

# General Reference-1.Data Types & Altibase Properties

---

## Altibase 7.1

Altibase® Administration

Altibase Administration General Reference

Release 7.1

Copyright © 2001~2023 Altibase Corp. All Rights Reserved.

본 문서의 저작권은 (주)알티베이스에 있습니다. 이 문서에 대하여 당사의 동의없이 무단으로 복제 또는 전용할 수 없습니다.

**(주)알티베이스**

08378 서울시 구로구 디지털로 306 대륭포스트타워II 10층

전화 : 02-2082-1114

팩스 : 02-2082-1099

고객서비스포털 : <http://support.altibase.com>

홈페이지 : <http://www.altibase.com>

# 목차

---

- [서문](#)
  - [이 매뉴얼에 대하여](#)
- [1.자료형](#)
  - [자료형의 개요](#)
  - [문자형 데이터 타입](#)
  - [숫자형 데이터 타입](#)
  - [날짜형 데이터 타입](#)
  - [이진 데이터 타입](#)
  - [LOB 데이터 타입](#)
  - [공간 데이터 타입](#)
- [2.Altibase 프로퍼티](#)
  - [환경 설정 방법](#)
  - [프로퍼티 요약](#)
  - [데이터베이스 초기화 프로퍼티](#)
  - [성능 관련 프로퍼티](#)
  - [세션 관련 프로퍼티](#)
  - [타임아웃 관련 프로퍼티](#)
  - [트랜잭션 관련 프로퍼티](#)
  - [백업 및 복구 관련 프로퍼티](#)
  - [이중화 프로퍼티](#)
  - [네트워크 관련 프로퍼티](#)
  - [메시지 로그 관련 프로퍼티](#)
  - [데이터베이스 링크 관련 프로퍼티](#)
  - [감사 관련 프로퍼티](#)
  - [에이전트 관련 프로퍼티](#)
  - [사용자 계정 보안 관련 프로퍼티](#)
  - [기타 프로퍼티](#)

# 서문

## 이 매뉴얼에 대하여

이 매뉴얼은 Altibase의 기능, 제품 구성요소, 그리고 사용법에 대해 설명한다.

### 대상 사용자

이 매뉴얼은 다음과 같은 Altibase 사용자를 대상으로 작성되었다.

- 데이터베이스 관리자
- 성능 관리자
- 데이터베이스 사용자
- 응용 프로그램 개발자
- 기술지원부

다음과 같은 배경 지식을 가지고 이 매뉴얼을 읽는 것이 좋다.

- 컴퓨터, 운영 체제 및 운영 체제 유틸리티 운용에 필요한 기본 지식
- 관계형 데이터베이스 사용 경험 또는 데이터베이스 개념에 대한 이해
- 컴퓨터 프로그래밍 경험
- 데이터베이스 서버 관리, 운영 체제 관리 또는 네트워크 관리 경험

### 소프트웨어 환경

이 매뉴얼은 데이터베이스 서버로 Altibase 버전 7.1을 사용한다는 가정 하에 작성되었다.

### 이 매뉴얼의 구성

이 매뉴얼은 다음과 같이 구성되어 있다.

- 제 1장 자료형  
이 장은 Altibase에서 지원하는 데이터 타입에 대해 설명한다.
- 제 2장 Altibase 프로퍼티  
이 장은 Altibase 프로퍼티에 대해 설명한다.
- 제 3장 데이터베이스 객체 및 권한 관리  
이 장은 Altibase 데이터 디렉터리리에 대해 설명한다. Altibase의 데이터 디렉터리는 데이터베이스 객체 정보를 저장하는 메타 테이블과 시스템 프로세스 정보를 저장하는 프로세스 테이블로 나뉘어진다.
- 제 4장 샘플 스키마  
이 장은 샘플로 제공되는 테이블 정보와 ER 다이어그램을 제공한다.

### 문서화 규칙

이 절에서는 이 매뉴얼에서 사용하는 규칙에 대해 설명한다. 이 규칙을 이해하면 이 매뉴얼과 설명서 세트의 다른 매뉴얼에서 정보를 쉽게 찾을 수 있다.




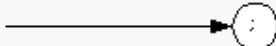
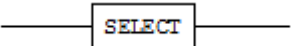
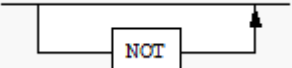
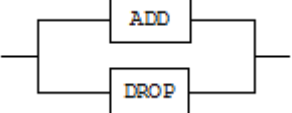
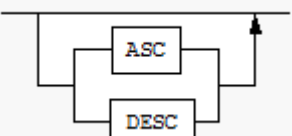
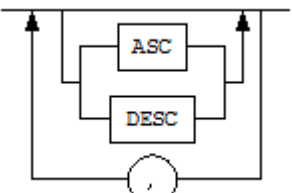
여기서 설명하는 규칙은 다음과 같다.

- 구문 다이어그램

- 샘플 코드 규칙

## 구문 다이어그램

이 매뉴얼에서는 다음 구성 요소로 구축된 다이어그램을 사용하여, 명령문의 구문을 설명한다.

구성 요소	의미
	명령문이 시작한다. 완전한 명령문이 아닌 구문 요소는 화살표로 시작한다.
	명령문이 다음 라인에 계속된다. 완전한 명령문이 아닌 구문 요소는 이 기호로 종료한다.
	명령문이 이전 라인으로부터 계속된다. 완전한 명령문이 아닌 구문 요소는 이 기호로 시작한다.
	명령문이 종료한다.
	필수 항목
	선택적 항목
	선택사항이 있는 필수 항목. 한 항목만 제공해야 한다.
	선택사항이 있는 선택적 항목.
	선택적 항목. 여러 항목이 허용된다. 각 반복 앞부분에逗마가 와야 한다.

## 샘플 코드 규칙

코드 예제는 SQL, Stored Procedure, iSQL, 또는 다른 명령 라인 구문들을 예를 들어 설명한다.

아래 테이블은 코드 예제에서 사용된 인쇄 규칙에 대해 설명한다.

규칙	의미	예제
[ ]	선택 항목을 표시	VARCHAR [(size)][[FIXED \]] VARIABLE]
{ }	필수 항목 표시. 반드시 하나 이상을 선택해야 되는 표시	{ ENABLE   DISABLE   COMPILE }

규칙	의미	예제
	선택 또는 필수 항목 표시의 인자 구분 표시	{ ENABLE   DISABLE   COMPILE } [ ENABLE   DISABLE   COMPILE ]
...	그 이전 인자의 반복 표시 예제 코드들의 생략되는 것을 표시	<pre>SQL&gt; SELECT ename FROM employee; ENAME ----- SWNO HJNO HSCHOI . . . 20 rows selected.</pre>
그 밖에 기호	위에서 보여진 기호 이 외에 기호들	EXEC :p1 := 1; acc NUMBER(11,2);
기울임꼴	구문 요소에서 사용자가 지정해야 하는 변수, 특수한 값을 제공해야만 하는 위치	<pre>SELECT * FROM <i>table_name</i>; CONNECT <i>userID/password</i>;</pre>
소문자	사용자가 제공하는 프로그램의 요소들, 예를 들어 테이블 이름, 칼럼 이름, 파일 이름 등	SELECT ename FROM employee;
대문자	시스템에서 제공하는 요소들 또는 구문에 나타나는 키워드	DESC SYSTEM.SYS_INDICES;

## 관련 자료

자세한 정보를 위하여 다음 문서 목록을 참조한다.

- Installation Guide
- Getting Started Guide
- Administrator's Manual
- Replication Manual

## Altibase는 여러분의 의견을 환영합니다.

이 매뉴얼에 대한 여러분의 의견을 보내주시기 바랍니다. 사용자의 의견은 다음 버전의 매뉴얼을 작성하는데 많은 도움이 됩니다. 보내실 때에는 아래 내용과 함께 고객센터포털(<http://support.altibase.com/kr/>)로 보내주시기 바랍니다.

- 사용 중인 매뉴얼의 이름과 버전
- 매뉴얼에 대한 의견
- 사용자의 성함, 주소, 전화번호

이 외에도 Altibase 기술지원 설명서의 오류와 누락된 부분 및 기타 기술적인 문제들에 대해서 이 주소로 보내주시면 정성껏 처리하겠습니다. 또한, 기술적인 부분과 관련하여 즉각적인 도움이 필요한 경우에도 고객센터포털을 통해 서비스를 요청하시기 바랍니다.

여러분의 의견에 항상 감사드립니다.

# 1.자료형

SQL을 사용하여 데이터베이스에 데이터를 저장, 변경하고 질의하기 위해서는 데이터베이스의 자료형에 대한 이해가 선행되어야 한다. 이 장에서는 Altibase가 지원하는 데이터형에 대해서 자세히 설명한다.

## 자료형의 개요

### 데이터형의 종류

Altibase에서 지원하는 데이터형은 다음과 같다.

#### 문자형 데이터형

M : 정의된 칼럼 길이 L : 입력 문자열의 길이		
타입	Length	Size
CHAR(M)	1 ~ 32000	M + 2
VARCHAR(M)	1 ~ 32000	length + 2 여기서: 입력 값이 가변영역에 저장되면, length = L 입력 값이 고정영역에 저장되면, length = M
NCHAR(M)	1 ~ 16000(UTF16) 1 ~ 10666(UTF8)	M*2 + 2(UTF16) M*3 + 2(UTF8)
NVARCHAR(M)	1 ~ 16000(UTF16) 1 ~ 10666(UTF8)	length*2 + 2(UTF16) length*3 + 2(UTF8) 여기서: 입력 값이 가변영역에 저장되면, length = L 입력 값이 고정영역에 저장되면, length = M

NCHAR와 NVARCHAR는 유니코드 문자형 타입이다. UTF16으로 인코딩된 문자열의 최대 길이는 UTF8로 인코딩된 문자열의 최대 길이와 다르다.



## 숫자형 데이터형

	타입	Precision	Scale	Size (bytes)	비 고
Non-native	NUMERIC	38	0	3+ ((precision)+2)/2	*고정 소수점 숫자 *DECIMAL은 NUMERIC과 동일한 데이터 타입이다.
	NUMERIC(p)	1 ~ 38	0		
	NUMERIC(p,s)	1 ~ 38	-84 ~ 128		
	DECIMAL	38	0		
	DECIMAL(p)	1 ~ 38	0		
	DECIMAL(p,s)	1 ~ 38	-84 ~ 128		
	NUMBER(p)	1 ~ 38	0		
	NUMBER(p,s)	1 ~ 38	-84 ~ 128		
	NUMBER	38	X	3+ ((precision)+2)/2	*부동 소수점 숫자
	FLOAT	38	X		
	FLOAT(p)	1 ~ 38	X		
Native	타입	호환 C Type		Size(bytes)	비고
	DOUBLE	double		8	*부동 소수점 숫자
	REAL	float		4	
	BIGINT	long 또는 long long		8	*정수형
	INTEGER	int		4	
	SMALLINT	short		2	

## 예제

예제1. 고정 소수점 숫자 크기 계산 :  $(3 + ((p) + 2) / 2)$

- NUMERIC  
NUMERIC(38, 0): 크기 =  $3 + 40/2 = 23$  bytes
- NUMERIC(p) / NUMERIC(p, 0)  
NUMERIC(10): 크기 =  $3 + 12/2 = 9$  bytes
- NUMERIC(p, s)  
NUMERIC(10, 9): 크기 =  $3 + 12/2 = 9$  bytes

- DECIMAL: NUMERIC과 동일
- DECIMAL(p): NUMERIC(p)과 동일
- DECIMAL(p,s): NUMERIC(p,s)과 동일
- NUMBER(p): NUMERIC(p)과 동일
- NUMBER(p,s): NUMERIC(p,s)과 동일

예제2. 부동 소수점 숫자 크기 계산:  $(3 + ((p) + 2) / 2)$

- FLOAT  
FLOAT(38): 크기 =  $3 + 40/2 = 23$  bytes
- FLOAT(p)  
FLOAT(20): 크기 =  $3 + 22/2 = 14$  bytes
- NUMBER: FLOAT과 동일

### 날짜 데이터형

타입	Size (byte)
DATE	8

### 이진 데이터형

M : 정의된 칼럼 길이  
L : 입력 문자열의 길이

타입	Length	Size
BLOB/CLOB		1 ~ 4294967295
BYTE	1 ~ 32000	M + 2
VARBYTE	1 ~ 32000	length + 2 여기서 입력 값이 가변영역에 저장되면, length = L 입력 값이 고정영역에 저장되면, length = M
NIBBLE	1 ~ 254	M/2 + 1
BIT	1 ~ 64000	M/8 + 4
VARBIT	1 ~ 64000	length/8 + 4 여기서 입력 값이 가변영역에 저장되면, length = L 입력 값이 고정영역에 저장되면, length = M

## 공간 데이터형

타입	Length	Size (byte)
GEOMETRY	8~104857600	length + 40

실제 레코드의 크기는 위에 각 데이터형 별로 명시된 크기(bytes)에서 헤더 정보 크기 만큼 추가된다. 헤더 정보는 운영 체제에 따라 다를 수 있다.

## NULL

행을 테이블에 삽입할 때 열의 값을 모르거나 값이 아직 결정되지 않은 경우, 즉 값이 존재 하지 않는 것을 나타내는 경우에 널(NULL)이 사용된다. 널(NULL)은 0 또는 공백과는 다르며, 비교연산이나 저장시 특별하게 취급된다.

NVL() 함수, IS NULL 조건, IS NOT NULL 조건을 제외한 수식 연산에 널이 포함되면, 최종 연산의 결과는 널이 된다. 즉, 수식에 널이 포함되면 비교 또는 연산이 의미가 없어지게 된다.

테이블 생성시 NOT NULL 또는 PRIMARY KEY로 정의되지 않은 모든 데이터 유형의 컬럼에는 널을 입력할 수 있다.

## 묵시적 데이터 타입 변환

서로 다른 데이터 타입에 대한 연산을 수행할 때 정확한 연산을 위해 데이터 타입이 변환되어 수행된다. 변환 방법은 묵시적인 방법과 명시적인 방법이 있다.

묵시적 변환이란, 타입이 다른 데이터를 연산할 때에는 내부적으로 데이터 타입을 변환하지만 데이터 타입의 속성은 유지되는 것을 의미한다. 같은 데이터 타입의 두 값을 비교 연산할 때, 어떤 변환 없이 직접 그 값에 대해 비교 연산이 수행된다. 그러나 비교되는 두 값의 데이터 타입이 다른 경우 한 쪽 값을 다른 값의 데이터 타입으로 변환한 후 비교 연산이 수행된다.

다음의 테이블은 묵시적 데이터 타입의 변환 가능한 행렬을 나타낸다 (O: 데이터 타입이 변환되어도 속성이 유지되는 것에 표시). 기존 테이블의 데이터 타입을 MODIFY 구문으로 변환하는 방법은 SQL Reference의 *modify\_column\_clause* 구문을 참조한다.

변환후 변환 전	char	var char	nchar	nvarchar	clob	big int	decim al	dou ble	float	int eger	num ber	num eric	real	small int	date	blob	byte	varbyte	nibble	bit	varbit	geometry
char	o	o	o	o		o	o	o	o	o	o	o	o	o	o							
varchar	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o							
nchar	o	o	o	o		o	o	o	o	o	o	o	o	o	o							
nvarchar	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o							
clob					o																	
bigint	o	o	o	o		o	o	o	o	o	o	o	o	o								
decimal	o	o	o	o		o	o	o	o	o	o	o	o	o								
double	o	o	o	o		o	o	o	o	o	o	o	o	o								
float	o	o	o	o		o	o	o	o	o	o	o	o	o								
integer	o	o	o	o		o	o	o	o	o	o	o	o	o								
number	o	o	o	o		o	o	o	o	o	o	o	o	o								
numeric	o	o	o	o		o	o	o	o	o	o	o	o	o								
real	o	o	o	o		o	o	o	o	o	o	o	o	o								
smallint	o	o	o	o		o	o	o	o	o	o	o	o	o								
date	o	o	o	o											o							
blob																o						
byte																o	o	o				
varbyte																o	o	o				
nibble																			o			
bit																				o	o	
varbit		o																		o	o	
geometry																						o

## 묵시적 데이터 타입 변환 규칙

테이블 t10 테이블에 bit 타입의 '1000'을 입력하면 integer '1000'으로 변환은 성공하지만, 데이터 타입의 속성이 달라지기 때문에 묵시적인 데이터 타입 변환이라고 볼 수 없다.

```
isQL> create table t10 (i1 integer);
Create success.
isQL> insert into t10 values (bit'1000');
1 row inserted.
isQL> select * from t10;
I1
-----
1000
1 row selected.
```

따라서 묵시적 데이터 타입의 변환은 아래와 같은 규칙을 따른다.

- 숫자형 데이터 타입과 문자형 데이터 타입의 비교 연산 또는 사칙 연산시, 문자형 데이터 타입을 숫자형 데이터 타입으로 변환하여 수행한다.
- 날짜형 데이터 타입과 문자형 데이터 타입의 비교 연산시, 문자형 데이터 타입을 날짜형 데이터 타입으로 변환하여 비교 연산을 수행한다.
- 데이터 타입의 변환이 불가능한 연산은 무효화 된다.
- 함수에서 사용되는 인자는 함수에서 정의된 인자의 데이터 타입으로 변환한다.
- 문자형 데이터 타입이나 십진 정밀도(decimal precision)를 사용하는 수치형 데이터 타입을 이진 정밀도(binary precision)를 사용하는 부동소수점 수치형 데이터 타입으로 변환하면 값이 손실될 수 있다.
- INSERT, UPDATE 실행시 INSERT, UPDATE 되는 칼럼의 데이터 타입으로 데이터형이 변환된다.

## 예제

<질의> 숫자형 데이터 타입과 문자형 데이터 타입의 비교 연산시 문자형 데이터 타입 '10'은 숫자형 데이터로 변환된다.

```
isQL> create table emp (empno integer, name varchar(10), hire_date date);
insert into emp values (10,'altibase', '10-nov-2015');

isQL> select name from emp where empno = '10';
NAME
-----
altibase
1 row selected.
```

<질의> 숫자형 데이터 타입과 문자형 데이터 타입 간 사칙연산시 문자형 데이터 타입 '10'은 숫자형 데이터 타입으로 변환된다.

```
isQL> select empno + '10' from emp;
EMPNO+'10'
-----
20
1 row selected.
```

<질의> 날짜형 데이터 타입과 문자형 데이터 타입의 비교 연산시 문자형 데이터 타입 '10-nov-2015'은 날짜형 데이터 타입으로 변환된다.

```
iSQL> select hire_date from emp where hire_date = '10-nov-2015';
HIRE_DATE
-----
10-NOV-2015
1 row selected.
```

<질의> 숫자형 데이터 타입과 문자형 데이터 타입 간 사칙연산시 이진 데이터 타입은 숫자형 데이터 타입으로 변환할 수 없어 연산이 무효화된다.

```
iSQL> select empno + cast(12345 as nibble(6)) from emp;
[ERR-2100C : Conversion not applicable.
0001 : select EMPNO + CAST(12345 as NIBBLE(6)) from EMP
          ^                               ^
]

```

<질의> 함수 SUM에 문자형 데이터 타입 '10'을 인자로 받는 경우 숫자형 데이터로 변환된다.

```
iSQL> select sum('10') from dual;
SUM('10')
-----
10
1 row selected.
```

<질의> 문자형 데이터 타입 '12.123456789'을 부동소수점 숫자형 데이터 타입인 float으로 변환하면, 유효 자릿수는 float(11)로 되어 값의 손실이 발생한다.

```
iSQL> select float'12.123456789' from dual;
FLOAT'12.123456789'
-----
12.1234568
1 row selected.
```

<질의> INSERT 되는 숫자형 데이터의 값은 INSERT 되는 칼럼의 데이터 유형에 맞게 변환되어 값이 INSERT 된다.

```
iSQL> create table t1 ( i1 char(10), i2 integer, i3 double);
Create success.
iSQL> insert into t1 values (integer'1020', char'1928', float'123.1234');
1 row inserted.
iSQL> select * from t1;
I1      I2      I3
-----
1020    1928    123.1234
1 row selected.
```

## 명시적 데이터 타입 변환

명시적 데이터 타입 변환은 SQL 변환 함수 또는 아래와 같이 타입 캐스팅을 사용해서 명시적으로 수행될 수 있다. 데이터 타입을 변환하는 SQL 함수는 SQL Reference에서 설명한다.

### 구문

```
datatype '문자 또는 상수 literal '
```

### 예제

어떤 데이터 타입의 상수 데이터를 명시적으로 다른 데이터 타입으로 변환한다. 다음은 157.27의 숫자 값을 '157.27'의 문자열로 변환하는 예제이다.

```
CHAR '157.27'
```

## 문자열 표기

SQL 쿼리에서 문자열을 표기할 시에는 홑 따옴표(')로 문자열을 묶어서 사용한다. 홑 따옴표를 문자열로 표기할 시에는 홑 따옴표가 escape 문자가 되므로 앞에 홑 따옴표를 붙여줘야 한다.

### 예제

```
SELECT * FROM EMPLOYEE WHERE NAME = 'KIM';
INSERT INTO EMPLOYEE VALUES ('GILDONG'''); //GILDONG' 값을 삽입
SELECT * FROM REMOTE_TABLE(link1, 'SELECT * FROM EMPLOYEE WHERE NAME='''KIM'''; //'''는 쌍 따옴표가 아니라 홑 따옴표 두 개임
```

## FIXED/VARIABLE 옵션

FIXED와 VARIABLE은 칼럼의 데이터가 어느 영역에 저장될 지를 지정하는 키워드이다.

한 레코드 전체가 연속된 공간에 저장될 때, 이 공간을 고정(FIXED) 영역이라고 한다. 칼럼들 중의 하나가 그 레코드의 나머지 연속된 고정 영역이 아닌 다른 분리된 공간에 저장될 때, 이 칼럼은 가변(VARIABLE) 영역에 저장된다고 말한다.

한 칼럼이 가변 영역에 저장될 때, 데이터 길이와 실제 데이터가 저장된 위치를 가리키는 포인터 같은 그 칼럼의 헤더 정보는 고정 영역에 저장된다. 반면 칼럼의 데이터는 가변 영역에 저장된다.

디스크 테이블스페이스에 테이블 생성시, 사용자가 FIXED 또는 VARIABLE을 지정하더라도 이는 무시되고 테이블의 모든 칼럼은 FIXED로 처리된다. 그러나 메모리 테이블스페이스에 테이블을 생성할 때는 사용자가 명시한 옵션이 그대로 사용된다.

그러나, 모든 LOB 데이터 타입 칼럼의 데이터는 항상 VARIABLE로 처리되고, 그 데이터는 IN ROW 절에 지정된 값에 따라서 고정 또는 가변 영역에 저장될 수 있다.

다음의 데이터 타입에 대해 VARIABLE을 지정할 수 있다.

: CHAR, VARCHAR, NCHAR, NVARCHAR, BYTE, VARBATE, NIBBLE, BIT, VARBIT, BLOB, 및 CLOB

## IN ROW 절

이 절은 가변 영역에 저장되는 칼럼 데이터에만 관련이 있다. 테이블 생성시 FIXED와 IN ROW 절이 모두 명시되면, IN ROW 절은 무시된다. VARIABLE로 지정된 칼럼에 데이터가 입력될 때, 데이터의 길이가 IN ROW 절에 명시된 값 이하이면 데이터는 고정 영역에 저장될 것이다. 반면 데이터의 길이가 IN ROW 절에 명시된 값보다 크면, 데이터는 가변 영역에 저장될 것이다.

여기서 “데이터의 길이”는 입력된 데이터의 길이가 아니고, 메모리 또는 디스크에 실제로 저장될 데이터의 길이를 의미하는데, 이는 입력 데이터의 길이보다 다소 크다. 예를 들어, 칼럼이 'VARCHAR(400) IN ROW 200'으로 정의 되었다면, 입력 데이터의 길이가 198 (데이터 저장시 2바이트가 추가로 더 필요하다) 이하일 때 데이터는 고정 영역에 저장될 것이다.

고정 영역에 저장되는 LOB 데이터의 기본 크기는 메모리 테이블을 위한 MEMORY\_LOB\_COLUMN\_IN\_ROW\_SIZE 프로퍼티와 디스크 테이블을 위한 DISK\_LOB\_COLUMN\_IN\_ROW\_SIZE 프로퍼티를 사용해서 지정할 수 있다. 또한, VARIABLE 옵션이 지정된 다른 데이터 타입의 칼럼을 위한 기본 크기는 MEMORY\_VARIABLE\_COLUMN\_IN\_ROW\_SIZE 프로퍼티를 사용해서 명시할 수 있다. 이들 프로퍼티를 지정하면 테이블 생성시 각 칼럼에 반복적으로 IN ROW 절을 사용할 필요가 없다.

이들 프로퍼티에 대한 상세한 설명은 2장을 참고하기 바란다.

## 문자형 데이터 타입

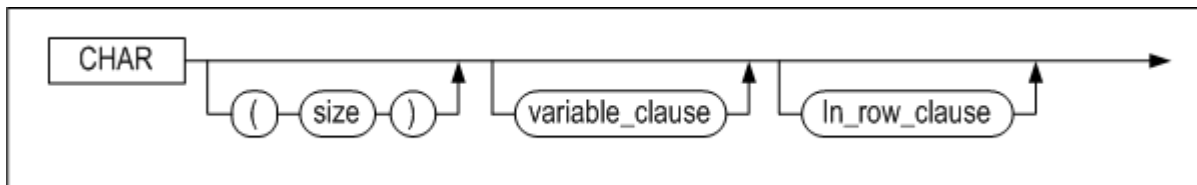
문자 데이터 타입은 데이터베이스 문자 집합 또는 국가 문자 집합의 문자 데이터를 저장하는데 사용된다.

Altibase는 아래의 타입을 지원한다.

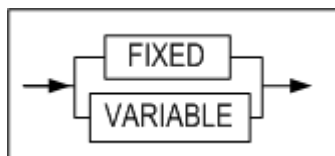
- CHAR
- VARCHAR
- NCHAR
- NVARCHAR

## CHAR

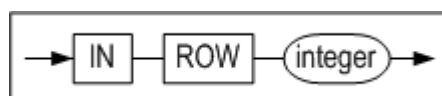
흐름도



variable\_clause::=



in\_row\_clause::=



## 구문

```
CHAR [(size)] [ FIXED | VARIABLE [IN ROW size] ]
```

## 설명

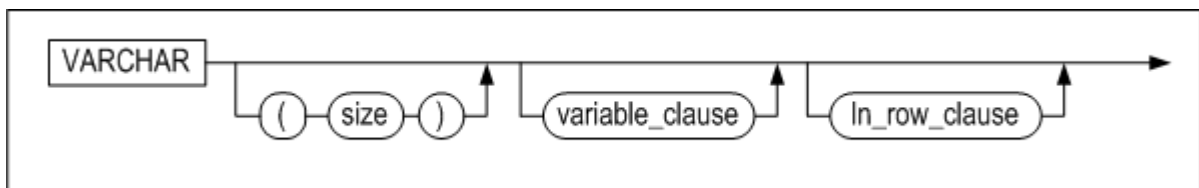
명시된 크기(size)만큼 고정 길이를 가지는 문자형 데이터 타입이다. 명시된 전체 크기에 비해 입력 값의 길이가 작을 경우 뒷부분은 공백으로 채워진다.

CHAR 칼럼의 기본 크기는 1 바이트이며 최대 길이는 32000바이트이다.

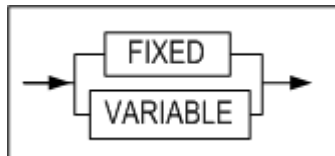
FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

## VARCHAR

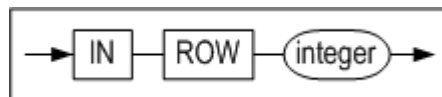
## 흐름도



**variable\_clause::=**



**in\_row\_clause::=**



## 구문

```
VARCHAR [(size)] [ FIXED | VARIABLE [IN ROW size] ]
```

## 설명

명시된 크기 내에서 가변 길이를 가지는 문자형 데이터 타입이다.

VARCHAR 칼럼의 기본 크기는 1 바이트이며 최대 길이는 32000바이트이다.

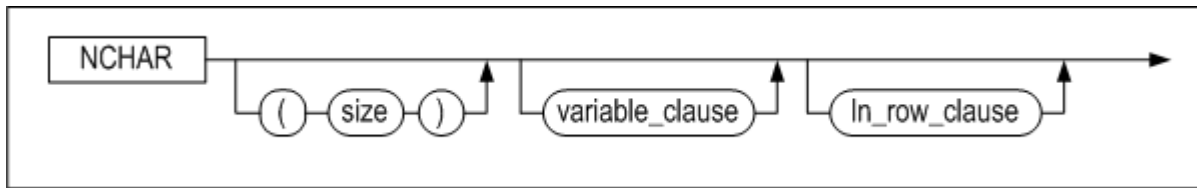
VARCHAR는 가변 길이 데이터 타입이다. 즉, 입력 데이터의 길이가 정의된 칼럼 크기보다 작을 경우, 실제로 입력된 데이터만 저장된다. 반면, CHAR 데이터 타입의 경우 입력 데이터의 길이가 칼럼 크기보다 작으면 그 칼럼의 남은 공간은 공백으로 채워진다. 예를 들어, CHAR(10)으로 정의된 칼럼에 단어 "magic"이 입력되면, 이 데이터는 "magic\_"으로 저장될 것이다. 여기서 "\_"은 공백을 나타낸다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

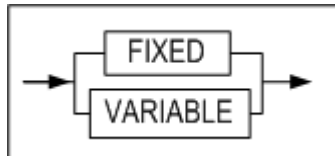


## NCHAR

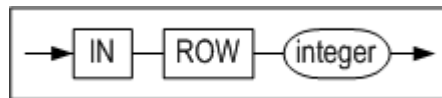
흐름도



**variable\_clause::=**



**in\_row\_clause::=**



구문

```
NCHAR [(size)] [ FIXED | VARIABLE [IN ROW size] ]
```

설명

명시된 길이만큼 고정 길이를 가지는 문자형 데이터 타입이다. 정의된 칼럼 크기에 비해 입력 값의 길이가 작을 경우, 남은 공간은 공백으로 채워진다.

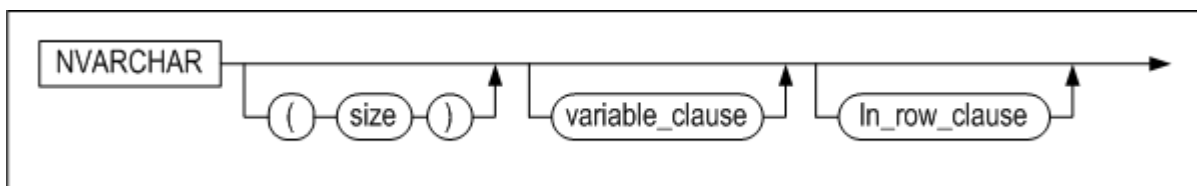
NCHAR 칼럼의 문자 하나당 크기는 국가 문자 집합(national character set)이 UTF16인 경우에는 고정길이 2바이트이고(가변길이 없음), UTF8인 경우에는 가변길이 3바이트(고정길이 없음)이다. UTF16은 한 문자에 대해 고정 2바이트를 사용하는 반면, UTF8은 1~3바이트까지 가변으로 사용하기 때문이다.

국가 문자 집합이 UTF16이면 최대 크기는 16000 bytes이다.

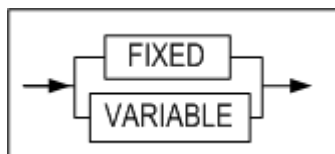
FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

## NVARCHAR

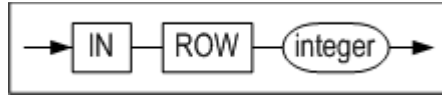
흐름도



**variable\_clause::=**



**in\_row\_clause::=**



## 구문

```
NVARCHAR [(size)] [ FIXED | VARIABLE [IN ROW size] ]
```

## 설명

명시된 길이 내에서 가변 길이를 가지는 유니코드 문자형 데이터 타입이다.

NVARCHAR 컬럼의 문자 하나당 크기는 국가 문자 집합(national character set)이 UTF16인 경우에는 고정길이 2바이트이고(가변길이 없음), UTF8인 경우에는 가변길이 3바이트(고정길이 없음)이다. UTF16은 한 문자에 대해 고정 2바이트를 사용하는 반면, UTF8은 1~3바이트까지 가변으로 사용하기 때문이다.

나머지 속성은 VARCHAR와 동일하므로, 더 상세한 설명은 VARCHAR 타입을 참조한다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

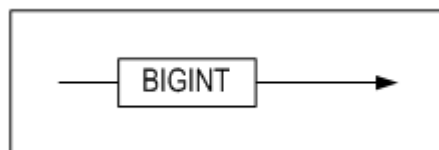
## 숫자형 데이터 타입

Altibase는 다음의 숫자형 데이터 타입을 지원한다.

- BIGINT
- DECIMAL
- DOUBLE
- FLOAT
- INTEGER
- NUMBER
- NUMERIC
- REAL
- SMALLINT

## BIGINT

### 흐름도



## 구문

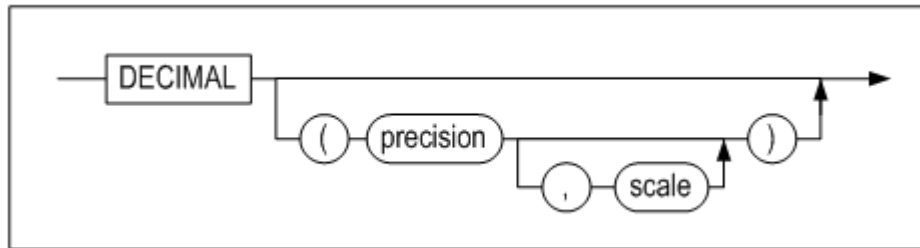
```
BIGINT
```

**설명**

8 바이트 정수형 데이터 타입이다.

C언어의 long(64 bit 시스템에서) 또는 long long(32 bit 시스템에서)과 동일하다.

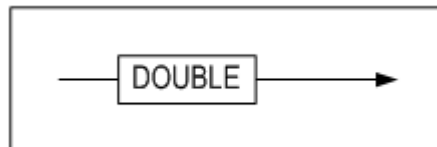
범위:  $-2^{63} + 1(-9223372036854775807) \sim 2^{63} - 1(9223372036854775807)$

**DECIMAL****흐름도****구문**

```
DECIMAL [(precision[, scale])]
```

**설명**

DECIMAL은 NUMERIC 데이터 타입과 동일하다.

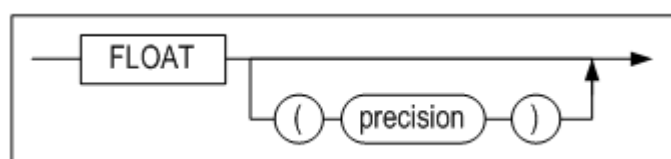
**DOUBLE****흐름도****구문**

```
DOUBLE
```

**설명**

8 바이트 부동 소수점 숫자형 데이터 타입이다.

C언어의 double과 동일한 데이터 타입이다.

**FLOAT****흐름도**

**구문**

```

FLOAT [(precision)]

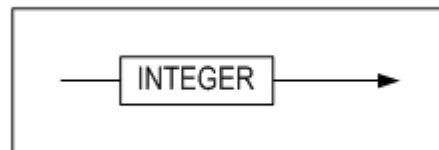
```

**설명**

-1E-120에서 1E+120까지 내의 부동 소수점 숫자 데이터 타입이다.

*Precision*은 정밀도 표시하기 위해 부동 소수점 숫자의 가수를 유효숫자 표기법으로 저장하는 데 사용되는 유효숫자의 자릿수이다.

*Precision*의 범위는 1에서 38까지이며, 39번째 자릿수에서 값이 반올림된다. *Precision*이 생략되면 기본값으로 38이 설정된다.

**INTEGER****흐름도****구문**

```

INTEGER

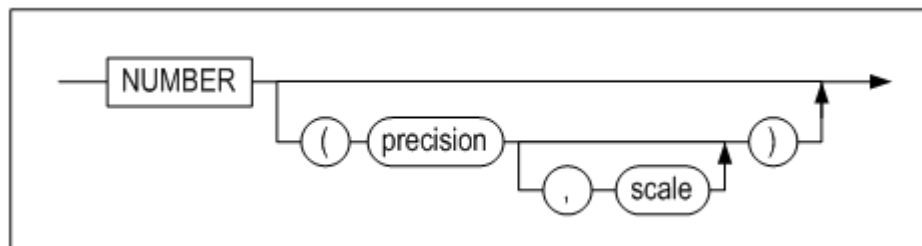
```

**설명**

4 바이트 크기의 정수형 데이터 타입이다.

