

ALTIBASE® HDB™

Disaster Recovery Manual

Release 6.5.1

May 28, 2015



ALTIBASE HDB Disaster Recovery Manual

Release 6.5.1

Copyright © 2001~2015 Altibase Corporation. All rights reserved.

This document contains proprietary information of Altibase® Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright patent and other intellectual property law. Reverse engineering of the software is prohibited.

All trademarks, registered or otherwise, are the property of their respective owners.

Altibase Corporation

10F, Daerung PostTower II, 182-13,

Guro-dong Guro-gu Seoul, 152-847, South Korea

Telephone: +82-2-2082-1000 Fax: 82-2-2082-1099

Homepage: <http://www.altibase.com>

Contents

Preface	v
About This Manual	vi
Audience.....	vi
Software Environment.....	vi
Organization.....	vi
Documentation Conventions	vii
Related Documents	ix
Online Manuals	ix
Altibase Welcomes Your Opinions.....	ix
1. Disaster Recovery Overview	1
1.1 Introduction.....	2
1.1.1 Concepts	2
1.1.2 Terminology.....	2
1.1.3 Data Synchronization.....	4
1.1.4 Failover	5
1.1.5 Failure Server Repair	5
2. Managing DR	7
2.1 Steps of the DR Procedure	8
2.2 Troubleshooting	9
2.2.1 Abnormal Shutdown of the Active Server	9
2.2.2 Interruption of Communication Between the Active Server and the Standby Server	10
2.2.3 Network Failure	11
2.3 Related Performance Views	12
3. Deploying DR.....	13
3.1 Considerations.....	14
3.1.1 System Requirements	14
3.1.2 Data Requirements.....	14
3.1.3 Configuration Requirements.....	14
3.2 Configuring DR.....	15
3.2.1 Syntax.....	15
3.2.2 Description	15
3.2.3 Error Codes.....	16
3.2.4 Examples	16
3.3 Starting, Stopping and Modifying DR.....	18
3.3.1 Syntax.....	18
3.3.2 Description	18
3.3.3 Error Code.....	19
3.3.4 Example.....	19
3.4 Disabling Disaster Recovery	21
3.4.1 Syntax.....	21
3.4.2 Description	21
3.4.3 Error Codes.....	21
3.4.4 Example.....	21
3.5 DR-related Properties.....	22
4. Failover	23
4.1 Failover Overview.....	24
4.1.1 Concept	24
4.2 Using Failover	25
4.2.1 Auto-failover.....	25
4.2.2 Manual Failover	25
5. Failure Server Repair	27
5.1 What is Failure Server Repair?.....	28
5.2 Full Sync	29
5.2.1 Performing Full Sync.....	29
5.2.2 Examples	29
5.3 Partial Sync.....	31

5.3.1 Example 32

Preface

About This Manual

This manual describes the disaster recovery feature in ALTIBASE HDB.

Audience

This manual has been prepared for the following ALTIBASE HDB users:

- Database administrators
- Technical support staff

It is recommended that those reading this manual possess 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 server management, operating systems or network management
- Knowledge in data storage, management and processing in distributed environments

Software Environment

This manual has been prepared assuming that ALTIBASE HDB 6.5.1 is used as the database server.

Organization

This manual has been organized as follows:

- [Chapter1: Disaster Recovery Overview](#)

This chapter introduces the disaster recovery feature in ALTIBASE HDB. Its concept and terminology are explained, and data synchronization in a disaster recovery environment is described.

- [Chapter2: Managing DR](#)

This chapter describes the steps of the ALTIBASE HDB DR procedure and gives explanations on how the DR feature operates in various failure situations which can occur in the DR environment.

- [Chapter4: Failover](#)

This chapter describes the failover feature supported in an ALTIBASE HDB DR environment and how to use it.

- [Chapter5: Failure Server Repair](#)

This chapter explains Failure Server Repair, which is the process of repairing a server on which

a failure has occurred in the DR environment and reinstating it.

Documentation Conventions



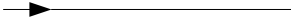
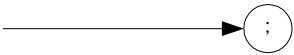


This section describes the conventions used in this manual. Understanding these conventions will make it easier to find information in this manual and other manuals in the series.

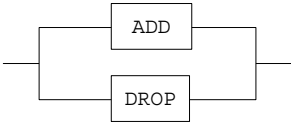
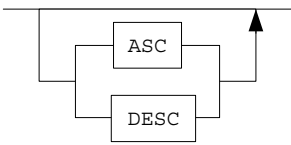
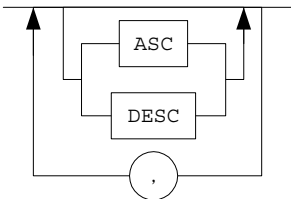
There are two sets of conventions:

- syntax diagram conventions
- sample code conventions

Syntax Diagram Conventions

This manual describes command syntax using diagrams composed of the following elements:

Elements	Meaning
	Indicates the start of a command. If a syntactic element starts with an arrow, it is not a complete command.
	Indicates that the command continues to the next line. If a syntactic element ends with this symbol, it is not a complete command.
	Indicates that the command continues from the previous line. If a syntactic element starts with this symbol, it is not a complete command.
	Indicates the end of a statement.
	Indicates a mandatory element.
	Indicates an optional element.

Elements	Meaning
	<p>Indicates a mandatory element comprised of options. One, and only one, option must be specified.</p>
	<p>Indicates an optional element comprised of options.</p>
	<p>Indicates an optional element in which multiple elements may be specified. A comma must precede all but the first element.</p>

Sample Code Conventions

The code examples explain SQL statements, stored procedures, iSQL statements, and other command line syntax.

The following table describes the printing conventions used in the code examples.

