DOWNTOWN LINE

Contract C955

InteGRATED SUPERVISORY

CONTROL SYSTEM (iscS)

DB\_Synch Configuration specificaiton

**AMENDMENT RECORDS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Rev*** | ***Date*** | ***Descriptions*** | ***Writer*** | ***Function*** |
| 1.0 | 22/05/2012 | Ouyang, zhilin | Draft |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Reviewed by** |  | **Approval by** |  |
| **Entity / Function** |  | **Entity / Function** |  |
| **Date & Sign.** |  | **Date & Sign.** |  |
| **Reserved to the Engineer** |  |  |  |

**Table of Contents**

[1 Introduction 5](#_Toc360627969)

[1.1 System Overview 5](#_Toc360627970)

[1.2 DB Synch Data Flow 5](#_Toc360627971)

[2 Database Configuration 8](#_Toc360627972)

[2.1 Configuration File 9](#_Toc360627973)

[2.2 Multi\_Queue Configuration 9](#_Toc360627974)

[2.2.1 Create a new queue; 9](#_Toc360627975)

[2.2.2 Create a payload type for the new created queue; 10](#_Toc360627976)

[2.2.3 Start up the queue; 10](#_Toc360627977)

[2.2.4 Enqueue message to the queue; 10](#_Toc360627978)

[2.2.5 Dequeue job for dequeue message from the queue. 12](#_Toc360627979)

[2.3 Failover Configuration 12](#_Toc360627980)

[2.4 Load balance configuration 12](#_Toc360627981)

[3 Station server installation guideline 13](#_Toc360627982)

[3.1 File path structure & rights 13](#_Toc360627983)

[3.2 Environment settings 13](#_Toc360627984)

[3.2.1 Library path to LD\_LIBRARY\_PATH 14](#_Toc360627985)

[3.3 DataSync Configuration settings 14](#_Toc360627986)

[3.3.1 Configuration file 14](#_Toc360627987)

[3.3.1.1 c\_cfg.cnf 14](#_Toc360627988)

[3.3.1.2 g\_datasyn\_synid 14](#_Toc360627989)

[4 DBS installation guideline 15](#_Toc360627990)

[4.1 File path structure & rights 15](#_Toc360627991)

[4.1.1 Path structure & rights 15](#_Toc360627992)

[4.1.2 Folder and components 15](#_Toc360627993)

[4.2 Environment settings 16](#_Toc360627994)

[4.2.1 Library path to LD\_LIBRARY\_PATH 16](#_Toc360627995)

[4.3 DataSync Configuration settings 16](#_Toc360627996)

[4.3.1 Configuration file 16](#_Toc360627997)

[4.3.1.1 s\_cfg.cnf 16](#_Toc360627998)

[4.3.1.2 g\_datasyn\_synid 17](#_Toc360627999)

[5 Appendix A Queue Tables 17](#_Toc360628000)

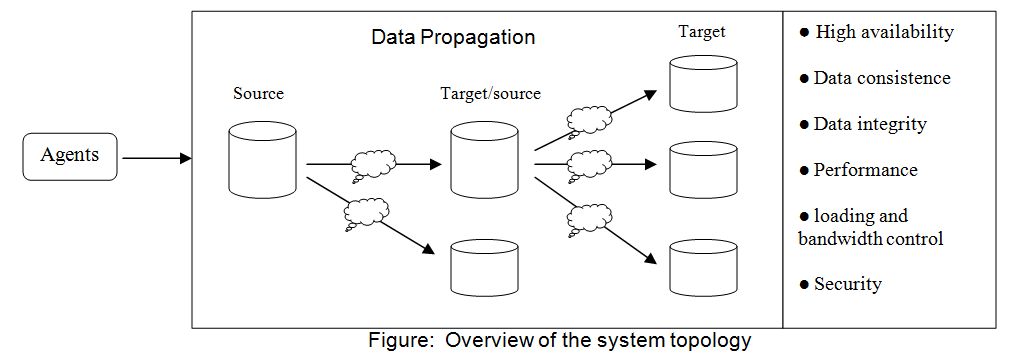
[5.1 Queue tables 17](#_Toc360628001)

# Introduction

The purpose of this document is to provide adequate information for configuring DB Synch Module which will synch data between MySQL and Oracle instead of using Oracle AQ.