C언어의 int와 동일한 데이터 타입이다.

-2,147,483,647에서 2,147,483,647까지의 정수값을 가질 수 있다.

**NUMBER****흐름도****구문**

```

NUMBER [(precision, scale)]

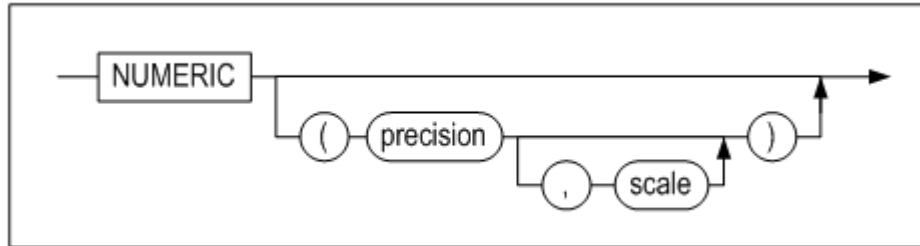
```

## 설명

NUMERIC 데이터 타입의 alias이다. 단, *precision*과 *scale*이 명시되지 않으면 FLOAT 데이터 타입과 동일하게 취급된다. FLOAT 데이터 타입은 39번째 자릿수에서 값이 반올림된다.

## NUMERIC

### 흐름도



### 구문

```
NUMERIC [(precision, scale)]
```

## 설명

NUMERIC은 Precision과 scale을 가지는 숫자형 데이터 타입으로 precision 만큼의 유효 숫자와 scale 만큼의 소수점 이하 정밀도를 가지는 고정 소수점형이다. FLOAT 데이터 타입이 실수를 표현하는 형식인 부동 소수점 형식인 반면 NUMERIC 데이터 타입은 precision과 scale이 모두 생략되면 precision은 38, scale은 0인 정수를 표현하는 고정 소수점형으로 사용된다.

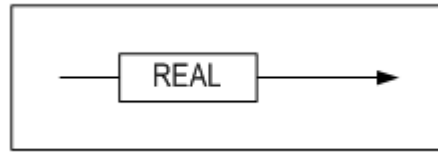
- Precision은 1부터 38까지의 값을 명시할 수 있다.
- Scale은 -84에서 128까지의 값을 명시할 수 있다.
- Precision이 생략되면 기본값으로 38이 설정된다.
- Scale이 생략되면 기본값으로 0이 설정된다.

다음은 각각 정의된 NUMERIC 타입의 칼럼에 입력 값이 1234567.89일때의 변환된 값을 보여준다.

- NUMERIC => 1234568
- NUMERIC(9) => 1234568
- NUMERIC(9, 2) => 1234567.89
- NUMERIC(9, 1) => 1234567.9
- NUMERIC(6) => precision 초과
- NUMERIC(7, -2) => 1234500
- NUMERIC(7, 2) => precision 초과

## REAL

### 흐름도



## 구문

REAL

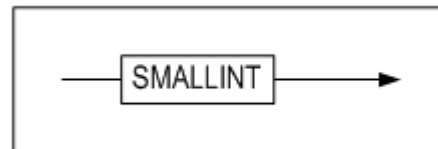
## 설명

4 바이트 크기의 부동 소수점형이다.

C언어의 float과 동일한 데이터 타입이다.

## SMALLINT

### 흐름도



## 구문

SMALLINT

## 설명

2 바이트 크기의 정수형 데이터 타입이다.

C언어의 short와 동일한 데이터 타입이다.

$-2^{15} + 1$  (-32,767)에서  $2^{15} - 1$  (32,767) 까지의 정수값을 가질 수 있다.

## 숫자형 데이터 형식

TO\_CHAR 나 TO\_NUMBER 등의 타입 변환 함수를 사용할 때 숫자형 데이터에 대하여 다음과 같이 형식을 지정할 수 있다. 숫자 데이터 형식은 하나 이상의 숫자를 표시하는 요소로 구성된다. 이장에서는 각각의 요소와 관련된 데이터 형식의 예를 설명한다.

### , (쉼표)

#### 설명

지정한 위치에 쉼표를 출력한다. 쉼표는 여러 번 사용할 수 있다.

#### 제한사항

쉼표는 숫자의 끝, 마침표의 오른쪽, 또는 숫자의 맨 앞자리에 올 수 없다.

**예제**

```

iSQL> SELECT TO_CHAR (1234, '99,99') FROM dual;
TO_CHAR (1234, '99,99')
-----
12,34
1 row selected.

iSQL> SELECT TO_NUMBER ( '12,34', '99,99') FROM dual;
TO_NUMBER ( '12,34', '99,99')
-----
1234
1 row selected.

```

**.(마침표)****설명**

지정한 위치에 마침표를 추가로 반환한다.

**제한사항**

한 숫자 내에서 마침표는 한번만 사용할 수 있다.

**예제**

```

iSQL> SELECT TO_CHAR (1.234, '99.999') FROM dual;
TO_CHAR (1.234, '99.999')
-----
1.234
1 row selected.

iSQL> SELECT TO_NUMBER ( '1.234', '99.999') FROM dual;
TO_NUMBER ( '1.234', '99.999')
-----
1.234
1 row selected.

```

**\$****설명**

숫자 앞에 \$ 기호를 붙인다.

## 예제

```
iSQL> SELECT TO_CHAR (123, '$9999') FROM dual;
TO_CHAR (123, '$9999')
-----
$123
1 row selected.

iSQL> SELECT TO_NUMBER ( '$0123', '09$99') FROM dual;
TO_NUMBER ( '$0123', '09$99')
-----
123
1 row selected.
```

## 0(숫자 0)

### 설명

정수 부분의 유효 자리수가 실제 숫자의 자리수 보다 많을 경우 실제 숫자 앞에 0을 채워서 반환한다. 그 외의 특성은 9와 같다.

### 예제

```
iSQL> SELECT TO_CHAR (123, '0999') FROM dual;
TO_CHAR (123, '0999')
-----
0123
```

## 9(숫자 9)

### 설명

출력할 숫자의 자릿수를 숫자 9를 이용해서 표시한다. 실제 숫자의 자릿수 보다 9의 개수가 더 많으면 앞에 공백문자를 출력하여 길이를 맞추고, 정수 부분의 9의 개수가 실제 숫자보다 더 적으면 숫자의 길이만큼 #를 출력한다. #의 개수는 사용자가 지정한 형식에 쓰인 문자의 개수 + 1(부호문자)이다.

9사이에 오는 마침표는 숫자의 정수 부분과 소수 부분을 구분하여 출력하게 한다.

첫 번째 인자에 소수가 있는데 사용자가 지정한 형식에 소수를 표현하는 부분이 없거나, 더 적으면 반올림해서 사용자가 지정한 형식의 소수 부분의 길이에 맞춘다.

### 예제

```
iSQL> SELECT TO_CHAR (123, '99999') FROM dual;
TO_CHAR (123, '99999')
-----
123

iSQL> SELECT TO_CHAR (123.55, '999') FROM dual;
TO_CHAR (123.55, '999')
-----
124
1 row selected.
```



```

iSQL> SELECT TO_CHAR (123.4567, '999999') FROM dual;
TO_CHAR (123.4567, '999999')
-----
      123
1 row selected.

```

```

iSQL> SELECT TO_CHAR (1234.578, '9999.99') FROM dual;
TO_CHAR (1234.578, '9999.99')
-----
    1234.58
1 row selected.

```

```

iSQL> SELECT TO_CHAR (1234.578, '999.99999') FROM dual;
TO_CHAR (1234.578, '999.99999')
-----
#####
1 row selected.

```

```

iSQL> SELECT TO_NUMBER ( '123', '999999') FROM dual;
TO_NUMBER ( '123', '999999')
-----
      123
1 row selected.

```

```

iSQL> SELECT TO_NUMBER ( '1234.58', '9999.99') FROM dual;
TO_NUMBER ( '1234.58', '9999.99')
-----
    1234.58
1 row selected.

```

## FM

### 설명

출력 문자열의 왼쪽 부분의 공백이나 0을 제거한다.

### 예제

```

iSQL> select to_char(00123.100,'99999.999') from dual;
TO_CHAR(00123.100,'99999.999')
-----
    123.100
1 row selected.
iSQL> select to_char(00123.100,'FM99999.999') from dual;
TO_CHAR(00123.100,'FM99999.999')
-----
    123.100
1 row selected.

```

**B****설명**

결괏값이 0일 경우, 0을 공백(Blank)으로 반환한다.

**예제**

```
iSQL> SELECT TO_CHAR (0.4, 'B9') FROM T1;
TO_CHAR (0.4, 'B9')
-----
1 row selected.
```

**C****설명**

ISO 통화 기호(NLS\_ISO\_CURRENCY 프로퍼티에 설정된 값)를 지정한 위치에 반환한다.

**예제**

```
iSQL> SELECT TO_CHAR (4000, 'C9999') FROM dual;
TO_CHAR (4000, 'C9999')
-----
KRW4000
1 row selected.
```

**D****설명**

소수점 문자(NLS\_NUMERIC\_CHARACTER 프로퍼티에 설정된 값)를 지정한 위치에 반환한다. 기본값은 마침표(.)이다.

**제한사항**

숫자 형식을 명시할 때 한 개의 소수점 문자만 포함할 수 있다.

**예제**

```
iSQL> SELECT TO_CHAR (24.06, '99D99') FROM dual;
TO_CHAR (24.06, '99D99')
-----
24.06
1 row selected.
iSQL> SELECT TO_CHAR (206, '999D99') FROM dual;
TO_CHAR (206, '999D99')
-----
206.00
1 row selected.
```

**EEEE****설명**

입력 받은 숫자를 지수 표기법을 이용하여 표기한다.

**제한사항**

EEEE는 항상 오른쪽 끝에 와야 한다. 단 S, PR, MI보다는 왼쪽에 오는 것이 가능하다. 쉼표와 같이 사용할 수 없다. TO\_NUMBER 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (1234, '9.9EEEE') FROM dual;
TO_CHAR (1234, '9.9EEEE')
-----
1.2E+03
1 row selected.
```

**G****설명**

그룹 구분자(NLS\_NUMERIC\_CHARACTER 프로퍼티에 설정된 값)를 지정한 위치에 반환한다. 하나의 숫자형 데이터 형식에 그룹 구분자를 여러 번 지정할 수 있다.

**제약사항**

그룹 구분자는 하나의 숫자형 데이터 형식에서 소수점 문자 또는 마침표의 오른쪽에 올 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (2534.3, '999G999D99') FROM dual;
TO_CHAR (2534.3, '999G999D99')
-----
2,534.30
1 row selected.
```

**L****설명**

지역 통화 기호(NLS\_CURRENCY 프로퍼티에 설정된 값)를 지정한 위치에 반환한다.

**예제**

```
iSQL> SELECT TO_CHAR (4000, 'L9999') FROM dual;
TO_CHAR (4000, 'L9999')
-----
?4000
1 row selected.
```

**MI****설명**

MI를 숫자 표현 형식의 오른쪽 끝에 사용하면 입력 받은 수가 음수일 경우 마이너스(-) 기호를 숫자 끝에 붙여서 반환한다. 양수일 경우에는 공백문자가 들어간다.

**제한사항**

MI는 항상 숫자 형식 표현의 오른쪽 끝에 와야 한다. S, PR과 같이 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (-123, '999MI') FROM dual;
TO_CHAR (-123, '999MI')
-----
123-
1 row selected.

iSQL> SELECT TO_NUMBER ( '123-', '999MI') FROM dual;
TO_NUMBER ( '123-', '999MI')
-----
-123
1 row selected.
```

**PR****설명**

PR를 숫자 표현 형식의 오른쪽 끝에 사용하면 입력 받은 수가 음수일 경우 마이너스 기호(-) 대신 <숫자> 형태로 출력된다.

**제한사항**

PR은 항상 숫자 표현 형식의 오른쪽 끝에 와야 한다. S, MI와 같이 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (-123, '999PR') FROM dual;
TO_CHAR (-123, '999PR')
-----
<123>
1 row selected.

iSQL> SELECT TO_NUMBER ( '<123>', '999PR') FROM dual;
TO_NUMBER ( '<123>', '999PR')
-----
-123
1 row selected.
```

**RN****설명**

입력 받은 수를 로마 숫자로 변환한다. 입력할 수 있는 수는 1부터 3999까지이다. 숫자 표현 형식을 소문자 rn으로 사용하면 로마 숫자가 소문자로 출력된다.

**제한사항**

다른 숫자 형식과 같이 사용할 수 없다. TO\_NUMBER 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (14, 'RN') FROM dual;
TO_CHAR (14, 'RN')
-----
XIV
1 row selected.
```

**S****설명**

숫자 표현 형식의 처음이나 끝에 와서 입력 받은 수의 기호에 따라서 마이너스(-) 또는 플러스(+) 기호를 붙인다.

**제한사항**

S는 숫자 표현 형식의 맨 앞이나 맨 뒤에만 올 수 있다. MI, PR과 같이 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (123, 'S999.99') FROM dual;
TO_CHAR (123, 'S999.99')
-----
+123.00
1 row selected.

iSQL> SELECT TO_CHAR (-123, '999.99S') FROM dual;
TO_CHAR (-123, '999.99S')
-----
123.00-
1 row selected.

iSQL> SELECT TO_NUMBER ( '+123', 'S999.99') FROM dual;
TO_NUMBER ( '+123', 'S999.99')
-----
123
1 row selected.

iSQL> SELECT TO_NUMBER ( '123.00-', '999.99S') FROM dual;
TO_NUMBER ( '123.00-', '999.99S')
-----
-123
1 row selected.
```

**V****설명**

V 다음에 있는 9의 개수와 10을 곱하고 그 값을 인자로 받은 숫자와 곱한다. V 앞의 9의 개수는 첫 번째 인자의 유효숫자의 개수를 의미한다.

**제한사항**

마침표와 같이 사용할 수 없다. TO\_NUMBER 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR (12, '99V99') FROM dual;
TO_CHAR (12, '99V99')
-----
1200
1 row selected.

iSQL> SELECT TO_CHAR (1200, '99V99') FROM dual;
TO_CHAR (1200, '99V99')
-----
#####
1 row selected.

iSQL> SELECT TO_CHAR (-123.456, '999V999EEEEMI') from dual;
TO_CHAR (-123.456, '999V999EEEEMI')
-----
1235E+02-
1 row selected.
```

**XXXX****설명**

입력 받은 수를 16진수로 변환한다. 만약 정수가 아니라면 반올림하여 16진수로 변환한다. xxxx는 16진수 중 문자를 소문자로 반환한다.

**제한사항**

다른 숫자 표현 형식과 같이 사용할 수 없다. 변환할 수는 0 이상이어야 한다.

**예제**

```
iSQL> SELECT TO_CHAR (123, 'XXXX') FROM dual;
TO_CHAR (123, 'XXXX')
-----
7B
1 row selected.

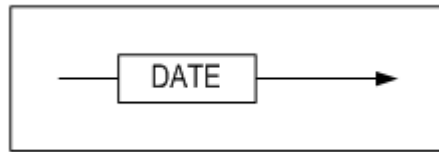
iSQL> SELECT TO_NUMBER ('ABC', 'XXXX') FROM dual;
TO_NUMBER ('ABC', 'XXXX')
-----
2748
1 row selected.
```

## 날짜형 데이터 타입

날짜형 타입은 날짜와 시간 데이터를 저장하는데 사용된다.

### DATE

#### 흐름도



#### 구문

DATE

#### 설명

8바이트 크기의 날짜 값을 저장하는 데이터 타입이다.

저장할 수 있는 날짜의 범위는 시스템에 따라 다르다. 일반적으로 1년 1월 1일부터 9999년 12월 31일의 범위 내에서 사용 가능하다.

날짜 값은 날짜형 데이터 형식을 사용해서 다양한 포맷으로 출력할 수 있다.

### 날짜형 데이터 형식

날짜형 데이터 타입의 데이터는 데이터베이스 내부적으로는 숫자형 데이터로 관리하지만 사용자는 TO\_CHAR또는 TO\_DATE 변환 함수 등을 사용해서 문자열로 표시할 수 있다. 변환 함수를 사용할 때 사용자는 보고자 하는 형식에 맞게 날짜형 데이터 형식 문자열을 지정해 주어야 한다.

날짜형 데이터 형식은 다음과 같은 기본요소들로 구성된다.

- AM, PM
- SCC, CC
- D, DD, DDD, DAY, DY
- HH, HH12, HH24
- MM, MON, MONTH
- MI
- Q
- SS, SSSSS, SSSSSS, SSSSSSSS, FF[1..6]
- WW, WW2, W, IW
- Y, YYYY
- SYYYY, YYYY, YYY, YY, Y, RR, RRRR
- IYYY, IYY, IY, I

위의 기본 요소들과 함께 다음의 다음의 구두점과 특수 문자들도 날짜형 데이터 형식을 구성하는 요소이다.

- 하이픈(-)

- 슬래시(/)
- 쉼표(,)
- 마침표(.)
- 콜론(:)
- 홑따옴표(')

각각의 기본 요소들이 의미하는 바와 활용 예를 다음에서 살펴보자.

## AM, PM

### 설명

정오를 기준으로 오전/오후를 구분한다. ('AM' 또는 'PM')

### 예제

```
iSQL> SELECT TO_CHAR ( TO_DATE( '13', 'HH' ), 'AM' ) FROM dual;
TO_CHAR ( TO_DATE( '13', 'HH' ), 'AM' )
-----
PM
1 row selected.

iSQL> SELECT TO_DATE('1980-12-28 PM', 'YYYY-MM-DD AM') FROM dual;
TO_DATE('1980-12-28 PM', 'YYYY-MM-DD AM'
-----
1980/12/28 12:00:00
1 row selected.
```

## SCC

### 설명

세기를 표시한다.

- 4자리 년도 중 뒤의 2자리 값이 01~99이면, 4자리 년도 중 앞의 두 자리의 값에 1을 더해서 반환한다.
- 4자리 년도 중 뒤의 2자리 값이 00이면, 4자리 년도 중 앞의 두 자리 값을 그대로 반환한다.

기원전은 년도 앞에 '-'가 표시된다.

- 년도가 0000인 경우 기원전 1년이며, -0001인 경우 기원전 2년이다.
- 0000~ -0099는 -1세기이며, -01로 표시된다.

TO\_DATE 함수에서는 인자로 사용할 수 없다.



**예제**

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'SCC' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'SCC' )
-----
20
1 row selected.

iSQL> SELECT TO_CHAR ( DATE'01-JAN-0001' - 1, 'SCC' ) FROM dual;
TO_CHAR ( DATE'01-JAN-0001' - 1, 'SCC' )
-----
-01
1 row selected.

```

**CC****설명**

세기를 표시한다.

- 4자리 년도 중 뒤의 2자리 값이 01~99이면, 4자리 년도 중 앞의 두 자리의 값에 1을 더해서 반환한다.
- 4자리 년도 중 뒤의 2자리 값이 00이면, 4자리 년도 중 앞의 두 자리 값을 그대로 반환한다.

TO\_DATE 함수에서는 인자로 사용할 수 없다.

**예제**

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'CC' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'CC' )
-----
20
1 row selected.

```

**D****설명**

일주일 중 몇 번째 날인지를 나타내는 1 ~ 7까지의 숫자이다. 일요일부터 1로 시작한다.

TO\_DATE 함수에서 인자로 사용할 수 없다.

**예제**

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'D' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'D' )
-----
1
1 row selected.

```

**DAY****설명**

요일의 영문이름을 나타낸다. (SUNDAY, MONDAY,...)

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'DAY' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'DAY' )
-----
SUNDAY
1 row selected.
```

**DD****설명**

한달 중 몇 번째 날인지를 나타낸다. (1 ~ 31)

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'DD' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'DD' )
-----
28
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE( '1980-12-28', 'YYYY-MM-DD') FROM dual;
TO_DATE( '1980-12-28', 'YYYY-MM-DD')
-----
1980/12/28 00:00:00
1 row selected.
```

**DDD****설명**

일 년 중 몇 번째 날인지를 나타낸다. (1 ~ 366)

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'DDD' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'DDD' )
-----
363
1 row selected.
```

**DY****설명**

요일의 이름을 약자로 나타낸다. (SUN, MON, TUE, ...)

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'DY' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'DY' )
-----
SUN
1 row selected.
```

**FF [1..6]****설명**

FF다음의 1 ~ 6까지의 숫자를 이용하여 마이크로초의 자리 수를 나타낸다. (0 ~ 999999). FF 형식은 FF6과 같은 같은 결과를 반환한다.

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR ( SYSDATE, 'FF5' ) FROM dual;
TO_CHAR ( SYSDATE, 'FF5' )
-----
34528
1 row selected.
```

**HH, HH24****설명**

시간을 24시간 단위로 나타낸다.(0 ~ 23)

**예제**

```
iSQL> SELECT TO_CHAR ( TO_DATE( '2008-12-28 17:30:29', 'YYYY-MM-DD HH:MI:SS' ), 'HH' ) FROM
dual;
TO_CHAR ( TO_DATE( '2008-12-28 17:30:29'
-----
17
1 row selected.

iSQL> SELECT TO_CHAR ( TO_DATE( '2008-12-28 17:30:29', 'YYYY-MM-DD HH24:MI:SS' ), 'YYYY-MM-
DD HH24:MI:SS' ) FROM dual;
TO_CHAR ( TO_DATE( '2008-12-28 17:30:29',
-----
2008-12-28 17:30:29
1 row selected.
```

## HH12

### 설명

시간을 12시간 단위로 나타낸다.(1 ~ 12)

### 예제

```

iSQL> SELECT TO_CHAR ( TO_DATE( '2008-12-28 17:30:29', 'YYYY-MM-DD HH:MI:SS' ), 'HH12' )
FROM dual;
TO_CHAR ( TO_DATE( '2008-12-28 17:30:29',
-----
05
1 row selected.

```

```

iSQL> SELECT TO_CHAR( TO_DATE ( '08-12-28 05:30:29', 'RR-MM-DD HH12:MI:SS' ), 'RR-MM-DD
HH12:MI:SS') FROM dual;
TO_CHAR( TO_DATE ( '08-12-28 05:30:29', 'R
-----
08-12-28 05:30:29
1 row selected.

```

## MI

### 설명

분 (0 ~ 59)

### 예제

```

iSQL> SELECT TO_CHAR ( TO_DATE( '1980-12-28 17:30:29', 'YYYY-MM-DD HH:MI:SS' ), 'HH' ) FROM
dual;
TO_CHAR ( TO_DATE( '1980-12-28 17:30:29'
-----
17
1 row selected.

```

-- 아래 출력은 DEFAULT\_DATE\_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.

```

iSQL> SELECT TO_DATE ( '05-12-28 14:30:29', 'RR-MM-DD HH:MI:SS' ) FROM dual;
TO_DATE ( '05-12-28 14:30:29', 'RR-MM-DD
-----
2005/12/28 14:30:29
1 row selected.

```

## MM

### 설명

월 (01 ~ 12)

## 예제

```

iSQL> SELECT TO_CHAR ( TO_DATE( '1980-12-28 17:30:29', 'YYYY-MM-DD HH:MI:SS' ), 'HH' ) FROM
dual;
TO_CHAR ( TO_DATE( '1980-12-28 17:30:29'
-----
17
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '05-12-28 14:30:29', 'RR-MM-DD HH:MI:SS' ) FROM dual;
TO_DATE ( '05-12-28 14:30:29', 'RR-MM-DD
-----
2005/12/28 14:30:29
1 row selected.

```

## MON

## 설명

월의 이름을 약자로 표시한다.(JAN, FEB, MAR, ...)

## 예제

```

SQL> SELECT TO_CHAR (TO_DATE ('1995-12-05', 'YYYY-MM-DD'), 'MON') FROM dual;
TO_
---
DEC

```

## MONTH

## 설명

월의 이름을 표시한다.(JANUARY, FEBRUARY, ...)

## 예제

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'Month' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'Month' )
-----
December
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '05-APRIL-28 14:30:29', 'RR-MONTH-DD HH:MI:SS' ) FROM dual;
TO_DATE ( '05-APRIL-28 14:30:29', 'RR-MO
-----
2005/04/28 14:30:29
1 row selected.

```

## Q

### 설명

분기를 표시한다. (1 ~ 4)

TO\_DATE 함수에서 사용할 수 없다.

### 예제

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'Q' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'Q' )
-----
4
1 row selected.
```

## RM

### 설명

로마 숫자로 월을 나타낸다. (I, II, III, IV,...)

### 예제

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'RM' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'RM' )
-----
XII
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '28-V-1980', 'DD-RM-YYYY' ) FROM dual;
TO_DATE ( '28-V-1980', 'DD-RM-YYYY' )
-----
1980/05/28 00:00:00
1 row selected.
```

## RR

### 설명

년도를 두자리 정수로 표시한다. 날짜를 표기할 때, 두 자리만 표기한 경우 50미만인 경우에만 21세기라고 가정하여 2000을 더하고, 50 경우에는 1900을 더해서 연도를 표시한다. 따라서 표시 가능한 년도는 1950 ~ 2049까지 이다.

## 예제

```

iSQL> SELECT TO_CHAR ( '28-DEC-80', 'RR' ) FROM dual;
TO_CHAR ( '28-DEC-80', 'RR' )
-----
80
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '28-DEC-80', 'DD-MON-RR' ) FROM dual;
TO_DATE ( '28-DEC-80', 'DD-MON-RR' )
-----
1980/12/28 00:00:00
1 row selected.

```

## RRRR

## 설명

연도 (0 ~ 9999)

네자리, 두자리의 년도를 모두 입력으로 받아서, 숫자가 50미만인 경우 2000을 더하고, 50이상 100 미만인 경우 1900을 더해서 연도를 나타낸다. 4자리의 숫자인 경우 그대로가 년도로 표시된다.

## 예제

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'RRRR' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'RRRR' )
-----
1980
1 row selected.

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'DD-MON-RRRR' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'DD-MON-RRRR' )
-----
28-DEC-1980
1 row selected.

```

## SS

## 설명

초 (0 ~ 59)

## 예제

```

iSQL> SELECT TO_CHAR ( TO_DATE( '1980-12-28 17:30:29', 'YYYY-MM-DD HH:MI:SS' ), 'HH' ) FROM
dual;
TO_CHAR ( TO_DATE( '1980-12-28 17:30:29'
-----
17
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '05-12-28 14:30:29', 'RR-MM-DD HH:MI:SS' ) FROM dual;
TO_DATE ( '05-12-28 14:30:29', 'RR-MM-DD
-----
2005/12/28 14:30:29
1 row selected.

```

## SSSSS

## 설명

자정부터 현재까지 몇 초가 경과 되었는지 나타낸다. (0 ~ 86399)

## 예제

```

iSQL> SELECT TO_CHAR ( TO_DATE( '1980-12-28 17:30:29', 'YYYY-MM-DD HH24:MI:SS' ), 'SSSSS' )
FROM dual;
TO_CHAR ( TO_DATE( '1980-12-28 17:30:29'
-----
63029
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE('1980-12-28 12345', 'YYYY-MM-DD SSSSS') FROM dual;
TO_DATE('1980-12-28 12345', 'YYYY-MM-DD
-----
1980/12/28 03:25:45
1 row selected.

```

## SSSSSS

## 설명

날짜 데이터 타입의 값의 마이크로초를 표시한다.(0 ~ 999999)



## 예제

```

iSQL> SELECT TO_CHAR (SYSDATE, 'SSSSSS') FROM dual;
TO_CHAR (SYSDATE, 'SSSSSS')
-----
490927
1 row selected.

iSQL> SELECT TO_CHAR ( TO_DATE('1980-12-28 123456', 'YYYY-MM-DD SSSSSS'), 'SSSSSS' ) FROM
dual;
TO_CHAR ( TO_DATE('1980-12-28 123456', '
-----
123456
1 row selected.

```

## SSSSSSSS

### 설명

초 + 마이크로초를 나타낸다. 앞의 2개의 숫자는 초를 나타내고, 나머지 6개의 숫자가 마이크로초를 나타낸다. (0 ~ 599999999)

### 예제

```

iSQL> SELECT TO_CHAR (SYSDATE, 'SSSSSSSS') FROM dual;
TO_CHAR (SYSDATE, 'SSSSSSSS')
-----
48987403
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '12.345678', 'ss.SSSSSS') FROM dual;
TO_DATE ( '12.345678', 'ss.SSSSSS')
-----
2005/12/01 00:00:12
1 row selected.

iSQL> SELECT TO_CHAR( TO_DATE( '12.345678', 'ss.SSSSSS'), 'SSSSSS') FROM dual;
TO_CHAR( TO_DATE( '12.345678', 'ss.SSSSS
-----
345678
1 row selected.

```

## WW

### 설명

일 년 중 몇 번째 주인지를 나타낸다. 1월 1일부터 그 주의 토요일까지가 그 해의 첫 번째 주이다. (1 ~ 54)

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'ww' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'ww' )
-----
53
1 row selected.

```

**WW2****설명**

요일과 상관없이 일 년 중 몇 번째 주인지를 나타낸다. 첫 번째 주는 1월 1일부터 시작하며, 7일 단위로 구분한다. (1~53주)

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'ww2' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'ww2' )
-----
52
1 row selected.

```

**W****설명**

한 달 중 몇 번째 주인지를 나타낸다. 1일부터 그 주의 토요일까지가 그 달의 첫 번째 주이다. (1 ~ 6)

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'w' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'w' )
-----
5
1 row selected.

```

**IW****설명**

ISO 8601에 따라 일 년 중 몇 번째 주인지를 나타낸다(1~52 또는 1~53). 주의 시작은 월요일이며, 첫 번째 주는 해당 연도의 첫 번째 목요일(1월 4일)을 포함한다.

TO\_DATE 함수에서 사용할 수 없다.

- 2012.12.31(월) ~ 2013.01.06(일) : 2013년 01주
- 2014.12.29(월) ~ 2015.01.04(일) : 2015년 01주
- 2015.12.28(월) ~ 2016.01.03(일) : 2015년 53주
- 2016.12.26(월) ~ 2017.01.01(일) : 2016년 52주

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'IW' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'IW' )
-----
52
1 row selected.
```

**Y,YYY****설명**

연도를 나타내는 숫자 중 임의의 위치에 ,(comma)를 삽입할 수 있다. 맨 앞이나 뒤에 와도 상관 없다.

TO\_DATE 함수에서 사용할 수 없다.

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'Y,YYY' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'Y,YYY' )
-----
1,980
1 row selected.
```

**SYYYYY****설명**

네 자리 숫자를 그대로 연도로 간주한다. 기원전은 '-'가 표시된다 (-9999 ~ 9999).

년도가 0000인 경우 기원전 1년, -0001은 기원전 2년을 나타낸다.

**예제**

```
iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'SYYYYY' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'SYYYYY' )
-----
1980
1 row selected.

iSQL> SELECT TO_CHAR ( DATE'01-JAN-0000' - 1, 'SYYYYY-MM-DD' ) FROM dual;
TO_CHAR ( DATE'01-JAN-0000' - 1, 'SYYYYY-MM
-----
-0001-12-31
1 row selected.
```

**YYYY****설명**

네 자리 숫자를 그대로 연도로 간주한다. (0 ~ 9999)

## 예제

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'YYYY' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'YYYY' )
-----
1980
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '28-DEC-1980', 'DD-MON-YYYY' ) FROM dual;
TO_DATE ( '28-DEC-1980', 'DD-MON-YYYY' )
-----
1980/12/28 00:00:00
1 row selected.

```

## YY

## 설명

연도의 마지막 숫자 2자리를 반환한다. 21세기라고 가정하고 2000을 더한 값을 연도로 간주한다. (2000 ~ 2099)

## 예제1

```

iSQL> SELECT TO_CHAR ( '28-DEC-1980', 'YY' ) FROM dual;
TO_CHAR ( '28-DEC-1980', 'YY' )
-----
80
1 row selected.

