Altibase Tools & Utilities

Audit User's Manual

Release 6.1.1

April 24, 2012

ALTIBASE

PERFORMANCE SOLUTIONS

Altibase Tools & Utilities Audit User’s Manual

Release 6.1.1

Altibase Corporation
10F, Daerung PostTower II, 182-13,
Guro-dong Guro-gu Seoul, 152-847, Korea
Telephone: +82-2-2082-1000 Fax: 82-2-2082-1099
E-mail: support@altibase.com www: http://www.altibase.com

Contents

Preface

About This Manual

This manual explains how to compare and check databases on replicated servers on a table-by-table basis and reconcile any inconsistencies that may be found between them.

Intended Audience

The following users of ALTIBASE® HDB™ will find this manual useful:

- Database administrators
- Database users
- Technical support workers

It is recommended that those reading this manual possess an understanding of the following background knowledge:

- Basic knowledge in the use of computers, operating systems, and operating system utilities
- Experience in using relational databases and an understanding of database concepts
- Computer programming experience
- Experience in database server, operating system or network administration

Software Environment

This manual has been written assuming that ALTIBASE HDB 5.5.1 is used as the database server.

How This Manual is Structured

This manual has been organized as follows:

- Chapter1: Overview

  This manual explains the functions for comparing databases on replication servers on a table-by-table basis and reconciling any inconsistencies that may be found between them.

- Chapter2: Resolving Data Inconsistencies

  This chapter describes the Audit environment file, which contains information for executing Audit, and then describes the comparison (DIFF) and synchronization (SYNC) functions.

- Chapter3: Examples

  This chapter describes examples of the use of the comparison (DIFF) and synchronization (SYNC) features.

References

For more detailed information, please refer to the following documents.

- ALTIBASE HDB Administrator's Manual
- ALTIBASE HDB Replication Manual
- ALTIBASE HDB iSQL User's Manual
- ALTIBASE HDB Utilities Manual
- ALTIBASE HDB Error Message Reference

Online Manual

Online versions of our manuals (PDF or HTML) are available from the Altibase Download Center (http://atc.altibase.com/).

Altibase Welcomes Your Opinions

Please feel free to send us your comments and suggestions regarding this manual. Your comments and suggestions are important to us, and may be used to improve future versions of the manual. Please send your feedback to support@altibase.com, making sure to include the following information:

- The name and version of the manual that you are using
- Your comments and suggestions regarding the manual
- Your full name, address, and phone number

In addition to suggestions, this address may also be used to report any errors or omissions discovered in the manual, which we will address promptly. If you need immediate assistance with technical issues, please contact the Altibase Customer Support Center.

We always appreciate your comments and suggestions.

1 Overview

This Chapter describes Audit and the consistency control methods for ALTIBASE HDB Replication Servers.

1.1 Introduction

The intended use of the ALTIBASE HDB Audit utility is as a means of monitoring the progress of replication between two Altibase databases and resolving data inconsistencies that arise during the course of replication.

Audit compares an Altibase database with either an Altibase database or an Oracle database on a table-by-table basis, and outputs information about any inconsistencies it finds. It also includes functionality for synchronizing two databases in the event that data inconsistencies are discovered.

Note: The Windows version of the ALTIBASE HDB server does not include the Audit utility.

1.1.1 Audit Terminology

1.1.1.1 Master Server

This is the server whose contents are accepted as correct in the event that a record is found to be inconsistent between two servers. Either server can be designated as the master server when Audit is executed.

1.1.1.2 Master DB

The database on the master server.

1.1.1.3 Slave Server

This is the server whose contents are updated with the contents of the other server in the event that a record is found to be inconsistent between two servers. Either server can be designated as the slave server when Audit is executed.

1.1.1.4 Slave DB

The database on the slave server.