Rules	Meaning	Example
[]	Indicates an optional item.	VARCHAR [(size)] [[FIXED] VARIABLE]
{ }	Indicates a mandatory field for which one or more items must be selected.	{ ENABLE DISABLE COMPILE }
	A delimiter between optional or mandatory arguments.	{ ENABLE DISABLE COMPILE } [ENABLE DISABLE COMPILE]

Rules	Meaning	Example
.	Indicates that the previous argument is repeated, or that sample code has been omitted.	iSQL> select e_lastname from employees; E_LASTNAME ----- Moon Davenport Kobain . . . 20 rows selected.
Other Symbols	Symbols other than those shown above are part of the actual code.	EXEC :p1 := 1; acc NUMBER(11,2);
Italics	Statement elements in italics indicate variables and special values specified by the user.	SELECT * FROM table_name; CONNECT <i>userID/password</i> ;
Lower case words	Indicate program elements set by the user, such as table names, column names, file names, etc.	SELECT e_lastname FROM employees;
Upper case words	Keywords and all elements provided by the system appear in upper case.	DESC SYSTEM_.SYS_INDEX_;

Related Documents

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

- Installation Guide
- Getting Started Guide
- SQL Reference
- iSQL User's Manual
- Administrator's Manual
- Error Message Reference

Online Manuals

Online versions of our manuals (PDF or HTML) are available from Altibase's Customer Support site (<http://support.altibase.com/>).

Altibase Welcomes Your Opinions

Please feel free to send us your comments and suggestions regarding this manual. Your comments

About This Manual

and suggestions are important to us, and may be used to improve future versions of the manual. When you send your feedback, please make sure to include the following information:

- The name and version of the manual 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 Altibase's Customer Support site (<http://support.altibase.com/>).

We always appreciate your comments and suggestions.

1 Disaster Recovery Overview

This chapter introduces the disaster recovery feature in ALTIBASE HDB. Its concept and terminology are explained, and data synchronization in a disaster recovery environment is described.

1.1 Introduction

This section explains the concept and terminology of disaster recovery, and describes the disaster recovery feature in ALTIBASE HDB.

1.1.1 Concepts

Disaster recovery(DR) aims to provide high availability by minimizing data loss and recovery time in the event of main server failure.

To configure an ALTIBASE HDB DR environment, apart from the main HDB server which provides database services to the user, it is necessary for an additional HDB server to be on standby in preparation of failure. Under normal circumstances, data of the standby server and the main server are synchronized; if it becomes impossible for the main server to provide regular services due to a physical or logical abnormality, the standby server resumes services.

The ALTIBASE HDB DR process mainly consists of the following three parts, and a description for each part is provided below

- Data Synchronization (Log Synchronization) Failover
- Failover
- Failure Server Repair

1.1.2 Terminology

This section explains key terms of the ALTIBASE HDB DR feature.

- **Active Server**
This is the server which provides services to the user and on which the insertion, alteration, deletion, retrieval of data through user transactions occur.
- **Standby Server**
This is the server which is being synchronized with data of the active server as a backup against active server failure and on which the user cannot perform SQL operations.
- **Role Manager**
This is the process which monitors the active and standby servers, and performs failover and Failure Server Repair.
- **Listener**
This is the thread on the active or standby server which handles requests from the role manager or another server.
- **Sender Thread**
This is the thread which sends logs which are generated on the active server to the standby server; it is identical to the sender.

- **Receiver Thread**

This is the thread which receives logs sent from the active server; it is identical to the receiver.
- **Sync Mode**

This is one of the synchronization modes between the active and standby servers, and offers zero data loss in the event of failure.
- **Async Mode**

This is one of the synchronization modes between the active and standby servers, and offers fast synchronization; however, data loss can occur in the event of failure.
- **Failover**

This is the process of a standby server resuming services by replacing the role of the active server in the event of active server failure.
- **Auto-failover**

This is the process of the role manager automatically performing failover without the user's command.
- **Failure Server Repair**

This is the process of a failed server being repaired and redeployed to the DR environment. At redeployment, the repaired server acts as the standby server.
- **Role Switching (Switchover)**

This is the role reversal between the active server and the standby server.
- **Full Synchronization**

This is the building of a standby server with the data files, log anchors, and log files of the active server.
- **Partial Synchronization**

This is the data synchronization between the active and standby servers using only parts of data (pages and logs) of the active server.
- **Page Synchronization**

This is one process of partial synchronization, where pages are received from the active server.
- **Last Synced SN**

This is the serial number (SN) of the valid log which was most recently received from the active server and applied to the database.
- **Gap**

This is the synchronization gap between the active and standby servers.
- **Invalid Server**

1.1 Introduction

If an error occurs while repairing a failed server, the server is determined to be an invalid server. To repair an invalid server, full synchronization must be performed, instead of partial synchronization.

1.1.3 Data Synchronization

Data synchronization in the DR environment is the data synchronization between the active and standby servers under normal circumstances, for the continuous provision of database services by the standby server in the event of active server failure.

The data change history which is transmitted between HDB servers for the purpose of synchronization, indicates the logs generated by user transactions. It is also called log synchronization since it performs data synchronization through transmitting redo logs.

The ALTIBASE HDB servers taking over the roles of the active and standby servers create the sender or receiver threads responsible for transmitting logs. The sender thread sends logs to the standby servers and the receiver thread of the standby server apply the received logs to its database. Also, the sender/receiver threads of the HDB servers and the role manager deployed in the DR environment detect the normal or abnormal termination of another server within the DR environment and perform the necessary operations.

The ALTIBASE HDB DR feature offers both sync and async modes for data synchronization. The following table categorizes the characteristics of each mode.

Mode	Performance	Delayed Synchronization	Data Consistency
Async	High	Possible	Low
Sync	Low	Impossible	High

1.1.3.1 Async Mode

In async mode, when logs are generated on the active server, the sender thread sends the logs to the standby servers, which are then received by the receiver thread of the standby server to be applied to its database.