-- 아래 출력은 DEFAULT_DATE_FORMAT 프로퍼티가 YYYY/MM/DD HH:MI:SS 일 때 예이다.
iSQL> SELECT TO_DATE ( '28-DEC-80', 'DD-MON-YY' ) FROM dual;
TO_DATE ( '28-DEC-80', 'DD-MON-YY' )
-----
2080/12/28 00:00:00
1 row selected.

```

## 예제2

```

iSQL> CREATE TABLE timetbl(i1 INTEGER, t1 DATE, etc VARCHAR(10));
Create success.

iSQL> INSERT INTO timetbl VALUES (1, SYSDATE, 'Start');
1 row inserted.

iSQL> INSERT INTO timetbl VALUES (2, TO_DATE('2003-02-20 12:15:50', 'YYYY-MM-DD HH:MI:SS'),
'The end');
1 row inserted.

iSQL> SELECT TO_CHAR(T1, 'YYYY YY MM MON Mon mon DD HH MI SS SSSSSS D DDD') Date_format FROM
timetbl WHERE I1 = 2;
DATE_FORMAT
-----
2003 03 02 FEB Feb feb 20 12 15 50 000000 5 051

```

1 row selected.

## RR, RRRR, YY, YYYY 비교

각 형식 요소에 대한 설명을 참고한다.

- [YYYY]: 숫자를 그대로 연도로 간주

'23-FEB-5': 0005년 2월 23일

'23-FEB-05': 0005년 2월 23일

'23-FEB-2005':

'23-FEB-95': 0095년 2월 23일

- [YY]: 2000 + YY

'23-FEB-5':

'23-FEB-05':

'23-FEB-2005': 예러

'23-FEB-95':

- [RRRR]: 4자리 숫자를 그대로 연도로 간주, 숫자가 50 미만인 경우 2000을, 50 이상 100 미만인 경우 1900을 더한다.

'23-FEB-5':

'23-FEB-05':

'23-FEB-2005':

'23-FEB-95':

'23-FEB-100': 0100년 2월 23일

'23-FEB-0005': 0005년 2월 23일

- [RR]: 숫자가 50 미만인 경우 2000을, 50 이상 100 미만인 경우 1900을 더한다.

'23-FEB-5':

'23-FEB-05':

'23-FEB-2005': 예러

'23-FEB-95': 1995년 2월 23일

## YYY

### 설명

연도의 마지막 숫자 3자리를 반환한다. 21세기라고 가정하고 2000을 더한 값을 연도로 간주한다. (2000 ~ 2099)

**Y****설명**

연도의 마지막 숫자를 반환한다. 21세기라고 가정하고 2000을 더한 값을 연도로 간주한다. (2000 ~ 2099)

**IYYY, IYY, IY, I**

**설명**

ISO 8601 표준에 따른 연도를 나타낸다.

주의 시작은 월요일이며, 첫 번째 주는 해당 연도의 첫 번째 목요일을 포함한다.

IYYY는 ISO 표준 연도를 반환한다.

IYY는 ISO 표준 연도의 마지막 3자리를 반환한다.

IY는 ISO 표준 연도의 마지막 2자리를 반환한다.

I는 ISO 표준 연도의 마지막 1자리를 반환한다.

TO\_DATE 함수에서 사용할 수 없다.

- 2012.12.31(월) ~ 2013.01.06(일) : 2013년
- 2014.12.29(월) ~ 2015.01.04(일) : 2015년
- 2015.12.28(월) ~ 2016.01.03(일) : 2015년
- 2016.12.26(월) ~ 2017.01.01(일) : 2016년

**예제**

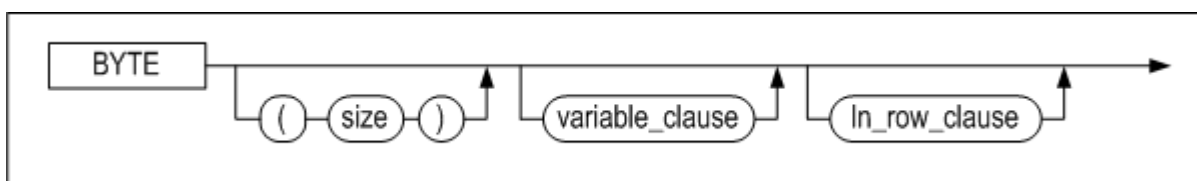
```
isql> SELECT TO_CHAR ( '1-JAN-2017', 'IYYY IYY IY I' ) FROM dual;
TO_CHAR('1-JAN-2017','IYYY IYY IY I')
-----
2016 016 16 6
1 row selected
```

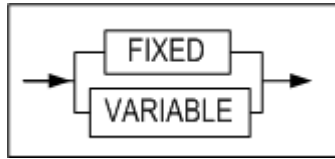
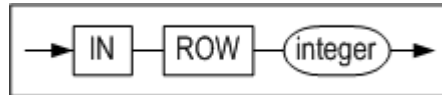
**이진 데이터 타입**

텍스트, 비디오, 및 공간 데이터 같은 비정형 데이터를 저장하는데 사용된다.

Altibase는 다음의 이진 데이터 타입을 지원한다.

- BYTE
- VARBYTE
- NIBBLE
- BIT
- VARBIT

**BYTE****흐름도**

**variable\_clause::=****in\_row\_clause::=**

## 구문

```
BYTE [(size)] [[FIXED |] VARIABLE ( IN ROW size ) ]
```

## 설명

명시된 크기만큼 고정된 길이를 가지는 이진 데이터 타입이다. 정의된 크기보다 짧은 길이의 문자열을 입력하면 입력된 데이터의 오른쪽이 '0'으로 채워진다.

BYTE 칼럼의 기본 크기는 1 바이트이다. 최대 길이는 32000 바이트이다. 데이터는 '0FAE13'과 같이 알파벳과 숫자 문자의 조합을 사용해서 16진수 형식으로 표현 가능하다. 이 때 사용 가능한 문자는 0에서 9까지, A에서 F까지의 문자이다.

BYTE 칼럼에 입력 또는 검색 시 반드시 정의한 크기에 맞추어야 한다. 1바이트에 2개의 문자를 입력할 수 있다. 예를 들어 BYTE(3) 이라고 정의 하였으면 '000000' 부터 'FFFFFF' 까지의 값을 입력할 수 있다.

소문자 'a'부터 'f'를 입력할 경우 대문자로 변환되어 저장된다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

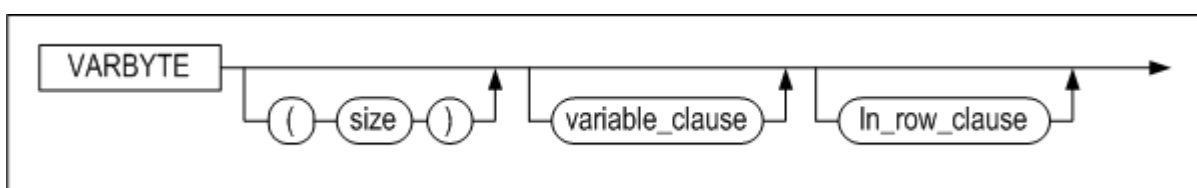
## 예제

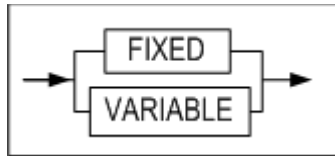
```

iSQL> CREATE TABLE T1 (I1 BYTE(1), I2 BYTE(5));
Create success.
iSQL> INSERT INTO T1 VALUES (BYTE'11', BYTE'0011');
1 row inserted.
iSQL> SELECT TO_CHAR(I1), TO_CHAR(I2) FROM T1;
TO_CHAR(I1)  TO_CHAR(I2)
-----
11  0011000000
1 row selected.
  
```

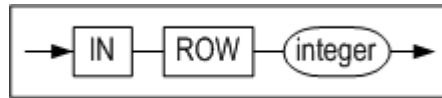
## VARBYTE

### 흐름도

**variable\_clause::=**



**in\_row\_clause::=**



**구문**

```
VARBYTE [(size)] [[FIXED |] VARIABLE ( IN ROW size ) ]
```

**설명**

가변 길이를 갖는 이진 데이터 타입이다. VARBYTE 칼럼의 기본 크기는 1바이트이며, 최대 길이는 32000바이트이다.

데이터는 '0FAE13'과 같이 알파벳과 숫자 문자의 조합을 사용해서 16진수 형식으로 표현 가능하다. 이 때 사용 가능한 문자는 0에서 9까지, A에서 F까지의 문자이다.

BYTE 칼럼에 입력 또는 검색 시 반드시 정의한 크기에 맞추어야 한다. 1바이트에 2개의 문자를 입력할 수 있다. 예를 들어 BYTE(3) 이라고 정의 하였으면 '000000' 부터 'FFFFFF' 까지의 값을 입력할 수 있다.

소문자 'a'부터 'f'를 입력할 경우 대문자로 변환되어 저장된다.

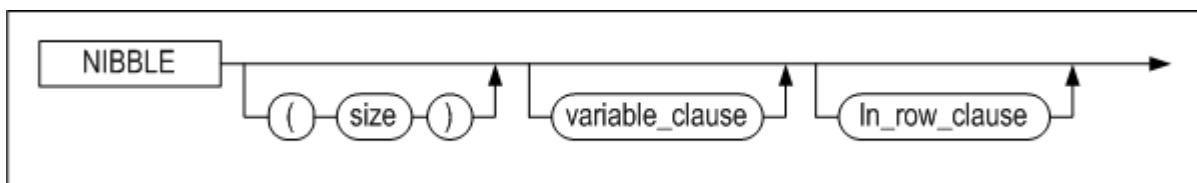
FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

**예제**

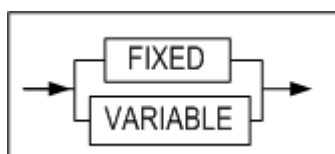
```
iSQL> CREATE TABLE T1 (I1 VARBYTE(1), I2 VARBYTE(5) );
Create success.
iSQL> INSERT INTO T1 VALUES (VARBYTE'11', VARBYTE'0011');
1 row inserted.
iSQL> SELECT TO_CHAR(I1), TO_CHAR(I2) FROM T1;
TO_CHAR(I1)  TO_CHAR(I2)
-----
11  0011
1 row selected.
```

## NIBBLE

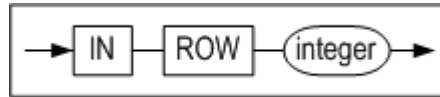
**흐름도**



**variable\_clause::=**





**in\_row\_clause::=****구문**

```
NIBBLE [(size)] [[FIXED |] VARIABLE ( IN ROW size ) ]
```

**설명**

명시된 크기만큼 가변 길이를 가지는 이진 데이터 타입이다.

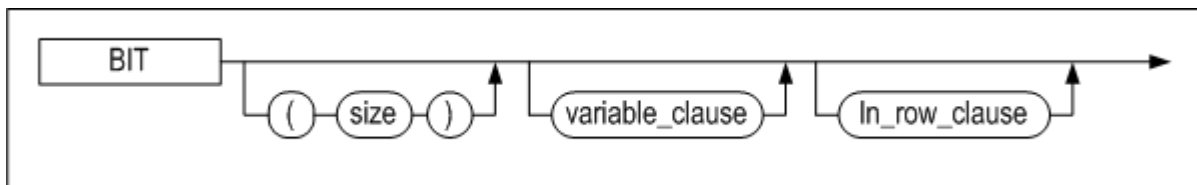
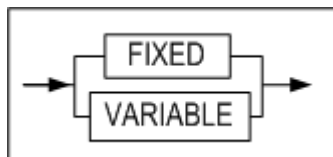
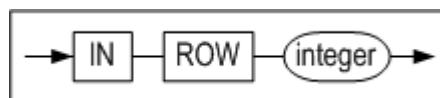
NIBBLE 칼럼의 기본 크기는 한 개의 문자 크기이며, 최대 254nibbles까지 허용된다.

데이터는 알파벳과 숫자 문자의 조합을 사용해서 16진수 형식으로 표현 가능하다. 이 때 사용 가능한 문자는 0에서 9 까지, A에서 F까지의 문자이다. BYTE와 달리 한 nibble에 한 개의 문자만을 입력할 수 있다.

예를 들어 NIBBLE(6) 이라고 정의 하였으면 '000000' 부터 'FFFFFF' 까지 입력할 수 있다.

소문자 'a'부터 'f'를 입력할 경우 대문자로 변환되어 저장된다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

**BIT****흐름도****variable\_clause::=****in\_row\_clause::=****구문**

```
BIT [(size)] [[FIXED |] VARIABLE ( IN ROW size ) ]
```

**설명**

고정 길이를 갖는 이진 데이터 타입으로, 데이터는 0과 1로만 이루어진다.

BIT 칼럼의 기본 크기는 1 비트이며, 최대 크기는 64000 비트이다.

정의된 크기보다 긴 문자열을 입력할 경우 'Invalid data type

length' 에러가 발생한다. 정의된 크기보다 짧은 길이의 문자열을 입력하면 입력 데이터의 오른쪽이 0으로 채워진다. 0과 1 이외의 값이 입력될 경우 'Invalid literal' 에러가 발생한다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

## 예제

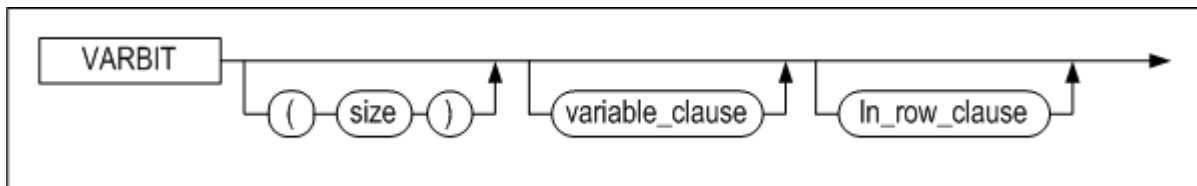
```

iSQL> CREATE TABLE T1 ( I1 BIT(1), I2 BIT(5) );
Create success.
iSQL> INSERT INTO T1 VALUES ( BIT'1', BIT'011' );
1 row inserted.
iSQL> SELECT TO_CHAR(I1), TO_CHAR(I2) FROM T1;
TO_CHAR(I1)  TO_CHAR(I2)
-----
1    01100
1 row selected.
iSQL> INSERT INTO T1 VALUES ( BIT'1111', BIT'011' );
[ERR-2100D : Invalid data type length]
iSQL> INSERT INTO T1 VALUES ( BIT'1', BIT'1234' );
[ERR-21011 : Invalid literal]

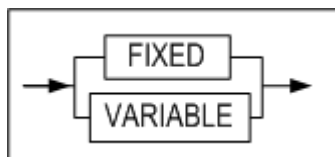
```

## VARBIT

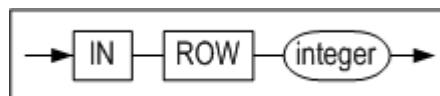
### 흐름도



**variable\_clause::=**



**in\_row\_clause::=**



### 구문

```
VARBIT [(size)] [[FIXED |] VARIABLE ( IN ROW size ) ]
```

### 설명

가변 길이를 갖는 이진 데이터 타입으로, 데이터는 0과 1로만 이루어진다.

BIT 칼럼의 기본 크기는 1Bit이며, 최대 크기는 64000비트이다. 정의된 크기보다 긴 문자열을 입력할 경우 'Invalid data type length' 에러가 발생한다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 "FIXED/VARIABLE 옵션"과 "IN ROW 절"을 참고한다.

## 예제

```

iSQL> CREATE TABLE T1 ( I1 VARBIT(1), I2 VARBIT(5) );
Create success.
iSQL> INSERT INTO T1 VALUES ( VARBIT'1', VARBIT'011' );
1 row inserted.
iSQL> SELECT TO_CHAR(I1), TO_CHAR(I2) FROM T1;
TO_CHAR(I1)  TO_CHAR(I2)
-----
1    011
1 row selected.
iSQL> INSERT INTO T1 VALUES ( VARBIT'1111', VARBIT'011' );
[ERR-2100D : Invalid data type length]
iSQL> INSERT INTO T1 VALUES ( VARBIT'1', VARBIT'1234' );
[ERR-21011 : Invalid literal]

```

## LOB 데이터 타입

### 개요

LOB(Large Object) 데이터 타입은 대용량 데이터를 저장할 수 있는 데이터 타입이다. 하나의 LOB칼럼에 저장 가능한 데이터의 크기는 최대 4GB-1byte 이다. 테이블을 생성할 때 다른 타입들과 달리 사용자가 LOB 칼럼의 크기를 명시할 필요가 없다. 그리고 하나의 테이블에 하나 이상의 LOB 타입 칼럼을 정의할 수 있다.

LOB 데이터 타입은 이미지, 동영상 파일들과 같은 이진 데이터를 저장하는 BLOB(Binary Large Object)과 문자열 데이터를 저장하는 CLOB(Character Large Object)으로 구분된다.

### LOB의 특징

Altibase가 제공하는 LOB은 다음과 같은 특징이 있다.

- 데이터 저장 기능
- 부분 읽기(Partial Read)
- 디스크 LOB 파티셔닝

#### 데이터 저장 기능

ODBC의 SQLPutLob 함수 또는 JDBC의 setBlob 또는 setClob 메소드를 이용하여 CLOB, BLOB 데이터를 저장할 수 있다.

#### 부분 읽기(Partial Read)

LOB 데이터의 특정 구간에 대한 데이터 조각을 읽는 기능이다. Altibase ODBC의 SQLGetLob 함수를 이용하여 특정 오프셋의 크기를 읽는다.

#### 디스크 LOB의 파티셔닝

디스크 LOB 데이터는 테이블이 속한 테이블스페이스가 아닌 다른 디스크 테이블스페이스로 저장이 가능하다. 이는 테이블 파티셔닝 방법과 유사하다.

## LOB 칼럼의 저장

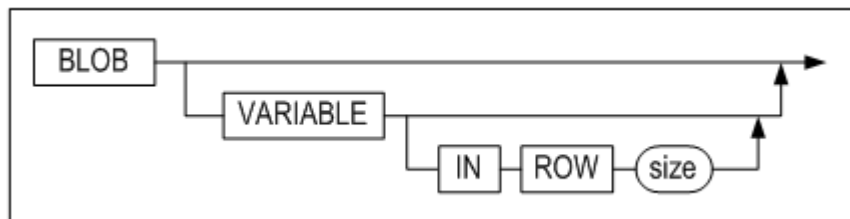
LOB 데이터는 대부분의 경우 레코드 영역 밖의 가변 영역에 저장된다. 또한 LOB 칼럼의 크기가 크지 않을 때에는 in row 옵션을 사용하여 레코드 영역(고정 영역) 안에 저장하기도 한다. 그러나 디스크 테이블의 LOB 데이터는 크기와 상관없이 항상 가변 영역에 저장된다.

가변 영역에 저장되는 LOB 칼럼의 데이터는 크기가 매우 크기 때문에, 레코드의 나머지가 속하는 테이블스페이스에 같이 저장되는 것은 공간 사용 측면에서 효율성이 떨어진다.

디스크 테이블의 경우 LOB 칼럼 데이터를 LOB 칼럼이 속한 테이블과 별도의 테이블스페이스에 저장할 수 있다. 그러나 메모리 테이블의 경우에는 LOB 칼럼 데이터를 별도로 저장할 수 없고 테이블과 동일한 테이블스페이스에만 저장할 수 있다.

## BLOB

### 흐름도



### 구문

```
BLOB [ VARIABLE ( IN ROW size ) ]
```

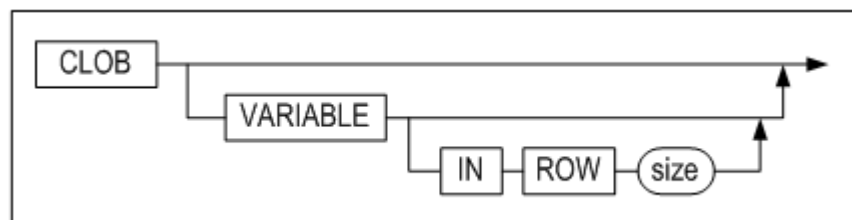
### 설명

BLOB은 이진형 대용량 데이터를 저장하기 위한 이진형 데이터 타입으로, 4GB-1byte 크기까지 저장 가능하다.

FIXED와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 “FIXED/VARIABLE 옵션”과 “IN ROW 절”을 참고한다.

## CLOB

### 흐름도



### 구문

```
CLOB [ VARIABLE ( IN ROW size ) ]
```

## 설명

CLOB은 문자형 대용량 데이터를 저장하기 위한 문자형 데이터 타입으로, 4GB-1byte 크기까지 저장 가능하다.

FIXED 와 VARIABLE 절에 대한 자세한 설명은 앞서 기술한 “FIXED/VARIABLE 옵션”과 “IN ROW 절”을 참고한다.

## 제한 사항

- 커서에서 LOB 타입 칼럼을 사용할 수 없다.
- 휘발성 테이블스페이스 또는 디스크 임시 테이블스페이스에 LOB 타입 칼럼을 사용할 수 없다.
- 폐기된 테이블스페이스의 LOB 타입 칼럼은 접근할 수 없다.
- 파티션 키 칼럼은 대소 비교가 가능해야 하기 때문에 LOB 타입 칼럼은 파티션 키 칼럼으로 사용될 수 없다.
- LOB 타입 칼럼에는 인덱스를 생성할 수 없다.
- LOB 타입 칼럼에 NOT NULL 제약조건을 사용하면 안 된다.  
LOB 타입 칼럼에 NOT NULL 제약조건을 정의할 수 있으나 APRE, CLI, JDBC 애플리케이션에서 LOB 데이터 처리 시 제약조건 위배 에러가 발생할 수 있다.

## 공간 데이터 타입

Altibase에서 SQL로 사용할 수 있도록 지원하는 공간 데이터 타입은 Geometry 한가지 뿐이다. 이 Geometry 타입은 내부적으로 다음 7개의 하위 데이터 타입으로 이루어진다:

- Point
- LineString
- Polygon
- GeomCollection
- MultiPolygon
- MultiLineString
- MultiPoint

공간 데이터 타입에 관한 자세한 내용은 *Spatial SQL Reference* 를 참조한다

## 2.Altibase 프로퍼티

사용자는 Altibase 서버를 다양한 모드로 운영할 수 있다. Altibase 서버의 환경 설정은 Altibase 프로퍼티 파일을 이용하는 것이다. 프로퍼티 파일은 Altibase 서버의 운용 방식과 튜닝에 관한 모든 구성 요소를 포함하고 있다.

이 장에서는 Altibase를 사용자 업무에 적합한 데이터베이스로 구성하고 운영하기 위해서 설정하고 관리해야 하는 Altibase 프로퍼티들에 대해서 설명한다.

### 환경 설정 방법

Altibase 서버와 관련된 환경을 설정하기 위한 방법은 세 가지가 있다.

첫째, Altibase 프로퍼티 파일, \$ALTIBASE\_HOME/conf/altibase.properties를 변경하는 방식이다. 이 방식은 Altibase 서버가 실행되지 않은 상태에서 할 수 있는 정적인 환경 설정 방법으로, 프로퍼티 파일에서 해당 구성 요소를 특정 값으로 설정한 후 Altibase 서버를 재구동해야 수정된 값이 Altibase 서버에 반영된다.

둘째, Altibase 서버가 운영 중이더라도 Altibase 관련 환경 설정을 변경할 수 있는 동적인 방식이다. Altibase 서버를 내리지 않고도 변경할 수 있다는 장점이 있으나, 모든 프로퍼티에 가능하지는 않다. 프로퍼티 속성의 동적 변경이 가능한 경우, ALTER SYSTEM 문 혹은 ALTER SESSION 문을 이용하여 변경할 수 있으며 Altibase 서버 전체 혹은 세션 별로 환경 설정 값이 적용된다.

셋째, 운영체제의 환경변수를 이용하여 설정할 수 있다. 이는 프로퍼티 파일과 마찬가지로 정적인 방법이다. 프로퍼티 속성이 읽기 전용, 단일 값일 경우 환경변수로 값을 설정할 수 있다. 환경변수의 이름은 ALTIBASE\_프로퍼티명 형식이며, 설정 후 데이터베이스 서버를 재구동하여야 수정된 값이 서버에 반영된다.

예를 들면 다음과 같다.

```
$ export ALTIBASE_DEFAULT_DATE_FORMAT=YYYY/MM/DD
```

프로퍼티 설정 방법에 따른 우선순위는 다음과 같다.

1. 환경변수로 설정
2. Altibase 프로퍼티 파일에서 설정
3. 시스템의 기본값 사용

다음과 같이 환경변수와 프로퍼티가 설정되었을 경우 환경변수의 우선순위가 높으므로, Altibase 프로퍼티 파일의 DEFAULT\_DATE\_FORMAT의 값은 무시되고 환경변수의 값이 사용된다.

```
$ export ALTIBASE_DEFAULT_DATE_FORMAT=YYYY-MM-DD
DEFAULT_DATE_FORMAT=YYYY-MM-DD (altibase.properties)
```

다음 예에서도 프로퍼티와 환경변수가 다르게 설정된 경우, 프로퍼티 파일의 NLS\_USE 값은 무시되고, 환경변수의 UTF-8 이 우선 설정된다.