1.1.2 Inconsistent Records

An inconsistent record is a record in which a disagreement between column values is found when a designated table in the Master DB is compared with the corresponding table in the Slave DB on the basis of a primary key. There are three types of inconsistency:

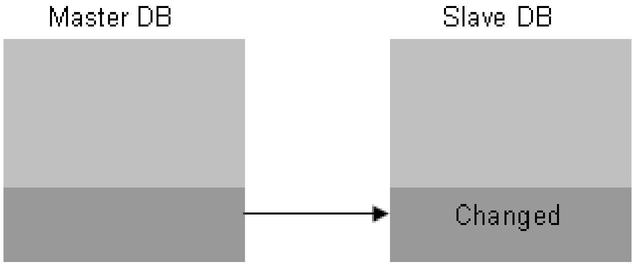
- MOSX Inconsistency: when a record based on a primary key can be found in the Master DB but not in the Slave DB.
- MOSO Inconsistency: when a record based on a primary key can be found in both the master and slave tables but the record contents are different.
- MXSO Inconsistency: when a record based on a primary key can be found in the Slave DB but not in the Master DB.

1.1.3 Synchronization Policy

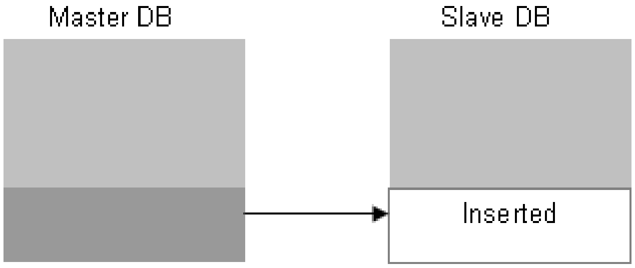
A Synchronization Policy is a policy that specifies how to synchronize inconsistent records. The Audit application usually treats the Master DB as the reference DB and synchronizes the Slave DB with it.

ALTIBASE HDB provides four synchronization policies:

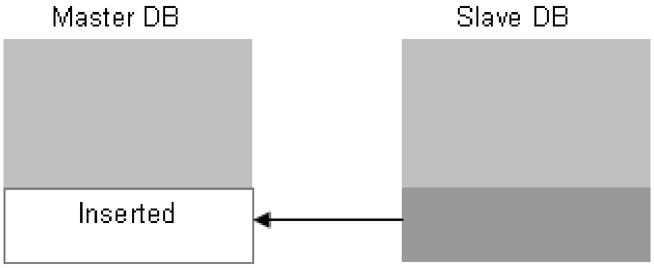
- SU Policy: This policy resolves MOSO inconsistencies by updating the Slave DB with the contents of the Master DB.



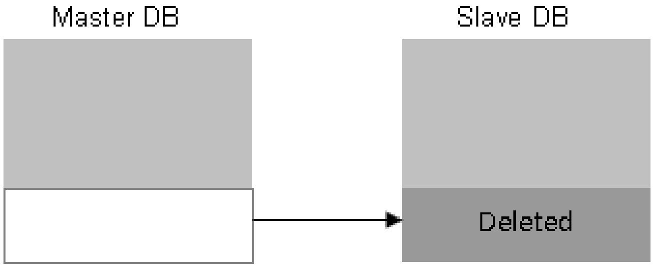
- SI Policy: This policy resolves MOSX inconsistencies by inserting records from the Master DB into the Slave DB.



- MI Policy: This policy resolves MXSO inconsistencies by inserting records from the Slave DB into the Master DB.



- SD Policy: This policy resolves MXSO inconsistencies by deleting records from the Slave DB.



The SU policy, SI policy, MI policy, and SD policy are set in the Audit environment file. Note that the MI policy and the SD policy are mutually exclusive, meaning that they cannot both be enabled at the same time.

1.1.3.1 DIFF

Creates an execution result file that identifies inconsistent records found during replication between the Master DB and the Slave DB.

1.1.3.2 SYNC

Identifies inconsistent records between the Master DB and the Slave DB, bidirectionally resolves the inconsistencies according to the synchronization policy set in the Audit environment file, and creates an execution result file including execution summary information and error information.

1.1.3.3 Audit Environment File

An environment file for setting options for Audit execution. This file includes connection information, Audit function settings, synchronization policies, and the like.