## System Overview

The DB SynchEngine is design to imitate the Oracle AQ mechanism to propagate messages between distributed database servers. The overview of the SynchEngine show below:



## DB Synch Data Flow

Currently, the Audit Data message will be propagated using the SynchEngine in C955 project. The general strategy for audit data is that it is written to and read from a local database server. Any audit data that is required immediately at the central server shall be immediately propagated to the central database server. The following figure illustrates a category of data flow and data access for Audit Data in a distributed heterogeneous database architecture.



DT18

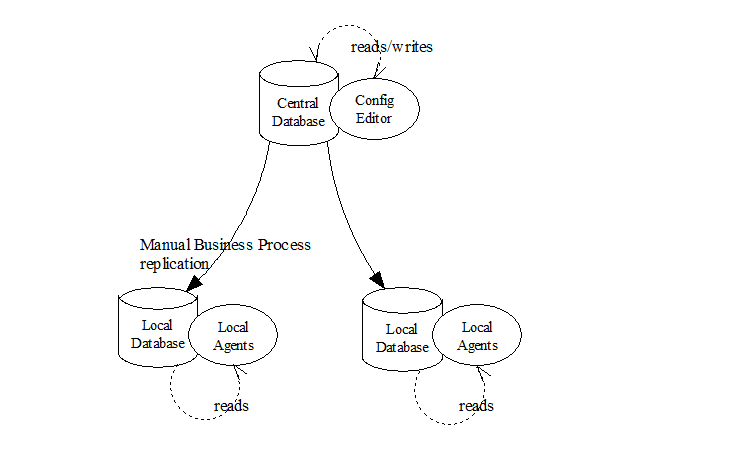
DT17

TRANSACT

Figure: AD data flow (Local to Central)

All the audit data shall be read from and write to at a local database, for the purpose of replication, these data shall be propagated to the central server database.

There are also data maybe propagate from central database to local database while the local database(s) offline and the agent(s) writes/reads from central database. The data will be recovery from central database when the local database came to online. The figure below show the data flow for this situation:



DT18

TRANSACT

DT18

DT17

TRANSACT

Figure: AD data flow (Central to Local)

SynchEngine also supports another message type: GROUP. The figure below depicts the data flow of group message.



Audit data

This left figure depicts the data flow that TRANSACT & DT18 are Group member of DT17

DT17

Group data

Group data

## Document overview

**Chapter 1:** provides the overview of the SynchEngine, such as topology, data flow.

**Chapter 2:** depicts the database configuration required for running SynchEngine correctly. It provides information which needed by SynchEngine such as tables. All these configuration data packed in the database patch. Therefore, the database patch need applied before installing the SynchEngine. This chapter is only for reference on trouble-shooting the database configuration.

**Chapter 3:** provides the SynchEngine installation/configuration guide for station servers.

**Chapter 4:** provides the SynchEngine installation/configuration guide for DBS central servers.

**Chapter 5:** appendix and lists some useful supplementary information.

# Database Configuration

For the database configuration, the following steps should be applied before deploying.

1. Applied database patch first. (including all the procedures & configuration tables)
2. Setup database tables (including recipient, recipient\_group & recipient\_info table), *DT17* station for example:

Table recipient (MySQL): for Insert, update and delete qualifier type

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PKEY | NAME | ADDRESS | DB\_LINK\_NAME | STATUS | TIMEOUT |
| 1 | TRANSACT | TRANSACT |  | 0 | 0 |

Table: setup example of MySQL DT17

Table recipient\_group (MySQL): for group qualifier type, there are only one group in station.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PKEY | NAME | MEMBER | DB\_LINK\_NAME | STATUS | TIMEOUT |
| 1 | TRA\_DT17 | TRANSACT |  |  | 0 |

Table: setup example of MySQL DT17

Table recipient\_info (MySQL): store all recipient information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PKEY | NAME | HOSTNAME | ADDRESS | DB\_LINK\_NAME | STATUS |
| 1 | TRA\_DT17 | dt17-sms-01 | 192.168.0.171:5001 | TRA\_DT17\_AQ | 0 |
| 2 | TRANSACT | occ-sms-01 | 192.168.0.3:2005 | TRANSACT\_AQ | 0 |