```
$ export ALTIBASE_NLS_USE=UTF8
NLS_USE=KO16KSC5601 (altibase.properties)
```

## 프로퍼티 요약

Altibase 서버의 환경 설정에 관한 프로퍼티 파일은 ALTIBASE\_HOME의 conf 디렉토리 밑에 있는 altibase.properties 이며, 프로퍼티의 내용은 크게 다음과 같이 분류할 수 있다.

- 데이터베이스 초기화 관련 프로퍼티
- 성능 관련 프로퍼티
- 세션 관련 프로퍼티
- 트랜잭션 관련 프로퍼티
- 백업 및 복구 관련 프로퍼티
- 이중화 관련 프로퍼티
- 네트워크 및 보안 관련 프로퍼티
- 메시지 로그 관련 프로퍼티
- 데이터베이스 링크 관련 프로퍼티
- 감사(Auditing) 관련 프로퍼티
- C/C++ 외부 프로시저용 에이전트 관련 프로퍼티
- 사용자 계정 보안 관련 프로퍼티
- 기타 프로퍼티

다음의 표는 위 분류기준에 의해서 Altibase 프로퍼티를 정리한 표이다. 참고로 표의 각 분류는 다음과 같은 의미를 지닌다.

- D: 데이터베이스 초기화
- P: 성능
- S: 세션
- T: 트랜잭션
- B: 백업 및 복구
- R: 이중화
- NM: 네트워크 관리 및 보안(Network Management)
- M: 메시지 로그
- L: 데이터베이스 링크
- U: 감사(Auditing)
- A: 에이전트(Agent)
- AS: 사용자 계정 보안(Account Security)
- E: 기타

"변경 레벨" 열은 다음의 의미를 지닌다.

- SESSION : ALTER SESSION 문을 이용하여 프로퍼티 값 변경 가능
- SYSTEM : ALTER SYSTEM 문을 이용하여 프로퍼티 값 변경 가능
- BOTH : ALTER SESSION, ALTER SYSTEM 문을 이용하여 프로퍼티 값 변경 가능
- NONE: 동적 변경 불가능

분류	소분류	프로퍼티	변경 레벨
----	-----	------	-------



D	BUFFER_AREA_CHUNK_SIZE	
	BUFFER_AREA_SIZE	SYSTEM
	BUFFER_CHECKPOINT_LIST_CNT	
	BUFFER_FLUSHER_CNT	
	BUFFER_FLUSH_LIST_CNT	
	BUFFER_HASH_BUCKET_DENSITY	
	BUFFER_HASH_CHAIN_LATCH_DENSITY	
	BUFFER_LRU_LIST_CNT	
	BUFFER_PREPARE_LIST_CNT	
	COMPRESSION_RESOURCE_GC_SECOND	
	DB_NAME	
	DDL_SUPPLEMENTAL_LOG_ENABLE	SYSTEM
	DEFAULT_DISK_DB_DIR	
	DEFAULT_MEM_DB_FILE_SIZE	
	DEFAULT_SEGMENT_MANAGEMENT_TYPE	
	DEFAULT_SEGMENT_STORAGE_INITEXTENTS	
	DEFAULT_SEGMENT_STORAGE_MAXEXTENTS	
	DEFAULT_SEGMENT_STORAGE_MINEXTENTS	
	DEFAULT_SEGMENT_STORAGE_NEXTTEXTENTS	
	DIRECT_PATH_BUFFER_PAGE_COUNT	SYSTEM
	DISK_INDEX_UNBALANCED_SPLIT_RATE	SYSTEM
	DISK_LOB_COLUMN_IN_ROW_SIZE	
	DISK_MAX_DB_SIZE	
	DOUBLE_WRITE_DIRECTORY	
	DOUBLE_WRITE_DIRECTORY_COUNT	
	DRDB_FD_MAX_COUNT_PER_DATAFILE	SYSTEM
	EXPAND_CHUNK_PAGE_COUNT	
	LOB_OBJECT_BUFFER_SIZE	
	LOCK_MGR_CACHE_NODE	

	LOCK_MGR_DETECTDEADLOCK_INTERVAL	SYSTEM
	LOCK_MGR_MAX_SLEEP	SYSTEM
	LOCK_MGR_MIN_SLEEP	SYSTEM
	LOCK_MGR_SPIN_COUNT	SYSTEM
	LOCK_MGR_TYPE	
	LOCK_NODE_CACHE_COUNT	
	LOGANCHOR_DIR	
	LOG_DIR	
	LOG_FILE_SIZE	
	MAX_CLIENT	
	MEM_DB_DIR	
	MEM_MAX_DB_SIZE	
	MEMORY_INDEX_BUILD_RUN_SIZE	SYSTEM
	MEMORY_INDEX_BUILD_VALUE_LENGTH_THRESHOLD	SYSTEM
	MEMORY_INDEX_UNBALANCED_SPLIT_RATE	SYSTEM
	MEMORY_LOB_COLUMN_IN_ROW_SIZE	
	MEMORY_VARIABLE_COLUMN_IN_ROW_SIZE	
	MEM_SIZE_CLASS_COUNT	
	MIN_COMPRESSION_RESOURCE_COUNT	
	MIN_LOG_RECORD_SIZE_FOR_COMPRESS	SYSTEM
	MIN_PAGES_ON_DB_FREE_LIST	
	MIN_PAGES_ON_TABLE_FREE_LIST	SYSTEM
	MIN_TASK_COUNT_FOR_THREAD_LIVE	SYSTEM
	PCTFREE	
	PCTUSED	
	QP_MEMORY_CHUNK_SIZE	
	RECYCLEBIN_DISK_MAX_SIZE	SYSTEM
	RECYCLEBIN_ENABLE	SESSION
	RECYCLEBIN_MEM_MAX_SIZE	SYSTEM

	REDUCE_TEMP_MEMORY_ENABLE	SYSTEM
	SECURITY_ECC_POLICY_NAME	SYSTEM
	SECURITY_MODULE_LIBRARY	SYSTEM
	SECURITY_MODULE_NAME	SYSTEM
	SERVICE_THREAD_INITIAL_LIFESPAN	SYSTEM
	SMALL_TABLE_THRESHOLD	SYSTEM
	ST_OBJECT_BUFFER_SIZE	BOTH
	SYS_DATA_FILE_INIT_SIZE	
	SYS_DATA_FILE_MAX_SIZE	
	SYS_DATA_FILE_NEXT_SIZE	
	SYS_DATA_TBS_EXTENT_SIZE	
	SYS_TEMP_FILE_INIT_SIZE	
	SYS_TEMP_FILE_MAX_SIZE	
	SYS_TEMP_FILE_NEXT_SIZE	
	SYS_TEMP_TBS_EXTENT_SIZE	
	SYS_UNDO_FILE_INIT_SIZE	
	SYS_UNDO_FILE_MAX_SIZE	
	SYS_UNDO_FILE_NEXT_SIZE	
	SYS_UNDO_TBS_EXTENT_SIZE	
	TABLE_BACKUP_FILE_BUFFER_SIZE	
	TABLE_COMPACT_AT_SHUTDOWN	SYSTEM

		TEMP_HASH_BUCKET_DENSITY
		TEMP_PAGE_CHUNK_COUNT
		USER_DATA_FILE_INIT_SIZE
		USER_DATA_FILE_MAX_SIZE
		USER_DATA_FILE_NEXT_SIZE
		USER_DATA_TBS_EXTENT_SIZE
		USER_TEMP_FILE_INIT_SIZE
		USER_TEMP_FILE_MAX_SIZE
		USER_TEMP_FILE_NEXT_SIZE
		USER_TEMP_TBS_EXTENT_SIZE
		VOLATILE_MAX_DB_SIZE

P		AGER_WAIT_MAXIMUM	
		AGER_WAIT_MINIMUM	
		BUFFER_VICTIM_SEARCH_INTERVAL	SYSTEM
		BUFFER_VICTIM_SEARCH_PCT	SYSTEM
		BULKIO_PAGE_COUNT_FOR_DIRECT_PATH_INSERT	SYSTEM
		CHECKPOINT_BULK_SYNC_PAGE_COUNT	SYSTEM
		CHECKPOINT_BULK_WRITE_PAGE_COUNT	SYSTEM
		CHECKPOINT_BULK_WRITE_SLEEP_SEC	SYSTEM
		CHECKPOINT_BULK_WRITE_SLEEP_USEC	SYSTEM
		CHECKPOINT_FLUSH_COUNT	SYSTEM
		CHECKPOINT_FLUSH_MAX_GAP	SYSTEM
		CHECKPOINT_FLUSH_MAX_WAIT_SEC	SYSTEM
		CM_BUFFER_MAX_PENDING_LIST	
		CM_DISPATCHER_SOCK_POLL_TYPE	
		DATABASE_IO_TYPE	
		DATAFILE_WRITE_UNIT_SIZE	SYSTEM
		DB_FILE_MULTIPAGE_READ_COUNT	SYSTEM
		DEDICATED_THREAD_CHECK_INTERVAL	
		DEDICATED_THREAD_INIT_COUNT	
		DEDICATED_THREAD_MAX_COUNT	
		DEDICATED_THREAD_MODE	
		DEFAULT_FLUSHER_WAIT_SEC	SYSTEM
		DELAYED_FLUSH_LIST_PCT	SYSTEM
		DELAYED_FLUSH_PROTECTION_TIME_MSEC	SYSTEM
		DIRECT_IO_ENABLED	
		DISK_INDEX_BUILD_MERGE_PAGE_COUNT	SYSTEM
		EXECUTE_STMT_MEMORY_MAXIMUM	SYSTEM
		EXECUTOR_FAST_SIMPLE_QUERY	
		FAST_START_IO_TARGET	SYSTEM

	FAST_START_LOGFILE_TARGET	SYSTEM
	FAST_UNLOCK_LOG_ALLOC_MUTEX	
	HASH_AREA_SIZE	SYSTEM
	HASH_JOIN_MEM_TEMP_AUTO_BUCKET_COUNT_DISABLE	SYSTEM
	HASH_JOIN_MEM_TEMP_PARTITIONING_DISABLE	SYSTEM
	HIGH_FLUSH_PCT	SYSTEM
	HOT_LIST_PCT	SYSTEM
	HOT_TOUCH_CNT	SYSTEM
	INDEX_BUILD_THREAD_COUNT	SYSTEM
	INDEX_INITTRANS	
	INDEX_MAXTRANS	
	LFG_GROUP_COMMIT_INTERVAL_USEC	
	LFG_GROUP_COMMIT_RETRY_USEC	
	LFG_GROUP_COMMIT_UPDATE_TX_COUNT	
	LOB_CACHE_THRESHOLD	BOTH
	LOCK_ESCALATION_MEMORY_SIZE	SYSTEM
	LOG_CREATE_METHOD	
	LOG_IO_TYPE	
	LOW_FLUSH_PCT	SYSTEM
	LOW_PREPARE_PCT	SYSTEM
	MATHEMATICS_TEMP_MEMORY_MAXIMUM	SYSTEM
	MAX_FLUSHER_WAIT_SEC	SYSTEM
	MEM_INDEX_KEY_REDISTRIBUTION	SYSTEM
	MEM_INDEX_KEY_REDISTRIBUTION_STANDARD_RATE	SYSTEM
	MULTIPLEXING_CHECK_INTERVAL	SYSTEM
	MULTIPLEXING_MAX_THREAD_COUNT	SYSTEM
	MULTIPLEXING_THREAD_COUNT	
	NORMALFORM_MAXIMUM	BOTH
	OPTIMIZER_AUTO_STATS	BOTH

	OPTIMIZER_DELAYED_EXECUTION	SESSION
	OPTIMIZER_FEATURE_ENABLE	SYSTEM
	OPTIMIZER_MODE	BOTH
	OPTIMIZER_PERFORMANCE_VIEW	
	OPTIMIZER_UNNEST_AGGREGATION_SUBQUERY	SYSTEM
	OPTIMIZER_UNNEST_COMPLEX_SUBQUERY	SYSTEM
	OPTIMIZER_UNNEST_SUBQUERY	SYSTEM
	OUTER_JOIN_OPERATOR_TRANSFORM_ENABLE	SYSTEM
	PARALLEL_LOAD_FACTOR	
	PARALLEL_QUERY_THREAD_MAX	SYSTEM
	PARALLEL_QUERY_QUEUE_SIZE	SYSTEM
	PREPARE_STMT_MEMORY_MAXIMUM	SYSTEM
	QUERY_REWRITE_ENABLE	BOTH
	REFINE_PAGE_COUNT	
	RESULT_CACHE_ENABLE	BOTH
	RESULT_CACHE_MEMORY_MAXIMUM	SYSTEM
	SECONDARY_BUFFER_ENABLE	
	SECONDARY_BUFFER_FILE_DIRECTORY	
	SECONDARY_BUFFER_FLUSHER_CNT	
	SECONDARY_BUFFER_SIZE	
	SECONDARY_BUFFER_TYPE	
	SORT_AREA_SIZE	SYSTEM
	SQL_PLAN_CACHE_BUCKET_CNT	
	SQL_PLAN_CACHE_HOT_REGION_LRU_RATIO	SYSTEM
	SQL_PLAN_CACHE_PREPARED_EXECUTION_CONTEXT_CNT	SYSTEM
	SQL_PLAN_CACHE_SIZE	SYSTEM
	STATEMENT_LIST_PARTIAL_SCAN_COUNT	SYSTEM
	TABLE_INITTRANS	
	TABLE_LOCK_ENABLE	SYSTEM

		TABLE_LOCK_MODE	
		TABLE_MAXTRANS	
		TABLESPACE_LOCK_ENABLE	BOTH
		TEMP_STATS_WATCH_TIME	SYSTEM
		THREAD_CPU_AFFINITY	
		THREAD_REUSE_ENABLE	
		TIMED_STATISTICS	SYSTEM
		TIMER_RUNNING_LEVEL	
		TIMER_THREAD_RESOLUTION	SYSTEM
		TOP_RESULT_CACHE_MODE	BOTH
		TOTAL_WA_SIZE	SYSTEM
		INIT_TOTAL_WA_SIZE	SYSTEM
		TOUCH_TIME_INTERVAL	SYSTEM
		TRANSACTION_SEGMENT_COUNT	SYSTEM
		TRX_UPDATE_MAX_LOGSIZE	BOTH



S	일반	CM_DISCONN_DETECT_TIME	
		CONCURRENT_EXEC_DEGREE_DEFAULT	SYSTEM
		CONCURRENT_EXEC_DEGREE_MAX	
		CONCURRENT_EXEC_WAIT_INTERVAL	SYSTEM
		DEFAULT_THREAD_STACK_SIZE	
		IPC_CHANNEL_COUNT	
		IPC_FILEPATH	
		IPC_SEM_KEY	
		IPC_SHM_KEY	
		IPCDA_CHANNEL_COUNT	SYSTEM
		IPCDA_DATABLOCK_SIZE	
		IPCDA_FILEPATH	
		IPCDA_SEM_KEY	
		IPCDA_SHM_KEY	
		MAX_LISTEN	
		MAX_STATEMENTS_PER_SESSION	BOTH
		NET_CONN_IP_STACK	
		NLS_COMP	
		NLS_CURRENCY	SESSION
		NLS_ISO_CURRENCY	SESSION
		NLS_NCHAR_CONV_EXCP	SESSION
		NLS_NCHAR_LITERAL_REPLACE	SESSION
		NLS_NUMERIC_CHARACTERS	SESSION
		NLS_TERRITORY	SESSION
		PORT_NO	
		PSM_CURSOR_OPEN_LIMIT	SYSTEM
		PSM_FILE_OPEN_LIMIT	SYSTEM
		TIME_ZONE	SESSION

		UNIXDOMAIN_FILEPATH	
		USE_MEMORY_POOL	
		USER_LOCK_POOL_INIT_SIZE	SYSTEM
		USER_LOCK_REQUEST_CHECK_INTERVAL	SYSTEM
		USER_LOCK_REQUEST_LIMIT	SYSTEM
		USER_LOCK_REQUEST_TIMEOUT	SYSTEM
		XA_HEURISTIC_COMPLETE	
	타임아웃	BLOCK_ALL_TX_TIME_OUT	SYSTEM
		DDL_LOCK_TIMEOUT	SYSTEM
		DDL_TIMEOUT	BOTH
		FETCH_TIMEOUT	BOTH
		IDLE_TIMEOUT	BOTH
		LOGIN_TIMEOUT	SYSTEM
		MULTIPLEXING_POLL_TIMEOUT	SYSTEM
		QUERY_TIMEOUT	BOTH
		SHUTDOWN_IMMEDIATE_TIMEOUT	SYSTEM
		UTRANS_TIMEOUT	BOTH
		XA_INDOUBT_TX_TIMEOUT	
T		AUTO_COMMIT	SYSTEM
		ISOLATION_LEVEL	
		TRANSACTION_TABLE_SIZE	

B	ARCHIVE_DIR	
	ARCHIVE_FULL_ACTION	
	ARCHIVE_MULTIPLEX_COUNT	
	ARCHIVE_MULTIPLEX_DIR	
	ARCHIVE_THREAD_AUTOSTART	
	CHECKPOINT_ENABLED	
	CHECKPOINT_INTERVAL_IN_LOG	SYSTEM
	CHECKPOINT_INTERVAL_IN_SEC	SYSTEM
	COMMIT_WRITE_WAIT_MODE	BOTH
	INCREMENTAL_BACKUP_CHUNK_SIZE	
	INCREMENTAL_BACKUP_INFO_RETENTION_PERIOD	
	LOG_BUFFER_TYPE	
	LOG_MULTIPLEX_COUNT	
	LOG_MULTIPLEX_DIR	
	PREPARE_LOG_FILE_COUNT	
	SNAPSHOT_MEM_THRESHOLD	SYSTEM
	SNAPSHOT_DISK_UNDO_THRESHOLD	SYSTEM

R	REPLICATION_ACK_XLOG_COUNT	
	REPLICATION_ALLOW_DUPLICATE_HOSTS	SYSTEM
	REPLICATION_BEFORE_IMAGE_LOG_ENABLE	SYSTEM
	REPLICATION_COMMIT_WRITE_WAIT_MODE	SYSTEM
	REPLICATION_CONNECT_RECEIVE_TIMEOUT	SYSTEM
	REPLICATION_CONNECT_TIMEOUT	SYSTEM
	REPLICATION_DDL_ENABLE	SYSTEM
	REPLICATION_DDL_ENABLE_LEVEL	
	REPLICATION_DDL_SYNC	SYSTEM
	REPLICATION_DDL_SYNC_TIMEOUT	
	REPLICATION_EAGER_PARALLEL_FACTOR	
	REPLICATION_EAGER_RECEIVER_MAX_ERROR_COUNT	SYSTEM
	REPLICATION_FAILBACK_INCREMENTAL_SYNC	
	REPLICATION_GAP_UNIT	
	REPLICATION_GAPLESS_ALLOW_TIME	SYSTEM
	REPLICATION_GAPLESS_MAX_WAIT_TIME	SYSTEM
	REPLICATION_GROUPING_AHEAD_READ_NEXT_LOG_FILE	SYSTEM
	REPLICATION_GROUPING_TRANSACTION_MAX_COUNT	SYSTEM
	REPLICATION_HBT_DETECT_HIGHWATER_MARK	SYSTEM
	REPLICATION_HBT_DETECT_TIME	SYSTEM
	REPLICATION_IB_LATENCY	
	REPLICATION_IB_PORT_NO	
	REPLICATION_INSERT_REPLACE	SYSTEM
	REPLICATION_KEEP_ALIVE_CNT	
	REPLICATION_LOCK_TIMEOUT	SYSTEM
	REPLICATION_LOG_BUFFER_SIZE	
	REPLICATION_MAX_COUNT	
	REPLICATION_MAX_LISTEN	
	REPLICATION_MAX_LOGFILE	SYSTEM

	REPLICATION_POOL_ELEMENT_COUNT	SYSTEM
	REPLICATION_POOL_ELEMENT_SIZE	SYSTEM
	REPLICATION_PORT_NO	
	REPLICATION_RECEIVE_TIMEOUT	SYSTEM
	REPLICATION_RECEIVER_APPLIER_ASSIGN_MODE	SYSTEM
	REPLICATION_RECEIVER_APPLIER_QUEUE_SIZE	SYSTEM
	REPLICATION_PREFETCH_LOGFILE_COUNT	SYSTEM
	REPLICATION_RECOVERY_MAX_LOGFILE	
	REPLICATION_RECOVERY_MAX_TIME	
	REPLICATION_SENDER_AUTO_START	
	REPLICATION_SENDER_COMPRESS_XLOG	SYSTEM
	REPLICATION_SENDER_ENCRYPT_XLOG	
	REPLICATION_SENDER_SEND_TIMEOUT	SYSTEM
	REPLICATION_SENDER_SLEEP_TIME	
	REPLICATION_SENDER_SLEEP_TIMEOUT	SYSTEM
	REPLICATION_SENDER_START_AFTER_GIVING_UP	SYSTEM
	REPLICATION_SERVER_FAILBACK_MAX_TIME	
	REPLICATION_SQL_APPLY_ENABLE	
	REPLICATION_SYNC_APPLY_METHOD	SYSTEM
	REPLICATION_SYNC_LOCK_TIMEOUT	SYSTEM
	REPLICATION_SYNC_LOG	
	REPLICATION_SYNC_TUPLE_COUNT	SYSTEM
	REPLICATION_TIMESTAMP_RESOLUTION	SYSTEM
	REPLICATION_TRANSACTION_POOL_SIZE	SYSTEM
	REPLICATION_UPDATE_REPLACE	SYSTEM

NM		IB_CONCHKSPIN	
		IB_ENABLE	
		IB_LATENCY	
		IB_LISTEN_DISABLE	
		IB_MAX_LISTEN	
		IB_PORT_NO	SYSTEM
		SNMP_ALARM_QUERY_TIMEOUT	
		SNMP_ALARM_FETCH_TIMEOUT	
		SNMP_ALARM_UTRANS_TIMEOUT	
		SNMP_ALARM_SESSION_FAILURE_COUNT	
		SNMP_ENABLE	
		SNMP_MSGLOG_FLAG	SYSTEM
		SNMP_PORT_NO	
		SNMP_TRAP_PORT_NO	
		SNMP_RECV_TIMEOUT	
		SNMP_SEND_TIMEOUT	
		SSL_CA	
		SSL_CAPATH	
		SSL_CERT	
		SSL_CIPHER_LIST	
		SSL_CLIENT_AUTHENTICATION	
		SSL_ENABLE	
		SSL_KEY	
		SSL_MAX_LISTEN	
		SSL_PORT_NO	SYSTEM
		TCP_ENABLE	

M	ALL_MSGLOG_FLUSH	SYSTEM
	COLLECT_DUMP_INFO	SYSTEM
	DK_MSGLOG_COUNT	
	DK_MSGLOG_FILE	
	DK_MSGLOG_FLAG	SYSTEM
	DK_MSGLOG_RESERVE_SIZE	
	DK_MSGLOG_SIZE	
	DUMP_MSGLOG_COUNT	
	DUMP_MSGLOG_SIZE	
	DUMP_MSGLOG_RESERVE_SIZE	
	ERROR_MSGLOG_COUNT	
	ERROR_MSGLOG_FILE	
	ERROR_MSGLOG_SIZE	
	ERROR_MSGLOG_RESERVE_SIZE	
	LB_MSGLOG_COUNT	
	LB_MSGLOG_FILE	
	LB_MSGLOG_FLAG	
	LB_MSGLOG_SIZE	
	MM_MSGLOG_COUNT	
	MM_MSGLOG_FILE	
	MM_MSGLOG_RESERVE_SIZE	
	MM_MSGLOG_SIZE	
	MM_SESSION_LOGGING	SYSTEM
	NETWORK_ERROR_LOG	SYSTEM
	QP_MSGLOG_COUNT	
	QP_MSGLOG_FILE	
	QP_MSGLOG_FLAG	SYSTEM
	QP_MSGLOG_RESERVE_SIZE	
	QP_MSGLOG_SIZE	

	QUERY_PROF_FLAG	SYSTEM
	QUERY_PROF_LOG_DIR	SYSTEM
	RP_CONFLICT_MSGLOG_COUNT	
	RP_CONFLICT_MSGLOG_DIR	
	RP_CONFLICT_MSGLOG_ENABLE	
	RP_CONFLICT_MSGLOG_FILE	
	RP_CONFLICT_MSGLOG_FLAG	SYSTEM
	RP_CONFLICT_MSGLOG_SIZE	
	RP_MSGLOG_COUNT	
	RP_MSGLOG_FILE	
	RP_MSGLOG_FLAG	SYSTEM
	RP_MSGLOG_RESERVE_SIZE	
	RP_MSGLOG_SIZE	
	SERVER_MSGLOG_COUNT	
	SERVER_MSGLOG_DIR	
	SERVER_MSGLOG_FILE	
	SERVER_MSGLOG_FLAG	SYSTEM
	SERVER_MSGLOG_RESERVE_SIZE	
	SERVER_MSGLOG_SIZE	
	SM_MSGLOG_COUNT	
	SM_MSGLOG_FILE	
	SM_MSGLOG_FLAG	SYSTEM
	SM_MSGLOG_RESERVE_SIZE	
	SM_MSGLOG_SIZE	
	TRC_MSGLOG_RESERVE_SIZE	
	TRCLOG_DETAIL_PREDICATE	SYSTEM
	XA_MSGLOG_COUNT	
	XA_MSGLOG_FILE	
	XA_MSGLOG_FLAG	SYSTEM



		XA_MSGLOG_RESERVE_SIZE	
		XA_MSGLOG_SIZE	
L		DBLINK_ALTLINKER_CONNECT_TIMEOUT	
		DBLINK_DATA_BUFFER_ALLOC_RATIO	
		DBLINK_DATA_BUFFER_BLOCK_COUNT	
		DBLINK_DATA_BUFFER_BLOCK_SIZE	
		DBLINK_ENABLE	
		DBLINK_GLOBAL_TRANSACTION_LEVEL	
		DBLINK_RECOVERY_MAX_LOGFILE	SYSTEM
		DBLINK_REMOTE_STATEMENT_AUTOCOMMIT	BOTH
		DBLINK_REMOTE_TABLE_BUFFER_SIZE	
U		AUDIT_FILE_SIZE	SYSTEM
		AUDIT_LOG_DIR	SYSTEM
		AUDIT_OUTPUT_METHOD	SYSTEM
		AUDIT_TAG_NAME_IN_SYSLOG	SYSTEM
A		EXTPROC_AGENT_CONNECT_TIMEOUT	SYSTEM
		EXTPROC_AGENT_CALL_RETRY_COUNT	SYSTEM
		EXTPROC_AGENT_IDLE_TIMEOUT	SYSTEM
		EXTPROC_AGENT_SOCKET_FILEPATH	
AS		CASE_SENSITIVE_PASSWORD	SYSTEM
		FAILED_LOGIN_ATTEMPTS	
		PASSWORD_LOCK_TIME	
		PASSWORD_LIFE_TIME	
		PASSWORD_GRACE_TIME	
		PASSWORD_REUSE_TIME	
		PASSWORD_REUSE_MAX	
		PASSWORD_VERIFY_FUNCTION	

E		ACCESS_LIST	
		ACCESS_LIST_FILE	
		ADMIN_MODE	SYSTEM
		ARITHMETIC_OPERATION_MODE	SESSION
		CHECK_MUTEX_DURATION_TIME_ENABLE	SYSTEM
		COERCE_HOST_VAR_IN_SELECT_LIST_TO_VARCHAR	
		DEFAULT_DATE_FORMAT	
		EXEC_DDL_DISABLE	SYSTEM
		GROUP_CONCAT_PRECISION	SYSTEM
		JOB_SCHEDULER_ENABLE	SYSTEM
		JOB_THREAD_COUNT	
		JOB_THREAD_QUEUE_SIZE	
		MSG_QUEUE_PERMISSION	
		PSM_CASE_SENSITIVE	SYSTEM
		PSM_CHAR_DEFAULT_PRECISION	
		PSM_IGNORE_NO_DATA_FOUND_ERROR	SYSTEM
		PSM_NCHAR_UTF16_DEFAULT_PRECISION	
		PSM_NCHAR_UTF8_DEFAULT_PRECISION	
		PSM_NVARCHAR_UTF16_DEFAULT_PRECISION	
		PSM_NVARCHAR_UTF8_DEFAULT_PRECISION	
		PSM_PARAM_AND_RETURN_WITHOUT_PRECISION_ENABLE	
		PSM_VARCHAR_DEFAULT_PRECISION	
		QUERY_STACK_SIZE	BOTH
		RECURSION_LEVEL_MAXIMUM	SESSION
		REMOTE_SYSDBA_ENABLE	SYSTEM
		SELECT_HEADER_DISPLAY	BOTH
		SYS_CONNECT_BY_PATH_PRECISION	SYSTEM
		SERIAL_EXECUTE_MODE	BOTH

이 장에서는 각 프로퍼티를 다음과 같은 형식으로 설명하고 있다.

- 프로퍼티 이름
- 데이터 타입
- 기본값
- 속성 (읽기 전용/변경 가능, 단일 값/다중 값 허용)
- 값의 범위 (최대, 최소값)
- 설명

## 데이터베이스 초기화 프로퍼티

### **BUFFER\_AREA\_CHUNK\_SIZE (단위: 바이트)**

#### 데이터 타입

Unsigned Long

#### 기본값

33554432 (32M)

#### 속성

읽기 전용, 단일 값

#### 값의 범위

[8192,  $2^{64}-1$ ]

#### 설명

버퍼 영역의 크기가 늘어나는 단위를 명시한다. 버퍼 크기를 늘리면 이 크기의 배수로 늘어난다.

서버 운영중에는 변경할 수 없다.

### **BUFFER\_AREA\_SIZE (단위: 바이트)**

#### 데이터 타입

Unsigned Long

#### 기본값

134217728 (128M)

#### 속성

변경 가능, 단일 값

#### 값의 범위

[8 \* 1024 \* 10,  $2^{64}-1$ ]

**설명**

Altibase 버퍼 풀이 사용하는 총 메모리 크기를 명시한다. 설정되는 크기는 BUFFER\_AREA\_CHUNK\_SIZE 값의 배수 중 사용자가 명시한 값과 가장 가까운 값이다. 버퍼 풀의 크기는 하나의 트랜잭션이 사용할 수 있는 페이지의 크기와 동시 진행되는 트랜잭션의 개수에 영향을 받는다. 따라서 DISK\_INDEX\_BUILD\_MERGE\_PAGE\_COUNT 프로퍼티 등의 페이지 개수와 관련된 프로퍼티를 변경할 때 BUFFER\_AREA\_SIZE의 값이 작지않도록 해야한다.

**BUFFER\_CHECKPOINT\_LIST\_CNT****데이터 타입**

Unsigned Integer

**기본값**

4

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 64]

**설명**

체크포인트 리스트의 개수를 명시한다. 이 개수가 많을수록 트랜잭션 간의 체크포인트 리스트에 대한 락(LOCK) 경합이 줄어든다.

**BUFFER\_FLUSH\_LIST\_CNT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 64]

**설명**

플러시 리스트의 개수를 명시한다. 이 개수가 많을수록 트랜잭션 간의 플러시 리스트 관련 락(LOCK) 경합이 준다.

**BUFFER\_FLUSHER\_CNT****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 16]

**설명**

버퍼 플러셔의 개수를 결정한다. 서버 운영중에는 변경할 수 없다.

**BUFFER\_HASH\_BUCKET\_DENSITY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 100]

**설명**

한 버킷 안에 들어갈 수 있는 BCBs (Buffer Control Blocks) 개수의 백분율을 명시한다. 예를 들어 BCBs의 개수가 100일 때, 이 값이 1이면 버킷의 개수는 버퍼 풀의 버퍼 프레임 개수와 같아지기 때문에 락(LOCK) 경합은 최소화된다. 이 값이 2이면 버킷의 개수는 프레임 개수의 절반이 되며, 100이면 버킷은 하나가 된다. 이 값이 커질수록 메모리는 적게 사용하지만, 한 버킷당 관리해야 할 버퍼 프레임의 수가 증가하기 때문에 연산 비용이 증가한다.

**BUFFER\_HASH\_CHAIN\_LATCH\_DENSITY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 100]

**설명**

한 해쉬 테이블 내에서 한 개의 래치가 담당해야 할 버킷의 백분율을 명시한다. 예를 들어 버킷이 1000개일 때, 이 값이 1이면 버킷 10개당 한 개의 래치를 두며, 2이면 버킷 20개가 하나의 래치를 공유한다. 그리고 100이면 해쉬 테이블 전체에 걸쳐 한 개의 래치가 존재한다.

이 프로퍼티는 해쉬 테이블에 BCB가 삽입, 삭제될 때 동시성을 제어하기 위해 사용된다. 래치 수가 많을 수록 해쉬 테이블 내 래치의 경합이 줄어든다.

**BUFFER\_LRU\_LIST\_CNT****데이터 타입**

Unsigned Integer

**기본값**

7

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 64]

**설명**

LRU 리스트의 개수를 명시한다. 이 개수가 많을수록 트랜잭션 간의 LRU 리스트 관련 락(LOCK) 경합이 줄어든다.

**BUFFER\_PREPARE\_LIST\_CNT****데이터 타입**

Unsigned Integer

**기본값**

7

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 64]

**설명**

Prepare 리스트의 개수를 명시한다. 이 개수가 많을수록 트랜잭션 간의 Prepare 리스트 관련 락(LOCK) 경합이 줄어든다.

**BULKIO\_PAGE\_COUNT\_FOR\_DIRECT\_PATH\_INSERT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

변경 가능, 단일 값

**값의 범위**

[128, 12800]

**설명**

Direct-Path INSERT 방식으로 데이터를 입력할 때 한 번에 몇 개의 페이지를 디스크에 기록할 것인지 결정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**COMPRESSION\_RESOURCE\_GC\_SECOND (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

3600

**속성**

읽기 전용, 단일 값

**값의 범위**

[1,  $(2^{64}-1)/1000000$ ]

**설명**

로그 압축 리소스 풀에서 리소스가 몇 초 이상 사용되지 않을 경우 가비지 콜렉션(Garbage Collection)할 것인지를 결정한다.

**DB\_NAME****데이터 타입**

String

**기본값**

mydb

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

데이터베이스 이름을 명시한다. 데이터베이스 생성시 데이터베이스 이름은 이 프로퍼티에 명시한 것과 같은 이름을 사용해야 한다. 만약 데이터베이스 이름을 변경하려면, 데이터베이스를 다시 생성해야 한다.

**DDL\_SUPPLEMENTAL\_LOG\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

DDL 연산시 로그를 추가적으로 남길 것인지 여부를 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: Disabled (로그를 남기지 않음)

1: Enabled (로그를 남김)

**DEFAULT\_DISK\_DB\_DIR****데이터 타입**

String



**기본값**

\$ALTIBASE\_HOME/dbs

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

디스크 데이터베이스 파일을 저장할 디렉토리 경로를 지정한다. 디스크 기능을 사용하지 않더라도 반드시 지정해야 한다. 이 경로의 기본값은 \$ALTIBASE\_HOME/dbs이다.

**DEFAULT\_MEM\_DB\_FILE\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

1073741824 bytes (1G)

**속성**

읽기 전용, 단일 값

**값의 범위**[4194304 (4M),  $2^{64}-1$ ]**설명**

메모리 테이블스페이스를 위한 체크포인트 이미지 파일의 기본 크기를 나타낸다.

**DEFAULT\_SEGMENT\_MANAGEMENT\_TYPE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

디스크 테이블스페이스를 생성할 때 세그먼트 관리 방법을 결정한다.

0: MANUAL - 사용자가 테이블스페이스의 가용 공간 관리 방식을 프리 리스트(Freelist) 기반으로 하는 세그먼트 생성

1: AUTO - 사용자가 테이블스페이스의 가용 공간 관리 방식을 비트맵(Bitmap) 인덱스 기반으로 하는 세그먼트 생성

**DEFAULT\_SEGMENT\_STORAGE\_INITEXTENTS (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

$[1, 2^{32}-1]$

**설명**

기본 세그먼트의 초기 익스텐트(extent) 개수를 명시한다.

**DEFAULT\_SEGMENT\_STORAGE\_MAXEXTENTS (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

$2^{32}-1$

**속성**

읽기 전용, 단일 값

**값의 범위**

$[1, 2^{32}-1]$

**설명**

기본 세그먼트의 최대 익스텐트 개수를 명시한다.

**DEFAULT\_SEGMENT\_STORAGE\_MINEXTENTS (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위** $[1, 2^{32}-1]$ **설명**

기본 세그먼트의 최소한의 익스텐트 개수를 명시한다.

**DEFAULT\_SEGMENT\_STORAGE\_NEXTTEXTENTS (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위** $[1, 2^{32}-1]$ **설명**

기본 세그먼트의 확장할 수 있는 익스텐트 개수를 명시한다.

**DIRECT\_PATH\_BUFFER\_PAGE\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

변경 가능, 단일 값

**값의 범위**

[1024,  $2^{32}-1$ ]

**설명**

Direct=Path INSERT 버퍼의 페이지 개수를 나타낸다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DISK\_INDEX\_UNBALANCED\_SPLIT\_RATE (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

90

**속성**

변경 가능, 단일 값

**값의 범위**

[50, 99]

**설명**

디스크 B+ tree인덱스에서 최하위 리프 노드의 마지막 차일드 노드(child node)가 분할될 때, 분할을 발생시키는 노드와 생성되는 노드 사이에 키를 분배하는 비율을 명시할 수 있다. 이 값을 기본값인 90으로 지정할 경우, 두 노드 사이에 키의 비율은 90:10으로 배분된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DISK\_LOB\_COLUMN\_IN\_ROW\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

4000

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,4000]

**설명**

디스크 테이블에 사용된 LOB 타입 데이터의 기본 in row 크기를 지정한다. in row 크기는 LOB 데이터 타입의 칼럼에 데이터가 들어갈 때 데이터 길이가 이 값보다 작거나 같으면 고정(fixed) 영역에 저장하고, 이 보다 긴 경우에는 가변(variable) 영역에 들어가도록 지정하는 속성이다. 디스크 테이블에만 해당하는 것으로 메모리 테이블은 이 프로퍼티를 참조하지 않는다.

in row 크기나 LOB 타입에 대한 자세한 사항은 1장의 데이터 타입 부분을 참조한다.

**DISK\_MAX\_DB\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값** $2^{64}-1$ **속성**

읽기 전용, 단일 값

**값의 범위**64비트: [2097152,  $2^{64}$ ]**설명**

Altibase에서 디스크 데이터베이스를 설정할 수 있는 최대 크기이다.

DISK\_MAX\_DB\_SIZE를 초과해서 데이터베이스 크기가 확장될 경우 그 트랜잭션은 오류 처리되며, 이후 수행되는 SELECT 문을 제외한 모든 SQL 문은 오류 처리된다.

**DOUBLE\_WRITE\_DIRECTORY****데이터 타입**

String

**기본값**

없음

**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

더블 라이트 파일이 저장될 디렉토리를 지정한다. 이 프로퍼티는 DOUBLE\_WRITE\_DIRECTORY\_COUNT에 의해 복수로 명시할 수 있다.

**DOUBLE\_WRITE\_DIRECTORY\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 16]

**설명**

더블 라이트 파일이 저장되는 디렉토리의 개수를 지정한다. 더블 라이트 파일들은 각각 다른 디스크에 저장될 수 있다. 플러셔(Flusher)마다 별도의 더블 라이트 파일을 사용하기 때문에 서로 다른 디스크로 디렉토리를 지정하면, 플러시 성능을 높일 수 있다.

**DRDB\_FD\_MAX\_COUNT\_PER\_DATAFILE****데이터 타입**

Unsigned Integer

**기본값**

8

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 1024]

**설명**

하나의 디스크 데이터 파일에서 I/O가 발생할 때 열 수 있는 최대 FD(File Descriptors) 개수다. FD가 프로퍼티에서 설정한 최대 숫자까지 열려있다면, 다른 I/O가 완료될 때까지 대기한다.

**EXPAND\_CHUNK\_PAGE\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위**[64,  $2^{32}-1$ ]**설명**

메모리 데이터베이스의 Expand Chunk가 가질 수 있는 페이지의 개수를 나타낸다. Expand Chunk는 메모리 데이터베이스에서 페이지를 확장할 수 있는 단위이다. 데이터베이스를 생성할 때 설정할 수 있으며, 만약 페이지 개수를 변경하려면 데이터베이스를 다시 생성해야 한다.

**LOB\_OBJECT\_BUFFER\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

32000

**속성**

읽기 전용, 단일 값

**값의 범위**

[32000, 104857600]

**설명**

파라미터, 내부 변수 또는 반환값의 타입이 LOB으로 선언된 저장 프로시저/함수 또는 트리거를 동작하는 중에, 이러한 LOB 값 처리를 위해 Altibase 서버에서 내부적으로 사용하는 LOB 데이터의 최대 크기를 지정하는 프로퍼티이다.

**LOCK\_MGR\_CACHE\_NODE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 2]

**설명**

테이블 락(LOCK) 노드의 캐시 타입을 명시한다.

서버 운영중에는 변경할 수 없다.

0: 캐시하지 않는다.

1: 링크드 리스트(Linked List)로 캐시한다.

2: 배열로 캐시하고, 64개로 고정된다.

**LOCK\_MGR\_DETECTDEADLOCK\_INTERVAL (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

3

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 10]

**설명**

Spin lock 모드에서 데드락을 검출하는 간격을 명시한다.

(Altibase 7.1.0.3.2부터 Spin lock 모드가 삭제되어 해당 프로퍼티 또한 지원하지 않는다.)

**LOCK\_MGR\_MAX\_SLEEP (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

1000



**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1000000]

**설명**

Spin lock 모드에서 재시도 횟수만큼 시도했음에도 불구하고 Lock 획득에 실패한 경우에 sleep하는 최대 시간을 명시한다.

(Altibase 7.1.0.3.2부터 Spin lock 모드가 삭제되어 해당 프로퍼티 또한 지원하지 않는다.)

**LOCK\_MGR\_MIN\_SLEEP (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

50

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1000000]

**설명**

Spin lock 모드에서 재시도 횟수만큼 시도했음에도 불구하고 Lock 획득에 실패한 경우에 sleep하는 시간을 명시한다.

(Altibase 7.1.0.3.2부터 Spin lock 모드가 삭제되어 해당 프로퍼티 또한 지원하지 않는다.)

**LOCK\_MGR\_SPIN\_COUNT (단위: 횟수)****데이터 타입**

Unsigned Integer

**기본값**

1000

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 3000]

**설명**

Spin lock 모드에서 Lock 획득에 실패했을 경우 재시도 횟수를 명시한다.

(Altibase 7.1.0.3.2부터 Spin lock 모드가 삭제되어 해당 프로퍼티 또한 지원하지 않는다.)

**LOCK\_MGR\_TYPE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 2]

**설명**

테이블 록(Table Lock) 관리자 타입을 선택한다. 서버 운영중에는 변경할 수 없다.

0: Mutex 모드

1: Spin lock 모드 (Altibase 7.1.0.3.2부터 지원 중단)

2: light Mutex (Altibase 7.1.0.3.2부터 지원)

**LOCK\_NODE\_CACHE\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1024]

**설명**

LOCK\_MGR\_CACHE\_NODE=1 일 경우 캐시할 테이블 락 노드의 개수를 명시한다. 서버 운영중에는 변경할 수 없다.

**LOGANCHOR\_DIR****데이터 타입**

String

**기본값**

\$ALTIBASE\_HOME/logs

**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

로그앵커 파일이 존재할 경로를 지정한다. 반드시 3개가 지정되어야 하며, 기본으로 3개가 동일하게 기본값으로 지정된다.

**LOG\_DIR****데이터 타입**

String

**기본값**

\$ALTIBASE\_HOME/logs

**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

로그 파일이 존재할 경로를 지정한다.

**LOG\_FILE\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

10 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**[1024 \* 1024, 2<sup>64</sup>-1]**설명**

로그 파일의 크기를 지정한다. 지정된 로그 파일의 크기 만큼 차게 되면 새로운 로그 파일에 기록한다. 데이터베이스 생성 시에 설정할 수 있으며, 생성 후에는 변경할 수 없다. 만약 사용자가 로그 파일 크기를 변경하려면, 데이터베이스를 다시 생성해야 한다.

**\* 제약사항**

- 오프라인 이중화를 사용하기 위해서는 지역 서버(Active)와 원격 서버(Standby)의 이 프로퍼티의 값이 반드시 동일해야 한다.

**MAX\_CLIENT****데이터 타입**

Unsigned Integer

**기본값**

1000

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 65535]

단, JOB\_THREAD\_COUNT 프로퍼티가 0보다 큰 값으로 설정된 경우에 최댓값은 65535에서 JOB\_THREAD\_COUNT 값을 뺀 값이다.

**설명**

Altibase에 접속할 수 있는 클라이언트의 최대 개수를 명시한다.

**MEM\_DB\_DIR****데이터 타입**

String

**기본값**

\$ALTIBASE\_HOME/dbs

**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

메모리 데이터베이스 파일이 존재할 경로를 지정한다.

최소 1개, 최대 8개 경로 지정이 가능하다. 여러 개의 경로가 지정될 경우, 데이터베이스 파일은 각 경로에 분산되어 저장된다. 이 프로퍼티로 지정되는 모든 경로는 실제 존재하는 경로여야 한다. 경로의 기본 개수는 2개, 각 경로의 기본값은 \$ALTIBASE\_HOME/dbs로 동일하게 지정되어 있다.

**MEM\_MAX\_DB\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값** $2^{31}$ **속성**

읽기 전용, 단일 값

**값의 범위**32비트:  $[2097152, 2^{32}+1]$ 64비트:  $[2097152, 2^{64}]$ **설명**

서비스 과정 중에 동적으로 늘어날 수 있는 모든 메모리 데이터베이스 크기의 합에 대한 최댓값을 명시한다. 32비트와 64비트 모드에 관계없이 기본 값은 2G이다.

MEM\_MAX\_DB\_SIZE를 초과해서 데이터베이스 크기가 확장될 경우 그 트랜잭션은 오류 처리되며, 이후 수행되는 SELECT 문을 제외한 모든 SQL 문은 오류 처리된다.

**MEMORY\_INDEX\_BUILD\_RUN\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

131072

**속성**

변경 가능, 단일 값

**값의 범위**

[1024,  $2^{64}-1$ ]

**설명**

메모리 인덱스를 구축할 때 in-memory sorting 영역의 크기를 지정할 수 있다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MEMORY\_INDEX\_BUILD\_VALUE\_LENGTH\_THRESHOLD (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

64

**속성**

변경 가능, 단일 값

**값의 범위**

[0,  $2^{64}-1$ ]

**설명**

메모리 인덱스 구축시, 중간 정렬을 위해 사용가능한 키 값의 최대 길이를 설정한다.

키 값의 길이가 이 프로퍼티보다 작으면 중간 정렬시 키 값을 사용하지만, 이 프로퍼티의 값이 0이면 레코드에 대한 포인터를 사용한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MEMORY\_INDEX\_UNBALANCED\_SPLIT\_RATE (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

90

**속성**

변경 가능, 단일 값

**값의 범위**

[50, 99]

**설명**

메모리 B+ tree인덱스에서 최하위 리프 노드의 마지막 차일드 노드(child node)가 분할될 때, 분할을 발생시키는 노드와 생성되는 노드 사이에 키를 분배하는 비율을 명시할 수 있다. 이 값을 기본값인 50으로 지정할 경우, 두 노드 사이에 키의 비율은 50:50으로 배분된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MEMORY\_LOB\_COLUMN\_IN\_ROW\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

64

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,4000]

**설명**

메모리 테이블에 사용된 LOB 타입 데이터의 기본 in row 크기를 지정한다. in row 크기는 LOB 데이터 타입의 칼럼에 데이터가 들어갈 때 데이터 길이가 이 값보다 작거나 같으면 고정(fixed) 영역에 저장하고, 이 보다 긴 경우에는 가변(variable) 영역에 들어가도록 지정하는 속성이다. 메모리 테이블에만 해당하는 것으로 디스크 테이블은 이 프로퍼티를 참조하지 않는다.

in row 크기나 LOB 타입에 대한 자세한 사항은 1장의 데이터 타입 부분을 참조한다.

**MEMORY\_VARIABLE\_COLUMN\_IN\_ROW\_SIZE(단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

32

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,4000]

**설명**

메모리 테이블에 사용된 가변 크기 타입 데이터의 기본 in row 크기를 지정한다. in row 크기는 가변 크기 타입의 칼럼에 데이터가 들어갈 때 데이터 길이가 이 값보다 작거나 같으면 고정(fixed) 영역에 저장한다. 이보다 긴 경우에는 가변(variable) 영역에 들어가도록 지정하는 속성이다. 메모리 테이블에만 해당하는 것으로 디스크 테이블은 이 프로퍼티를 참조하지 않는다.

in row 절에 대한 자세한 사항은 1장의 데이터 타입 부분을 참조한다.

**MEM\_SIZE\_CLASS\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

4

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 4]

**설명**

메모리 페이지의 빈 공간(free space)을 몇 개의 클래스로 구분한 것인지를 나타낸다.

**MIN\_COMPRESSION\_RESOURCE\_COUNT (단위: 개수)****데이터 타입**

unsigned integer

**기본값**

16

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 16384]

**설명**

로그 관리자가 로그를 압축하기 위해 사용하는 버퍼 청크(buffer chunk)의 최소한의 개수를 나타낸다 (한 개의 압축 버퍼 청크는 약 16KB이다).



**MIN\_LOG\_RECORD\_SIZE\_FOR\_COMPRESS (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본 값**

512

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

로그 압축을 수행의 기준이 되는 로그의 크기를 설정할 수 있다. 이 값이 0으로 설정된 경우 로그 압축을 수행하지 않으며, 로그 크기가 설정된 값보다 큰 경우 로그 압축을 수행한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MIN\_PAGES\_ON\_DB\_FREE\_LIST****데이터 타입**

Unsigned Integer

**기본값**

16

**속성**

읽기 전용, 단일 값

**값의 범위** $[1, 2^{32}-1]$ **설명**

데이터베이스의 다중화된 사용가능한 페이지 리스트에서 페이지를 분배할 때 각 페이지 리스트에 남겨 놓아야할 최소 페이지 개수이다.

**MIN\_PAGES\_ON\_TABLE\_FREE\_LIST****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경가능, 단일 값

**값의 범위**[ 1, 2<sup>32</sup>-1]**설명**

테이블에서 데이터베이스로 빈 페이지를 반납하거나, 다중화된 사용 가능한 페이지 리스트에서 페이지를 분배할 때 각 페이지 리스트에 남겨 놓아야 할 최소 페이지 개수이다.

**MIN\_TASK\_COUNT\_FOR\_THREAD\_LIVE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 1024]

**설명**

이 프로퍼티는 서비스 스레드가 해제되지 않기 위해 할당 받아야 하는 최소한의 클라이언트 수를 설정한다. 만약 서비스 스레드가 할당 받은 클라이언트의 수가 이 프로퍼티의 값보다 적으면, 곧바로 해제되는 것이 아니라 일정 기간 대기 상태로 놓인다. 대기중인 서비스 스레드가 SERVICE\_THREAD\_INITIAL\_LIFESPAN에 설정된 주기동안 이 프로퍼티의 개수 이상 클라이언트를 할당 받지 못하면 해제된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**PCTFREE (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 99]

**설명**

이 프로퍼티는 테이블스페이스의 각 페이지에서 삽입 가능한 상태를 유지하기 위한 여유 공간의 최소 비율을 나타낸다. 이 여유 공간은 기존 레코드들을 갱신하기 위해 필요하다.

만약 테이블스페이스의 전체 크기가 100MB인 경우 PCTFREE가 10이라면, 90%의 공간인 90MB까지만 삽입 연산이 가능하다.

디스크 테이블 생성 시 CREATE TABLE 구문에서 PCTFREE 값이 지정되지 않은 경우 기본값이 사용된다.

**PCTUSED (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

40

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 99]

**설명**

이 프로퍼티는 테이블스페이스의 페이지가 갱신 연산만 가능한 상태에서 다시 삽입 연산이 가능한 상태로 가기 위한 페이지의 사용 공간 비율을 나타낸다.

페이지의 사용 공간은 PCTFREE에서 지정한 값까지 입력할 경우 갱신 연산만이 가능한 상태로 된다. 이 때 갱신과 삭제로 빈 공간이 다시 확보되어 PCTUSED에서 정한 값보다 낮아지면, 새로운 행을 삽입할 수 있는 상태가 된다.

디스크 테이블 생성시 CREATE TABLE 구문에서 PCTUSED 값이 지정되지 않은 경우 기본값으로 사용된다.

**QP\_MEMORY\_CHUNK\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

65536

**속성**

읽기 전용, 단일 값

**값의 범위**

[1024,  $2^{64}-1$ ]

**설명**

질의 처리기에서 필요한 메모리를 시스템에서 가져올 때 확장 단위를 정한다.

**RECYCLEBIN\_DISK\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

$2^{64}-1$

**속성**

변경 가능, 단일 값

**값의 범위**

[0,  $2^{64}-1$ ]

**설명**

디스크 테이블이 버려지는(drop) 휴지통의 사이즈를 명시할 수 있다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**RECYCLEBIN\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

테이블이 DROP 구문으로 삭제된 경우 휴지통으로 버려지거나, 데이터베이스 시스템에서 바로 삭제할 것인지를 설정할 수 있다. 기본값은 DROP 구문을 수행하면 시스템에서 테이블이 제거된다.

