provides the following examples:

- [Example Script for Oracle](#)
- [Example Script for Microsoft SQL](#)
- [Example Script for DB2](#)

Example Script for Oracle

```
declare
  OVERRIDE$ int := 1;
  CALLER$ varchar2(40) := 'singleIDBMerge';
  RESULT$ int;
  PREVCALLER$ varchar2(40);
  PREVAGE$ int;
begin
  gsysIRMerge2( OVERRIDE$, CALLER$, RESULT$,
               PREVCALLER$, PREVAGE$ );
  dbms_output.put_line('='||RESULT$);
end;
```

Example Script for Microsoft SQL

```
declare @OVERRIDE          int
declare @CALLER            varchar(40)
declare @RESULT            int
declare @PREVCALLER       varchar(40)
declare @PREVAGE          int
set @OVERRIDE = 1
set @CALLER = 'singleIDBMerge'
exec gsysIRMerge2 @OVERRIDE, @CALLER, @RESULT, @PREVCALLER, @PREVAGE
```

Example Script for DB2

```
create procedure gsysIRMergeTest
DYNAMIC RESULT SETS 1
language SQL
begin
  declare OVERRIDE          int default 1;
  declare CALLER            varchar(40) default 'singleIDBMerge';
  declare RESULT            int;
  declare PREVCALLER       varchar(40);
  declare PREVAGE          int;

  call gsysIRMerge2( OVERRIDE, CALLER, RESULT,
                   PREVCALLER, PREVAGE );

  begin
    declare c_cur cursor with return for
      select RESULT, PREVCALLER, PREVAGE from sysibm.sysdummy1;
    open c_cur;
  end;

end;
call gsysIRMergeTest;
drop procedure gsysIRMergeTest;
```

Improving Merge Procedure Performance

Database configuration, database settings, and merge procedure scheduling can significantly impact merge procedure performance.

Computing Statistics

Periodically gathering statistics is a generic requirement for good database performance. For optimal

The Merge Stored Procedure

performance of the merge procedure, ensure that the available database statistics are representative (in other words, relatively fresh).

Troubleshooting the Merge Procedure

For information about merge procedure execution problems and their solutions, see [Troubleshooting](#) in the *Interaction Concentrator Deployment Guide*.

To minimize the occurrence of deadlocks and to optimize merge procedure performance, carefully review the information in [Setting Up the Merge Procedure](#) and in [Troubleshooting](#) in the *Interaction Concentrator Deployment Guide*. However, deadlocks are inevitable, and the downstream reporting application must be capable of handling them.

If the procedure raises any exceptions, issue a rollback statement before performing any other actions on the connection.

Resetting the Merge Procedure

For convenience, Interaction Concentrator provides a stored procedure, `gsysIRMergeReset`, that resets the merge procedure to a clean state. Execute the reset procedure if the merge procedure does not complete successfully and you want to rerun it.

To execute the merge procedure, use the following statement (the exact syntax depends on the RDBMS):

```
EXEC gsysIRMergeReset
```