Table: setup example of MySQL DT17

Table recipient (Oracle), all stations’ db recipients should be defined.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PKEY | NAME | ADDRESS | DB\_LINK\_NAME | STATUS | TIMEOUT |
| 1 | TRA\_DT17 | TRA\_DT17 | TRA\_DT17\_AQ | 0 | 0 |
| 2 | TRA\_DT18 | TRA\_DT18 | TRA\_DT18\_AQ | 0 | 0 |

Table: setup example of Oracle side

Table recipient\_group (Oracle) all stations’ db group recipients should be defined.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PKEY | NAME | MEMBER | DB\_LINK\_NAME | STATUS | TIMEOUT |
| 1 | TRANSACT | TRA\_DT17 | TRA\_DT17\_AQ | 0 | 0 |
| 2 | TRANSACT | TRA\_DT18 | TRA\_DT18\_AQ | 0 | 0 |
| 3 | TRA\_DT17 | TRA\_DT17 | TRA\_DT17\_AQ | 0 | 0 |
| 4 | TRA\_DT18 | TRA\_DT18 | TRA\_DT18\_AQ | 0 | 0 |

Table: setup example of Oracle side

Table recipient\_info (Oracle), store all recipient information

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PKEY | NAME | HOSTNAME | ADDRESS | DB\_LINK\_NAME | STATUS |
| 1 | TRA\_DT17 | dt17-sms-01 | 192.168.0.171:5001 | TRA\_DT17\_AQ | 0 |
| 2 | TRA\_DT18 | dt18-sms-01 | 192.168.0.181:5002 | TRA\_DT18\_AQ | 0 |
| 3 | TRANSACT | occ-sms-01 | 192.168.0.3:2005 | TRANSACT\_AQ | 0 |

All the above tables’ configuration is just an example for station DT17. The data in the table should be changed based on the actual running environment.

## Configuration File

As the SynchEngine running both in MySQL & Oracle sides, the configuration file should be properly changed to meet the current working environment. Details shown below:

1. For station server(s) (usually working with MySQL database)

* Configuration

cp  ./dbsynconfig.cnf   ./c\_cfg.cnf    // for running as client/MySQL side

Open file c\_cfg.cnf and modify the following sections (example for DT17   SrcDbConStr =Oracle:OracleServiceName:UserName:Password)

SrcDbConStr = MySQL:TRA\_DT17:IT271350\_5:IT271350\_5:192.168.0.171

SkipErrors = 1061,1062

LogFileName = /u01/app/datasyn/log/c\_SynchEngine.log

Please refer to the comments of the configuration file to set the configurations.

1. For DBS server(s) (usually working with Oracle database)

Please refer to .

## Multi\_Queue Configuration

For Multi-Queue support, to create a new queue, the following steps should be applied both at MySQL & Oracle database as below:

### Create a new queue;

● Oracle

1. Create queue table using the Oracle Queue administration interface

DBMS\_AQADM.CREATE\_QUEUE\_TABLE ()

1. Create queue using the Oracle Queue administration interface

SYS.DBMS\_AQADM.CREATE\_QUEUE ()

For details, please ask the DBA to assist.

● MySQL

Use the below command to create a new queue, *test\_data\_queue\_1* for example:

*call prc\_create\_queue(‘test\_data\_queue\_1’, 0, 0, 10);*

### Create a payload type for the new created queue;

● Oracle

Use the Oracle SQL statement to create a new payload type for the new created Queue.

CREATE OR REPLACE TYPE “TEST\_DATA\_QUEUE\_PAYLOAD\_TYPE”

COLUMN\_1 VARCHAR2 (200),

COLUMN\_2 VARCHAR2 (2000) )

● MySQL

For MySQL, the payload type is specified during enqueue message. For the payload format, please refer to step 4 for enqueuing message.

### Start up the queue;

● Oracle

Use the below Oracle AQ administration interface to start up the queue which new created.

SYS.DBMS\_AQADM.START\_QUEUE

● MySQL

Currently, MySQL database no operation need to perform.