휴지통으로 버려진 테이블은 이름이 변경되어 저장된다. 그리고 테이블 타입이 'R' 타입으로 변경 되어 일체 다른 DDL 및 INSERT/UPDATE/DELETE 등을 수행 할 수 없다. 단 SELECT는 가능하다.

휴지통에 테이블이 존재한다면 프로퍼티의 값을 0으로 변경하더라도 휴지통의 테이블을 조회하거나 FLASHBACK 또는 PURGE 구문을 사용하여 복구 및 제거할 수 있다.

Altibase 운영 중 ALTER SESSION문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0 : Disable(기본값)-테이블이 데이터베이스에서 삭제된다.

1 : Enable - 테이블이 휴지통으로 버려진다.

**RECYCLEBIN\_MEM\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

4 GB

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2<sup>64</sup>-1]

**설명**

메모리 테이블이 버려지는(drop) 휴지통의 사이즈를 명시할 수 있다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REDUCE\_TEMP\_MEMORY\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

가변 길이 칼럼의 데이터가 메모리 테이블스페이스에 임시로 저장될 때 차지하는 공간을 줄일 수 있는 프로퍼티이다.

Altibase 서버가 디스크 테이블이나 뷰에 대한 질의를 처리할 때 중간 결과를 임시로 저장하는 공간은 기본적으로 디스크 임시 테이블스페이스이다. 그러나 성능 향상을 목적으로 TEMP\_TBS\_MEMORY 힌트(*Administrator's Manual, Performance Tuning Guide* 참고)를 사용하여 중간 결과가 메모리 테이블스페이스에 저장되도록 할 수 있다.

기본값인 0은 가변 길이 칼럼의 데이터도 고정 길이 칼럼처럼 고정된 길이의 임시 저장 공간을 사용하여, 불필요한 공간을 차지할 수 있다. 이러한 공간 낭비를 줄이기 위해 이 프로퍼티를 1로 설정하면 가변 길이 칼럼의 실제 데이터만큼 공간을 사용한다. 그러나, 메모리 사용량은 줄어들지만 질의 처리 속도가 떨어질 수 있다.

0: 가변 길이 칼럼이 정의된 길이만큼 임시 저장 공간을 사용

1: 가변 길이 칼럼이 실제 데이터만큼 임시 저장 공간을 사용

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**SECURITY\_ECC\_POLICY\_NAME****데이터 타입**

String

**기본값**

없음

**속성**

변경 가능, 단일 값

**값의 범위**

없음

**설명**

암호화 칼럼을 위한 보안 모듈을 수행할 때 사용하는 ECC (Encrypted Comparison Code) 알고리즘의 이름을 지정한다.

**SECURITY\_MODULE\_LIBRARY****데이터 타입**

String

**기본값**

없음

**속성**

변경 가능, 단일 값

**값의 범위**

없음

**설명**

보안 모듈의 라이브러리 파일 이름을 지정한다. 보안 모듈을 수행할 때 이 파일이 사용된다.

**SECURITY\_MODULE\_NAME****데이터 타입**

String

**기본값**

없음

**속성**

변경 가능, 단일 값

**값의 범위**

없음

**설명**

보안 모듈을 수행할 때 명시하는 보안 모듈의 이름을 지정한다.

**SERVICE\_THREAD\_INITIAL\_LIFESPAN****데이터 타입**

Unsigned Integer

**기본값**

6000

**속성**

변경 가능, 단일 값

**값의 범위**

[30,  $2^{32}-1$ ]

**설명**

서비스 스레드의 해제 주기를 설정한다. 해제 주기란, 스레드를 해제하기 위해 스레드 매니저가 대기중인 서비스 스레드들의 상태를 확인해야 하는 횟수이다. 대기중인 서비스 스레드는 스레드 매니저가 이 프로퍼티에 설정된 확인 횟수 동안 클라이언트를 할당 받지 못하면 해제된다.

대기하는 서비스 스레드는 MIN\_TASK\_COUNT\_FOR\_THREAD\_LIVE 프로퍼티에 설정한 값보다 적은 클라이언트를 할당 받을 때 정해진다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## SMALL\_TABLE\_THRESHOLD

### 데이터 타입

Unsigned Integer

### 기본값

128

### 속성

변경 가능, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

디스크 테이블에 FULL SCAN 이 수행될 때, 그 테이블의 페이지 개수가 이 프로퍼티에 지정된 페이지 개수보다 작거나 같으면 "다중 페이지 읽기"로 디스크에서 버퍼로 읽혀온 페이지가 full scan이 끝난 후에 그대로 버퍼에 남겨진다. 테이블의 페이지 개수가 이 프로퍼티에 지정된 페이지 개수보다 크면 페이지는 버퍼에 남겨지지 않는다.

이 값을 0으로 지정한 경우, 테이블의 페이지 개수에 상관없이 읽혀온 페이지는 버퍼에 남겨지지 않는다.

이 값이 최댓값인  $2^{32}-1$ 로 지정된 경우, 테이블의 페이지 개수에 상관없이 읽혀온 페이지는 무조건 버퍼에 남겨진다.

## ST\_OBJECT\_BUFFER\_SIZE (단위: 바이트)

### 데이터 타입

Unsigned long

### 기본값

32000 (32KByte)

### 속성

변경 가능, 단일 값

### 값의 범위

[32000, 104857600]

### 설명

단일 공간 객체(Geometry Object)의 최대 크기를 지정할 수 있다.



**SYS\_DATA\_FILE\_INIT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

100M (100 \* 1024 \* 1024)

**속성**

읽기 전용, 단일 값

**값의 범위**

[8\*8K, 32GB]

**설명**

SYS\_TBS\_DISK\_DATA (시스템 디스크 테이블스페이스) 생성시 데이터 파일(system001.dbf)의 기본 크기를 명시한다. 또한 SYS\_TBS\_DISK\_DATA에 데이터 파일(system001.dbf를 제외한 사용자가 명시한 데이터 파일)을 추가할 때 초기 크기를 지정하지 않은 경우 기본값으로 사용된다.

**SYS\_DATA\_FILE\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

2 \* 1024 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8K, 32GB]

**설명**

SYS\_TBS\_DISK\_DATA (시스템 디스크 데이터 테이블스페이스) 생성 시 할당될 데이터 파일의 최대 크기를 명시한다. 최소 SYS\_DATA\_FILE\_INIT\_SIZE 이상의 크기를 가져야 하며, 최솟값은 64K이다.

또한 SYS\_TBS\_DISK\_DATA 테이블스페이스에 데이터 파일을 추가할 때 최대 크기를 지정하지 않았을 경우 이 값을 기본값으로 사용한다.

**SYS\_DATA\_FILE\_NEXT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

1 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8K, 32GB]

**설명**

시스템 디스크 데이터 테이블스페이스 (SYS\_TBS\_DISK\_DATA)의 속성이 “autoextend on”으로 설정되어 있는 경우, 데이터량의 증가에 따라서 명시된 값만큼 데이터 파일의 크기가 자동으로 확장된다.

데이터 파일의 크기가 SYS\_DATA\_FILE\_MAX\_SIZE에 도달한 상태에서 다른 데이터 파일에도 SYS\_DATA\_FILE\_NEXT\_SIZE에 설정한 만큼의 유효 공간이 없다면 테이블스페이스 공간 부족 오류가 발생한다.

**SYS\_DATA\_TBS\_EXTENT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

512 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[40K, 32G]

**설명**

SYS\_TBS\_DISK\_DATA (시스템 디스크 데이터 테이블스페이스<sup>1)</sup> 생성 시 익스텐트의 크기<sup>2</sup>를 명시한다. 최소 5개 이상의 페이지를 갖도록 설정해야 하고 최솟값은 40kB (5\*8kB)이다.

[<sup>1</sup>] 시스템 디스크 데이터 테이블스페이스 (System Disk Data Tablespace): 데이터베이스 생성 시 기본적으로 생성되는 디스크 데이터 테이블스페이스이다. 데이터베이스 객체 중 디스크 테이블과 디스크 인덱스만 저장된다.

[<sup>2</sup>] 하나의 테이블스페이스의 EXTENT의 크기(SIZE)는 생성 시 지정하면 이후 변경할 수 없다. 명시하지 않을 경우 기본값은 32개의 페이지 크기로 고정되어 있다.

**SYS\_TEMP\_FILE\_INIT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

100M (100 \* 1024 \* 1024)

**속성**

읽기 전용, 단일 값

**값의 범위**

[ 8 \* 8kB, 32GB]

**설명**

SYS\_TBS\_DISK\_TEMP 생성 시 임시 데이터 파일(temp001.dbf)의 초기 크기를 명시한다. 또한 SYS\_TBS\_DISK\_TEMP에 임시 데이터 파일을 추가할 때 초기 크기를 지정하지 않았을 경우, 이 값이 기본값으로 사용된다.

**SYS\_TEMP\_FILE\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

2 \* 1024 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8K, 32GB]

**설명**

SYS\_TBS\_DISK\_TEMP 생성 시 할당될 데이터 파일(temp001.dbf)의 최대 크기를 명시한다.

최소 SYS\_TEMP\_FILE\_INIT\_SIZE 이상의 크기를 가져야 하며, 가능한 최솟값은 64kB이다. 또한, SYS\_TBS\_DISK\_TEMP 테이블스페이스에 임시 데이터 파일을 추가할 때 최대 크기를 지정하지 않았을 경우, 여기에 명시한 값이 기본 최대 크기가 된다.

**SYS\_TEMP\_FILE\_NEXT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

1 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

SYS\_TBS\_DISK\_TEMP 테이블스페이스에 데이터 파일의 공간이 부족한 경우 명시된 값만큼 데이터 파일의 크기를 확장한다.

**SYS\_TEMP\_TBS\_EXTENT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

512 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[40kB, 32GB]

**설명**

시스템 디스크 임시 테이블스페이스<sup>3</sup> SYS\_TBS\_DISK\_TEMP 생성 시 익스텐트의 크기를 명시한다.

[<sup>3</sup>] 시스템 디스크 임시 테이블스페이스 (System disk temporary tablespace): 데이터베이스 생성 시 기본적으로 생성되며, 데이터베이스 각종 연산의 임시 저장소로 사용되는 테이블스페이스이다. 모든 사용자의 디스크 객체를 위한 기본(DEFAULT) 임시 테이블스페이스로 지정된다. 데이터베이스 객체 중 디스크 테이블과 디스크 인덱스만 저장된다.

최소 5개 페이지(40kB = 5 \* 8K) 이상의 크기를 가져야 한다.

**SYS\_UNDO\_FILE\_INIT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

100 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[32 \* 8kB, 32GB]

**설명**

SYS\_TBS\_DISK\_UNDO 테이블스페이스 생성 시 데이터 파일(undo001.dbf)의 기본 크기를 명시한다. 또한 SYS\_TBS\_DISK\_UNDO에 데이터 파일을 추가할 때 초기 크기를 지정하지 않았을 경우 기본값으로 사용된다.

**SYS\_UNDO\_FILE\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

2 \* 1024 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[32 \* 8kB, 32GB]

**설명**

SYS\_TBS\_DISK\_UNDO 테이블스페이스 생성 시 할당될 데이터 파일(undo001.dbf)의 최대 크기를 명시한다. 최소 SYS\_UNDO\_FILE\_INIT\_SIZE 이상의 크기를 가져야 한다. 가능한 최솟값은 256K이다. SYS\_TBS\_DISK\_UNDO 테이블스페이스에 데이터 파일을 추가할 때 최대 크기를 지정하지 않았을 경우, 기본 최대 크기로 사용된다.

**SYS\_UNDO\_FILE\_NEXT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

1 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

SYS\_TBS\_DISK\_UNDO테이블스페이스의 데이터 파일에 공간이 부족한 경우 명시된 값만큼 데이터 파일의 크기를 확장한다.

**SYS\_UNDO\_TBS\_EXTENT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

256 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[40kB, 32GB]

**설명**

시스템 디스크 언두 테이블스페이스<sup>4</sup> SYS\_TBS\_DISK\_UNDO 생성 시 익스텐트의 크기를 명시한다.

[<sup>4</sup>] 시스템 디스크 언두 테이블스페이스 (System disk undo tablespace): 데이터베이스 생성 시 기본적으로 생성되며 언두(undo) 정보를 저장하기 위해 유일하게 사용되는 특수한 테이블스페이스이다. 사용자는 디스크 언두 테이블스페이스 내에 테이블이나 인덱스 등을 생성할 수 없다. 데이터베이스 내에 오직 하나만 존재하며, 사용자가 생성하거나 삭제할 수 없다.

**TABLE\_BACKUP\_FILE\_BUFFER\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1048576]

**설명**

메모리 테이블의 칼럼을 추가하거나 삭제할 경우 사용되는 테이블 백업 파일의 I/O 버퍼 크기를 나타낸다.

**TABLE\_COMPACT\_AT\_SHUTDOWN****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

데이터베이스 종료 시 테이블을 컴팩트할지 여부를 나타낸다. 데이터베이스 재시작시 테이블을 위한 메모리 낭비를 줄이기 위해 1로 설정하기를 권장한다.

**TEMP\_HASH\_BUCKET\_DENSITY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 100]

**설명**

해시 버킷 하나가 관리해야 하는 임시 테이블의 페이지 프레임 개수를 백분율로 명시한다. 이 값이 커질수록 필요한 해시 버킷의 개수가 줄어들어 메모리는 적게 사용되지만, 한 버킷당 관리해야 할 임시 페이지 프레임의 수가 증가하기 때문에 연산 비용이 커진다.

예를 들어 임시 페이지 프레임의 개수가 100일 때, 이 값이 1이면 버킷의 개수는 프레임의 개수와 같아진다. 이 값이 2이면 버킷의 개수는 프레임 개수의 절반인 50이 된다.

**TEMP\_PAGE\_CHUNK\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위**[1, 2<sup>32</sup>-1]**설명**

임시 데이터 페이지를 한번에 할당하는 개수이다.

**USER\_DATA\_FILE\_INIT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

100 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

사용자 디스크 데이터 테이블스페이스에 사용자 정의 데이터 파일을 생성하거나 추가할 때 데이터 파일의 초기 크기를 명시한다. 데이터 파일을 생성하거나 추가할 때 초기 크기를 명시하지 않은 경우 기본값으로 사용된다.

**USER\_DATA\_FILE\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

2 \* 1024 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

사용자 디스크 데이터 테이블스페이스에 사용자 정의 데이터 파일을 생성하거나 추가할 때 데이터 파일의 최대 크기를 명시한다.

최소 USER\_DATA\_FILE\_INIT\_SIZE 이상의 크기를 가져야 하며, 가능한 최솟값은 64K이다. 테이블 스페이스 생성시 최대 크기를 명시하지 않은 경우 기본 최대 크기를 의미한다.



**USER\_DATA\_FILE\_NEXT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

1 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

사용자 디스크 데이터 테이블스페이스의 사용자 정의 데이터 파일에 데이터 파일 공간이 부족한 경우, 명시된 값만큼 데이터 파일의 크기를 확장한다.

**USER\_DATA\_TBS\_EXTENT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

512 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**[2 \* 8kB, 2<sup>64</sup>-1]**설명**

사용자 디스크 데이터 테이블스페이스<sup>5</sup> 생성 시 익스텐트의 크기를 명시한다.

[<sup>5</sup>] 사용자 디스크 데이터 테이블스페이스 (User disk data tablespace): 사용자의 객체를 저장하기 위한 테이블스페이스이다. 데이터베이스 객체 중 디스크 테이블과 디스크 인덱스만 저장된다.

**USER\_TEMP\_FILE\_INIT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

100 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

사용자 임시 테이블스페이스에 사용자 정의 임시 데이터 파일을 생성하거나 추가할 때 데이터 파일의 초기 크기를 명시한다. 임시 데이터 파일을 생성하거나 추가할 때 초기 크기를 명시하지 않은 경우 기본 크기로 사용된다.

**USER\_TEMP\_FILE\_MAX\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

2 \* 1024 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

사용자 임시 테이블스페이스에 사용자 정의 임시 데이터 파일을 생성하거나 추가할 때 할당 될 데이터파일의 최대 크기를 명시한다.

최소 USER\_DATA\_FILE\_INIT\_SIZE 이상의 크기를 가져야 하며, 가능한 최솟값은 64kB이다. 임시 데이터 파일을 생성하거나 추가할 때 최대 크기를 지정하지 않은 경우 이 프로퍼티가 최대 크기로 사용된다.

**USER\_TEMP\_FILE\_NEXT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

1 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[8 \* 8kB, 32GB]

**설명**

사용자 임시 테이블스페이스에 사용자 정의 임시 데이터 파일에 데이터 파일의 공간이 부족한 경우 명시된 값만큼 데이터 파일의 크기를 확장한다.

**USER\_TEMP\_TBS\_EXTENT\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

512 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**[5 \* 8kB,  $2^{64}-1$ ]**설명**

사용자 임시 테이블스페이스(User temporary tablespace) 생성 시 익스텐트의 크기를 명시한다. 최소 2개 페이지 (16kB = 2\*8kB) 이상의 크기를 가져야 한다.

**VOLATILE\_MAX\_DB\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값** $2^{32}+1$ **속성**

읽기 전용, 단일 값

**값의 범위**32비트: [2097152,  $2^{32}+1$ ]64비트: [2097152,  $2^{64}$ ]**설명**

모든 휘발성 테이블스페이스 크기의 합에 대한 최댓값을 설정한다. 운영 체제에서 제공하는 메모리 공간의 크기를 초과할 수 없다.

## 성능 관련 프로퍼티

### AGER\_WAIT\_MAXIMUM (단위: 마이크로초)

#### 데이터 타입

Unsigned Integer

#### 기본값

1000000

#### 속성

읽기 전용, 단일 값

#### 값의 범위

$[0, 2^{32}-1]$

#### 설명

가비지 콜렉터(garbage collector, 혹은 Ager) 관련 스레드들이 가비지 콜렉터 sleep 시, 시스템 호출인 sleep의 과도한 사용으로 인하여 (특히, HP 시스템) 발생하는 서버의 성능 저하를 막기 위한 것이다. 이 값을 이용하여 서버 운영 중에 가비지 콜렉터 sleep time을 적절히 조절할 수 있도록 한다.

### AGER\_WAIT\_MINIMUM (단위: 마이크로초)

#### 데이터 타입

Unsigned Integer

#### 기본값

200000

#### 속성

읽기 전용, 단일 값

#### 값의 범위

$[0, 2^{32}-1]$

#### 설명

가비지 콜렉터(garbage collector, 혹은 Ager) 관련 스레드들이 가비지 콜렉터 sleep 시 시스템 호출인 sleep의 과도한 사용으로 인하여 (특히, HP 시스템) 발생하는 서버의 성능 저하를 막기 위한 것이다. 이 값을 이용하여 서버 운영 중에 가비지 콜렉터 sleep time을 적절히 조절할 수 있도록 한다.

**BUFFER\_VICTIM\_SEARCH\_INTERVAL (단위: 밀리초)****데이터 타입**

Unsigned Integer

**기본값**

3000

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 86400000]

**설명**

이 프로퍼티는 버퍼 교체 대상 검색이 실패한 후 플러셔가 플러시 작업을 할 것을 기다리는 시간을 명시한다. 이 시간을 대기하여도 교체 대상 버퍼를 찾지 못하면, V\$BUFFPOOL\_STAT 성능 뷰의 VICTIM\_SEARCH\_WARP의 값이 증가한다.

**BUFFER\_VICTIM\_SEARCH\_PCT(단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

5

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

버퍼 교체 대상을 LRU 리스트에서 검색할 때 얼마나 탐색할 것인지를 명시한다. 이 프로퍼티는 하나의 LRU 리스트 전체를 100이라고 할 때, LRU Cold last를 기준으로 명시한 값의 퍼센트만큼 검색하는 것을 나타낸다.

**CHECKPOINT\_BULK\_SYNC\_PAGE\_COUNT (단위: 페이지 개수)****데이터 타입**

Unsigned Integer

**기본값**

3200

**속성**

변경 가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

체크포인트를 할 때 메모리와 디스크의 데이터를 일치시키기 위해 한 번에 몇 개의 페이지 단위로 내용을 일치시킬 것 인지를 나타낸다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**CHECKPOINT\_BULK\_WRITE\_PAGE\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

체크포인트시 메모리의 더티 페이지들을 여러 번에 나눠서 디스크로 저장할 수 있다. 이 때 이 프로퍼티를 사용해서 한 번에 저장할 수 있는 더티 페이지의 개수를 설정할 수 있다. 이 값이 0일 때는 한 번에 모든 더티 페이지가 디스크 데이터베이스로 저장될 것이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**CHECKPOINT\_BULK\_WRITE\_SLEEP\_SEC (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2592000]

**설명**

CHECKPOINT\_BULK\_WRITE\_PAGE\_COUNT의 값이 0이 아닐 때 더티 페이지들을 디스크로 저장 후 대기하는 시간(초)을 나타낸다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**CHECKPOINT\_BULK\_WRITE\_SLEEP\_USEC (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 60000000]

**설명**

CHECKPOINT\_BULK\_WRITE\_PAGE\_COUNT의 값이 0이 아닐 때 더티 페이지들을 디스크로 저장 후 대기하는 시간(마이크로초)을 나타낸다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**CHECKPOINT\_FLUSH\_COUNT (단위: 프레임 개수)****데이터 타입**

Unsigned Integer

**기본값**

64

**속성**

변경 가능, 단일 값

**값의 범위**[1,  $2^{64}-1$ ]

**설명**

플러셔가 한 번의 주기에서 체크포인트 플러시를 할 때 플러시 할 수 있는 버퍼 페이지(프레임)의 개수를 명시한다.

**CHECKPOINT\_FLUSH\_MAX\_GAP (단위: 로그 파일 개수)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

체크포인트 플러시를 발생시키는 조건 중의 하나가 된다. 체크포인트 리스트에 있는 갱신된 버퍼들 중에 갱신 LSN(recovery LSN)이 가장 작은 값과 현재 로그 LSN과의 차이가 이 프로퍼티에서 지정한 값이 되면 체크포인트 플러시를 수행한다.

이 값은 서버 재구동시의 복구 시간을 결정한다. 이 값이 클수록 체크포인트 플러시가 적게 발생하며, 서버의 재구동시 복구 시간은 길어진다. 서버 운영 중에도 변경이 가능하다.

**CHECKPOINT\_FLUSH\_MAX\_WAIT\_SEC (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

체크포인트 플러시를 발생시키는 조건 중의 하나이다. 마지막 플러시 작업이 끝난 이후 이 프로퍼티에서 지정한 시간이 지나면 체크포인트 플러시가 발생한다.



**CM\_BUFFER\_MAX\_PENDING\_LIST****데이터 타입**

Unsigned Integer

**기본값**

512

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 512]

**설명**

메모리가 급격하게 증가하는 것을 방지하기 위해, 이 프로퍼티를 사용해서 한 세션에서 할당받을 수 있는 최대 통신 버퍼 블록의 수를 제한할 수 있다.

**CM\_DISPATCHER\_SOCK\_POLL\_TYPE****데이터 타입**

Unsigned Integer

**기본값**

3

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 2, 3]

**설명**

Altibase 서버의 디스패처(Dispatcher)에서 소켓 감지를 위한 시스템 콜을 선택하는 프로퍼티이다. 클라이언트의 동시 접속이 1,000개 이상일 때 POLL() 시스템 콜이 사용되면 성능 향상에 도움이 된다.

- 1: SELECT() 시스템 콜
- 2: POLL() 시스템 콜; 지원되는 운영시스템에서만 사용이 가능함
- 3: EPOLL() 시스템 콜; 지원되는 운영시스템에서만 사용이 가능함

**DATABASE\_IO\_TYPE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

데이터파일과 관련하여 디스크 I/O가 수행될 때, Direct I/O와 Buffered I/O 두 가지 방법이 사용될 수 있다. Direct I/O를 사용하기 위해서는 이 프로퍼티의 값을 1로 설정하며 Buffered I/O를 사용하려면 0으로 설정한다.

Direct I/O는 디스크 I/O가 발생하는 동안 CPU 점유율을 줄인다는 장점이 있다. Buffered I/O는 read-ahead, asynchronous write 기법을 사용하므로 디스크 I/O 요구가 있을 때마다 실제로 디스크에 접근하지 않을 수도 있으므로 응용 프로그램 수준에서 볼 때 디스크 I/O가 훨씬 빠르다는 장점이 있다.

**DATAFILE\_WRITE\_UNIT\_SIZE (단위: 개수)****데이터 타입**

Unsigned Long

**기본값**

1024

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 1024]

**설명**

데이터 파일을 생성할 때 데이터의 기본 단위를 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DB\_FILE\_MULTIPAGE\_READ\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

8

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 128]

**설명**

디스크 테이블을 풀 스캔(Full Scan)할 때 이 값에 정해진 페이지 개수 단위로 IO를 수행한다.

이 때 디스크 테이블의 익스텐트 사이즈 즉 익스텐트 내의 페이지 개수가 여기에 명시한 값의 배수이면서 이 값보다 크면 MPR(Multi Page Read)을 한다.

그러나 여기에 명시한 값의 배수가 아니고, 이 값보다 작으면 SPR(Single Page Read)로 수행한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DEDICATED\_THREAD\_CHECK\_INTERVAL (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

3600

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,  $2^{64}-1$ ]

**설명**

전용 쓰레드(Dedicated thread) 모드에서 서비스 쓰레드가 정리되는 시간 간격을 설정하는 프로퍼티이다. 이 프로퍼티는 DEDICATED\_THREAD\_MODE 프로퍼티가 1일 때에만 유효하다.

이 프로퍼티에 설정한 주기로 작업이 없는 서비스 쓰레드가 해제된다. 단, DEDICATED\_THREAD\_INIT\_COUNT에 설정된 개수만큼의 서비스 쓰레드는 유지된다. 이 프로퍼티의 값이 0인 경우, 서비스 쓰레드가 정리되지 않는다.

**DEDICATED\_THREAD\_INIT\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 65535]

**설명**

전용 스레드(Dedicated thread) 모드에서 초기에 생성할 서비스 스레드의 개수를 지정하는 프로퍼티이다. 이 프로퍼티는 DEDICATED\_THREAD\_MODE 프로퍼티가 1일 때에만 유효하다.

DEDICATED\_THREAD\_INIT\_COUNT 값은 DEDICATED\_THREAD\_MAX\_COUNT 프로퍼티에 설정한 값보다 작아야 한다.

**DEDICATED\_THREAD\_MAX\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 65535]

**설명**

전용 스레드(Dedicated thread) 모드에서 최대 생성할 수 있는 서비스 스레드의 개수를 지정하는 프로퍼티이다. 이 프로퍼티는 DEDICATED\_THREAD\_MODE 프로퍼티가 1일 때에만 유효하다.

초기에 생성된 서비스 스레드가 전부 클라이언트에 할당된 경우, 서비스 스레드의 개수는 DEDICATED\_THREAD\_MAX\_COUNT까지 증가하게 된다.

DEDICATED\_THREAD\_MAX\_COUNT 값은 MAX\_CLIENT 프로퍼티에 설정한 값보다 작아야 한다.

**DEDICATED\_THREAD\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

서비스 스레드가 클라이언트에 대해 전용 스레드(Dedicated thread) 모드로 동작할지 여부를 설정하는 프로퍼티이다.

1인 경우 전용 스레드 모드로 동작하고, 0인 경우 멀티플렉싱 스레드(Multiplexing thread) 모드로 동작한다.

**DEFAULT\_FLUSHER\_WAIT\_SEC (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**[1, 2<sup>32</sup>-1]**설명**

플러셔의 최소 대기 시간을 명시한다. 플러셔는 특정 조건을 제외하고, 항상 이 시간만큼 대기 후 플러시 작업을 수행한다. 플러셔가 대기에서 풀려나 아무런 플러시를 하지 않는다면, 대기 시간은 1초씩 계속 늘어난다.

**DELAYED\_FLUSH\_LIST\_PCT (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

30

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

지연된 플러시 목록의 최대 비율을 명시한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DELAYED\_FLUSH\_PROTECTION\_TIME\_MSEC (단위: 밀리초)****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100000]

**설명**

최근에 사용된 Page로 판단하기 위한 기준 시간이다. 마지막으로 사용된 시간이 설정 값 이하라면 최근에 사용된 Page로 판단한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DIRECT\_IO\_ENABLED****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

데이터베이스를 디스크로 직접 입출력할 것인지 여부를 나타낸다.

0 : disable (직접 입출력하지 않음)

1 : enable (직접 입출력함)

**DISK\_INDEX\_BUILD\_MERGE\_PAGE\_COUNT (단위: 페이지 개수)****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

변경 가능, 단일 값

**값의 범위** $[2, 2^{32}-1]$ **설명**

디스크 인덱스를 생성할 때 데이터에서 추출된 키들을 메모리에서 한번에 정렬할 수 없을 경우, 외부 정렬에 사용될 페이지의 수를 나타낸다.

운영 중 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.

**EXECUTE\_STMT\_MEMORY\_MAXIMUM (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

1073741824

**속성**

변경 가능, 단일 값

**값의 범위** $[1024*1024, 2^{64}-1]$ **설명**

하나의 질의문(statement)이 사용할 수 있는 execute 메모리의 양을 제한한다.

이 프로퍼티는 가동중에 ALTER SYSTEM 구문으로 변경 가능하다.

**EXECUTOR\_FAST\_SIMPLE\_QUERY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2]

**설명**

이 프로퍼티는 단순한 DML 구문을 최적화하거나 Nested Loop Join을 사용하는 구문의 실행 계획을 최적화하여 성능을 향상시킬 수 있다. DA 모드에서 사용할 때 성능을 기대할 수 있으며, SIMPLE QUERY PLAN으로 실행되면 PLAN TREE에서 확인할 수 있다. 단 SIMPLE QUERY 최적화를 사용할 때에는 이전 동작과 같은 동작을 보장할 수 없으므로, 주의를 기울여야 한다.

0: SIMPLE QUERY PLAN을 사용하지 않는다.

1: SIMPLE QUERY PLAN을 MEMORY TABLE에 사용한다.

2: SIMPLE QUERY PLAN을 MEMORY PARTITION TABLE에 사용한다.

**FAST\_START\_IO\_TARGET (단위: 페이지 개수)****데이터 타입**

Unsigned Long

**기본값**

10000

**속성**

변경 가능, 단일 값

**값의 범위**[1, 2<sup>64</sup>-1]**설명**

서버를 재구동 단계에서 복구할 때 읽을 리두(Redo) 페이지의 개수를 명시한다.

운영중에 플래서가 체크포인트 플러시를 할 때 버퍼에 남은 더티 페이지가 이 프로퍼티에서 지정한 페이지보다 많다면, 그 차이만큼 오래된 더티 페이지부터 디스크에 반영한다.

이 값은 서버 재구동시에 복구 시간을 결정해준다. 이 값이 작을수록 플러시할 페이지 수가 많아지기 때문에 서버를 재구동할 때 소요되는 복구 시간을 줄일 수 있다.

서버 운영중에 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.



**FAST\_START\_LOGFILE\_TARGET (단위: 로그 파일 개수)****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

변경 가능, 단일 값

**값의 범위** $[1, 2^{32}-1]$ **설명**

서버를 재구동하여 복구할 때 읽을 로그 파일의 개수를 명시한다.

운영중에 플러셔가 체크포인트 플러시를 할 때, 체크포인트 리스트에 있는 더티 페이지 중 페이지 LSN의 LogFileNo와 현재 로그 LSN의 LogFileNo와의 차이가 이 프로퍼티에서 지정한 값보다 크면 페이지를 플러시한다.

이 값은 서버를 재구동할 때 복구 시간을 결정한다. 이 값이 작을수록 플러시 할 페이지 수는 많아지기 때문에 서버를 재구동할 때 소요되는 복구 시간을 줄일 수 있다.

서버 운영중에 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.

**FAST\_UNLOCK\_LOG\_ALLOC\_MUTEX****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 1]$ **설명**

서버의 여러 스레드가 로그 버퍼에 동시에 접근할 수 있는지를 설정하는 프로퍼티이다. 이 프로퍼티의 값을 0으로 설정하면, 서버의 여러 스레드가 로그 버퍼에 순차적으로 접근할 수 있다. 그리고 이 값을 1로 설정하면, 서버의 스레드들이 로그 버퍼로 동시에 접근할 수 있다.

0: 동시 접근 불가능

1: 동시 접근 가능

**HASH\_AREA\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

4MB

**속성**

변경 가능, 단일 값

**값의 범위**[3M, 2<sup>64</sup>-1]**설명**

서버가 해싱(hashing) 작업을 위해 사용하는 임시 테이블 하나의 메모리 크기를 지정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**HASH\_JOIN\_MEM\_TEMP\_AUTO\_BUCKET\_COUNT\_DISABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

해쉬 조인(Hash Join)에 사용되는 메모리 해쉬 임시 테이블의 필요한 버킷(Bucket) 개수를 설정한다.

0 : 해쉬 테이블에 삽입된 실제 레코드 개수 ( DISTINCT 해싱방식에 의해 처리 되는 경우 /\*+ HASH BUCKET COUNT () \*/ HINT로 HSDS 노드의 해시 버킷 수 지정 가능)

1 : 쿼리 옵티마이저가 예측한 버킷 개수 또는 힌트 /\*+ HASH BUCKET COUNT () \*/로 지정된 버킷 개수

HASH\_JOIN\_MEM\_TEMP\_PARTITIONING\_DISABLE 프로퍼티의 값이 0(파티셔닝 방식)이면, 실제 레코드 개수에 기반하여 버킷 개수만큼 구역을 나누는 것이 효과적이다.

HASH\_JOIN\_MEM\_TEMP\_PARTITIONING\_DISABLE 프로퍼티의 값이 1(버킷 방식)이면, 이 프로퍼티의 값과 관계 없이 쿼리 옵티마이저가 예측하거나 힌트로 지정한 버킷 개수를 사용한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**HASH\_JOIN\_MEM\_TEMP\_PARTITIONING\_DISABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

해쉬 조인(Hash Join)에 사용되는 메모리 해쉬 임시 테이블의 레코드를 어떠한 방식으로 삽입하고 탐색할 것인지를 설정할 수 있다.

0 : 파티셔닝 방식

1 : 버킷 방식

파티셔닝 방식은 임시 공간에 삽입된 레코드의 개수를 기반으로 파티션 개수를 자동 설정하여 레코드를 파티션 단위로 탐색한다. 버킷 방식은 레코드를 버킷에 리스트 형태로 저장하여 버킷 단위로 레코드를 탐색하는 방식이다. 파티셔닝 방식을 사용할 경우 추가적인 메모리를 요구하지만, 대용량 데이터베이스를 사용할 때에는 버킷 방식보다 삽입 및 탐색 속도가 효과적이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**HIGH\_FLUSH\_PCT (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

5

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

플러셔가 대기 상태에서 풀렸을 때 플러시 리스트의 길이가 전체 버퍼 크기에서 명시된 값 이상이 되면 교체 플러시를 수행한다. 이 때 해당되는 플러시 리스트의 모든 갱신된 버퍼들이 대기없이 연속적으로 플러시된다. 이 프로퍼티는 서버 운영중에 변경이 가능하다.

**HOT\_LIST\_PCT (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

LRU 리스트 내에서 hot 영역의 비중을 명시한다. 이 프로퍼티는 서버 운영중에 변경이 가능하다.

**HOT\_TOUCH\_CNT****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

변경 가능, 단일 값

**값의 범위**[1,  $2^{32}-1$ ]**설명**

버퍼가 hot하다고 판단하는 접근 횟수의 기준을 명시한다. 이 속성에서 명시한 값 이상으로 버퍼에 접근되면 그 버퍼는 hot이라고 판단되며, hot 버퍼는 교체 대상 검색시 hot 리스트로 이동한다.

**INDEX\_BUILD\_THREAD\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

물리 코어 수

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 512]

**설명**

실행시간(runtime)에 인덱스를 재구축(rebuilding)하는 과정에서 생성되는 인덱스를 구축(index build)하는 쓰레드의 개수를 조절한다.

주석으로 처리할 경우 기본값으로 시스템의 물리 코어 수(N) 만큼 병렬 작업 쓰레드가 생성된다.

**INDEX\_INITTRANS (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

8

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 30]

**설명**

인덱스 페이지에 유지될 TTS(Touched Transaction Slots)의 초기 개수를 나타낸다.

**INDEX\_MAXTRANS (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

30

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 30]

**설명**

인덱스 페이지에 유지될 수 있는 TTS(Touched Transaction Slots)의 최대 개수를 나타낸다.

**LFG\_GROUP\_COMMIT\_INTERVAL\_USEC (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

1000

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

본 프로퍼티는 그룹 커밋 관련 프로퍼티이다.

트랜잭션 커밋을 위해 로그를 디스크에 기록하는 디스크 I/O를 수행한 마지막 시각을 가지고 있다. 이 시각을 기준으로 이 프로퍼티가 지정한 시간이 지난 후에 로그를 디스크에 기록하는 디스크 I/O가 수행된다.

이를 통해, 여러 트랜잭션이 동시 다발적으로 커밋하면서 요청하는 디스크 I/O가 한꺼번에 수행된다.

**LFG\_GROUP\_COMMIT\_RETRY\_USEC (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 60000000]$

**설명**

본 프로퍼티는 그룹커밋 관련 프로퍼티이다.

마지막으로 로그 기록을 위한 디스크 I/O를 수행한 시각 이후로 LFG\_GROUP\_COMMIT\_INTERVAL\_USEC 만큼의 시간이 지나지 않은 경우, 커밋을 하려는 트랜잭션은 사용자가 이 프로퍼티에 지정한 값 만큼 기다린 후 디스크 I/O를 수행할 수 있는 충분한 시간이 지났는지를 다시 체크한다.

**LFG\_GROUP\_COMMIT\_UPDATE\_TX\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

80

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

본 프로퍼티는 그룹커밋 관련 프로퍼티이다.

데이터베이스에 변경(UPDATE)을 가한 트랜잭션의 수(V\$LFG의 UPDATE\_TX\_COUNT 칼럼에서 조회 가능)가 이 값보다 크거나 같은 값이 되면 그룹 커밋이 활성화된다. 만약 이 프로퍼티가 0으로 설정되면 그룹커밋은 작동하지 않는다.

**LOB\_CACHE\_THRESHOLD (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

8192

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 524288]$ **설명**

클라이언트 LOB 캐시에 저장될 수 있는 LOB 데이터의 최대 크기를 지정하는 프로퍼티이다. 이 프로퍼티에 0보다 큰 값을 지정하면, 설정한 크기 이하의 LOB 데이터는 클라이언트에 임시로 저장된다. 0으로 설정하면 클라이언트에 LOB 데이터가 임시 저장되지 않는다.

이 프로퍼티를 이용해서 임계치를 적절히 조절하면, LOB 데이터가 클라이언트 LOB 캐시에 저장되어 Bulk-select 속도가 향상된다.

Altibase 운영 중 ALTER SYSTEM 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## LOCK\_ESCALATION\_MEMORY\_SIZE (단위: 바이트)

### 데이터 타입

Unsigned Integer

### 기본값

100M

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1000MB]

### 설명

메모리 테이블에 대해 다량의 변경(UPDATE) 배치 작업을 할 때, 버전(versioning)으로 인해 메모리 사용량이 크게 증가하는 것을 방지하기 위한 프로퍼티이다. 어떤 DML에 의해 생성되는 로그의 크기가 이 프로퍼티에 설정된 값보다 커지면 버전을 만들지 않고 "inplace update"<sup>6</sup>를 하여 메모리 사용량이 증가하는 것을 막는다.

[<sup>6</sup>] inplace update란 변경 대상이 되는 원래 레코드에 대해 버전(version)을 만들지 않고 해당 칼럼의 값이 바로 수정되는 것을 의미한다.

변경시 버전 기법을 사용하면, 레코드 레벨의 X 잠금(lock)을 획득하고 테이블 레벨의 IX 잠금을 획득한다. 그러나 inplace update시에는 테이블 레벨에도 X 잠금 즉, 배타적 잠금을 획득한다. 따라서, 이 값을 너무 작게 설정하면 해당 테이블에 대한 확장성(scalability)이 떨어질 수 있으므로 유의해야 한다.

Altibase 운영 중 ALTER SYSTEM 문으로 이 프로퍼티의 값을 변경할 수 있다.

## LOG\_IO\_TYPE

### 데이터 타입

Unsigned Integer

### 기본값

1

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 1]

### 설명

로그를 기록할 때의 I/O 모드를 나타낸다.

0: buffered I/O

1: direct I/O



**LOG\_CREATE\_METHOD****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

로그 파일을 생성할 때 사용되는 시스템 콜을 선택하는 프로퍼티이다. awrite 유틸리티로 출력되는 값을 참고하여 선택적으로 사용하면 성능 향상에 도움이 된다. awrite 유틸리티에 대한 자세한 내용은 Utilities Manual을 참고한다.

0 : write() 시스템 콜

1: fallocate() 시스템 콜 ; 지원되는 운영 시스템에서만 사용이 가능함

**LOW\_FLUSH\_PCT (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

플러시 리스트의 길이가 전체 버퍼 크기에서 명시된 값 이상이 되면 교체 플러시를 수행한다. 이 때 해당 플러시 리스트의 모든 갱신 버퍼들은 플러시한다.

**LOW\_PREPARE\_PCT (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

플러셔가 대기에서 깨어났을 때 Prepare 리스트의 길이가 전체 버퍼에서 명시한 값 이하가 되면 교체 플러시를 수행한다. 이 때 해당 플러시 리스트의 모든 갱신 버퍼들은 플러시한다.

**MATHEMATICS\_TEMP\_MEMORY\_MAXIMUM (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**[0,  $2^{64}-1$ ]**설명**

시스템 전체에서 분석 함수가 사용하는 MATHEMATICS\_TEMP의 메모리 양을 제한한다. ( 분석 함수 : listagg, percentile\_cont, percentile\_disc, ... ) 사용한 메모리 양이 MATHEMATICS\_TEMP\_MEMORY\_MAXIMUM과 같거나 크다면 에러가 발생한다. 속성 값이 0 인 경우에 메모리 사용량을 검사하지 않는다.

**MAX\_FLUSHER\_WAIT\_SEC (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**[1, 2<sup>32</sup>-1]**설명**

플러셔의 최대 대기 시간을 명시한다. 플러셔의 대기 시간은 작업 빈도에 따라 계속 늘어날 수 있지만, 이 값을 넘어가지는 못한다.

**MEM\_INDEX\_KEY\_REDISTRIBUTION****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

메모리 인덱스의 키를 재분배하는 프로퍼티이다.

이 프로퍼티는 역순의 인덱스 키가 다수 삽입될 때, 리프 노드의 분열이 많이 발생할 때, 데이터에 비해 인덱스 범위가 클 때 사용하면 효과적이다.

0: 메모리 인덱스의 키를 재분배하지 않는다.

1: 메모리 인덱스의 키를 재분배한다. (기본값)

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MEM\_INDEX\_KEY\_REDISTRIBUTION\_STANDARD\_RATE****데이터 타입**

Unsigned Integer

**기본값**

50

**속성**

읽기 전용, 단일 값

**값의 범위**

[10, 90]

**설명**

메모리 인덱스의 키를 재분배하기 위해 이웃 노드의 미사용 공간 비율의 최솟값을 설정한다. 예를 들어 이 프로퍼티 값이 30이면 리프 노드의 빈 공간이 최소 30% 이상이 있을 때 메모리 인덱스의 키를 재분배한다.

MEM\_INDEX\_KEY\_REDISTRIBUTION 프로퍼티가 0이면 이 프로퍼티는 동작하지 않는다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MULTIPLEXING\_CHECK\_INTERVAL (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

200000

**속성**

변경 가능, 단일 값

**값의 범위**

[100000, 10000000]

**설명**

쓰레드 매니저가 서비스 쓰레드 분산을 위해 세션을 확인하는 주기를 나타낸다. 시간 단위는 마이크로초이다.

쓰레드 매니저는 주기적으로 세션의 상태를 확인하여 세션의 통계 정보를 갱신하며, 서비스 쓰레드를 추가 또는 삭제한다.

**MULTIPLEXING\_MAX\_THREAD\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 1024]

**설명**

멀티플렉싱을 하는 쓰레드의 최대 개수다.

기존에 존재하는 쓰레드들의 부하가 증가하면 쓰레드가 자동으로 증가한다. 그러나 쓰레드들이 계속 증가하게 되면 오히려 성능이 저하될수 있으므로 적절한 값을 설정해야 한다.

단 큐(QUEUE)를 사용하는 경우에는 이 프로퍼티에서 설정한 값보다 더 많은 쓰레드가 생성될수 있다.

**MULTIPLEXING\_THREAD\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

호스트 장비의 논리 코어 수

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 1024]

**설명**

Altibase가 유지하는 서비스 쓰레드의 최소 개수이다. 기본값은 장비의 논리 코어 수이다.

이 값은 서버가 구동된 후에는 변경할 수 없다.

**NORMALFORM\_MAXIMUM****데이터 타입**

Unsigned Integer

**기본값**

2048

**속성**

변경 가능, 단일 값

**값의 범위**

[1,  $2^{32}-1$ ]

**설명**

조건절을 정규화할 때 정규화 형(Normal Form) 노드의 최대 개수를 지정할 수 있는 프로퍼티이다.

SELECT 질의문의 WHERE절에 존재하는 predicate들이 논리 연산자(AND, OR)로 복잡하게 연결되어 있을 때, Altibase는 테이블을 더 빠르게 탐색하기 위하여 predicate을 정규화시킨다.

정규화 방법으로 CNF(Conjunctive Normal Form)와 DNF(Disjunctive Normal Form)가 있으며, 이 프로퍼티에 명시한 노드 개수를 초과하면 해당 정규화 형으로 더 이상 정규화를 시도하지 않는다.

만약 두 가지 정규화 형이 명시한 노드 개수를 초과하면, 조건절을 정규화하지 않고 NNF(Not Normal Form)를 수행한다. NNF를 사용하면 조건절이 정규화되지 않아서 인덱스를 사용할 수 없다.

그러나 조건절을 지나치게 복잡하게 작성한 경우 NNF를 피하고자 이 프로퍼티의 값을 지나치게 크게 설정하면, 복잡한 조건절 처리를 위해 정규화 비용과 메모리 사용량이 증가한다.

따라서 조건절 작성시에 논리연산자를 지나친 사용을 자제하고 정규화 형태로 조건절을 작성하는 것이 중요하다.

이 규칙은 ON 조건 조인의 ON predicate에도 동일하게 적용된다.

## OPTIMIZER\_AUTO\_STATS

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 10]

### 설명

옵티마이저가 사용할 수 있는 통계 정보가 없을 때 자동으로 통계 정보를 수집하는 프로퍼티이다. 수집할 수 있는 Page개수는 1~10까지의 값에 따라 아래의 표와 같다. 0으로 설정하면 통계 정보를 자동으로 수집할 수 없다.

Altibase 운영 중 ALTER SESSION, SYSTEM 문을 이용하여 변경할 수 있다.

값	PAGE 개수	통계 정보 수집 여부
0(기본값)	0	OFF
1	32	ON
2	64	ON
3	128	ON
4	256	ON
5	512	ON
6	1024	ON
7	4096 (32M)	ON
8	16384 (128M)	ON
9	65536 (512M)	ON
10	ALL	ON

## OPTIMIZER\_DELAYED\_EXECUTION

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

실행 계획의 그래프를 기준으로 hierarchy, sorting, windowing, grouping, set, distinction의 실행(execute)을 패치(fetch)에서 수행시킨다. 사용자는 실행 계획에서 상위 PROJECTION 아래에 DELAY 플랜이 추가된 것을 확인할 수 있다.

이 프로퍼티는 쿼리 수행 시 실행(execute)을 지연시켜 첫 번째 패치(fetch)에서 수행하도록 하여, 실행 시간을 줄이고 첫 번째 패치 시간을 늘린다. 첫 번째 패치 이후의 수행 시간은 동일하다. 관련된 수행 시간은 v\$statement 성능 뷰로 확인 가능하다.

0 : disable (비활성화, 기본값)

1 : enable (활성화)

Altibase 운영 중 ALTER SESSION문을 이용하여 변경할 수 있다.

## OPTIMIZER\_FEATURE\_ENABLE

### 데이터 타입

String

### 기본값

Altibase 서버 버전

### 속성

변경 가능, 단일 값

### 값의 범위

[6.1.1.0.6 | 6.1.1.0.7 | 6.3.1.0.1 | 6.5.1.0.0 | 7.1.0.0.0]

### 설명

쿼리 옵티마이저와 관련된 프로퍼티들을 일괄적으로 제어하는 프로퍼티이다.

이 프로퍼티는 값의 범위에 속하는 버전에 한해 ALTER SYSTEM문으로 변경할 수 있다.