1.2 Executing the Audit Utility

To use the functions of Audit, an Audit environment file, which contains information about the table(s) on which DIFF or SYNC is to be executed, must first be created. The Audit environment file will be explained in the Audit Environment File section of Chapter 2: Resolving Data Inconsistencies.

Audit commands have the following form:

```
$ audit -f script_file_name
script_file_name : File name including the path of the environment file
or
If current directory is as follows: /user/charlie/altibase_home/audit
/user/charlie/altibase_home/audit> audit script_file_name
(/user/charlie/altibase_home/audit> audit ./script_file_name)
```

1.2 Executing the Audit Utility

2 Resolving Data Inconsistencies

In this chapter, the Audit environment file, which contains information necessary for executing Audit, is first explained, and then the comparison (DIFF) and synchronization (SYNC) features are described.

2.1 Configuring the Audit Utility

Each of the comparison and synchronization tasks, which are described in the environment file, has its own unique properties. The properties provide information necessary for running the Audit utility. (Please refer to the sample.cfg file in the ALTIBASE_HOME/audit directory.)

2.1.1 The Audit Environment File

Properties have the form “property name = property value” and are case-insensitive. The following symbols have special meanings when used in the environment file:

- "#" indicates a comment and causes the remainder of the line to be ignored.
- "{ }" (curly braces) used to indicate that a property value spans multiple lines.
- " ; " (semicolon) serves as a delimiter to separate multiple values.
- ‘ “ ‘ (double quotation marks) are used to enclose a string (such as a user name, password, table name, or column name) that includes one or more reserved words or special characters.

  In ALTIBASE HDB, the following are special characters:

  ```
  ~ ! @ # $ % ^ & * ( ) _ + |
  ```

2.1.1.1 Property Name

A property name consists of characters other than spaces, and identifies a property within a group.

2.1.1.2 Property Value

A property value be a single value, multiple values, or an expression.

- The expression may include blanks. Most properties have this format:

  ```
  e.g. 1) TABLE = EMPLOYEES
  ```

- A multiple value comprises several values separated by the “;” delimiter, and must be contained within “{}” when it occupies more than one line (see Example 2). The “EXCLUDE” group allows multiple values.

  ```
  e.g. 2) EXCLUDE = ENO; DNO; E_FIRSTNAME
  or EXCLUDE = {ENO; DNO; E_FIRSTNAME}
  ```

- Expressions are character strings, can include spaces, and must be enclosed within “{}”. The “WHERE” property is an expression.

  ```
  e.g. 3) WHERE = { ENO > ‘1000’ and ENO < ‘2000’ }
  ```

2.1.1.3 Data Type Support

ALTIBASE HDB does not support the use of CLOB and BLOB with Oracle databases. The EXCLUDE property is used as follows to exclude a particular column or columns from Audit processing.

```
Example 4) You should exclude certain column for AUDIT when CLOB column
exists on EMP table.
TABLE = EMP
EXCLUDE = { CCC }
```

2.1.2 Execution Options

These options specify the information for accessing the local server and the remote server, comparison (DIFF) and synchronization (SYNC) tasks, and synchronization policies governing the handling of inconsistent records. The properties that can be set in this group are as follows:

2.1.2.1 DB_MASTER

This is used to set the server whose contents are to be accepted as correct if inconsistent records are found between two servers. Set the user name and password, the name or IP address of the server, and NLS_USE. The property values must match the information in the property file in the home directory of ALTIBASE HDB.

2.1.2.2 DB_SLAVE

This is used to set the other server. Set the user name and password, the name or IP address of the server, and NLS_USE. The property values must match the information in the property file in the home directory of ALTIBASE HDB. Additionally, a text DB can be specified for the other server. In this case, the following format is used for this property (where "./log" is the directory containing the text DB).

```
DB_SLAVE = text://userID:PW@./log
```

2.1.2.3 OPERATION

This is set to "DIFF" for a comparison task, or to "SYNC" for a synchronization task.

2.1.2.4 INSERT_TO_SLAVE

Sets the SI policy used to resolve MOSX inconsistencies. Specifies whether to insert the record in question into the Slave DB. The property value is set to "ON" to specify that the record is to be inserted, and "OFF" to specify that it is not to be inserted.