### Enqueue message to the queue;

● Oracle

Use the below Oracle AQ interface to enqueue message to queue

dbms\_aq.enqueue(

queue\_name => queue\_name,

enqueue\_options => v\_enqueue\_options,

message\_properties => v\_message\_properties,

payload => v\_payload,

msgid => v\_enq\_msgid

);

For more details, please ask DBA for assistance.

● MySQL

Use the below MySQL AQ interface to enqueue message to queue

PROCEDURE prc\_enqueue(

IN p\_queue VARCHAR(80),

IN p\_payload VARCHAR(8050),

IN p\_msgid VARCHAR(38),

IN p\_sub\_cnt TINYINT(1),

IN p\_recipient VARCHAR(4000) )

**p\_queue(IN)**

the name of the queue which the message will be enqueue to.

**p\_payload(IN)**

the data buffer of the message. For object payload type, the format of the payload data as below: size of attribute one + attribute name=vaule...

*5var1=test15var2=test2*

**p\_msgid(IN)**

the sender message id of the received message. The message can be received from remote location and the message id is generated by remote, however, if the message is generated by local, then leave this parameter to empty.

**p\_sub\_cnt(IN)**

the tag to indicate the message is for local, or need propagate to other consumers by using the built-in propagation mechnism or consums directly by other consumers.

0 – the message is for local; !0 -- the message need propagate to others.

**p\_recipient(IN)**

the recipient lists of the message. One message can be consumed by one or more than one consumers. For message only need for local dequeue in which case leave the address to empty; for message need to propagate to recipients in which case the address should set. The rule of the recipient lists as below:

for only one recipient: *name,address*

for more than one recipients*: name1,address1;name2,address2;....nameN,addressN*

### Dequeue job for dequeue message from the queue.

For dequeue job, please ask the DBA for assistance. For performance issue, currently the Oracle server side applies directly the message to target tables. This option can be set in table: AQ\_SYS\_QUEUE\_M. column name: ENQ\_MODE; the default value is using AQ\_API.

1 – Stands for using AQ\_API and dequeue by dequeue job;

2 – Stands for using SQL procedure to apply the message to target tables directly;

3 – Stands for using prepared binding parameters to apply the message one by one;

4 – Stands for using prepared binding parameters to apply the message by batch.

## Failover Configuration

For failover support, each station will be running two SynchEngine for both database nodes: one with control mode and another with monitor mode.

1. The address of recipient in table recipient\_info should change into the following format for example:

192.168.0.1:5001|192.168.0.2:5001

2. The start up of SynchEngine adds more parameters to indicate the running mode shown below:

For control mode:

SynchEngine --SynID=1 --default-file=c\_cfg.cnf --mode=control &

For monitor mode:

SynchEngine --SynID=1 --default-file=c\_cfg.cnf --mode=monitor &

3. The stop of the SynchEngine command shown below:

To stop SynchEngine:

SynchEngine --SynID=1 --default-file=c\_cfg.cnf --mode=stop

## Load balance configuration

Currenty, the load balance is based on the configuration in stations database. We use the order position for the connection priority. The following is the setting of recipient\_info table when two SynchEngine servers and two clients to connect to each other respectively.

Station 1 (TRA\_DT17)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| pkey | name | hostname | address | db\_link\_name | status | db\_type |
| 1 | TRANSACT | occ-cms-01 | 192.168.0.3:2005|192.168.0.4:2005 |  | 0 | oracle |

Then tra\_dt17 will first connect to the SynchEngine which its address is: 192.168.0.3:2005

Station 2 (TRA\_DT18)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| pkey | name | hostname | address | db\_link\_name | status | db\_type |
| 1 | TRANSACT | occ-cms-01 | 192.168.0.4:2005|192.168.0.3:2005 |  | 0 | oracle |

Then tra\_dt18 will first connect to the SynchEngine which its address is: 192.168.0.4:2005

# Station server installation guideline

For stations, the SynchEngine should be deployed on the local database servers.

## File path structure & rights

* The owner of below folders and files in the folders should be set to user ‘mysql’.