```
e.g) ALTER SYSTEM SET optimizer_feature_enable = '6.1.1.0.6';
```

아래는 이 프로퍼티의 설정에 따라 쿼리 옵티마이저와 관련된 프로퍼티들이 어떤 값으로 설정되는지를 나타낸 표이다.

OPTIMIZER_FEATURE_ENABLE	6.1.1.0.6	6.1.1.0.7	6.3.1.0.1	6.5.1.0.0	7.1.0.0.0
QUERY_REWRITE_ENABLE	0 (off)	0 (off)	0 (off)	0 (off)	0 (off)
OPTIMIZER_UNNEST_SUBQUERY	0 (off)	0 (off)	1 (on)	1 (on)	1 (on)
OPTIMIZER_UNNEST_COMPLEX_SUBQUERY	0 (off)	0 (off)	1 (on)	1 (on)	1 (on)
OPTIMIZER_UNNEST_AGGREGATION_SUBQUERY	0 (off)	0 (off)	0 (off)	1 (on)	1 (on)
RESULT_CACHE_ENABLE	0 (off)	0 (off)	0 (off)	0 (off)	0 (off)
TOP_RESULT_CACHE_MODE	0 (off)	0 (off)	0 (off)	0 (off)	0 (off)
OPTIMIZER_AUTO_STATS	0 (off)	0 (off)	0 (off)	0 (off)	0 (off)
OPTIMIZER_PERFORMANCE_VIEW	0 (off)	0 (off)	0 (off)	0 (off)	1 (on)

## OPTIMIZER\_MODE

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

이 프로퍼티 값이 0으로 설정되면 질의문을 최적화하기 위하여 비용 기반 최적화(cost-based optimization)가 사용되고, 1이면 규칙 기반 최적화(rule-based optimization)가 사용된다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## OPTIMIZER\_PERFORMANCE\_VIEW

### 데이터 타입

Unsigned Integer

### 기본값

1



**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

성능 뷰를 최적화하기 위하여 메모리 제약 여부를 설정하는 프로퍼티이다. Altibase 시스템 정보를 모니터링 할 때 내 부적인 효율성 향상을 기대할 수 있다.

0: 적용 안함 - 메모리 제약이 없다.

1: 적용(기본값) - 메모리 제약이 있다. EXECUTE\_STMT\_MEMORY\_MAXIMUM 프로퍼티의 값을 초과하면 쿼리 실행이 실패한다.

**OPTIMIZER\_UNNEST\_AGGREGATION\_SUBQUERY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

SELECT 구문에 "GROUP BY절 없이 집계 함수만 사용한 중첩된 부질의(Nested Subquery)<sup>7</sup>"가 포함되어 있을 때, 쿼리 옵티마이저가 중첩 풀기(unnesting)를 할 것인지 여부를 지시하는 프로퍼티이다.

[<sup>7</sup>] 중첩된 부질의 (Nested Subquery): WHERE절에 포함된 부질의이다. 결과 집합을 한정하기 위해, 주로 메인 쿼리 (Main Query, 외부 질의)에 있는 칼럼을 참조하는 형태를 가진다. 이렇게 중첩된 부질의가 포함된 쿼리를 중첩되지 않은 조인 형태의 쿼리로 변환하는 것을 "Subquery Unnesting"이라고 한다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: unnesting을 시도하지 않는다.

1: unnesting을 시도한다.

**OPTIMIZER\_UNNEST\_COMPLEX\_SUBQUERY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

SELECT 구문에 복잡한 형태의 중첩된 부질의(Nested Subquery)가 포함되어 있을 때, 쿼리 옵티마이저가 중첩 풀기(unnesting)를 할 것인지 여부를 지시하는 프로퍼티이다. 복잡한 형태란 FROM절에 뷰 또는 두 개 이상의 테이블이 있거나, 또는 SELECT, FROM, WHERE절 이외의 다른 절이 포함된 것을 말한다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: unnesting을 시도하지 않는다.

1: unnesting을 시도한다.

**OPTIMIZER\_UNNEST\_SUBQUERY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

SELECT 구문에 "중첩된 부질의(Nested Subquery)"가 포함되어 있을 때, 쿼리 옵티마이저가 중첩 풀기(unnesting)를 할 것인지 여부를 지시하는 프로퍼티이다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: unnesting을 시도하지 않는다.

1: unnesting을 시도한다.

## OUTER\_JOIN\_OPERATOR\_TRANSFORM\_ENABLE

### 데이터 타입

Unsigned Integer

### 기본값

1

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 1]

### 설명

최적화된 쿼리의 실행 계획이 수행되도록, 사용자가 직접 ANSI/ISO SQL 표준 기법으로 Outer Join 연산을 수행하도록 설정할 수 있다.

이 값을 0으로 설정하면, ANSI/ISO SQL 표준 기법의 Outer Join 연산을 수행하며, 쿼리문에 Oracle Outer Join Operator (+)이 사용되었다면 에러가 발생한다. 기본값은 1이며, Oracle Outer Join Operator(+)를 사용한다.

0 : Disable

1 : Enable (기본값, Oracle Outer Join Operator 사용)

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## PARALLEL\_LOAD\_FACTOR

### 데이터 타입

Unsigned Integer

### 기본값

2N (N:논리 코어 수)

### 속성

읽기 전용, 단일 값

### 값의 범위

[1, 512]

### 설명

Altibase 서버 재가동 시 데이터베이스 정제 과정(database refining) 또는 인덱스 재구축(index rebuilding) 과정에서 생성되는 데이터베이스 정제 쓰레드/인덱스 구축 쓰레드 개수를 조절하는 프로퍼티이다.

주석으로 처리할 경우 기본값으로 시스템의 논리 코어 수(N)의 두 배(2N)만큼 병렬 작업 쓰레드가 생성된다.

**PARALLEL\_QUERY\_THREAD\_MAX****데이터 타입**

Unsigned Integer

**기본값**

논리 코어 수

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 1024]

**설명**

Altibase 서버가 병렬 질의를 처리하기 위해 생성할 수 있는 최대한의 작업 스레드 개수를 명시한다. 이 프로퍼티를 명시하지 않을 경우 기본값은 시스템의 논리 코어 수이다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**PARALLEL\_QUERY\_QUEUE\_SIZE****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

변경 가능 전용, 단일 값

**값의 범위**

[4, 1048576]

**설명**

Altibase 서버가 병렬 질의를 처리할 때, PARALLEL-QUEUE(PRLQ) 노드에서 데이터를 임시로 저장할 큐의 크기를 지정하는 프로퍼티이다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**PREPARE\_STMT\_MEMORY\_MAXIMUM (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

200M

**속성**

변경 가능, 단일 값

**값의 범위**[1024\*1024, 2<sup>64</sup>-1]**설명**

하나의 질의문(statement)이 사용할 수 있는 prepare 메모리의 양을 제한한다.

Altibase 운영 중 ALTER SYSTEM문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**QUERY\_REWRITE\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

Altibase 서버에서 쿼리 변환 시에 함수 기반 인덱스를 적용할지 여부를 지정하는 프로퍼티이다.

0: disable (미적용)

1: enable (적용)

**REFINE\_PAGE\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

50

**속성**

읽기 전용, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

Altibase 서버 가동 시 수행하는 단계 중 데이터베이스 정제(database refining) 단계가 있다. Altibase 서버가 이전 종료할 당시, 트랜잭션들이 생성한 버전 레코드(versioning record)들이 가비지 콜렉터에 의해 처리되지 못해서 불필요한 레코드들이 데이터베이스에 존재하고 있을 수 있으며 또한 서버 가동시의 회복 과정에서 생성된 버전 레코드들이 존재할 수 있다. 이들 레코드들을 재사용 가능하도록 처리하는 것이 데이터베이스 정제 단계이다.

데이터베이스 정제 대상이 되는 레코드들이 많을 경우 시간을 많이 소모할 수 있기 때문에 이 작업을 여러 스레드에 의해 병렬로 수행할 수 있다. 이 때 각 스레드가 처리하는 페이지 양을 지정한다.

**RESULT\_CACHE\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

Result Cache를 사용하여 중간 결과의 실행계획을 저장할 지 여부를 설정한다.

0: Disabled (기본값) - 캐시기능을 사용하지 않는다.

1: Enabled - 캐시기능을 사용한다.

Altibase 운영 중 ALTER SYSTEM 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**RESULT\_CACHE\_MEMORY\_MAXIMUM (단위:바이트)****데이터 타입**

Unsigned Long

**기본값**

10M

**속성**

변경 가능, 단일 값

**값의 범위**

[4096, ULONG MAX]

**설명**

Result Cache 및 Top Result Cache를 사용할 때, 저장하는 메모리 크기를 제한한다. 만약 이 크기를 넘으면 메모리에 저장되지 않고 해제된다. 이 프로퍼티는 쿼리 1개에 대한 제약이며 시스템 전체에 대한 제약은 제공하지 않는다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**SECONDARY\_BUFFER\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

보조 버퍼(Secondary Buffer) 기능을 사용하려면, 이 프로퍼티의 값을 설정한 후 반드시 SECONDARY\_BUFFER\_FILE\_DIRECTORY, SECONDARY\_BUFFER\_SIZE를 설정해야 서버가 구동된다.

보조 버퍼 기능을 사용하면 메모리 버퍼의 데이터 페이지를 디스크에 바로 플러시하지 않고 SSD(Solid-state Drive)의 보조 버퍼에 먼저 플러시한다. 그리고 Altibase 서버 내의 별도의 쓰레드가 보조 버퍼의 데이터 페이지를 디스크로 플러시한다.

0: 보조 버퍼 기능 비활성화

1: 보조 버퍼 기능 활성화

**SECONDARY\_BUFFER\_FILE\_DIRECTORY****데이터 타입**

String

**기본값**

""

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

보조 버퍼 파일이 존재할 디렉토리의 경로를 지정한다. 파일 이름은 sbuffer.sbf로 고정되며, 변경할 수 없다. 이 프로퍼티를 설정하지 않으면 SECONDARY\_BUFFER\_ENABLE 프로퍼티가 1이더라도 보조 버퍼 기능이 비활성화된다.

보조 버퍼 기능에 대한 설명은 SECONDARY\_BUFFER\_ENABLE 프로퍼티의 설명을 참고하도록 한다.

**SECONDARY\_BUFFER\_FLUSHER\_CNT****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 16]

**설명**

보조 버퍼의 데이터 페이지를 하드 디스크로 플러시하는 쓰레드(flusher)의 개수를 지정한다.

보조 버퍼 기능에 대한 설명은 SECONDARY\_BUFFER\_ENABLE 프로퍼티의 설명을 참고하도록 한다.

**SECONDARY\_BUFFER\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 32GB]

**설명**

보조 버퍼 파일의 크기를 명시한다. 이 값이 0이면 SECONDARY\_BUFFER\_ENABLE 프로퍼티가 1이더라도 보조 버퍼 기능이 비활성화된다.

보조 버퍼 기능에 대한 설명은 SECONDARY\_BUFFER\_ENABLE 프로퍼티의 설명을 참고하도록 한다.



**SECONDARY\_BUFFER\_TYPE****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 2]

**설명**

보조 버퍼에 저장할 데이터 페이지의 타입을 지정한다. 아래의 값 중 하나를 지정할 수 있다.

0: 모든 타입의 페이지

1: Dirty 페이지

2: Clean 페이지

보조 버퍼 기능에 대한 설명은 SECONDARY\_BUFFER\_ENABLE 프로퍼티의 설명을 참고하도록 한다.

**SORT\_AREA\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

1048576

**속성**

변경 가능, 단일 값

**값의 범위**[512,  $2^{64}-1$ ]**설명**

서버가 정렬 작업을 위해 사용하는 임시 테이블 하나의 메모리 크기를 지정한다.

시스템 운영 중에는 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.

**SQL\_PLAN\_CACHE\_BUCKET\_CNT****데이터 타입**

Unsigned Integer

**기본값**

127

**속성**

읽기 전용, 단일 값

**값의 범위**

[5, 4096]

**설명**

SQL 플랜 캐시에서 해쉬 테이블의 버킷 개수를 나타낸다.

**SQL\_PLAN\_CACHE\_HOT\_REGION\_LRU\_RATIO (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

50

**속성**

변경 가능, 단일 값

**값의 범위**

[10, 100]

**설명**

SQL 플랜 캐시에 있는 LRU 리스트의 HOT 영역 비율을 나타낸다. HOT 영역 LRU 리스트는 SQL 플랜 캐시에 있는 LRU 리스트에서 빈번하게 참조되는 플랜들을 별도의 HOT 영역으로 저장한 것이다.

시스템 운영 중 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.

**SQL\_PLAN\_CACHE\_PREPARED\_EXECUTION\_CONTEXT\_CNT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1024]

**설명**

플랜을 생성할 때 초기에 생성하는 execution context의 개수를 나타낸다.

플랜을 생성하기 전에 execution context의 개수를 지정하지만, 이는 초기에 생성하는 개수만을 결정할 뿐이다. 그리고 execution context는 실행 시간에 필요한 만큼 자동으로 증가되거나 감소된다.

그러나 하나의 플랜을 동시에 execute하는 경우 이 값을 증가시키는 것은 성능에 도움이 되지만, 그렇지 않은 경우에는 플랜의 크기만 증가시킬 뿐 성능 향상에 도움이 되지 않는다.

**SQL\_PLAN\_CACHE\_SIZE (단위: 바이트)****데이터 타입**

Unsigned long

**기본값**

64 M

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2<sup>64</sup>-1]

**설명**

SQL 플랜 캐시의 최대 크기를 나타낸다. 그러나 0으로 설정되면 캐시를 사용할 수 없게 된다. V\$SQL\_PLAN\_CACHE의 MAX\_CACHE\_SIZE로 확인할 수 있다.

시스템 운영 중에는 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.

**STATEMENT\_LIST\_PARTIAL\_SCAN\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

V\$STATEMENT, V\$SQLTEXT, 또는 V\$PLANTEXT 성능 뷰에 대한 SELECT 쿼리 결과로서 클라이언트에 반환할 statement의 최대 개수를 지정한다. 이 값이 0이면, 모든 statement에 대한 모든 레코드를 반환한다.

시스템 운영 중에는 ALTER SYSTEM 문을 이용해 프로퍼티의 값을 변경할 수 있다.

**TABLE\_INITTRANS****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 120]

**설명**

테이블 페이지에 유지될 TTS(Touched Transaction Slots)의 초기 개수를 나타낸다.

**TABLE\_LOCK\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

잠금(Lock) 수준을 제어하는 프로퍼티이다.

0: 테이블 잠금을 지원하지 않으며 레코드 잠금을 지원한다. 단순 DML 사용시에 성능이 향상된다.

1: 테이블 잠금과 레코드 잠금을 지원한다.

이 프로퍼티 값을 0으로 설정하면 다음과 같은 제약이 따른다.

- DDL문을 수행할 수 없다.
- CREATE DATABASE질의문을 수행할 수 없다.
- 이중화로 사용할 시에병렬 동기화 (parallel SYNC)를 사용할 수 없다.

이 프로퍼티는 ALTER SYSTEM 문으로 변경이 가능하다.

## TABLE\_LOCK\_MODE

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 1]

### 설명

테이블 파티션의 잠금(Lock) 수준을 설정하는 프로퍼티이다. TABLE\_LOCK\_ENABLE프로퍼티가 1인 경우에만 이 프로퍼티를 사용할 수 있다.

0: 테이블 파티션 단위의 잠금을 지원한다. 파티션 단위의 잠금을 사용하므로 ALTER TABLE...TRUNCATE PARTITION와 같이 파티션 단위로 수행되는 질의문 사용 시에 동시성이 향상된다.

1: 테이블 파티션의 잠금을 지원하지 않으며 파티션드 테이블의 논 파티션드 테이블 잠금과 동일하다. 잠금 횟수가 줄어들어 DML 성능이 향상된다.

## TABLE\_MAXTRANS

### 데이터 타입

Unsigned Integer

### 기본값

120

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 120]

**설명**

테이블 페이지에 유지될 수 있는 TTS(Touched Transaction Slots)의 최대 개수를 나타낸다.

**TABLESPACE\_LOCK\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

테이블스페이스 잠금(lock) 레벨을 지정하는 프로퍼티이다.

1: 테이블스페이스 잠금을 지원한다.

0: 테이블스페이스 잠금을 지원하지 않는다. 이 경우 단순 DML 수행 시 처리 속도 향상의 이점이 있지만, 다음과 같은 제약이 있다.

- DDL문을 수행할 수 없다.
- CREATE DATABASE 구문을 수행할 수 없다.
- 이중화 시 병렬 동기화(Parallel Synchronization)를 허용하지 않는다.

Altibase 운영 중 ALTER SYSTEM 또는 ALTER SESSION문을 이용하여 이 프로퍼티의 값을 변경할 수 있다. 단, 활성화된 트랜잭션이 없을 경우에만 변경할 수 있다.

**TEMP\_STATS\_WATCH\_TIME****데이터 타입**

Unsigned Int

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2<sup>32</sup>-1]

**설명**

통계 정보에 등록되는 기준 시간을 지정한다. 임시 테이블을 사용하는 연산 중 이 프로퍼티의 설정된 시간보다 오래 걸리는 연산은 통계정보로 등록된다. 임시 테이블에

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**THREAD\_CPU\_AFFINITY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

서비스 스레드를 CPU와 연계할지 여부를 설정하는 프로퍼티이다. CPU 연계 기능을 활성화하면 하나의 서비스 스레드는 동일한 CPU 코어에만 할당될 것이다. 또한, 한 CPU 코어에서 다른 CPU 코어로 스레드 정보를 이동하는 비용이 줄어들 것이다.

- 0: CPU 연계 기능을 사용하지 않는다.
- 1: CPU 연계 기능을 사용한다.

하지만 HP-UX IA64 플랫폼에서는 CPU 연계 기능을 사용하는 경우에 로깅 병목이 발생할 수 있다. 따라서 HP-UX IA64 플랫폼에서는 이 프로퍼티의 값을 1로 설정(altibase.properties 파일에서)하여도 Altibase 서버는 CPU 연계 기능을 활성화하지 않는다.

**THREAD\_REUSE\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

Altibase 내부에서 사용하는 스레드를 재사용할 것인지 여부를 설정한다.

0: 스레드 재사용 불가

1: 스레드 재사용 가능(기본값)

스레드를 재사용하면, 메모리에 대기중인 스레드를 사용할 수 있어 스레드를 생성하는 비용을 줄일 수 있다.

**TIMED\_STATISTICS****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

Wait 발생시 대기 시간과 SQL 연산의 소요 시간을 측정할 것인지 여부를 설정한다. 이 프로퍼티를 측정하는 것으로 설정할 경우 성능이 떨어질 수 있다.

0: 측정하지 않음

1: 측정함

**TIMER\_RUNNING\_LEVEL****데이터 타입**

Unsigned Integer

**기본값**

플랫폼 별로 기본값이 아래와 같이 상이하다.

1 : 하위에 기술되지 않은 모든 플랫폼

2 : IBM-AIX

3 : x86-linux, Amd64-linux

**속성**

읽기 전용, 단일 값



**값의 범위**

[1, 3]

**설명**

Wait event를 대기하는 시간과 SQL 연산에 소요되는 시간을 측정하는 방법을 설정한다.

1: 시간을 측정하는 스레드가 TIMER\_THREAD\_RESOLUTION에서 지정한 규칙적인 간격으로 시간을 측정한다.

2: 플랫폼 별로 지원하는 라이브러리 함수를 이용하여 측정한다.

3: 1번과 비슷하지만 시스템 시계(clock)를 이용하여 측정한다. 따라서 다른 방법에 비해 성능 저하가 낮다.

LINUX, PA-RISC-HP-64 이외의 장비에서 이 값을 3으로 설정할 경우, 시간을 제대로 측정하지 못하여 Altibase 서버가 기동이 안 될 수도 있다. 이 경우 Altibase는 altibase\_boot.log에 경고 메시지를 남기고, 자동으로 이 프로퍼티의 값을 기본값으로 변경한 후 기동한다. 기본값이 1인 장비에서는 아래와 같은 메시지가 altibase\_boot.log에 기록될 것이다.

[Warning] Because a TIMER\_RUNNING\_LEVEL of 3 is not supported on this platform, it has been set to the default(=1) for this platform.

**TIMER\_THREAD\_RESOLUTION (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

1000

**속성**

변경 가능, 단일 값

**값의 범위**

[50, 10000000]

**설명**

TIMER\_RUNNING\_LEVEL을 1로 설정한 경우 이 프로퍼티는 시간을 측정하는 주기를 의미한다.

**TOP\_RESULT\_CACHE\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 3]

**설명**

Top Result Cache를 사용하여 최종 결과의 실행계획을 저장할 지 여부를 설정한다.

0: Disabled (기본값) - Top Result Cache기능을 사용하지 않는다.

1: MEMORY - 메모리 테이블만 사용한 쿼리의 최종결과를 캐시한다

2: DISK - 디스크 테이블만 사용한 쿼리의 최종결과를 캐시한다

3: ALL - 테이블 타입과 상관없이 최종 결과를 캐시한다.

Altibase 운영 중 ALTER SYSTEM 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**TOTAL\_WA\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

128MB

**속성**

변경 가능, 단일 값

**값의 범위**[0,  $2^{64}-1$ ]**설명**

정렬 또는 해싱 작업을 위해 할당할 수 있는 메모리의 최대 크기를 지정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**INIT\_TOTAL\_WA\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값** $2^{64}-1$ **속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>64</sup>-1]**설명**

정렬 또는 해싱 작업을 위해 미리 할당 할 메모리의 크기를 지정한다.

TOTAL\_WA\_SIZE 보다 더 클 경우 TOTAL\_WA\_SIZE 까지만 생성한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**TOUCH\_TIME\_INTERVAL (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

3

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

버퍼의 접근 횟수를 증가시키기 위한 최소 시간 간격을 의미한다. 버퍼에 대해 마지막으로 접근한 이후 이 프로퍼티에서 명시한 시간이 지나면, 접근 횟수가 증가한다.

이 값이 기본값 3으로 명시되면, 특정 버퍼의 마지막 접근 이후 3초 이내의 접근에 대해서는 접근 횟수를 갱신하지 않는다.

**TRANSACTION\_SEGMENT\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

256

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 16384]

**설명**

언두 테이블스페이스에서 관리하는 트랜잭션 세그먼트의 개수를 나타낸다. 트랜잭션 세그먼트에는 TSS 세그먼트와 언두 세그먼트가 있으며 기본값 256은 TSS 세그먼트 256개, 언두 세그먼트 256개를 의미한다. 트랜잭션 세그먼트는 Altibase 서버 구동 시 생성된다. 트랜잭션 수행 중 트랜잭션 세그먼트가 부족한 경우 altibase\_boot.log 에 "TRANSACTION\_SEGMENT\_COUNT is full" 로그를 남기고 사용 가능한 트랜잭션 세그먼트를 할당받을 때까지 트랜잭션은 대기한다.

Altibase 분산 데이터베이스 시스템에서 이 프로퍼티 값을 이전 값보다 작게 설정하기를 원하면 분산 트랜잭션을 모두 종료한 후 Altibase 서버를 중지해야 한다. 정리되지 않은 분산 트랜잭션이 있을 때 현재 값보다 작게 설정하고 Altibase 서버를 재시작한 경우 기존 언두 테이블스페이스의 공간에 접근할 수 없어 Altibase 서버 구동이 실패한다.

**TRX\_UPDATE\_MAX\_LOGSIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

10 MB

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2<sup>64</sup>-1]

**설명**

하나의 DML에 의해 생성되는 로그의 누적 크기가 이 프로퍼티에 설정된 값보다 커지면 해당 트랜잭션을 중단하고 오류를 반환한다.

사용자의 부주의로 대용량 배치 작업이 실행되어 시스템에 부하가 발생하는 것을 방지하기 위해 사용하는 프로퍼티이다.

해당 프로퍼티의 값을 0으로 설정하면 로그 크기에 대한 제한이 없어지기 때문에, 트랜잭션을 갱신할 때 발생하는 로그를 무한대로 사용할 수 있다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**SERIAL\_EXECUTE\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 프로퍼티는 SCAN PLAN내 FILTER를 최적화하여 FILTER 수행 성능을 향상시킨다. 이때 단순한 FILTER에 한하여 최적화하며, 최적화 여부는 PLAN TREE에서 확인할 수 있다.

0: 최적화하지 않음

1: 최적화함

**세션 관련 프로퍼티**

Altibase는 클라이언트-서버 구조로 사용 가능하며 세션 연결 프로퍼티는 클라이언트와 서버의 통신에 관한 프로퍼티를 규정하는 것이다. 다음과 같은 프로퍼티들이 있다.

**CM\_DISCONN\_DETECT\_TIME (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

3

**속성**

읽기 전용, 단일 값

**값의 범위**

[1,  $2^{32}-1$ ]

**설명**

세션 관리 쓰레드의 동작 주기를 설정하기 위한 프로퍼티이다. Altibase 서버에는 클라이언트와 서버의 연결이 단절되었는지 검사하기 위해 세션 관리 쓰레드가 존재한다.

일반적으로 클라이언트 프로세스가 비정상 종료하면 그 클라이언트와 연결된 세션은 곧바로 그 상태를 감지할 수 있다.

그러나 세션에서 수행 중인 작업이 세션 작업과는 무관한 Altibase 서버 내부의 작업이면서 오랜 시간을 요구하는 작업이라면, 해당 세션은 클라이언트 비정상 종료 여부를 확인할 수 없다. 즉 클라이언트와 연결이 종료되었는지를 해당 세션에서는 확인할 수 없기 때문에 클라이언트가 비정상 종료되었음에도 불구하고 Altibase 서버는 그 작업을 계속 진행하게 된다.

위와 같은 경우 그러한 세션을 감지하여 해당 트랜잭션들을 롤백시킬 필요가 있으며 이를 위해 세션 관리 쓰레드가 주기적으로 세션들의 상태를 검사하게 된다.

## CONCURRENT\_EXEC\_DEGREE\_DEFAULT

### 데이터 타입

Unsigned Integer

### 기본값

4

### 속성

변경 가능, 단일 값

### 값의 범위

[2, 1024]

### 설명

각 세션에서 DBMS\_CONCURRENT\_EXEC 패키지를 이용하여 동시에 실행시킬 수 있는 프로시저 개수를 설정한다. 만약 사용자가 저장 패키지의 INITIALIZE 함수로 병렬 처리할 프로시저의 개수를 지정하지 않으면, 이 프로퍼티의 값이 적용된다.

그러나 Altibase서버에서 저장 패키지를 이용하여 수행될 수 있는 프로시저의 개수는 CONCURRENT\_EXEC\_DEGREE\_MAX 프로퍼티의 값보다 클 수 없다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## CONCURRENT\_EXEC\_DEGREE\_MAX

### 데이터 타입

Unsigned Integer

### 기본값

논리 코어 수

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 1024]

### 설명

Altibase서버에서 DBMS\_CONCURRENT\_EXEC 패키지를 이용하여 실행시킬 수 있는 병렬 프로시저의 최대 개수를 설정한다.

이 프로퍼티의 값을 0으로 설정한 경우 DBMS\_CONCURRENT\_EXEC 패키지가 동작하지 않는다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 없다.

**CONCURRENT\_EXEC\_WAIT\_INTERVAL****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

변경 가능, 단일 값

**값의 범위**

[10, 1000000]

**설명**

DBMS\_CONCURRENT\_EXEC 패키지에서 요청한 프로시저가 정상적으로 동작하는지 여부를 검사하는 간격을 설정한다.

이 프로퍼티의 값은 DBMS\_CONCURRENT\_EXEC 패키지에서 REQUEST 함수와 WAIT\_REQ 함수에 영향을 준다. 이 함수들은 패키지를 이용하여 요청된 프로시저의 작업이 완료될 때까지 기다리기 때문이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DEFAULT\_THREAD\_STACK\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

10485760 (10MB)

**속성**

읽기 전용, 단일 값

**값의 범위**

[1048576, 134217728]

**설명**

모든 쓰레드의 스택 사이즈를 지정한다.

**IPC\_CHANNEL\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 65535]

**설명**

클라이언트와 서버 사이에 IPC 통신을 하기 위하여 반드시 설정되어야 하는 프로퍼티이다. IPC를 이용한 클라이언트와 서버 통신 채널의 최대 개수를 지정한다. 각 채널 수에 비례해서 공유 메모리와 세마포어를 할당받기 때문에 서버에 동시에 연결할 수 있는 최대 IPC 연결 개수를 설정하는 것은 중요하다.

**IPC\_FILEPATH****데이터 타입**

String

**기본값**

\$ALTIBASE\_HOME/trc/cm-ipc

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

Altibase 서버가 유닉스 환경에서 IPC 방식으로 클라이언트와 연결하기 위해 생성하는 소켓 파일이다.

서버를 구동하면 \$ALTIBASE\_HOME/trc 디렉토리 아래에 cm-ipc로 소켓 파일이 생성되며, 이 파일은 삭제되지 않도록 주의해야 한다.

**IPC\_SEM\_KEY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값



**값의 범위**

[0, 4294967294]

**설명**

IPC 채널을 생성하는 데 필요한 세마포어 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다.

기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 세마포어 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPC\_SEM\_KEY 값을 기준으로 IPC\_SEM\_KEY부터 IPC\_SEM\_KEY + (IPC\_CHANNEL\_COUNT + 1)만큼의 연속된 세마포어 키를 사용하여 IPC 채널을 생성한다. +1은 SYS 사용자가 관리자 모드(sysdba)로 접속하기 위해 예약된 IPC 채널이다. 예를 들어 IPC\_SEM\_KEY 값이 10000이고 IPC\_CHANNEL\_COUNT 값이 1000이면 세마포어 키로 10000부터 11000까지 사용한다.

IPC 채널은 Altibase 서버 구동 시 생성되는데, 세마포어 키가 사용 중이거나 다른 이유로 세마포어를 생성하지 못하면 Altibase 서버 구동은 실패한다. 이 경우 Altibase 서버 트레이스 로그 altibase\_boot.log에서 시스템 에러(errno)를 확인하고 그에 따른 적절한 처리를 해야 한다.

**IPC\_SHM\_KEY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 4294967294]

**설명**

IPC 채널을 생성하는 데 필요한 공유 메모리 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다.

기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 공유 메모리 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPC\_SHM\_KEY 값을 공유 메모리 키로 사용한다.

IPC 채널은 Altibase 서버 구동 시 생성되는데, 공유 메모리 키가 사용 중이거나 다른 이유로 공유 메모리를 생성하지 못하면 Altibase 서버 구동은 실패한다. 이 경우 Altibase 서버 트레이스 로그 altibase\_boot.log에서 시스템 에러(errno)를 확인하고 그에 따른 적절한 처리를 해야 한다.

**IPCD\_CHANNEL\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 65535]

**설명**

클라이언트와 서버 사이에 IPCDA 통신을 하기 위하여 반드시 설정되어야 하는 프로퍼티이다.

IPCDA를 이용한 클라이언트와 서버간 통신 채널의 최대 개수를 지정한다. 채널 개수에 비례하여 공유 메모리와 세마 포어를 할당받기 때문에 서버에 동시에 연결할 수 있는 최대 IPCDA 연결 개수를 설정하는 것이 중요하다. IPCDA의 채널 개수는 CPU 코어 개수의 1/2 값이 최적화된 값이다.

**IPCDA\_DATABLOCK\_SIZE (단위: 킬로바이트)****데이터 타입**

Unsigned Integer

**기본값**

20480

**속성**

읽기 전용, 단일 값

**값의 범위**

[32, 102400]

**설명**

IPCDA를 이용하는 통신 채널 한 개의 공유 메모리의 크기를 설정하는 프로퍼티이다. 이 값을 1000으로 하고, IPCDA\_CHANNEL\_COUNT를 24로 하는 경우 서버에서 통신 채널로 사용되는 전체 메모리의 크기 (IPCDA\_CHANNEL\_COUNT \* IPCDA\_DATABLOCK\_SIZE)는 1000KB \* 24 = 24000KB가 된다.

이 값은 시스템 메모리의 크기에 따라 공유 메모리를 사용하는 다른 프로그램의 운영에 장애가 되지 않도록 적절한 값을 설정해야 한다. 예를 들어 사용자의 시스템 메모리가 4GB인 시스템에서는 IPCDA\_CHANNEL\_COUNT를 24로 할 경우, IPCDA\_DATABLOCK\_SIZE는 최댓값으로 하여도, 실제 사용 메모리는 2457600KB를 사용하기 때문에 최댓값으로 설정할 수 있다.

**IPCDA\_FILEPATH****데이터 타입**

String

**기본값**`$ALTIBASE_HOME/trc/cm-ipcda`**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

Altibase 서버가 유닉스 환경에서 IPCDA 방식으로 클라이언트와 연결하기 위해 생성하는 소켓 파일이다.

서버를 구동하면 `$ALTIBASE_HOME/trc` 디렉토리 아래에 `cm-ipcda`로 소켓 파일이 생성되며, 이 파일은 삭제되지 않도록 주의해야 한다.

**IPCDA\_SEM\_KEY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 4294967294]

**설명**

IPCDA 채널을 생성하는 데 필요한 세마포어 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다.

기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 세마포어 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPCDA\_SEM\_KEY 값을 기준으로 IPCDA\_SEM\_KEY부터 IPCDA\_SEM\_KEY + IPC\_CHANNEL\_COUNT만큼의 연속된 세마포어 키를 사용하여 IPCDA 채널을 생성한다. 예를 들어 IPCDA\_SEM\_KEY 값이 10000이고 IPC\_CHANNEL\_COUNT 값이 1000이면 세마포어 키로 10000부터 10999까지 사용한다.

IPCDA 채널은 Altibase 서버 구동 시 생성되는데, 세마포어 키가 사용 중이거나 다른 이유로 세마포어를 생성하지 못 하면 Altibase 서버 구동은 실패한다. 이 경우 Altibase 서버 트레이스 로그 `altibase_boot.log`에서 시스템 에러(errno)를 확인하고 그에 따른 적절한 처리를 해야 한다.

**IPCDA\_SHM\_KEY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 4294967294]

**설명**

IPCD A 채널을 생성하는 데 필요한 공유 메모리 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다.

기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 공유 메모리 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPCDA\_SHM\_KEY 값을 기준으로 연속된 키 2개를 공유 메모리 키로 사용한다. 예를 들어 IPCDA\_SHM\_KEY=10000이면 10000, 10001을 공유 메모리 키 값으로 사용한다.

IPCD A 채널은 Altibase 서버 구동 시 생성되는데, 공유 메모리 키가 사용 중이거나 다른 이유로 공유 메모리를 생성하지 못하면 Altibase 서버 구동은 실패한다. 이 경우 Altibase 서버 트레이스 로그 altibase\_boot.log에서 시스템 에러(errno)를 확인하고 그에 따른 적절한 처리를 해야 한다.

**MAX\_LISTEN****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 16384]

**설명**

클라이언트와 Altibase 간의 통신 시 TCP/IP 또는 UNIX DOMAIN 소켓을 사용하는 경우 대기 큐(listen queue)의 크기를 지정하는 값이다.

**MAX\_STATEMENTS\_PER\_SESSION****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 65535]

**설명**

한 세션에서 실행 가능한 statement의 최대 개수를 지정한다.

Altibase 운영 중 ALTER SYSTEM 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**NET\_CONN\_IP\_STACK****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1, 2]

**설명**

클라이언트 서버간의 TCP/IP통신을 위해서 서버 측에 소켓을 생성할 때 사용하는 인터넷 프로토콜 스택을 지정한다.

0: IPv4만을 지원하는 인터넷 프로토콜 스택을 사용하게 된다.

1: 듀얼 스택 (IPv4와 IPv6 모두 지원하는 인터넷 프로토콜 스택)이 사용된다.

2: IPv6만을 지원하는 인터넷 프로토콜 스택을 사용하게 된다.

**NLS\_COMP****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

데이터베이스 생성시 지정된 캐릭터셋(character set)의 캐릭터들의 순서는 해당 국가에서 사용되는 사전과 그 이전 값의 순서가 동일하다고 보장할 수 없다.

이 프로퍼티의 값이 1로 설정되면, 각 캐릭터셋을 해당 언어의 사전 순서대로 비교한다. 현재 한국어에 대해서만 지원하기 때문에, 데이터베이스 캐릭터셋이 한글(KSC5601 완성형 또는 MS확장 완성형)로 설정됐을 때에만 지원한다.

**NLS\_CURRENCY****데이터 타입**

String

**기본값**

NLS\_TERRITORY의 값을 따른다.

**속성**

변경 가능, 단일 값

**값의 범위**

최대 10바이트 길이의 문자

**설명**

지역 통화 기호를 지정하는 프로퍼티이다. 숫자형 데이터 형식 L을 통해 지역 통화 기호를 표시할 때 이 프로퍼티의 값이 사용된다. 이 프로퍼티의 값은 +, -, <, > 기호로 시작할 수 없다.

Altibase 운영 중 아래와 같이 ALTER SESSION문을 이용하여 이 프로퍼티의 값을 변경할 수 있다:

```
ALTER SESSION SET NLS_CURRENCY = '$';
```

주의: 아랍권의 통화 기호가 정확히 표시되려면 클라이언트 응용 프로그램(또는 셸이나 에디터)이 아랍 문자의 표시를 지원해야 한다.

**NLS\_ISO\_CURRENCY****데이터 타입**

String

## 기본값

NLS\_TERRITORY의 값을 따른다.

## 속성

변경 가능, 단일 값

## 값의 범위

V\$NLS\_TERRITORY 성능 뷰에 존재하는 값

## 설명

ISO 통화 기호를 지정하는 프로퍼티이다. 숫자형 데이터 형식 C를 통해 국제 통화 기호를 표시할 때 이 프로퍼티의 값이 사용된다. 이 프로퍼티에 지정할 수 있는 값은 V\$NLS\_TERRITORY 성능 뷰에 존재하는 값(지역)에 한정된다.

Altibase 운영 중 아래와 같이 ALTER SESSION문을 이용하여 이 프로퍼티의 값을 변경할 수 있다:

```
ALTER SESSION SET NLS_ISO_CURRENCY = America;
```

## NLS\_NCHAR\_CONV\_EXCP

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

NCHAR 데이터 타입의 데이터를 다른 캐릭터셋으로 변환 시, 데이터 손실이 발생할 수 있다. 이 때 에러를 발생시킬 것인지 아니면 데이터 손실이 발생한 채로 변환을 할 것인지를 결정하는 프로퍼티이다.

이 프로퍼티는 서버에서 NCHAR타입 데이터를 다른 캐릭터셋으로 변환할 때에만 에러를 발생시킨다.

Altibase 운영 중 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: FALSE (에러를 발생시키지 않는다.)

1: TRUE

## NLS\_NCHAR\_LITERAL\_REPLACE

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

기본적으로 클라이언트는 쿼리 문 전체를 데이터베이스 문자 셋으로 변환하여 전송한다. 그러나, 특정 리터럴에 대해 이런 동작을 막으려면, 이 프로퍼티의 값을 1로 설정하고 그 리터럴 앞에 "N" 문자를 덧붙이면 된다. 즉, NCHAR 리터럴로 만드는 것이다.

이 프로퍼티의 값이 1일 때, 클라이언트는 쿼리 문 내의 모든 리터럴 앞에 "N" 문자가 있는지 찾는다. 만약 찾게 되면 클라이언트는 그 리터럴을 데이터베이스 문자 셋으로 변환하지 않고 그대로 전송하며 서버가 직접 내셔널 문자 셋으로 변환한다. 이것은 데이터베이스 문자 셋과는 다른 인코딩이 필요한 NCHAR 타입 데이터를 사용하고자 할 때 유용하다.

Altibase 운영 중 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: "N" 문자가 있는지 검사하지 않고 쿼리 문 전체를 데이터베이스 문자 셋으로 변환한다.

1: "N" 문자가 붙어있는 NCHAR 리터럴은 데이터베이스 문자 셋으로 변환하지 않는다.

주의: 이 값을 1로 설정하는 것은 클라이언트의 비용이 크게 발생하므로, 사용시 주의가 필요하다.

## NLS\_NUMERIC\_CHARACTERS

### 데이터 타입

String

### 기본값

NLS\_TERRITORY의 값을 따른다.

### 속성

변경 가능, 단일 값

### 값의 범위

없음



## 설명

소수점 문자와 그룹 구분자를 지정하는 프로퍼티이다. 그룹 구분자는 숫자에서 천단위로 구분해서 표기할 때 사용되며 일반적으로 쉼표(,)가 사용된다. 소수점 문자는 일반적으로 마침표(.)가 사용된다. 이 프로퍼티는 +, -, <, > 기호로 시작할 수 없다.

이 프로퍼티에 지정한 문자열에서 처음 두 개의 문자만 각각 소수점 문자와 그룹 구분자로 설정되고 나머지 문자는 무시된다.

Altibase 운영 중 아래와 같이 ALTER SESSION문을 이용하여 이 프로퍼티의 값을 변경할 수 있다:

```
ALTER SESSION SET NLS_NUMERIC_CHARACTERS='.,';
```

## NLS\_TERRITORY

### 데이터 타입

String

### 기본값

KOREA

### 속성

변경 가능, 단일 값

### 값의 범위

V\$NLS\_TERRITORY 성능 뷰에 존재하는 값

## 설명

지역의 이름을 지정하는 프로퍼티이다. 이 프로퍼티에 지정한 지역에 따라 NLS\_NUMERIC\_CHARACTER, NLS\_CURRENCY, 및 NLS\_ISO\_CURRENCY 프로퍼티의 값이 자동으로 변경된다. V\$NLS\_TERRITORY 성능 뷰를 조회해서 이 프로퍼티에 지정 가능한 지역의 이름을 확인할 수 있다.

Altibase 운영 중 아래와 같이 ALTER SESSION문을 이용하여 이 프로퍼티의 값을 변경할 수 있다. 지역의 이름은 대소문자를 구분하지 않으며, 작은 따옴표(')를 사용할 수 있다.