In this mode, user transactions on the active server and transactions performed by the receiver thread on the standby server are operated separately. In other words, user transactions on the active server are not affected by log transmissions or log replay on the standby server; thus, the transaction speed of the active server does not fall due to deployment in the DR environment.

However, data synchronization can be delayed on the standby server in situations where massive transactions occur, since the receiver thread of the standby server always follows transactions on the active server.

1.1.3.2 Sync Mode

In sync mode, when logs are generated on the active server, the user transaction which generated the logs directly sends the logs to the standby server, which are then received by the receiver thread

of the standby server to be applied to its database. For transaction commit logs, the corresponding transaction on the active server directly sends the logs and commits itself after confirming that the logs have been received by the standby server.

One advantage of sync mode is that no data loss exists between the active and standby servers, since transactions committed on the active server are guaranteed to have been received by the standby server. Also, unlike async mode, synchronization delay does not occur in sync mode even under situations where transactions occur frequently. However, the transaction speed of the active server falls, compared to async mode, since the active server waits for confirmation as to whether the standby server has received the logs.

In sync mode, if the client session mode is autocommit, it suffers more synchronization performance degradation than non-autocommit.

1.1.4 Failover

Failover in the DR feature is the provision of continuous database services by switching the role of the standby server to the active server in the event of active server failure.

For further information on failover, please refer to [Chapter4: Failover](#).

1.1.5 Failure Server Repair

Failure Server Repair is the process in which a failed server is repaired and redeployed in the DR environment. At redeployment, the given server acts as the standby server.

For further information on Failure Server Repair, please refer to [Chapter5: Failure Server Repair](#).

1.1 Introduction

2 Managing DR

This chapter describes the steps of the ALTIBASE HDB DR procedure and gives explanations on how the DR feature operates in various failure situations which can occur in the DR environment.

2.1 Steps of the DR Procedure

This section explains the steps that are taken by the respective roles in the DR environment before the active and standby servers are running and starting data synchronization between the servers.

1. The role manager starts.
2. The active server connects to the role manager server.
3. The standby server connects to the role manager server.
4. The role manager passes information of the standby server to the active server.
5. The active server attempts connection to the standby server.
6. Once connection between the active and standby server is established, the Failure Server Repair process is executed when necessary.
7. Once Failure Server Repair is complete, log transmission starts.

2.1.1 Starting the Role Manager

The role manager is operated as an ALTIBASE HDB process. You can start the role manager by using server script, just like the ALTIBASE HDB server.

```
$ server startRoleManager
```

You can also terminate the role manager by using server script.

```
$ server stopRoleManager
```

2.2 Troubleshooting

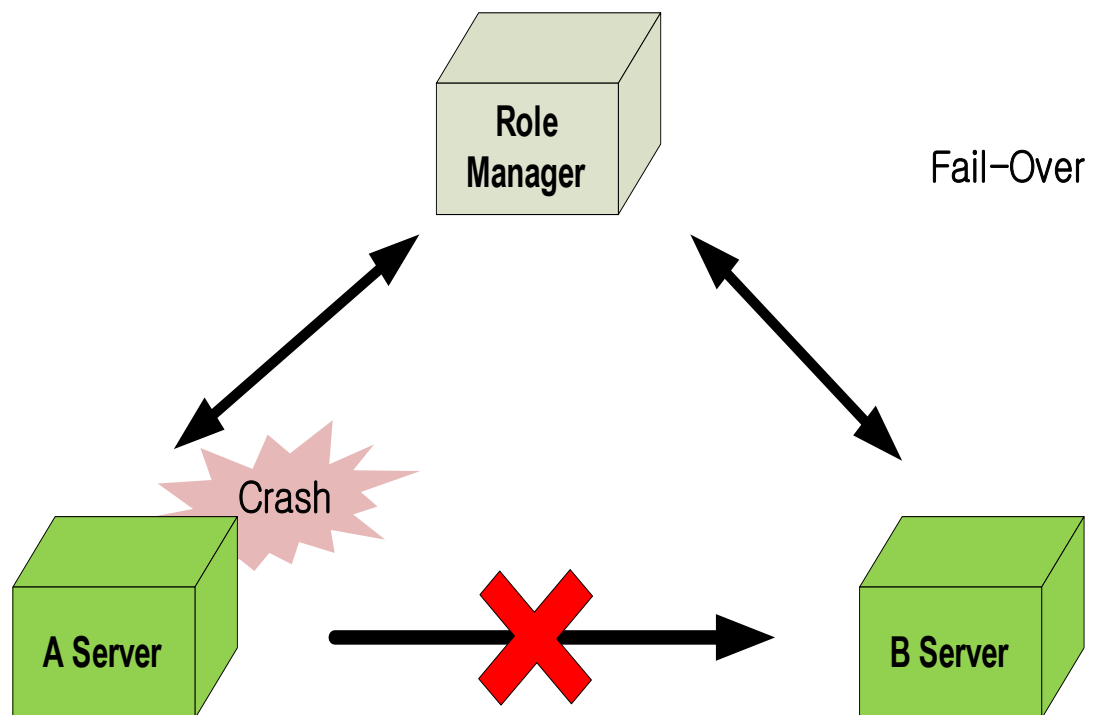
This section explains failures which can occur in the DR environment and recovery processes for each failure.

The following failures can occur in an ALTIBASE HDB DR environment.

- Abnormal shutdown of the active server or the standby server
- Interruption of communication between the active server and the standby server
- Network failure

2.2.1 Abnormal Shutdown of the Active Server

Figure 2-1 DR in the Event of Active Server Failure



1. Abnormal termination of Server A

If the active server (server A) terminates abnormally, the receiver thread on server B closes and waits for a new active server to connect.

2. Role manager process

The role manager detects that server A has failed and performs failover.

3. Failover execution

The role manager connects to server B and commands it to switch roles to the active server. Server B starts up in the SERVICE phase and starts operations as the active server.