XXXX/datasyn/bin (config file and executable components)

XXXX/datasyn/log (log folder)

Details show below table:

|  |  |
| --- | --- |
| Folder | components |
| /u01/app/datasyn/bin | SynchEngine  c\_cfg.cnf (Rename dbsynconfig.cnf to c\_cfg.cnf) |
| /u01/app/datasyn/log | All the SynchEngine logs will be generated here. |

* XXXX default value is set to ‘/u01/app/’,
* All files in the bin folder and the folder should be assigned ACL mask 755(owner with all rights, group and other with read and execute rights) by command ‘chmod 755 \*’.
* The log folder should be assigned ACL mask 777 by command ‘chmod 777’Environment settings

## Environment settings

### Library path to LD\_LIBRARY\_PATH

Environment variable:  add **the library path of ISCS** to LD\_LIBRARY\_PATH into ‘mysql’ and ‘transactive’ user

For example: LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/u01/app/transactive/lib

## DataSync Configuration settings

### Configuration file

#### c\_cfg.cnf

Rename dbsynconfig.cnf to c\_cfg.cnf if c\_cfg.cnf cannot be found.

Config section [dbsyn\_srcDb] and section [Log],

[dbsyn\_srcDb]

SrcDbConStr = **MySQL**: ***DatabaseName****:UserName:Password:NativeDBIP*

*; NativeDBIP has two possible values, station DB server 1 or station DB server 2.*

*; The SyncEngine should be deployed on all station DB server, the NativeDBIP is*

*; the machine’s IP which is deployed the SyncEngine.*

; SkipErrors, **no need to change it**.

[Log]

LogLevel = INFO

LogFileSize = 20000000

LogFileCount = 50

LogFileName = /u01/app/datasyn/log/c\_SynchEngine.log

DebugPidFilenames = 1

for example:

[dbsyn\_srcDb]

SrcDbConStr = MySQL:TRA\_DT17:IT271350\_5:IT271350\_5:192.168.0.1

[Log]

LogLevel = INFO

LogFileSize = 20000000

LogFileCount = 50

LogFileName = /u01/app/datasyn/log/c\_SynchEngine.log

DebugPidFilenames = 1

#### g\_datasyn\_synid

If SynchEngine needs be deployed in one machine with two different instances, the g\_datasyn\_synid should be different. It must not deploy two instances of SynchEngine in one station server for the database in the station server.

## Start and Stop

For start and stop SynchEngine on stations, please refer to Section *2.2.5*, Doc: *C955-Solaris\_10\_MySQL\_SSReplication\_Setup.doc*

# DBS installation guideline

For DBS (TRANSACT), the SynchEngine should be deployed on the central Oracle database servers.

## File path structure & rights

### Path structure & rights

* The owner of below folders and files in the folders should be set to user ‘oracle’.

XXXX/datasyn/bin (monitor scripts and executable components)

XXXX/datasyn/lib (dependant libraries)

XXXX/datasyn/log (log folder)

* XXXX default value is set to ‘/u01/app/’,
* All files in the bin and lib folder and the two folders should be assigned ACL mask 755(owner with all rights, group and other with read and execute rights) by command ‘chmod 755 \*’.
* The log folder should be assigned ACL mask 777 by command ‘chmod 777’.

### Folder and components

We should get the relevant components from build baseline, according to DBS CPU model (SPARC or Intel).

In C955, the DBS CPU model is SPARC.

|  |  |
| --- | --- |
| **Folder** | **Components** |
| /u01/app/datasyn/bin | SynchEngine  HA-DBSyn-Main.bsh  HA-DBSyn-Ora.sh  s\_cfg.cnf (Rename dbsynconfig.cnf to s\_cfg.cnf) |
| /u01/app/datasyn/log | All the SynchEngine logs will be generated here. |
| /u01/app/datasyn/lib | libACE.so  libACE.so.6.0.4  libCOS4.so  libCOS4.so.1  libCOS4.so.1.6  libTA\_mcl.so  libTA\_mcl.so.5  libTA\_mcl.so.5.1.1.0  libboost\_filesystem-gcc34-mt-1\_39.so  libboost\_filesystem-gcc34-mt-1\_39.so.1.39.0  libboost\_regex-gcc34-mt-1\_39.so  libboost\_regex-gcc34-mt-1\_39.so.1.39.0  libboost\_system-gcc34-mt-1\_39.so  libboost\_system-gcc34-mt-1\_39.so.1.39.0  libclntsh.so.10.1  libcrypto.so.1.0.0  libgcc\_s.so.1  libmysqlclient.so.18  libnnz10.so  libocilib.so  libocilib.so.3  libocilib.so.3.9.3  libomniDynamic4.so  libomniDynamic4.so.1  libomniDynamic4.so.1.6  libomniORB4.so  libomniORB4.so.1  libomniORB4.so.1.6  libomnithread.so  libomnithread.so.3  libomnithread.so.3.4  libstdc++.so.6 |