2.1.2.5 INSERT_TO_MASTER

Sets the MI policy used to resolve MXSO inconsistencies. Specifies whether to insert the record in question into the Master DB. The property value is set to "ON" to specify that the record is to be inserted, and "OFF" to specify that it is not to be inserted. This property and DELETE_IN_SLAVE cannot both be set to "ON" simultaneously.

2.1.2.6 DELETE_IN_SLAVE

Sets the SD policy used to resolve MXSO inconsistencies. Specifies whether to delete the record in question from the Slave DB. The property value is set to "ON" to specify that the record is to be deleted, and "OFF" to specify that it is not to be deleted. This property and INSERT_TO_MASTER cannot both be set to "ON" simultaneously.

2.1.2.7 UPDATE_TO_SLAVE

Sets the SU policy used to resolve MOSO inconsistencies. Specifies whether to update the record in question in the Slave DB. The property value is set to "ON" to specify that the record is to be changed, and "OFF" to specify that it is not to be changed.

2.1.2.8 AUTODETECT_UNIQ_INX

Specifies whether to delete the record in question from the Slave DB and repeat the insert or update action if a "Duplicate Key Values" error is raised in the Slave DB when inserting or updating data from the Master DB to the Slave DB. This property value may be set to "ON" or "OFF". "ON" signifies that the record is to be deleted, and "OFF" that it is not to be deleted. This option can be set to "ON" only when both the INSERT_TO_SLAVE and DELETE_IN_SLAVE properties are also set to "ON".

2.1.2.9 CHECK_INTERVAL

Sets the interval between the completion of a SYNC operation on a table and the start of a SYNC operation on the next table. Expressed in units of ms (milliseconds).

2.1.2.10 MAX_THREAD

Specifies the maximum number of threads that can run concurrently. Set to -1 to specify an unlimited number of threads.

2.1.2.11 FILE_MODE_MAX_ARRAY

If its value is greater than 1, `audit` writes the fetched data to a file and then starts a SYNC or DIFF operation on the file. This value is used to set the maximum size of array(s) for fetching data. `audit` fetches a number of records equal to this value and writes them to a csv file. This option can be used to realize better performance. However, when a target table has many LOB type columns, this option may not improve performance. This option can only be used between Altibase databases.

Example

```
FILE_MODE_MAX_ARRAY = 1000
```

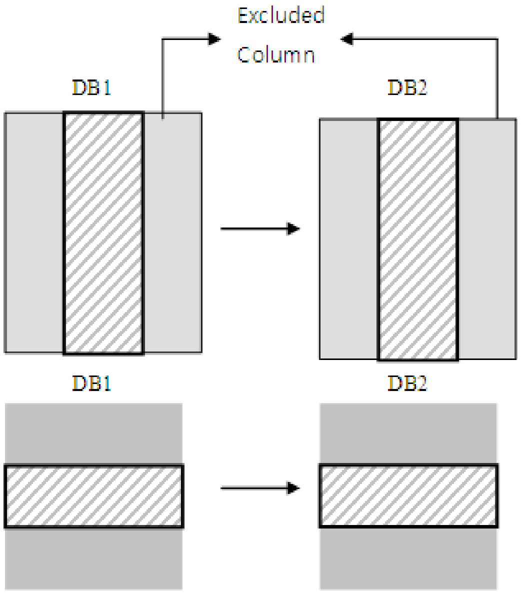
2.1.3 Table Group

For defining information related to target table(s). The number of descriptions in the group must be as many as the number of target tables, and the name of each group must correspond to the name of a table in the Master DB. The following properties can be set:

2.1.3.1 WHERE

Used to set conditions for selecting table records. This property is described in the same way as a WHERE clause of a SQL statement. Multiple values are permitted, but the ";" delimiter cannot be used to specify multiple values. Moreover, this property cannot be commented.