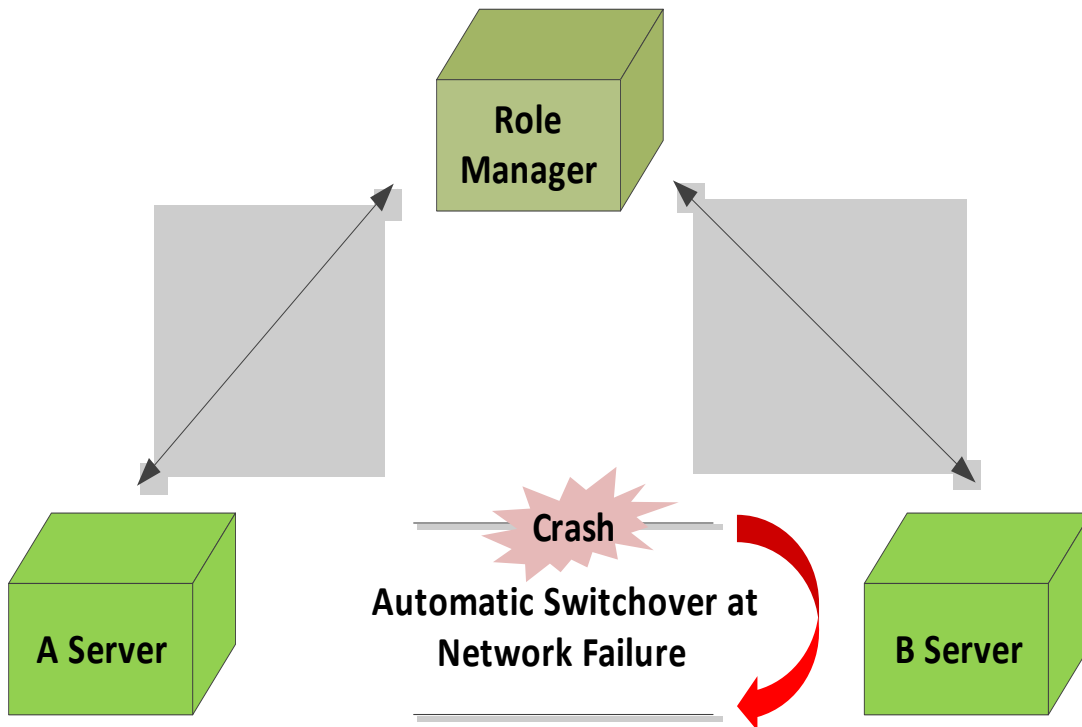
2.2 Troubleshooting

4. Server A restart (the sender thread on server B calls the receiver thread on the standby server)

When a server which has once failed in the DR environment is redeployed in the DR environment, it acts as the standby server, regardless of its former role. Therefore, server A restarts, undergoes Failure Server Repair if necessary, and is redeployed in the DR environment as the standby server.

2.2.2 Interruption of Communication Between the Active Server and the Standby Server

Figure 2-2 DR in Response to Communication Failure with the Standby Server



1. Communication failure between the active server (server A) and the standby server (server B)

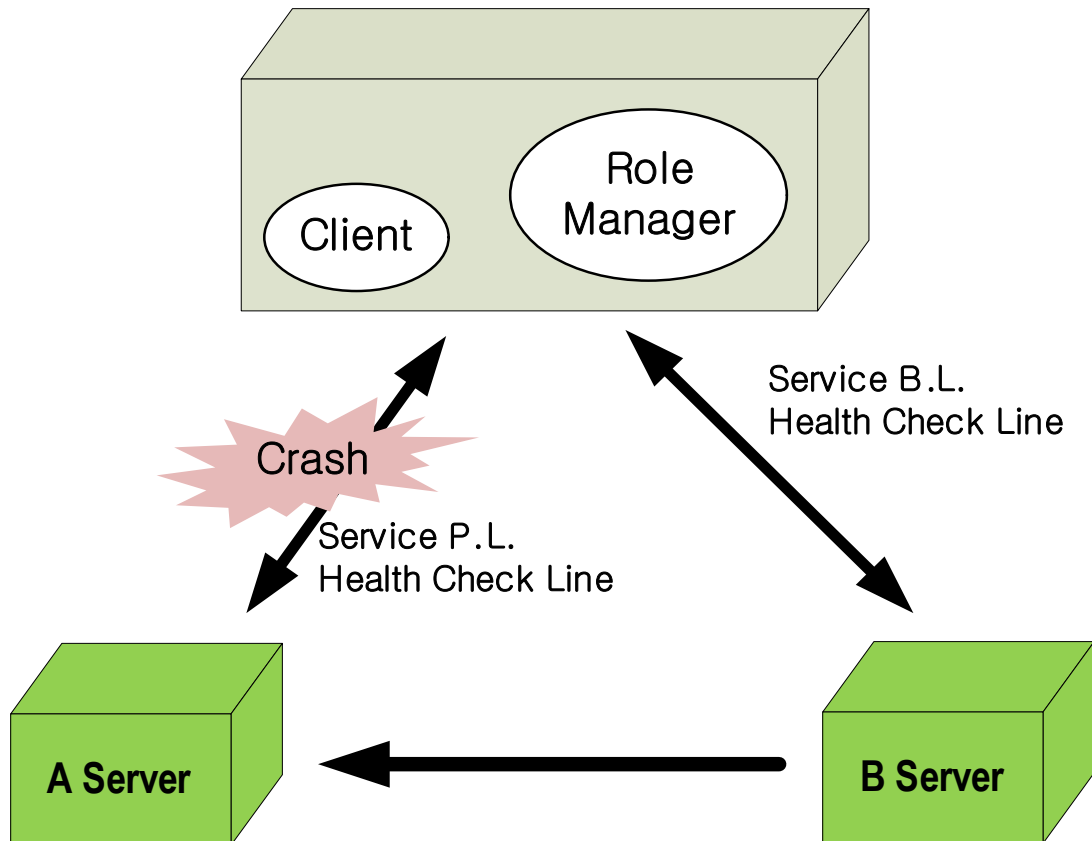
When communication fails between the servers, the receiver thread on the standby server (server B) closes. The sender thread on server A records the restart SN for log transmission and attempts connection to server B every 60 seconds.