## Environment settings

### Library path to LD\_LIBRARY\_PATH

Environment variable:  add the library path to LD\_LIBRARY\_PATH into ‘oracle’ user

For example: LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/u01/app/datasyn/lib

## DataSync Configuration settings

### Configuration file

#### s\_cfg.cnf

Rename dbsynconfig.cnf to s\_cfg.cnf if s\_cfg.cnf cannot be found.

Config section [dbsyn\_srcDb] and section [Log],

[dbsyn\_srcDb]

SrcDbConStr = **Oracle**:*ServiceName:UserName:Password*

; SkipErrors, **no need to change it**.

[Log]

LogLevel = INFO

LogFileSize = 20000000

LogFileCount = 50

LogFileName =/u01/app/datasyn/log/s\_SynchEngine.log

DebugPidFilenames = 1

for example:

[dbsyn\_srcDb]

SrcDbConStr = Oracle:TRANSACT:IT271350\_5:IT271350\_5

[Log]

LogLevel = INFO

LogFileSize = 20000000

LogFileCount = 50

LogFileName = =/u01/app/datasyn/log/s\_SynchEngine.log

DebugPidFilenames = 1

#### g\_datasyn\_synid

If SynchEngine needs be deployed in one machine with two different instances, the g\_datasyn\_synid should be different. It can deploy two instances of SynchEngine in DBS, the g\_datasyn\_synid of the two instances should be different, and it is better to deploy each instance in different folder and set the g\_datasyn\_syndid to different value.

## Start and stop

For start and stop SynchEngine on DBS central database, please refer to Section *2.2.5*, Doc: *C955-Solaris\_10\_MySQL\_SSReplication\_Setup.doc*

# Appendix A Queue Tables

## Queue tables

The design of *audit\_data\_queue\_t* as below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Data Length | Nullable | Default value | Comment |
| Q\_NAME | VARCHAR | 30Bytes | Y | NULL | The name of the queue. |
| ENQ\_HID | INT | 4Bytes | N | 0 | The high bit of message id. |
| ENQ\_LID | INT | 4Bytes | N | 0 | The low bit of message id. |
| MSGID | VARCHAR | 36Bytes | N | UUID() | The unique ID of the message. |
| PRIORITY | TINYINT | 1Bytes | Y |  | The priority of the message. |
| STATE | TINYINT | 1Bytes | N | 0 | 1 – ready; 2 – waiting;  3 – processed; 4 - expired |
| SUB\_CNT | TINYINT | 1Bytes | N | 0 | 0 – local; 1 – central; 2 – others |
| EXPIRATION | DATETIME |  | Y |  | The message expiration time. |
| ENQ\_TIME | DATETIME |  | N | sysdate() | The enqueue time of the message. |
| ENQ\_UID | VARCHAR | 30Bytes | Y |  | The enqueue user of the message. |
| DEQ\_TIME | DATETIME |  | Y |  | The dequeue time of the message. |
| RETRY\_COUNT | INT | 4Bytes | Y |  | The retry count of the message. |
| SENDER\_MSGID | VARCHAR | 36Bytes | Y |  | The original ID of the message. |
| SENDER\_NAME | VARCHAR | 30Bytes | Y |  | The sender name of the message. |
| SENDER\_ADDR | VARCHAR | 20Bytes | Y |  | The address of the sender. |
| USER\_DATA\_1 | VARCHAR | 4000Bytes | Y |  | The message data. |
| USER\_DATA\_2 | VARCHAR | 4000Bytes | Y |  | The Oracle message data |
| USER\_DATA\_3 | VARCHAR | 4000Bytes | Y |  | The MySQL message data |

Table: audit\_data\_queue\_t