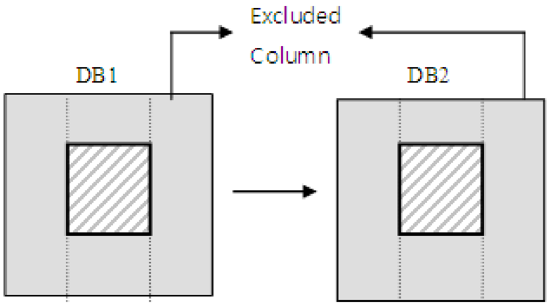
Applies to the comparison (DIFF) and synchronization (SYNC) functions.



2.1.3.2 EXCLUDE

For setting conditions for the projection of table records. The property may have multiple values. The specified columns are excluded from comparison and synchronization operations.

Applies to comparison (DIFF) and synchronization (SYNC) functions. By suitably combining WHERE and EXCLUDE properties, the result of a combined selection and projection operation can be obtained, and an auditing task can be conducted on the result.



2.1.3.3 TABLE

For setting the Slave DB table name. In cases where the table names on the Master DB and the Slave DB differ from each other, this must be explicitly described in order to use the comparison (DIFF) and synchronization (SYNC) functions.

When omitted, it is assumed that the table name on the Slave DB is the same as that on the Master DB. The table name can contain Roman alphabetic characters, numbers, and the following special characters:

space ~ ! @ # $ % ^ & * ( ) _ + |

However, it cannot contain Korean characters.

2.1.3.4 SCHEMA

This is specified for Audit to check table schema in Slave database. Table schema in ALTIBASE HDB contains information of user's account. If the schema of a user accessing the Slave database differs from the target table’s schema, this must be explicitly stated. If this is not stated, the error "Table not found" may occur when you try to execute Audit. If this property is omitted, the schema of the user accessing the Slave database will be used.

2.2 Comparison (DIFF) Function

This function identifies inconsistent records that are found during replication between the Master DB and the Slave DB, and creates an execution result file.

2.2.1 Setting the DIFF Option in the Environment File

In the Audit environment file, set the OPERATION property to "DIFF". All execution option properties must be specified, and the table group properties WHERE, EXCLUDE, TABLE, and SCHEMA can be optionally specified.

2.2.2 Execution

The comparison (DIFF) function is executed as follows:

```
$ audit -f script_file_name
```

script_file_name : File name including the path of the environment file

2.2.3 Execution Results

This function compares the contents of the Master Database and the Slave Database for each execution log file, which includes execution result summary information, and each table, and creates an execution result file that includes the contents of the inconsistent columns of the records in which inconsistencies were found. For example, when running an Audit command such as:

```
/user/charlie/altibase_home/audit> audit sample.cfg
```

if the Audit command is executed successfully, a file containing table information and named "mastertable-username.slavetable.log" is created in the audit directory for each table, alongside sample.log.

2.2.3.1 Execution log file

This file is created with the name "script_file_name.log" and contains the contents of the executed environment file along with a summary of the comparison (DIFF) task for each table in the TABLES group. The contents of the environment file are displayed as follows:

```
INFO[ MNG ] Tread # 0 init is OK!
INFO[ MNG ] Tread # 0 start is OK!
[TAB_2->TAB_2]
Fetch Rec In Master: 3
Fetch Rec In Slave : 2
MOSX = DF, Count : 1
MXSO = DF, Count : 0
MOSO = DF, Count : 1
 SCAN TPS: 20547.95
 Time: 0.00 sec
```

2.2.3.2 Execution Result File

This file is created with the name “mastertable username.slavetable.log” and displays the comparison results in the following format.

```
DF[m,n]-> COL_N (Vn_M, Vn_S):PK->{ PCOL_V }
```

- DF : the type of inconsistency (MOSX, MOSO, MXSO)
- m : the record number on the Master server
- n : the record number on the Slave server
- COL_N : the name of the first column found to contain differing values as the result of comparison
- Vn_M : the value in the corresponding column on the Master server
- Vn_S : the value in the corresponding column on the Slave Server

However, for records that have LOB type columns, the LOB column value is not output.

2.3 Synchronization (SYNC) Function

This function identifies records that are inconsistent between the Master DB and the Slave DB, bidirectionally resolves the inconsistencies according to the synchronization policy in the Audit configuration file, and creates an execution result file including execution summary information and error information.