2. Connection restoration

The sender thread on the active server (server A) awakes the receiver thread on the corresponding server and sends logs, starting with the restart SN, to the receiver thread. The receiver thread on server B is created by the connection of the sender thread on the corresponding server and performs synchronization by receiving logs from server A.

2.2.3 Network Failure

Figure 2-3 DR in the Event of Network Failure (the client and the role manager exist on the same server)



1. Disconnection of the primary line (server P.L. in the figure).
2. The role manager detects that server A has failed since communication from the role manager to server A is interrupted.
3. The role manager switches server B to the active server and performs failover.
4. Server B provides database services with a backup line (service B.L. in the figure).
5. Once server A is repaired, it operates as the standby server.

2.3 Related Performance Views

The following performance views can be used to view information of the currently configured DR environment and monitor the state of the synchronization process being executed for DR. For further information about each performance view, please refer to the *General Reference*.

- V\$DR_SERVERS
- V\$DR_CONNECTION_INFO
- V\$DR_GAP
- V\$DR_STATUS

3 Deploying DR

This chapter describes the statements which configure the ALTIBASE HDB DR environment and start, stop and alter it.

3.1 Considerations

The ALTIBASE HDB DR feature is accompanied with the following conditions. If these conditions are not met, the DR feature cannot be used.

3.1.1 System Requirements

Servers deployed in the DR environment must fulfill the following conditions.

- The active server, standby server and role manager server must be of an identical ALTIBASE HDB version.
- The Endianness (the byte order) must be identical for the active server and the standby server.
- The log file group of the active server and the standby server must be specified as one. If the log file group is specified as two or more, the DR feature does not operate.
- The database file path of the active and standby servers must be the same.
- The FAST_UNLOCK_LOG_ALLOC_MUTEX property must be set to 1(enable) to use sync mode.
- The incremental backup feature, including page change tracking, cannot be used with the DR feature.

3.1.2 Data Requirements

Logs are not recorded for changes made to data of tables in volatile tablespaces. Likewise, the DR feature cannot be used for data for which logs are not recorded.

3.1.3 Configuration Requirements

- Only one role manager can exist in the DR environment.
- Only one active server can exist in the DR environment.
- Only one standby server can exist in the DR environment.

3.2 Configuring DR

The DR environment must be configured on the database to deploy an ALTIBASE HDB server in the DR environment. This section explains statements which configure the DR environment on a database.

3.2.1 Syntax

```
ALTER DATABASE DISASTER RECOVERY SET my_server_name
ROLE [ ACTIVE | STANDBY ]
MODE [ ASYNC | SYNC ]
SERVER LIST server1_name('host_ip', host_port_no),
            server2_name('host_ip', host_port_no)
            ...;

ALTER DATABASE DISASTER RECOVERY
ADD ROLE MANAGER ('host_ip', host_port_no);
```

3.2.2 Description

The statement for configuring the DR environment can only be executed by the user granted the SYSDBA privilege in the CONTROL startup phase of the server.

The role of the server itself and information of other servers deployed in the DR environment must be input when configuring the DR environment on the database. Using the ALTER DATABASE DISASTER RECOVERY ADD ROLE MANAGER statement, information of the role manager must be input as well.

- ***my_server_name***
This specifies the server name to be used in the DR environment. A unique name within the DR environment should be input.
- **SERVER LIST**
This lists the servers deployed in the DR environment. The name, IP address, and the listening port of the receiver must be specified.
- ***host_ip***
The IP address.
- ***host_port_no***
The listening port number of the listener for receiving logs.
- **ACTIVE | STANDBY**
This specifies the role of the server itself as either ACTIVE or STANDBY.
- **ASYNC | SYNC**
This specifies the default synchronization mode of the server itself. Unless the server is the active server, this value is meaningless; standby servers conform to the synchronization mode

3.2 Configuring DR

of the active server.

3.2.3 Error Codes

Please refer to the *Error Message Reference*.

3.2.4 Examples

Assume that the list of servers deployed in the DR environment is as follows. Add the DR feature to each server.

- Role manager HDB server: IP address is 192.168.1.60 and port number is 15524
- Active server: IP address is 192.168.1.10, port number is 25524, server name is Seoul. Synchronization is executed in async mode.
- Standby server 1: IP address is 192.168.1.20, port number is 25535, server name is Daegu.

On the Seoul server

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY SET seoul
  ROLE active
  MODE async
  SERVER LIST seoul('192.168.1.10', 25524),
              daegu('192.168.1.20', 25535);
Alter success.
```

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY ADD ROLE
MANAGER ('192.168.1.60', 15524);
Alter success.
```

On the Daegu server

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY SET daegu
  ROLE standby
  MODE async
  SERVER LIST daegu('192.168.1.20', 25535),
              seoul('192.168.1.10', 25524),
              jeju('192.168.1.30', 25546),
              suwon('192.158.1.40', 25557);
Alter success.
```

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY ADD ROLE
MANAGER ('192.168.1.60', 15524);
Alter success.
```

3.3 Starting, Stopping and Modifying DR

3.3.1 Syntax

```
ALTER DATABASE DISASTER RECOVERY ADD SERVER
    server_name('host_ip', host_port_no);

ALTER DATABASE DISASTER RECOVERY MODIFY server_name
    ADD HOST 'host_ip', host_port_no;

ALTER DATABASE DISASTER RECOVERY MODIFY server_name
    REMOVE HOST 'host_ip', host_port_no;

ALTER DATABASE DISASTER RECOVERY MODIFY server_name
    SET HOST 'host_ip', host_port_no;

ALTER DATABASE DISASTER RECOVERY REMOVE ROLE MANAGER;

ALTER DATABASE DISASTER RECOVERY REMOVE SERVER server_name;

ALTER DATABASE DISASTER RECOVERY SET MODE = [ ASYNC | SYNC ];

ALTER DATABASE DISASTER RECOVERY SWITCH ROLE TO ACTIVE;

ALTER DATABASE DISASTER RECOVERY START STANDBY;

ALTER DATABASE DISASTER RECOVERY START STANDBY FORCE;
```

3.3.2 Description

The statement for starting or stopping the DR feature, or modifying the DR configuration can only be executed by the user granted the SYSDBA privilege in the CONTROL startup phase of the server.