2.3.1 Setting the SYNC Option in the Environment File

In the Audit environment file, set the OPERATION property to "SYNC". All execution option properties must be described, and the table group properties WHERE, EXCLUDE, TABLE, and SCHEMA can be optionally specified.

2.3.2 Execution

The synchronization (SYNC) function is executed as follows:

```
$ audit -f script_file_name
```

script_file_name : File name including the path of the configuration file

2.3.3 Execution Results

This function compares the Master DB with the Slave DB for each execution log file, which includes a summary of execution results, and each table, and outputs an execution result file that comprises information about synchronization tasks conducted on inconsistent records and an error log that includes information about the errors that occurred during execution.

2.3.3.1 Execution Log File

This file is created with the name "script_file_name.log" and displays the contents of the executed environment file as well as the summary of the synchronization (SYNC) task for the table(s) in each TABLES group. The contents of the environment file are written to the log file as follows:

```
INFO[ MNG ] Tread # 0 init is OK!
INFO[ MNG ] Tread # 0 start is OK!
[TAB_2->TAB_2]
Fetch Rec In Master: 3
Fetch Rec In Slave : 2
MOSX = -, SI
MXSO = -, -
MOSO = -, SU
MXSX = -, -

--------------------------------------------
 Operation      Type     MASTER      SLAVE
--------------------------------------------
 INSERT         Try      0           1
                Fail     0           0

 UPDATE         Try      X           1
                Fail     X           0
 DELETE         Try      X           0
                Fail     X           0
--------------------------------------------
 UPDATE         Try      0           2
                Fail     0           0
 OOP TPS:  13698.63
 SCAN TPS: 20547.95
     Time: 0.00 sec
```

If a failure occurs for any record, the cause of the error and the record contents are written to the log file.

3 Examples

This chapter describes examples of the use of the comparison (DIFF) and synchronization (SYNC) functions.

3.1 Comparison (DIFF) Example

In the following example, the EMP table on host1 is compared with the EMPLOYEES table on host2, and the DEPARTMENTS table on host1 is compared with the DEPARTMENTS table on host2.

3.1.1 DIFF Example 1

Assuming that DB_MASTER is set to host1, DB_SLAVE is set to host2, and that all records in each table are to be compared, the environment file would appear as follows:

```
DB_MASTER = “altibase://sys:man-
ager@DSN=host1;PORT_NO=10111;NLS_USE=US7ASCII”
DB_SLAVE = “altibase://sys:manager@DSN=host2;PORT_NO=20111;NLS_USE=US7ASCII”
OPERATION = DIFF
MAX_THREAD = -1

DELETE_IN_SLAVE = ON
INSERT_TO_SLAVE = ON
INSERT_TO_MASTER = ON
UPDATE_TO_SLAVE = ON
AUTODETECT_UNIQ_INX = ON

LOG_DIR = "./"
LOG_FILE = "sample.log"

[EMP]
TABLE = EMPLOYEES
SCHEMA = SYS

[DEPARTMENTS]
TABLE = DEPARTMENTS
SCHEMA = SYS
```

As can be seen in the above example, the target tables in the master server (host1) and the slave server (host2) can have different names.

3.1.2 DIFF Example 2

In the following example, the ENO column of the EMP table is used as the basis for record selection, and the JOIN_DATE and SEX columns are excluded from the comparison. According to the value of the CONDITION property, the records in EMP that will be compared are limited to those for which “ENO” is greater than or equal to 1 and less than or equal to 20. In addition, according to the value of the EXCLUDE property, JOIN_DATE and SEX will be excluded from the comparison. In other words, if only JOIN_DATE and SEX differ, as long as the other columns are the same, the records will be handled as though they were the same.

```
[EMP]
TABLE = EMPLOYEES
WHERE = {ENO >= 1 and ENO <= 20}
EXCLUDE = {JOIN_DATE; SEX}
[DEPARTMENTS]
```

3.1.3 DIFF Example 3

In this example, records are selected based on the ENO and JOIN_DATE columns of the EMP table, and the SEX column is excluded from the comparison:

```
[EMP]
TABLE = EMPLOYEES
WHERE = {(ENO >= 1 and ENO <= 20) or (JOIN_DATE >= '20001010')}
EXCLUDE = {SEX}
[DEPARTMENTS]
```

According to the WHERE property, the EMP records that will be compared are limited to those for which ENO is greater than or equal to 1 and less than or equal to 20, and for which JOIN_DATE is on or after October 10, 2000. Also, according to the EXCLUDE property, SEX will be excluded from the comparison.