- `ADD SERVER`
This adds one server to the DR environment.
- `ADD HOST`
This adds the host IP and port number of a particular server.
- `DROP HOST`
This drops a certain host IP and port number of a particular server.
- `SET HOST`
This sets a certain host IP and port number of a particular server as the main host of the server. When another server attempts to connect to a different server in the DR environment, the main host information of the server has priority.
- `REMOVE ROLE MANAGER`
This drops information of the role manager.
- `REMOVE SERVER`

3.3 Starting, Stopping and Modifying DR

This drops a particular server from the DR environment.

- `SET MODE [ASync | SYNC]`

This switches the synchronization mode of the server.

- `SWITCH ROLE TO ACTIVE`

This switches the server role to active.

- `START STANDBY`

This starts the standby server in a phase in which data (log) synchronization is possible. This statement can be executed in the CONTROL phase.

- `START STANDBY FORCE`

This starts a server as the standby server in the DR environment. Full synchronization must have been manually performed on this server.

3.3.3 Error Code

Please refer to the error messages related to DR in the *Error Message Reference*.

3.3.4 Example

- Add a server of the name Suwon with the IP address 192.168.1.60 and the port number 25524 to the DR environment.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY ADD SERVER  
suown('192.168.1.60', 25524);  
Alter success.
```

- Add the host information(IP address: 192.168.1.60, port number: 25524) to the server Suwon.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY MODIFY suown ADD HOST  
'192.168.1.60', 25524;  
Alter success.
```

- Drop the host information (IP address: 192.168.1.60, port number: 25524) from the server Suwon.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY MODIFY suown REMOVE HOST  
'192.168.1.60', 25524;  
Alter success.
```

- Set the information of a certain host (IP address: 192.168.1.60, port number:25524) as the main host on the server Suwon..

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY MODIFY suown SET HOST  
'192.168.1.60', 25524;  
Alter success.
```

- Drop role manager information from the DR environment.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY REMOVE ROLE MANAGER;
```

Alter success.

- Drop the server Suwon from the DR environment.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY REMOVE SERVER suown;  
Alter success.
```

- Switch the synchronization mode of the active server to sync mode.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY SET MODE = SYNC;  
Alter success.
```

- Switch the standby server to the active server.

```
ALTER DATABASE DISASTER RECOVERY SWITCH ROLE TO ACTIVE;
```

- Start the standby server with the following statement in the CONTROL phase.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY START STANDBY;  
[SM] Recovery Phase - 1 : Preparing Database  
                        : Dynamic Memory Version => Parallel Loading  
[SM] Recovery Phase - 2 : Loading Database  
[SM] Recovery Phase - 3 : Starting Standby Recovery  
                        Initializing Active Transaction List  
                        Redo  
...
```

- After performing full synchronization by copying files, start the standby server with the following statement in the CONTROL phase.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY START STANDBY FULL SYNC;  
[SM] Recovery Phase - 1 : Preparing Database  
                        : Dynamic Memory Version => Parallel Loading  
[SM] Recovery Phase - 2 : Loading Database  
[SM] Recovery Phase - 3 : Starting Standby Recovery  
                        Initializing Active Transaction List  
                        Redo  
...
```

3.4 Disabling Disaster Recovery

This section explains the statement for disabling the built DR configuration.

3.4.1 Syntax

```
ALTER DATABASE DISASTER RECOVERY UNSET;
```

3.4.2 Description

Disables the DR configuration of the database. The server on which this statement has been executed is excluded from the DR environment.

This statement can only be executed by the user granted the SYSDBA privilege in the CONTROL startup phase of the server.

3.4.3 Error Codes

Please refer to the *Error Message Reference*.

3.4.4 Example

Disable the DR configuration of the database.

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY UNSET;  
Alter success.
```

3.5 DR-related Properties

To use DR, the `altibase.properties` file should be modified to suit the purpose of its use. For the role manager server, however, the server must be restarted for the modified property to be applied, even if it is a DR-related property which can be modified while the server is running.

The following is a list of DR-related properties. For further information about each property, please refer to the *General Reference*.

- DR_ENABLE
- DR_MAX_LOGFILE
- DR_META_DIR
- DR_RM_META_DIR
- DR_RM_PORT_NO
- DR_PORT_NO
- DR_CONNECT_TIMEOUT
- DR_RECEIVE_TIMEOUT
- DR_HBT_DETECT_HIGHWATER_MARK
- DR_HBT_DETECT_TIME
- DR_PREFETCH_LOGFILE_COUNT
- DR_SENDER_SLEEP_TIME
- DR_KEEP_ALIVE_CNT
- DR_SENDER_NEXT_CONNECTION_TIMEOUT
- DR_STANDBY_WAIT_TIMEOUT

3.5 DR-related Properties

4 Failover

This chapter describes the failover feature supported in an ALTIBASE HDB DR environment and how to use it.

4.1 Failover Overview

In the event of failure while the database system is running, the failover feature aims to overcome failure and continue providing services, regardless of the failure that has occurred.

This section explains the failover concept supported in an ALTIBASE HDB DR environment.