3.2 Synchronization (SYNC) Example

An example that uses OPERATION and TABLE in a synchronization policy to resolve inconsistent records will now be presented.

3.2.1 SYNC Example 1

Suppose that records with MOSX inconsistencies (i.e. that exist in the master server but not in the slave server) are to be inserted into the slave server, and that records with MXSO inconsistencies (i.e. that exist in the slave server but not in the master server) are to be ignored.

```
Master Server = “altibase://sys:man-
ager@DSN=host1;PORT_NO=10111;NLS_USE=US7ASCII”

Slave Server = “altibase://sys:man-
ager@DSN=host2;PORT_NO=20111;NLS_USE=US7ASCII”

OPERATION = SYNC
MAX_THREAD = -1

DELETE_IN_SLAVE = OFF
INSERT_TO_SLAVE = ON
INSERT_TO_MASTER = OFF
UPDATE_TO_SLAVE = ON
AUTODETECT_UNIQ_INX = ON

LOG_DIR = "./"
LOG_FILE = "sample.log"

[EMP]
TABLE = EMPLOYEES
SCHEMA = SYS

[DEPARTMENTS]
TABLE = DEPARTMENTS
SCHEMA = SYS
```

As the SI policy is the synchronization policy used to resolve MOSX inconsistencies, the INSERT_TO_SLAVE property value has been set to “ON”. In contrast, because MXSO inconsistencies are to be ignored, the INSERT_TO_MASTER and DELETE_IN_SLAVE properties, which are related to the MI and SD policies, have been set to “OFF”.

3.2.2 SYNC Example2

Suppose that records with MOSX inconsistencies (i.e. that exist in the master server but not in the slave server) are to be inserted into the slave server and that records with MXSO inconsistencies (i.e. that exist in the slave server but not in the master server) are to be inserted into the master server.

```
Master Server = altibase://sys:man-
ager@DSN=host1;PORT_NO=10111;NLS_USE=US7ASCII
Slave Server = altibase://sys:man-
ager@DSN=host2;PORT_NO=20111;NLS_USE=US7ASCII
OPERATION = SYNC
MAX_THREAD = -1

DELETE_IN_SLAVE = OFF
INSERT_TO_SLAVE = ON
```

```
INSERT_TO_MASTER = ON
UPDATE_TO_SLAVE = ON
AUTODETECT_UNIQ_INX = ON

LOG_DIR = "./"
LOG_FILE = "sample.log"

[EMP]
TABLE = EMPLOYEES
SCHEMA = SYS

[DEPARTMENTS]
TABLE = DEPARTMENTS
SCHEMA = SYS
```

As the SI policy is the synchronization policy used to remove inconsistent MOSX records, the INSERT_TO_SLAVE property has been set to "ON". Moreover, because records with MXSO inconsistencies must be inserted into the master server, the MI policy is necessary. Therefore, the INSERT_TO_MASTER property, which is related to the MI policy, is set to "ON", and the DELETE_IN_SLAVE property is set to "OFF".