4.1.1 Concept

Failover in the DR environment refers to the standby server switching roles to the active server for the provision of continuous database services. For such a switchover, the role manager server is necessary in the DR environment.

Failover can be invoked by the user with a SQL statement or automatically executed by the role manager, without the user's manipulation. If the auto-failover feature, wherein failover is automatically executed, is activated, the role manager selects the standby server and executes failover automatically, when it detects an active server failure.

In the DR environment, physical logs are used for data synchronization between active and standby servers. When compared with replication, the DR feature shows better performance of data synchronization. In failure situations, however, it takes the standby server longer to perform failover than in replication.

4.2 Using Failover

Failover can be manually executed with the user's command, or automatically executed by the role manager.

4.2.1 Auto-failover

The user's manipulation is unnecessary for auto-failover. If a failure occurs on the active server in the DR environment, the role manager automatically performs the failover operation.

The role manager automatically executes failover in the following steps.

1. The role manager detects failure on the active server.
2. The role manager selects the standby server.
3. The chosen server is failed over as the active server.
4. After completing failover, the active server sends logs to another standby server and starts data synchronization.

4.2.2 Manual Failover

The user manually executes failover in the following steps.

1. The user connects to the standby server to be switched to the active server with the SYSDBA privilege.
2. When the following statement is executed, the role manager executes failover.

```
ALTER DATABASE db_name DR SWITCH ROLE TO ACTIVE;
```

3. After completing failover, the active server sends logs to another standby server and starts data synchronization.

4.2 Using Failover

5 Failure Server Repair

This chapter explains Failure Server Repair, which is the process of repairing a server on which a failure has occurred in the DR environment and reinstating it.

5.1 What is Failure Server Repair?

Failure Server Repair is the process of repairing a server on which a failure has occurred and has therefore, been excluded from the DR environment and reinstating it. When the server is repaired and reinstated in the DR environment, it operates as the standby server, regardless of its previous role.

A failure server is repaired in the following steps.

1. The failure server is restarted and connected to the current active server in the DR environment.
2. The role manager determines whether or not to repair the failure server.
3. If failure server repair is decided upon, the failed server which is restarted determines whether or not partial synchronization is possible by checking the `CURRENT_SN` and `FAILOVER_SN` of the active server, and either performs partial synchronization or the user manually performs full synchronization, if partial synchronization is impossible.
4. Logs are received from the active server and data synchronization is started.

5.2 Full Sync

Full sync in the ALTIBASE HDB DR environment is the process of the the user manually copying everything (data files, log files, log anchor files, etc.) to a certain server, and deploying that server to the DR environment.

The user can manually perform full sync, which is necessary in the following situations.

- When a new server is to be deployed in the DR environment which is already built and running.
- When repairing an invalid server.
- When an abnormally terminated server in the DR environment restarts and during this process, partial sync either fails or is rendered impossible.

5.2.1 Performing Full Sync

Full sync is performed in the following steps.

1. Copy the following directory and all files in it from the active server to the standby server.
 - a. Property file (e.g., \$ALTIBASE_HOME/conf/altibase.properties)
 - b. Memory and disk data directory (e.g., \$ALTIBASE_HOME/dbs)
 - c. Log anchors and log directory (e.g., \$ALTIBASE_HOME/logs)
 - d. Archive log directory (e.g., \$ALTIBASE_HOME/arch_logs)
2. Start up the standby server in the CONTROL phase.
3. Start up the standby server in the full sync option with the following statement.

```
ALTER DATABASE DISASTER RECOVERY START STANDBY FULL SYNC;
```

5.2.2 Examples

The following example performs full sync on the server Daegu using the active server, Seoul.

1. Copy files

```
[active_host] altibase: scp $ALTIBASE_HOME/conf/altibase.properties
altibase@standby_host:$ALTIBASE_HOME/conf/.
```

```
[active_host] altibase: scp -r $ALTIBASE_HOME/dbs alti-
base@standby_host:$ALTIBASE_HOME/.
```

```
[active_host] altibase: scp -r $ALTIBASE_HOME/logs alti-
base@standby_host:$ALTIBASE_HOME/.
```

2. Start up the standby server in the CONTROL phase.

```
iSQL(sysdba) > STARTUP CONTROL
Connecting to the DB server.... Connected.
```

5.2 Full Sync

```
TRANSITION TO PHASE : PROCESS
```

```
TRANSITION TO PHASE : CONTROL  
[DR] Listener thread Initialization : PASS  
Command executed successfully.
```