3.2.3 SYNC Example 3

Suppose that the slave server is to be synchronized with the master server.

```
Master Server = altibase://sys:man-
ager@DSN=host1;PORT_NO=10111;NLS_USE=US7ASCII
Slave Server = altibase://sys:man-
ager@DSN=host2;PORT_NO=20111;NLS_USE=US7ASCII
OPERATION = SYNC
MAX_THREAD = -1

DELETE_IN_SLAVE = ON
INSERT_TO_SLAVE = ON
INSERT_TO_MASTER = OFF
UPDATE_TO_SLAVE = ON
AUTODETECT_UNIQ_INX = ON

LOG_DIR = "./"
LOG_FILE = "sample.log"

[EMP]
TABLE = EMPLOYEES
SCHEMA = SYS

[DEPARTMENTS]
TABLE = DEPARTMENTS
SCHEMA = SYS
```

To synchronize the slave server with the master server, the SI and SD synchronization policies are necessary. Therefore, the INSERT_TO_SLAVE and DELETE_IN_SLAVE properties are set to "ON".

3.2.4 Example

This is a simple example that compares the *employees* table on the local server, host1, with the *employees* table (from which records having ENOs from 16 to 20 have been deleted) on the remote server, host2, and synchronizes (SYNC) the *departments* table on host1 with the *departments* table on host2 (please refer to "schema.sql" in $ALTIBASE_HOME/sample/APRE/schema directory). First, a replication connection is established between the local server and the remote server.

On the local server (IP: 192.168.1.11)

```
iSQL> CREATE REPLICATION rep1 WITH '127.0.0.1', 56342 FROM sys.employees TO
sys.employees, FROM sys.departments TO sys.departments;
Create Success
iSQL>
```

On the remote server (IP: 127.0.0.1)

```
iSQL> CREATE REPLICATION rep1 WITH '192.168.1.11', 65432 FROM sys.employees
TO sys.employees, FROM sys.departments TO sys.departments;
Create Success
iSQL>
```

Here, the current directory is:

/user/charlie/altibase_home/audit

```
$ vi sample.cfg
Master Server = altibase://sys:man-
ager@DSN=127.0.0.1;PORT_NO=20582;NLS_USE=US7ASCII
Slave Server = altibase://sys:man-
ager@DSN=192.168.1.11;PORT_NO=20582;NLS_USE=US7ASCII

OPERATION = SYNC
MAX_THREAD = -1

DELETE_IN_SLAVE = ON
INSERT_TO_SLAVE = ON
INSERT_TO_MASTER = OFF
UPDATE_TO_SLAVE = ON
AUTODETECT_UNIQ_INX = ON

LOG_DIR = "./"
LOG_FILE = "sample.log"

[ EMPLOYEES ]
WHERE = {ENO >= 1 and ENO <= 20}
TABLE = EMPLOYEES
SCHEMA = SYS

[ DEPARTMENTS ]
TABLE = DEPARTMENTS
SCHEMA = SYS
$ audit -f sample.cfg
$ cat sample.log
INFO[ MNG ] Tread # 0 init is OK!
INFO[ MNG ] Tread # 1 init is OK!
INFO[ MNG ] Tread # 0 start is OK!
INFO[ MNG ] Tread # 1 start is OK!

[DEPARTMENTS->DEPARTMENTS]
Fetch Rec In Master: 5
Fetch Rec In Slave : 5
MOSX = NO
MXSO = NO
MOSO = SU

--------------------------------------------
 Operation     Type      MASTER      SLAVE
--------------------------------------------
 INSERT        Try       0           0
```

```
               Fail      0           0
 UPDATE        Try       X           0
               Fail      X           0
 DELETE        Try       X           0
               Fail      X           0
--------------------------------------------
 UPDATE        Try       0           0
               Fail      0           0
 OOP TPS: 0.00
 SCAN TPS: 60240.96
 Time: 0.00 sec

[EMPLOYEES->EMPLOYEES]
Fetch Rec In Master: 20
Fetch Rec In Slave : 15
MOSX = NO
MXSO = NO
MOSO = SU

-------------------------------------------
 Operation    Type       MASTER      SLAVE
-------------------------------------------
 INSERT       Try        0           5
              Fail       0           0
 UPDATE       Try        X           0
              Fail       X           0
 DELETE       Try        X           0
              Fail       X           0
-------------------------------------------
 UPDATE       Try        0           5
              Fail       0           0
 OOP TPS: 576.04
 SCAN TPS: 2304.15
 Time: 0.01 sec
```

3.2 Synchronization (SYNC) Example

Index