3. Start up the standby server in the full sync option.

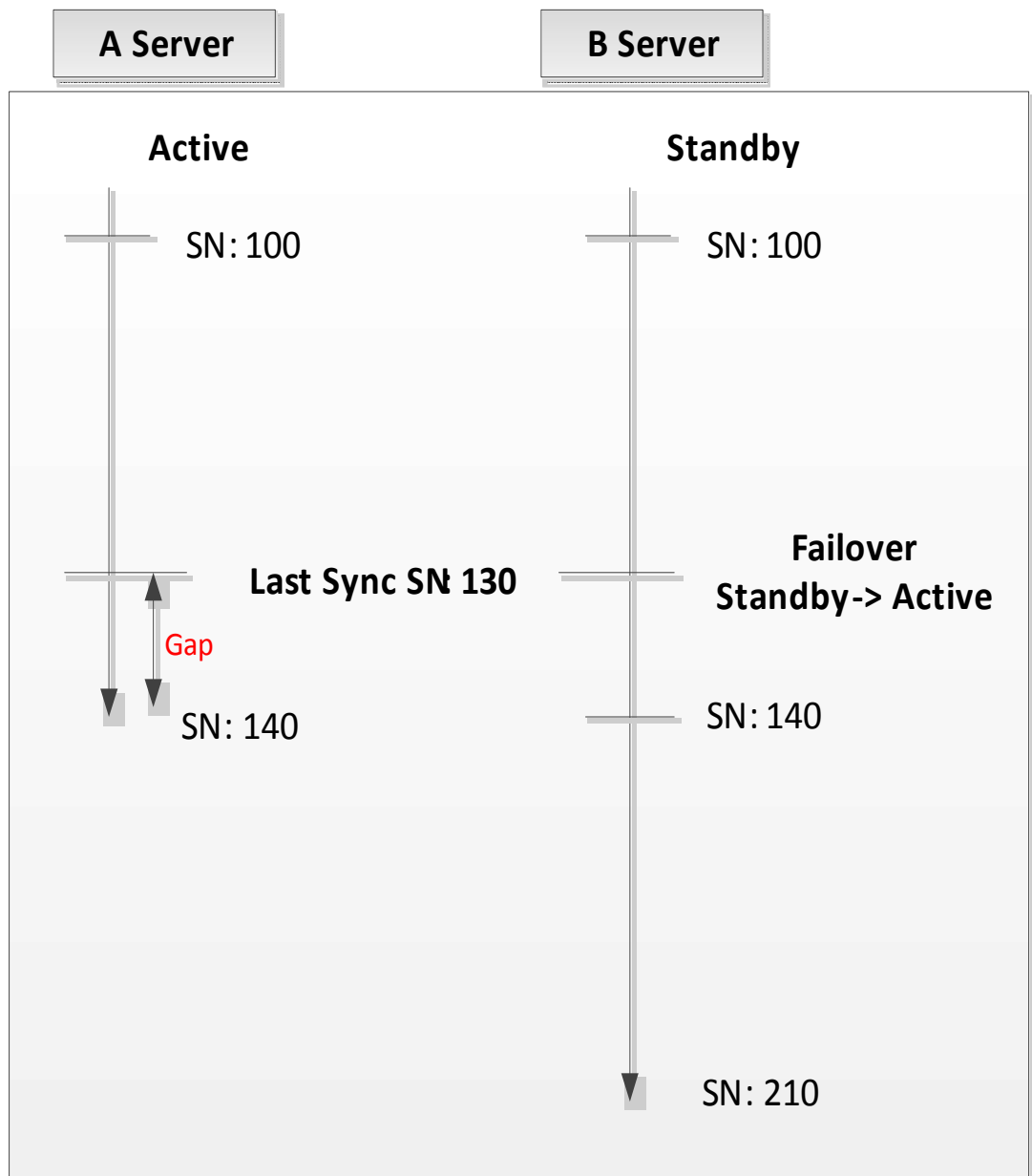
```
iSQL(sysdba)> ALTER DATABASE DISASTER RECOVERY START STANDBY FULL SYNC;  
[SM] Recovery Phase - 1 : Preparing Database  
                        : Dynamic Memory Version => Parallel Loading  
[SM] Recovery Phase - 2 : Loading Database  
[SM] Recovery Phase - 3 : Starting Standby Recovery  
                        Initializing Active Transaction List  
                        Redo  
...
```


5.3 Partial Sync

Partial sync is the process of synchronizing data in the DR environment using only part of the data pages and logs.

Partial sync is automatically performed during the process of restarting a server on which a failure has occurred in the DR environment. If a failure occurs during partial sync, the database becomes unusable. In this case, the user must perform full sync.

The process wherein partial sync (including page synchronization) can occur is described in the following example with a figure.



1. The log serial number (SN) is generated to 140 on the active server (server A).

5.3 Partial Sync

2. A failure occurs on server A at the point at which up to SN 130 has been sent to the standby server.
3. Server B is failed over and switched to the active server.
4. SN 131 to 210 is generated as server B provides database services.
5. Failure Server Repair is attempted as server A is restarted. Failure Server Repair is processed in the following steps.
 - a. Server A receives pages related to logs between SN 131 to 140 from server B and performs page synchronization.
 - b. Server A receives logs starting from SN 131 from server B and applies them to the database.
 - c. For the DR environment operated in async mode, partial sync is completed when transmission is completed, up to the most recent logs of server B at which page synchronization is complete.
 - d. For the DR environment operated in full sync, partial sync is completed when transmission up to the log which the service thread started to send (SN 210) is completed.

5.3.1 Example

After going through the Failure Server Repair process, start the server as the standby server.

```
iSQL(sysdba) > STARTUP CONTROL
Connecting to the DB server.... Connected.
```

```
TRANSITION TO PHASE : PROCESS
```

```
TRANSITION TO PHASE : CONTROL
[DR] Listener thread Initialization : PASS
Command executed successfully.
```

```
iSQL(sysdba) > ALTER DATABASE DISASTER RECOVERY START STANDBY;
[SM] Recovery Phase - 1 : Preparing Database
                        : Dynamic Memory Version => Parallel Loading
[SM] Recovery Phase - 2 : Loading Database
[SM] Recovery Phase - 3 : Starting Standby Recovery
                        Initializing Active Transaction List
                        Redo
...
```

Index

A

ACTIVE 15
Active Server 2
ADD HOST 18
ADD SERVER 18
ASYNC 15
Async Mode 3
Auto-failover 3

D

Data Synchronization 2
Disaster recovery 2
DROP HOST 18

F

Failover 3
Failure Server Repair 3
Full Synchronization 3

G

Gap 3

H

host_ip 15
host_port_no 15

I

Invalid Server 3

L

Last Synced SN 3
Listener 2

M

my_server_name 15

P

Page Synchronization 3
Partial Synchronization 3

R

Receiver Thread 3
REMOVE ROLE MANAGER 18
REMOVE SERVER 18
Role Manager 2
Role Switching 3

S

Sender Thread 2
SERVER LIST 15
SET HOST 18
SET MODE 19
STANDBY 15
Standby Server 2
START STANDBY 19
START STANDBY FORCE 19
SWITCH ROLE TO ACTIVE 19
Switchover 3
SYNC 15
Sync Mode 3