```
ALTER SESSION SET NLS_TERRITORY = America;
or
ALTER SESSION SET NLS_TERRITORY = 'America';
```

## PORT\_NO

### 데이터 타입

Unsigned Integer

### 기본값

20300

**속성**

읽기 전용, 단일 값

**값의 범위**

[1024, 65535]

**설명**

TCP/IP로 클라이언트와 서버가 통신할 때 사용하는 포트 번호이다. 사용자는 루트 영역(일반적으로 1023번까지)을 제외한 나머지 영역(최대 65535)에 대해 다른 프로그램에서 사용하지 않는 번호면 임의로 지정할 수 있다. Altibase 응용 프로그램은 이 포트 번호를 사용하여 서버와 연결할 수 있다.

**PSM\_CURSOR\_OPEN\_LIMIT****데이터 타입**

Unsigned Integer

**기본값**

32

**속성**

읽기 전용, 단일 값

**값의 범위**

[1,1024]

**설명**

세션에서 DBMS\_SQL 패키지를 사용하여 열 수 있는 커서의 개수를 지정한다.

**PSM\_FILE\_OPEN\_LIMIT****데이터 타입**

Unsigned Integer

**기본값**

16

**속성**

변경 가능, 단일 값

**값의 범위**

[0,128]

**설명**

세션당 최대로 열 수 있는 저장 프로시저 파일 핸들의 개수를 지정한다.

**TIME\_ZONE****데이터 타입**

String

**기본값**

OS\_TZ

**속성**

변경 가능, 단일 값

**값의 범위**

V\$TIME\_ZONE\_NAMES 성능 뷰에 존재하는 값

**설명**

타임 존을 설정하는 프로퍼티이다. 해당되는 지역 이름이나 약어 또는 UTC 오프셋(예. +09:00)과 같은 문자열로 지정할 수 있다.

Altibase 운영 중 ALTER SESSION 문을 이용하여 해당 세션의 타임 존을 변경할 수 있다.

**UNIXDOMAIN\_FILEPATH****데이터 타입**

String

**기본값**

\$ALTIBASE\_HOME/trc/cm-unix

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

Altibase 서버가 유닉스 도메인 방식으로 클라이언트와 연결하기 위해 생성하는 소켓 파일이다.

서버를 구동하면 \$ALTIBASE\_HOME/trc 디렉토리 아래에 cm-unix로 기본 소켓 파일이 생성되며, 이 파일은 삭제되지 않도록 주의해야 한다.

**USE\_MEMORY\_POOL****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,1]

**설명**

메모리 풀링 기능의 사용 유무를 지정한다. 메모리 풀링이란 서버에서 메모리를 미리 할당하여 사용하는 기능이다.

이 기능을 사용하면, 메모리를 미리 할당하기 때문에 메모리 사용량이 많아진다.

0: 사용하지 않음

1: 사용

**USER\_LOCK\_POOL\_INIT\_SIZE (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위**

[128, 10000]

**설명**

데이터베이스에서 사용자 잠금(User Lock)을 사용할 수 있는 개수를 설정할 수 있다.

이 프로퍼티의 값을 초과하여 사용자 잠금의 개수를 사용할 수 있으나, 성능 저하가 발생할 수 있다. 또한 한 번 생성된 사용자 잠금을 해제하여도 제거되는 것은 아니므로, 해제된 사용자 잠금을 재사용하는 것이 효과적이다.

**USER\_LOCK\_REQUEST\_CHECK\_INTERVAL (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

10000

**속성**

변경 가능, 단일 값

**값의 범위**

[10, 999999]

**설명**

다른 세션이 사용자 잠금(User Lock)을 획득할 수 있는지 검사하는 주기를 설정한다. 다른 세션에서 현재 사용중인 사용자 잠금을 요청하면, 사용자 잠금이 해제되기 전까지 세션이 대기한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**USER\_LOCK\_REQUEST\_LIMIT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 10000]

**설명**

하나의 세션이 요청할 수 있는 사용자 잠금(User Lock)의 개수를 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**USER\_LOCK\_REQUEST\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

세션에서 사용자 잠금(User Lock)을 요청한 후 획득할 때까지 대기하는 최대 시간을 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**XA\_HEURISTIC\_COMPLETE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 2]

**설명**

PREPARE 상태(IN\_DOUBT 상태)인 전역 트랜잭션을 일정 시간이 지나면 임의로 종료시킬 수 있는 프로퍼티이다. 이 때 COMMIT으로 종료시킬 것인지, ROLLBACK으로 종료시킬 것인지를 선택할 수 있으며, 트랜잭션을 종료시킬 때 명령을 기다리는 시간은 XA\_INDOUBT\_TX\_TIMEOUT 프로퍼티에 정의된 값을 사용한다.

0: 종료하지 않음.

1: 트랜잭션 커밋

2: 트랜잭션 롤백

분산 트랜잭션 환경에서는 2 단계 커밋 프로토콜(2 Phase Commit Protocol, XA)을 사용하여 트랜잭션을 수행한다. 트랜잭션을 수행중에 전역(global) 트랜잭션 관리자(Coordinator)로부터 PREPARE 명령을 받은 후 COMMIT이나 ROLLBACK 명령을 받지 못한 상태가 오래 지속될 경우, 데이터베이스 성능에 영향을 줄 수 있다.

## 타임아웃 관련 프로퍼티

### BLOCK\_ALL\_TX\_TIME\_OUT (단위: 초)

데이터 타입

Unsigned Integer

기본값

3

속성

변경 가능, 단일 값

값의 범위

$[0, 2^{32}-1]$

설명

버퍼 매니저가 해쉬 테이블의 크기를 조정할 때 트랜잭션의 접근을 제한할 수 있다. 이 프로퍼티의 최솟값 0은 대기하지 않고 즉시 오류 처리됨을 의미한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

### DDL\_LOCK\_TIMEOUT (단위: 초)

데이터 타입

Short integer

기본값

0

속성

변경 가능, 단일 값

값의 범위

$[-1, 65535]$

설명

DDL 문을 수행할 때 해당 테이블에 이미 다른 트랜잭션에 의해 잠금(Lock)이 획득되어 있는 경우 잠금을 대기하는 옵션을 설정하는 것이다. 잠금을 요구하여 곧바로 획득되지 않을 경우 이 프로퍼티의 값이 -1로 설정되어 있으면 무한정 대기하고 양수로 설정되어 있으면 지정된 값 만큼 대기하고 다시 시도한다.

기본값은 0으로, DDL 수행시 잠금을 요구한 시점에서 잠금을 획득할 수 없는 경우 해당 DDL은 즉시 오류 처리된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## DDL\_TIMEOUT (단위: 초)

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

DDL 문의 실행 시간이 이 프로퍼티에 설정한 시간(초)을 초과하면, 그 구문의 실행은 취소된다. 기본 설정 값은 0으로, Altibase는 DDL 구문 수행이 끝날 때까지 무한 대기한다.

이 프로퍼티의 값은 Altibase 실행 중에 ALTER SYSTEM 또는 ALTER SESSION 구문으로 변경 가능하다.

Note: Altibase 버전 5.5.1까지는, DDL구문의 실행 시간도 UTRANS\_TIMEOUT 과 QUERY\_TIMEOUT 프로퍼티의 영향을 받았다. DML과 DCL구문의 실행시간은 여전히 UTRANS\_TIMEOUT 과 QUERY\_TIMEOUT 프로퍼티의 영향을 받는다.

## FETCH\_TIMEOUT (단위: 초)

### 데이터 타입

Unsigned Integer

### 기본값

60

### 속성

변경 가능, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

응용 프로그램에서 SELECT 문을 수행하는 시간이 길어짐에 따라 데이터베이스 메모리가 비정상적으로 증가하는 것을 막기 위하여 이 값을 설정한다. 질의 수행시간이 프로퍼티 파일에 설정된 값보다 커지면 세션 연결을 해제하고 현재 트랜잭션을 철회한다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.



**IDLE\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

서버에 접속된 클라이언트가 비정상적으로 오랜 시간 연결을 맺고 있고, 만약 이러한 클라이언트의 수가 점차적으로 많아진다면 결국에는 서비스를 할 수 있는 연결 개수가 현저히 작아져, 나중에는 서비스가 불가능한 상황이 될 수 있다.

이러한 현상을 미리 방지하기 위해 이 값을 설정한다. 한 세션의 유효 시간이 프로퍼티 파일에 설정된 값보다 커지면 세션 연결을 해제하고 현재 트랜잭션을 철회한다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**LOGIN\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

Altibase의 포트로 접속이 이루어진 후 인증 절차가 완료될 때까지 허용된 시간이다. 이 시간 안에 인증이 이루어지지 않으면 서버는 접속을 끊는다.

**MULTIPLEXING\_POLL\_TIMEOUT (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

10000

**속성**

변경 가능, 단일 값

**값의 범위**

[1000, 1000000]

**설명**

멀티플렉싱을 하는 서비스 스레드가 세션을 감지하는 주기를 나타낸다.

**QUERY\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

600

**속성**

변경 가능, 단일 값

**값의 범위**[0,  $2^{32}-1$ ]**설명**

특정 질의들(정렬 혹은 긴 조인 등)의 수행 시간이 길어짐에 따라 데이터베이스 메모리가 비정상적으로 증가하는 것을 막기 위하여 이 값을 설정한다. 질의 수행 시간이 프로퍼티 파일에 설정된 값보다 커지면 현재 트랜잭션 연산을 부분 철회한다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**SHUTDOWN\_IMMEDIATE\_TIMEOUT****데이터 타입**

Unsigned Integer

**기본값**

60

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

IMMEDIATE 옵션으로 Altibase 서버를 종료할 때, 끝나지 않은 트랜잭션들은 롤백 처리된다. 이 프로퍼티는 트랜잭션을 롤백하기 위해 대기하는 시간을 지정한다. 이 시간을 초과하게 되면, 종료되지 않은 트랜잭션이 롤백되지 않은 채로 서버는 강제 종료된다. 이 값이 0이면, 모든 트랜잭션이 롤백될 때까지 대기한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**UTRANS\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

3600

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

변경 연산(UPDATE, INSERT, DELETE)을 수행하는 트랜잭션의 수행 시간이 길어짐에 따라 로그 파일의 개수가 비정상적으로 증가하는 것을 막기 위하여 이 값을 설정한다. 수행 시간이 프로퍼티 파일에 설정된 값보다 커지면 세션 연결을 해제하고 현재 트랜잭션을 철회한다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**XA\_INDOUBT\_TX\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

60

**속성**

읽기 전용, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

2 단계 커밋 프로토콜에서 IN-DOUBT 상태의 오래 실행되는 전역 트랜잭션을 임의로 종료시키는 시간 기준에 대한 프로퍼티이다.

**트랜잭션 관련 프로퍼티****AUTO\_COMMIT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

세션에서 SQL 문을 수행할 때 하나의 SQL 문을 하나의 트랜잭션으로 처리하여 커밋할 것인지를 결정하는 프로퍼티이다. 이 값이 1 이면 auto-commit 모드이고 0 이면 non-autocommit 모드이다. non-autocommit 모드인 경우 응용 프로그램에서 트랜잭션의 시작과 끝을 명시적으로 알려주어야 한다.

세션 단위로 이를 설정할 수 있는데, 서버 구동 시 AUTO\_COMMIT의 값이 1일지라도 세션 별로 이 모드를 변경할 수 있다. 예를 들어, 클라이언트에서 ALTER SESSION SET AUTOCOMMIT = FALSE (non-autocommit)를 실행하면 이 세션은 이 후에 발생하는 트랜잭션을 반영할 것인지 또는 롤백할 것인지를 사용자가 명시해 주어야 한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**ISOLATION\_LEVEL****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 2]

**설명**

트랜잭션의 고립화 수준(isolation level)을 지정한다. 고립화 수준에 따라 하나의 트랜잭션이 여러 번 같은 테이블에 대한 검색을 수행할 때의 결과가 다르다.

트랜잭션의 고립화 수준에 대한 자세한 설명은 *Administrator's Manual*을 참고하기 바란다.

고립화 수준	특징
0 (Committed Read)	Altibase의 기본 모드이다. 검색 트랜잭션이 읽은 데이터는 완료된 다른 트랜잭션에 의해 변경된 데이터임을 보장한다. 검색 트랜잭션이 한번 읽고 다시 읽을 때 다른 트랜잭션이 동시에 INSERT 혹은 DELETE를 수행하고 커밋했다면 그것이 반영되어 새로운 행(row)이 보이거나 혹은 보였던 행이 보이지 않게 될 수 있다.
1 (Repeatable Read)	한 트랜잭션 수행 동안 동일 레코드를 여러 번 반복해서 읽는 경우 항상 동일한 레코드의 내용을 검색하게 됨을 보장한다. 한번 읽을 때 읽은 행(row)에 잠금(lock)이 걸린다. 그래서 다시 읽을 때 이전에 보였던 행이 안 보이는 경우는 없으나, 그 사이에 새로 삽입된 행은 보일 수 있다.
2 (No Phantom)	여러 번 반복해서 읽어도 모두 동일한 결과가 보이는 것을 보장한다.

**TRANSACTION\_TABLE\_SIZE (단위: 트랜잭션 개수)****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[16,16384]

**설명**

Altibase 서비스 중에 동시에 생성될 수 있는 트랜잭션 개수를 데이터베이스를 생성할 때 설정할 수 있으며, 이에 대해 메모리가 미리 할당된다.

이 프로퍼티는 2n 크기만큼 값을 증감시켜야 하며, 감소시킬 때에는 반드시 데이터베이스를 다시 생성해야 한다.

## 백업 및 복구 관련 프로퍼티

데이터베이스가 변경될 때 변경 로그를 유지하는데, 이에 관한 처리를 어떻게 할 것인지를 정하는 프로퍼티들이 있다.

### ARCHIVE\_DIR

#### 데이터 타입

String

#### 기본값

\$ALTIBASE\_HOME/arch\_logs

#### 속성

읽기 전용, 다중 값

#### 값의 범위

없음

#### 설명

데이터베이스를 아카이브 로그(ArchiveLog) 모드로 운영하는 경우 아카이브 로그파일들이 백업될 디렉토리를 설정하는 프로퍼티이다. 사용자가 명시적으로 이 값을 지정하지 않으면 기본으로 Altibase가 설치된 디렉토리 밑의 arch\_logs 디렉토리에 아카이브 로그파일들이 백업된다. 이 프로퍼티의 개수는 LOG\_DIR 프로퍼티 개수와 정확히 일치해야 한다.

사용자가 이 값을 명시적으로 지정할 수 있으나 지정된 디렉토리는 미리 생성되어 있어야 하며, 그렇지 않은 경우 오류 메시지를 출력함과 동시에 Altibase 서버가 구동되지 않는다.

### ARCHIVE\_FULL\_ACTION

#### 데이터 타입

Unsigned Integer

#### 기본값

0

#### 속성

변경 가능, 단일 값

#### 값의 범위

[0, 2]

#### 설명

ARCHIVE\_DIR에 설정된 디렉토리가 속한 파일 시스템에 충분한 디스크 공간이 없는 경우 아카이브 로그 백업(archive log backup)을 수행하는 아카이브로그 쓰레드의 동작을 제어하는 프로퍼티이다.

디스크 공간 부족으로 로그 파일 백업이 실패하는 경우 설정값에 따른 아카이브로그 쓰레드의 동작 차이는 아래와 같다.

0 : 트레이스 로그(altibase\_sm.log)에 에러 메시지를 출력하고 다음 로그 파일의 백업을 시도한다. 로그 파일마다 차례로 이 과정을 거치므로 디스크 공간 부족이 발생한 시점부터 여유 공간이 생길 때까지 아카이브 로그 파일 경로에 백업하지 못한 로그 파일이 있을 수 있다. 이때 체크포인트가 발생하면 백업이 실패한 로그 파일도 불필요한 로그 파일로 판단하고 삭제하기 때문에 데이터베이스 복구가 불가능할 수 있으므로 0으로 설정하고 운영할 경우 주의가 필요하다.

1 : 충분한 디스크 공간이 확보되어 아카이브 로그파일을 백업할 수 있을 때까지 기다린다. 이 기간 동안은 체크포인트가 발생하더라도 아카이브 로그파일을 백업할 수 없기 때문에 로그 파일들은 삭제되지 않는다.

2 : 디스크 공간 부족을 포함한 백업 실패 상황이 발생하는 경우 트레이스 로그(altibase\_sm.log)에 에러 메시지를 출력하고 다음 로그 파일의 백업을 시도한다. 로그 파일마다 차례로 이 과정을 거치므로 로그 파일 백업 실패 현상이 발생한 시점부터 백업 실패 상황이 해소될 때까지 아카이브 로그 파일 경로에 백업하지 못한 로그 파일이 있을 수 있다. 이때 체크포인트가 발생하면 백업이 실패한 로그 파일도 불필요한 로그 파일로 판단하고 삭제하기 때문에 데이터베이스 복구가 불가능할 수 있으므로 2로 설정하고 운영할 경우 주의가 필요하다.

## ARCHIVE\_MULTIPLEX\_COUNT

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 10]

### 설명

Altibase 서버가 아카이브 로그 파일의 복사본<sup>9</sup>을 몇 개 유지할 것인지를 지정한다. 프로퍼티 설정 이후에 새로이 아카이브 되는 로그 파일부터 복사본이 유지된다.

[<sup>9</sup>] 아카이브 로그 파일 다중화(Multiplexing Archive Logfiles): 아카이브 로그 파일을 별도의 디스크에 복사해 두어, 아카이브 로그 파일 원본의 손상에 대비하는 기능이다.

ARCHIVE\_MULTIPLEX\_COUNT와 ARCHIVE\_MULTIPLEX\_DIR 프로퍼티는 서버가 중지(shutdown)된 상태에서 설정할 수 있다.

## ARCHIVE\_MULTIPLEX\_DIR

### 데이터 타입

String

### 기본값

""

**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

아카이브 로그 파일의 복사본이 위치할 경로를 지정한다. ARCHIVE\_MULTIPLEX\_DIR 프로퍼티는 ARCHIVE\_MULTIPLEX\_COUNT에 명시한 아카이브 로그 파일의 개수만큼 설정해야 한다. 이 때 각각의 경로는 서로 다른 디스크에 둘 것을 권장한다.

**ARCHIVE\_THREAD\_AUTOSTART****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

아카이브 로그파일을 주기적으로 백업하는 스레드인 아카이브로그 스레드를 Altibase 구동시 자동으로 활성화시킬 것 인지를 지정하는 프로퍼티이다. 이 값을 1로 하는 경우 아카이브 스레드가 자동으로 활성화된다.

이 프로퍼티는 디렉토리에 아카이브 로그파일 백업을 위한 충분한 디스크 공간이 없어서 아카이브로그 스레드가 비활 성화되면, 이후에 디스크 공간을 확보하여 아카이브로그 스레드를 다시 활성화하고자 할 때 사용할 수 있다.

**CHECKPOINT\_ENABLED****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값



**값의 범위**

[0, 1]

**설명**

체크포인트를 ON 또는 OFF 시키는 프로퍼티이다.

0: OFF

1: ON

이 값을 0(OFF)으로 지정하면 체크포인트 쓰레드가 동작하지 않으며, CHECKPOINT\_INTERVAL\_IN\_SEC와 CHECKPOINT\_INTERVAL\_IN\_LOG 프로퍼티에서 지정한 주기로 동작할 수 없다. 그러나 사용자가 명시적으로 체크포인트를 수행하면 수행할 수 있다.

**CHECKPOINT\_INTERVAL\_IN\_LOG****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

변경 가능, 단일 값

**값의 범위**[1, 2<sup>32</sup>-1]**설명**

체크포인트 주기를 설정하는 프로퍼티이다. 즉, 이 프로퍼티에 설정된 값만큼 로그 파일이 생성되면, 다음 체크포인트가 발생한다.

그러나 이 프로퍼티 값에 의해 체크포인트 수행이 요구될 때, 이미 체크포인트가 진행 중이거나 기타 다른 이유로 인하여 체크포인트가 수행되지 못하는 경우가 발생할 수 있다. 이 경우 이미 진행 중인 체크포인트가 끝난 후 바로 체크포인트를 수행하는 것이 아니라 현재의 체크포인트 요구는 바로 취소된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**CHECKPOINT\_INTERVAL\_IN\_SEC (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

6000

**속성**

변경 가능, 단일 값

**값의 범위**

[3, 2592000]

**설명**

체크포인트의 주기를 초 단위 시간으로 정하는 것이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**COMMIT\_WRITE\_WAIT\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

트랜잭션을 커밋할 때 로그가 로그 파일에 반영될 때까지 기다릴 것인지 여부를 설정하는 프로퍼티이다. Altibase는 기본으로 성능을 위해 기다리지 않는 값으로 설정된다.

이 프로퍼티는 시스템 전체에 대해서 혹은 사용자의 세션 단위로 설정할 수 있으며, Altibase 운영 중 ALTER SYSTEM 또는 ALTER SESSION 구문으로 변경할 수 있다.

0: Do Not Wait

1: Wait

**INCREMENTAL\_BACKUP\_CHUNK\_SIZE****데이터 타입**

Unsigned Integer

**기본값**

4

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

데이터의 페이지가 변경되는 것을 페이지 단위로 추적하기 위해 설정하는 프로퍼티이다. 예를 들어 이 프로퍼티의 값이 4로 설정되면, 4개의 페이지 단위로 변경 추적 정보를 기록하기 때문에, 한 묶음에 속하는 4개의 페이지 중 하나라도 변경된다면 그 묶음의 모든 페이지가 백업된다.

이 프로퍼티를 변경하려면 아래의 단계를 거쳐서 changeTracking 파일을 다시 생성해야 한다.

1. 서버 종료 후 프로퍼티 파일에서 값 변경
2. 서버 구동
3. 변경 추적 기능 비활성화 후 다시 활성화

**INCREMENTAL\_BACKUP\_INFO\_RETENTION\_PERIOD****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

backupInfo 파일에 저장되는 백업 정보가 유지되는 기간(단위: 일)을 지정하는 프로퍼티이다.

ALTER DATABASE DELETE OBSOLETE BACKUP FILES 구문으로 삭제되는 백업 정보는 유효 기간이 지난 가장 오래된 레벨 0 백업부터 유효 기간이 지나지 않은 가장 오래된 레벨 0 바로 이전의 레벨 1 백업까지이다.

참고: 증분 백업에 대한 복구에는 레벨 0로 백업된 백업 파일이 반드시 필요하다. 따라서 이 프로퍼티에 지정한 유지 기간이 지난 백업 정보와 거기에 대응하는 백업 파일이 있더라도, 레벨 0 백업이 유일하다면, 삭제 명령을 실행하더라도 유지 기간이 지난 백업 정보와 백업 파일이 삭제되지 않는다.

**LOG\_BUFFER\_TYPE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

로그 버퍼 타입을 결정하는 프로퍼티이다.

0: 운영체제 커널의 로그 버퍼를 사용

1: 프로세스 메모리의 로그 버퍼를 사용

이 프로퍼티는 시스템 운영 중에 변경할 수 없다.

**LOG\_MULTIPLEX\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 10]

**설명**

Altibase 서버가 로그 파일의 복사본<sup>10</sup>을 몇 개 유지할 것인지를 지정한다. 원본 로그 파일이 생성되거나 삭제될 때 로그 파일의 복사본도 동시에 생성되거나 삭제된다.

[<sup>10</sup>] 로그 파일 다중화(Multiplexing Log Files): 데이터베이스의 모든 변경 사항이 기록되는 로그 파일을 별도의 디스크에 복사해 두어, 로그 파일 원본의 손상에 대비하는 기능이다.

LOG\_MULTIPLEX\_COUNT와 LOG\_MULTIPLEX\_DIR 프로퍼티는 서버가 중지(shutdown)된 상태에서 설정할 수 있다.

**LOG\_MULTIPLEX\_DIR****데이터 타입**

String

**기본값**

""

**속성**

읽기 전용, 다중 값

**값의 범위**

없음

**설명**

로그 파일의 복사본이 위치할 경로를 지정한다. LOG\_MULTIPLEX\_COUNT에 명시한 개수만큼 LOG\_MULTIPLEX\_DIR 프로퍼티를 설정해야 한다. 또한, 각 경로를 서로 다른 디스크에 둘 것을 권장한다.

**PREPARE\_LOG\_FILE\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

5

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

로그 생성시 해당 로그파일에 충분한 공간이 없으면 새로운 로그 파일을 생성하며, 이 경우 트랜잭션의 응답 시간은 늦어지게 된다. 이처럼 로그파일 생성으로 인해 트랜잭션의 수행이 늦어지는 것을 막기 위해 Altibase는 여분의 로그파일을 미리 생성해 둔다. 이 여분의 로그파일의 개수를 지정하는 것이 이 프로퍼티이다.

**SNAPSHOT\_MEM\_THRESHOLD (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

80

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

스냅샷 설정(BEGIN SNAPSHOT) 이후 메모리 데이터베이스에서 사용할 수 있는 임계치(Threshold)를 설정하는 프로퍼티이다.

현재 사용되는 메모리의 크기는 프로퍼티 MEM\_MAX\_DB\_SIZE의 몇 퍼센트를 사용하고 있는지 확인하고, 설정한 임계치를 초과하면 스냅샷(snapshot)은 자동으로 중지된다.

**SNAPSHOT\_DISK\_UNDO\_THRESHOLD (단위: 백분율)****데이터 타입**

Unsigned Integer

**기본값**

80

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 100]

**설명**

스냅샷 설정(BEGIN SNAPSHOT) 이후 디스크에서 사용할 수 있는 임계치(Threshold)를 설정하는 프로퍼티이다.

현재까지 사용된 디스크 언두 테이블스페이스의 크기는 프로퍼티 SYS\_UNDO\_FILE\_MAX\_SIZE 에서 몇 퍼센트를 사용하는지 확인하고, 설정한 임계치를 초과하면 스냅샷(snapshot)은 자동으로 중지된다.

**이중화 프로퍼티**

다음 속성값들은 데이터베이스의 이중화 기능을 위한 값들이다. 데이터베이스 이중화에 대한 자세한 내용은 *Getting Started Guide*의 데이터베이스 이중화 장과 *Replication Manual*을 참조하기 바란다.

**REPLICATION\_ACK\_XLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

수신 스레드가 송신 스레드에게 ACK를 보내는 주기를 나타낸다.

수신 스레드는 XLog를 받아서 하나씩 반영하는 작업을 하는데, 반영한 XLog의 개수가 REPLICATION\_ACK\_XLOG\_COUNT를 넘게 되면 송신 스레드에게 ACK를 전송한다.

이 값이 너무 작으면 송신 스레드에 ACK를 자주 보내게 되어 성능 저하를 가져올 수 있다.

너무 큰 경우에는 송신 스레드가 ACK를 받기 위해 대기하는 시간을 초과하여 네트워크 장애로 판단할 수 있다. 또한, 오랜 시간 ACK를 받지 못하는 경우 이중화 재시작 SN이 갱신되지 않아, 체크포인트시 송신 스레드가 가장 최근의 로그 레코드부터 다시 시작되며, 이중화하지 못한 로그 파일이 삭제되는 현상이 발생할 수 있다.

**REPLICATION\_ALLOW\_DUPLICATE\_HOSTS****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 객체들의 원격 서버 IP 주소와 포트 번호를 서로 동일하게 설정하는 것을 허용할 것인지 지정한다.

0: 허용하지 않는다.

1: 허용한다.

**REPLICATION\_BEFORE\_IMAGE\_LOG\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 수신자의 "이전 이미지(before image)"가 송신자로부터 받은 XLog의 이전 이미지와 다르면 데이터 충돌이 발생할 수 있다.

사용자가 충돌이 발생한 데이터를 확인할 수 있도록, 이 프로퍼티를 설정하여 트레이스 로그에 이중화 수신자의 이전 이미지를 기록할 수 있다.

0: 기록하지 않음

1: 기록함

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_COMMIT\_WRITE\_WAIT\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 수신자가 수신한 XLOG를 복제 트랜잭션으로 수행 완료한 후에 디스크에 반영될 때까지 기다릴지 여부를 지정한다. 이 값이 0이면 기다리지 않으며, 1이면 디스크에 반영될 때까지 기다린다.

**REPLICATION\_CONNECT\_RECEIVE\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

60

**속성**

변경 가능, 단일 값



**값의 범위**[0, 2<sup>32</sup>-1]**설명**

이중화 시작시 대상 호스트에 접속을 시도한 후, 대기하는 시간이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_CONNECT\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

이중화를 수행하기 위해 대상 호스트에 대한 연결 수행 시, 이 프로퍼티에 설정된 시간 값 이상 응답이 없을 경우 연결을 더 이상 시도하지 않는다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_DDL\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 대상 테이블에 DDL 구문을 허용할 것인지를 나타낸다. 1로 설정한 경우 이중화 대상 테이블에 DDL을 실행할 수 있다.

DDL을 수행하기 전에 현재 세션에서 수행하는 트랜잭션의 이중화 프로퍼티를 NONE 이외의 값으로 설정해야 송신 스레드에서 DDL 실행을 알 수 있다. 이중화에서 허용하는 DDL 목록과 제약사항은 *Replication Manual*을 참조한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## REPLICATION\_DDL\_ENABLE\_LEVEL

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

이중화 대상 테이블에 사용할 수 있는 DDL 구문의 범위를 결정한다. 이 프로퍼티를 사용하려면 먼저 REPLICATION\_DDL\_ENABLE 프로퍼티의 값을 1로 설정해야 한다. REPLICATION\_DDL\_ENABLE\_LEVEL 값에 따라 허용되는 DDL 구문은

*Replication Manual*을 참조한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## REPLICATION\_DDL\_SYNC

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

이중화중 DDL 복제 여부를 나타낸다.

0 : 이중화중 DDL 복제를 허용하지 않음. DDL 수행 시 이중화 지역 서버 ( Local Server ) 에서만 수행된다.

1 : 이중화중 DDL 복제를 허용함. DDL 수행 시 이중화 원격 서버 ( Remote Server ) 에 DDL 이 복제된다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티 값을 변경할 수 있다.

## REPLICATION\_DDL\_SYNC\_TIMEOUT (단위: 초)

### 데이터 타입

Unsigned Integer

### 기본값

7200

### 속성

변경 가능, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

DDL 문장 실행시간이 이 프로퍼티에서 설정한 시간을 초과하면 해당 DDL 문장은 지역 서버, 원격 서버에서 모두 롤백된다.

DDL 복제 시 테이블 잠금 획득 실패(DDL\_LOCK\_TIMEOUT)와 같이 재시도 가능한 에러가 발생하면 이 프로퍼티에서 설정한 시간 동안 DDL 복제를 재시도한다. [REPLICATION\\_DDL\\_SYNC\\_TIMEOUT#1](#)

DDL 복제 대상 구문의 수행 시간은 DDL\_TIMEOUT 프로퍼티와 REPLICATION\_DDL\_SYNC\_TIMEOUT 프로퍼티 중 더 작은 값에 영향을 받는다.

Altibase 운영 중 ALTER SYSTEM 문 또는 ALTER SESSION 문을 이용하여 이 프로퍼티 값을 변경할 수 있다.

#### REPLICATION\_DDL\_SYNC\_TIMEOUT#1

DDL 복제 재시도 기능은 7.1.0.6.5 이상에서 지원된다.

## REPLICATION\_EAGER\_PARALLEL\_FACTOR

### 데이터 타입

Unsigned Integer

### 기본값

'논리 코어 수/2'와 '512' 중 작은 값

### 속성

읽기 전용, 단일 값

### 값의 범위

$[1, 512]$

### 설명

EAGER 모드로 이중화 수행시 여러 개의 송신 스레드가 병렬로 작업을 하게 된다. 이 프로퍼티는 병렬 작업할 송신 스레드의 개수를 지정한다. 이중화 송신 스레드의 개수를 늘리면 이중화 성능이 향상될 수 있다.

그러나, 이 프로퍼티를 사용하여 송신 스레드 개수를 늘릴 경우, 송신 스레드가 보내는 트랜잭션의 순서가 보장되지 않는다는 점을 주의해야 한다. 이에 대한 자세한 내용은 *Replication Manual*을 참고한다.

이 프로퍼티를 설정하지 않으면 기본값은 논리 코어 수와 512 중 더 작은 값이 된다.

## REPLICATION\_EAGER\_RECEIVER\_MAX\_ERROR\_COUNT

### 데이터 타입

Unsigned Integer

### 기본값

5

### 속성

변경 가능, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

Eager 모드로 이중화를 진행할 때, 이중화 수신자가 데이터 복제를 재시도하는 횟수를 설정할 수 있는 프로퍼티이다.

만약 이 값이 5로 설정되었을 때, Xlog 복제에 오류가 발생하면 이중화 수신자는 최대 5회까지 복제를 재시도한다. 재 시도에서도 성공하지 못하면 Altibase 서버를 강제로 종료한다. 이 값을 0으로 설정하면, 이중화 수신자가 Xlog 복제에 실패할 경우 성공할 때까지 복제를 재시도하며, 서버를 강제 종료하지 않는다.

Eager 모드의 이중화는 이중화 수신자와 송신자간의 데이터 일치가 중요한 환경에서 주로 사용된다. 따라서 사전에 데이터 불일치를 예방하기 위해 데이터 복제를 중단하고 서버를 종료하는 것이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## REPLICATION\_FAILBACK\_INCREMENTAL\_SYNC

### 데이터 타입

Unsigned Integer

### 기본값

1

### 속성

읽기 전용, 단일 값

### 값의 범위

$[0, 1]$

**설명**

이 프로퍼티는 증분 동기화(Incremental Sync) 수행 여부를 설정한다. 증분 동기화란 EAGER 모드의 이중화 작업 중 한 쪽 서버의 장애로 인해 데이터 불일치가 발생한 경우, 이 후 장애 서버가 다시 구동되는 중에 서버가 불일치 데이터를 동기화하는 것을 말한다.

- 0: 증분 동기화를 비활성화시킨다.
- 1: 증분 동기화를 활성화시킨다.

이중화에 참가하는 양쪽 Altibase 서버 모두 이 프로퍼티를 같은 값으로 설정해야 한다. 증분 동기화 및 Eager 이중화 장애 복구에 대한 자세한 내용은 *Replication Manual*을 참고한다.

**REPLICATION\_GAP\_UNIT (단위: 바이트)****데이터 타입**

Unsigned Long

**기본값**

1048576 (1MB)

**속성**

변경 가능, 단일 값

**값의 범위**

$[1, 2^{64}-1]$

**설명**

이중화 갭의 크기를 조회하는 REP\_GAP의 값을 표현하는 단위를 설정한다.

REP\_GAP의 값은 REP\_GAP\_SIZE의 값을 이 프로퍼티로 나눈 곱셈값이며, 나머지는 올림한다.

기본값이면 V\$REPGAP의 REP\_GAP\_SIZE는 바이트 단위로, REP\_GAP은 메가바이트 단위로 조회된다.

**REPLICATION\_GAPLESS\_ALLOW\_TIME (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

2000

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

갭 해소 옵션이 설정된 이중화 송신자는 이 프로퍼티에 설정한 시간만큼 이중화 격차(Replication Gap)를 허용한다. 이중화 송신자는 이 프로퍼티에 설정된 시간 안에 이중화 격차를 해소할 수 없다고 판단하면 서비스 트랜잭션의 커밋을 지연시키는 기능이 동작한다.

예를 들어 이 프로퍼티가 5이고, 송신자가 초당 처리하는 로그가 1,000건이라면 이중화 격차를 허용할 수 있는 로그의 개수는 5,000건이다. 만약 현재의 이중화 격차가 10,000건이라면, 허용하는 로그의 개수보다 5,000건을 초과하였으므로, 이중화 격차가 허용하는 로그 개수에 만족하기 위해서 트랜잭션이 5초간 지연되어야 한다는 것을 의미한다.

이 값이 0이면 이중화 격차를 허용하지 않는다는 의미이므로, 트랜잭션이 커밋되기 전에 이중화 격차를 해소시키고 커밋을 수행한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_GAPLESS\_MAX\_WAIT\_TIME (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

10,000,000

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

이중화 갭 해소(Gapless) 옵션이 설정된 이중화 송신자가 이중화 격차를 해소하기 위하여 트랜잭션 커밋을 지연시킬 수 있는 최대 시간을 의미한다.

이 값이 0이면 이중화 격차를 해소할 때까지 트랜잭션의 커밋을 지연시킨다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_GROUPING\_AHEAD\_READ\_NEXT\_LOG\_FILE (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

읽기 전용, 단일 값

**값의 범위**[1, 2<sup>32</sup>-1]**설명**

선행 분석(Ahead Analyzer) 스레드가 분석을 시작할 때, 송신자(Sender)가 현재 읽는 로그 파일 번호보다 얼마나 큰 번호의 로그 파일을 읽을 것인지 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_GROUPING\_TRANSACTION\_MAX\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

5

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 1000]

**설명**

Ahead Analyzer 스레드가 복수의 트랜잭션을 한 번에 최대 몇 개까지 그룹화하여 수신자에 전송할 것인지 정하는 프로퍼티이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_HBT\_DETECT\_HIGHWATER\_MARK****데이터 타입**

Unsigned Integer

**기본값**

5

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

연결(Connection) 응답을 하지 않는 경우 몇 회 이후에 장애로 판단할 것인지 결정한다. 따라서, 임의의 호스트의 장애를 판단하는 최대 시간은 REPLICATION\_HBT\_DETECT\_TIME \* REPLICATION\_HBT\_DETECT\_HIGHWATER\_MARK 이다.

즉, HeartBeat 스레드는 기본값인 30초 (5회 시도\* 6초) 동안 연결이 되지 않을 경우 장애로 판단한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_HBT\_DETECT\_TIME (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

6

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2592000]

**설명**

HeartBeat 스레드<sup>12</sup>의 검사 주기를 설정한다. 기본값인 6초 마다 HeartBeat 스레드는 해당 호스트에 대한 장애를 검사한다.

[<sup>12</sup>] HeartBeat 스레드: Altibase의 리플리케이션은 송신 스레드와 수신 스레드 간에 데이터 통신 수행 시 물리적인 장애가 발생했을 때, 신속하게 장애를 검출하기 위하여 HeartBeat 스레드를 만들어서 상대 호스트의 상태를 주기적으로 검사하는 기법을 도입했다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_IB\_LATENCY****데이터 타입**

Unsigned Integer

**기본값**

0



**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

rsocket의 RDMA\_LATENCY 옵션 값이다.

이 값이 1이면 CPU 자원을 소모하더라도, 적은 대기 시간을 사용한다.

**REPLICATION\_IB\_PORT\_NO****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 65535]

**설명**

이중화로 연결할 때 인피니밴드(InfiniBand)를 이용하여 접속할 경우 지역 서버의 이중화 포트 번호를 나타낸다.

인피니밴드를 이용하기 위해 IB\_ENABLE 프로퍼티의 값이 1이어야 한다. 이 값이 0인 경우에는 인피니밴드로 이중화를 연결할 수 없다.

**REPLICATION\_INSERT\_REPLACE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 프로퍼티는 이중화 중 INSERT 충돌이 발생했을 때 삽입된 내용을 유지할 지를 결정한다. 이 값이 0이면, INSERT는 커밋되지 않고 데이터 충돌은 에러 처리 된다. 반면에 이 값이 1이면 데이터 충돌은 무시되고 INSERT는 커밋된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_KEEP\_ALIVE\_CNT****데이터 타입**

Unsigned Integer

**기본값**

600

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

송신 스레드가 패킷을 전송하지 않고 (REPLICATION\_SENDER\_SLEEP\_TIME \* REPLICATION\_KEEP\_ALIVE\_CNT) 시간 동안 Sleep하면, KEEP\_ALIVE신호를 전송한다.

**REPLICATION\_LOCK\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

5

**속성**

변경 가능, 단일 값

**값의 범위**

$[0, 3600]$

**설명**

이중화 데드락이 발생하는 경우 수신 스레드 쪽에서는 무한정 잠금을 기다리게 되어 서비스가 중단될 수 있다. 이러한 경우를 방지하기 위해 수신 스레드는 해당 작업의 수행에 대해 잠금을 요구할 때, 최대 이 프로퍼티에 설정된 시간만큼만 잠금을 획득하기 위해 기다린다.

명시된 시간 내에 잠금을 획득하지 못하는 경우 해당 작업은 철회된다.

**REPLICATION\_LOG\_BUFFER\_SIZE (단위: 메가바이트)****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{12}-1]$ **설명**

이중화 전용 로그 버퍼를 사용함으로써 이중화 성능을 개선하는 프로퍼티이다. 이중화 전용 로그 버퍼에는 이중화에 필요한 로그만 필터링하여 저장하기 때문에 더 많은 로그가 버퍼에 존재한다.

송신 스레드는 로그를 읽기 위해서 로그 버퍼 또는 디스크에서 로그를 읽는다. 그러나 디스크에서 로그를 읽는 경우 송신 스레드의 처리 속도가 현저히 떨어질 수 있다. 더욱이 이중화를 실행하면 필요하지 않은 로그까지 읽어야 하는 부담이 발생한다. 이중화 전용 로그 버퍼는 이러한 부담을 완화시켜준다.

다수의 이중화 송신 스레드가 수행중일 때는 이중화 성능과 서비스 성능이 저하될 수 있다. 이중화 로그 버퍼가 하나이므로 다수의 송신 스레드가 접근한다면 동기화 오버헤드가 발생할 확률이 높아지기 때문이다.

이중화 로그 버퍼에서 읽은 로그는 REPLICATION\_SYNC\_LOG 값을 1로 하여도 디스크에 기록되지 않은 상태에서 수신 스레드로 전송될 수 있다.

REPLICATION\_LOG\_BUFFER\_SIZE를 너무 작게 설정하면, 사용하지 않는 것(0)보다 더 좋지 못한 성능을 낼 수 있다.

**REPLICATION\_MAX\_COUNT (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

32

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 10240]$

**설명**

Altibase에 생성할 수 있는 이중화 객체의 최대 개수를 지정한다.

기본값은 32이며, 한 서버에서 최대 32개의 원격 서버와 이중화로 연결할 수 있음을 의미한다.

**REPLICATION\_MAX\_LISTEN****데이터 타입**

Unsigned Integer

**기본값**

32

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 512]

**설명**

이중화 송신 쓰레드와 수신 쓰레드를 관리하는 Altibase 서버간의 통신 시 TCP/IP 소켓을 사용하는 경우 대기 큐(listen queue)의 크기를 지정하는 값이다.

**REPLICATION\_MAX\_LOGFILE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 65535]

**설명**

이중화를 위하여 재시작 리두 시점(Restart Redo Point)을 기준으로 삭제하지 않을 최대 로그파일의 개수이다.

이중화 시작 후, 지역 서버와 원격 서버간의 네트워크 속도 등의 문제로 지역 서버의 변경내용이 아직 원격 서버에 반영되지 않았을 경우, 체크포인트가 발생해도 서버는 로그 파일을 삭제할 수 없다. 이러한 문제가 발생하면, 지역서버의 로그파일 개수가 계속 증가하게 되고 결국 디스크가 가득 차버릴 수 있다(Disk Full).

따라서 체크포인트가 발생하였을 때, 재시작 리두 시점 (Restart Redo Point) 이전 로그 파일들의 개수가 이 프로퍼티로 정해 놓은 값보다 클 경우 이중화는 잠시 중단되며, Altibase는 중단된 시점의 일시 (날짜, 시각)와 “재시작 SN”을 SYS\_REPLICATION\_ 메타 테이블의 GIVE\_UP\_TIME과 GIVE\_UP\_XSN 칼럼에 각각 저장한다. 그리고 재시작 리두 시점 이전 로그파일들을 삭제한다. 현재 로그파일의 마지막(가장 큰) SN값이 새로운 이중화 “재시작 SN” 이 되며, 이 값도

SYS\_REPLICATION\_ 메타 테이블의 XSN 칼럼에 저장된다. 이중화는 이 새로운 “재시작SN”부터 다시 수행될 것이다. 이런 디폴트 작업 방식을 변경하고 싶다면, REPLICATION\_SENDER\_START\_AFTER\_GIVING\_UP 프로퍼티 값을 변경하면 된다. 또, SYS\_REPLICATION\_ 메타 테이블에서 특정 replication과 관련된 모든 정보들을 초기화하고 싶다면, “ALTER REPLICATION replication\_name RESET” 을 실행해라.

0으로 설정한 경우에는 이 기능을 적용하지 않는다. 참고로 체크포인트를 수행할 때 로그파일을 지우기 때문에, CHECKPOINT\_INTERVAL\_IN\_SEC와 CHECKPOINT\_IN\_LOG의 값을 함께 고려해야 한다.

## REPLICATION\_POOL\_ELEMENT\_COUNT (단위: 개수)

### 데이터 타입

Unsigned Integer

### 기본값

10

### 속성

변경 가능, 단일 값

### 값의 범위

[1, 1024]

### 설명

송신 스레드가 로그를 분석하여 칼럼 값을 복사할 때 사용하는 메모리의 개수이다. 이 때 메모리는 메모리 풀에서 미리 할당하며, 메모리 크기는 REPLICATION\_POOL\_ELEMENT\_SIZE에서 지정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## REPLICATION\_POOL\_ELEMENT\_SIZE (단위: 바이트)

### 데이터 타입

Unsigned Integer

### 기본값

256

### 속성

변경 가능, 단일 값

### 값의 범위

[128, 65536]

### 설명

송신 스레드가 로그를 분석하여 각각의 칼럼 값을 복사할 때 사용하는 메모리의 크기이다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## REPLICATION\_PORT\_NO

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 65535]

### 설명

이중화 연결을 할 때 지역 서버의 이중화 포트 번호이다. 이중화를 사용하지 않으려면 이 프로퍼티를 0으로 설정한다.

## REPLICATION\_PREFETCH\_LOGFILE\_COUNT

### 데이터 타입

Unsigned Integer

### 기본값

3

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1024]

### 설명

각각의 로그 파일 그룹에 대해서 미리 읽을 로그 파일의 수를 나타낸다. 로그 파일을 미리 읽어 둬으로써 송신 스레드가 파일에서 로그 레코드를 읽는 시간을 줄인다.

## REPLICATION\_RECEIVE\_TIMEOUT (단위: 초)

### 데이터 타입

Unsigned Integer

### 기본값

7200

**속성**

변경 가능, 단일 값

**값의 범위**

[0,  $2^{32}-1$ ]

**설명**

송신 스레드와 수신 스레드에서 공통적으로 사용하는 프로퍼티로 송신/수신 스레드로부터 어떤 메시지를 기다리는 최대 시간을 지정하는 프로퍼티이다.

송신 스레드에서는 상대방 수신 스레드로부터 응답을 기다리는 최대 시간이며 명시된 시간이 경과하게 되면 REPLICATION\_SENDER\_SLEEP\_TIMEOUT에서 지정된 시간만큼 sleep한 후 다시 상대방 수신 스레드와의 접속을 시도한다. 이 경우 기존 사용하던 소켓은 닫고 새로운 소켓을 생성하여 재연결을 시도한다.

수신 스레드에서는 상대방 송신 스레드로부터 어떤 메시지를 기다리는 최대 시간이다. 명시된 시간이 경과하게 되면 수신 스레드는 자동 종료되며 이후 상대방 송신 스레드가 어떤 메시지를 다시 보내게 되는 경우 다시 생성된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_RECEIVER\_APPLIER\_ASSIGN\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 수신자가 적용자(Applier)에게 XLog를 할당할 때 어떤 모드로 적용자에게 분배할 것인지를 선택하는 프로퍼티이다.

0: Transaction Count Mode

1: XLog Count Mode

이 값이 0이면 수신자는 할당된 트랜잭션이 가장 적은 적용자에게 XLog를 할당한다. 1인 경우에는 XLog 개수가 가장 적게 할당된 적용자에게 XLog를 할당한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_RECEIVER\_APPLIER\_QUEUE\_SIZE****데이터 타입**

Unsigned Integer

**기본값**

20

**속성**

변경 가능, 단일 값

**값의 범위** $[2, 2^{32}-1]$ **설명**

수신자가 적용자 스레드들의 대기 큐에 전달할 수 있는 최대 XLog의 개수이다.

이 값이 클수록 적용자(Applier)에 전달하는 XLog의 개수가 많아질 수 있으나, 대기 큐에 XLog 들이 쌓이므로 메모리를 보다 많이 사용하게 된다. 이 값은 적용자 숫자의 2배 가량으로 설정하는 것을 권고한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_RECOVERY\_MAX\_LOGFILE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

이중화를 이용한 데이터 복구를 위하여 재시작 리두 시점(Restart Redo Point)을 기준으로 삭제하지 않을 최대 로그 파일의 개수를 의미한다.

이중화를 이용한 데이터 복구를 하기 위하여 원격 서버에서 디스크에 반영(flush)되지 않은 로그에 해당하는 지역 서버의 로그를 삭제하지 않고 유지한다. 이 때 체크포인트가 발생해도 로그 파일을 삭제할 수 없어 지역 서버의 로그파일 개수가 계속 증가하게 되면 결국 디스크 풀이 발생할 수 있다.

따라서 체크포인트가 발생하였을 때 복구 옵션을 위한 로그 파일 개수의 최대 값을 넘는 경우, 이중화를 이용한 복구를 포기하고, 로그 파일들을 삭제한다. 그리고 이중화를 다시 시작한다.

0으로 설정한 경우, 이 기능을 적용하지 않는다.



체크포인트를 수행할 때 로그 파일을 지우기 때문에, CHECKPOINT\_INTERVAL\_IN\_SEC와 CHECKPOINT\_IN\_LOG의 값을 함께 고려해야 한다.

## REPLICATION\_RECOVERY\_MAX\_TIME (단위: 초)

### 데이터 타입

Unsigned Integer

### 기본값

$2^{32}-1$

### 속성

읽기 전용, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

이중화 모듈이 복구를 진행하는 중에 최대 시간이 지나면, 복구를 중단하고 지금까지 복구된 상태로 서비스를 진행할 수 있도록 한다.

이 값을 0으로 설정하는 경우, 이중화를 이용한 복구 과정을 진행하지 않는다.

Altibase는 이중화를 이용한 데이터 복구가 완료하기 전에 서비스 단계로 진행할 수 없어, 서비스 지연을 가져올 수 있다.

## REPLICATION\_SENDER\_AUTO\_START

### 데이터 타입

Unsigned Integer

### 기본값

1

### 속성

읽기 전용, 단일 값

### 값의 범위

$[0, 1]$

### 설명

Altibase가 서버 종료시 stop 시키지 않은 이중화 객체가 있었다면, 서버 재구동 시 이러한 이중화 객체들이 자동으로 시작되는 것이 기본 동작이다. 이 값을 0으로 설정하면, 서버 구동 시 이중화 객체가 자동으로 시작되는 것을 막을 수 있다.

**REPLICATION\_SENDER\_COMPRESS\_XLOG****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 프로퍼티는 이중화를 위해 데이터가 네트워크 상에서 전송될 때, 전송하는 패킷의 압축 여부를 지정한다.

0: 전송할 데이터를 압축하지 않음

1: 전송할 데이터를 압축함

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_SENDER\_ENCRYPT\_XLOG****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 프로퍼티는 이중화를 수행할 때 송신 쓰레드의 로그를 암호화하여 전송할 것인지 여부를 지정할 수 있다.

0: 전송 로그를 비암호화

1: 전송 로그를 암호화

## REPLICATION\_SENDER\_IP

### 데이터 타입

String

### 기본값

ANY

### 속성

읽기 전용, 다중 값

### 값의 범위

없음

### 설명

이중화 송신자의 IP 주소를 설정하는 프로퍼티이다. 값으로 ANY나 IP 주소를 입력할 수 있다.

기본값 ANY는 이중화 객체를 생성하는 지역 서버의 모든 IP 주소가 이중화 통신에 사용될 수 있으며 OS에서 할당한 IP 주소가 송신자 IP 주소로 사용된다. IP 주소를 값으로 설정하면 원격 서버(수신자)와 통신할 때 설정한 IP 주소만 사용된다. REPLICATION\_SENDER\_IP = *value*를 추가하여 여러 개의 IP 주소를 설정할 수 있으며 순서대로 송신자 IP 주소로 사용된다. IP 주소는 IPv4, IPv6, IPv6 확장 주소 형태로 입력할 수 있다.

### 설정 예시

```
REPLICATION_SENDER_IP = 10.0.0.1
REPLICATION_SENDER_IP = 0000:0000:0000:0000:0000:ffff:1400:0001
```

이 프로퍼티는 Altibase 7.1.0.8.0부터 지원하며 이중화 통신 방법이 TCP일 때 적용된다.

자세한 내용은 *Replication Manual*에서 [송신자 IP 주소 설정](#) 설명을 참고한다.

## REPLICATION\_SENDER\_SEND\_TIMEOUT (단위: 초)

### 데이터 타입

Unsigned Integer

### 기본값

7200

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 2<sup>32</sup>-1]

**설명**

송신 쓰레드가 원격 서버로 패킷을 전송할 때 대기하는 최대 시간을 지정하는 프로퍼티이다.

이 값은 REPLICATION\_RECEIVE\_TIMEOUT(기본값:7200초) 프로퍼티의 값과 동일한 값을 설정할 것을 권고한다. 이 값을 0으로 설정하면 송신 쓰레드가 블로킹 소켓(blocking socket)을 사용한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_SENDER\_SLEEP\_TIME (단위: 마이크로초)****데이터 타입**

Unsigned Integer

**기본값**

10000

**속성**

변경 가능, 단일 값

**값의 범위**

[0,  $2^{32}-1$ ]

**설명**

이 프로퍼티는 송신 쓰레드가 더 이상 읽을 로그가 없을 때, Sleep하는 시간을 지정한다. 특정 플랫폼에서는 짧은 시간의 Sleep이 무시되므로, 적당한 값을 지정해야 한다. REPLICATION\_KEEP\_ALIVE\_CNT와 함께 KEEP\_ALIVE를 전송하는데 사용된다.

**REPLICATION\_SENDER\_SLEEP\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

60

**속성**

변경가능, 단일 값

**값의 범위**

[0, 2592000]

**설명**

이중화 송신 쓰레드가 오류 상황에서 sleep을 해야 할 때 얼마나 할 것인지를 결정한다.

Altibase 운영 중에 ALTER SYSTEM 구문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_SENDER\_START\_AFTER\_GIVING\_UP****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 프로퍼티는 재시작 리두 시점 (Restart Redo Point) 이전 로그 파일들의 개수가 REPLICATION\_MAX\_LOGFILE 프로퍼티로 설정된 값을 초과하여 이중화가 잠시 중단된 이후, 다시 시작하는 방식을 결정한다.

0으로 설정된 경우에는, 이중화 "재시작 SN" (즉 SYS\_REPLICATIONS\_ 메타 테이블의 XSN 칼럼의 값)이 -1로 초기화되며, 이중화는 중지된다. 그리고, SYS\_REPLICATIONS\_ 메타 테이블의 IS\_STARTED 칼럼의 값이 0으로 바뀐다.

1로 설정된 경우, 이중화 "재시작 SN" 값은 현재 로그 파일의 마지막 (가장 큰) SN으로 변경되고, 이중화는 이 "재시작 SN" 부터 다시 수행된다.

**REPLICATION\_SERVER\_FAILBACK\_MAX\_TIME****데이터 타입**

Unsigned Integer

**기본값** $2^{32}-1$ **속성**

읽기 전용, 단일 값

**값의 범위**[0,  $2^{32}-1$ ]**설명**

EAGER 모드 이중화에서, 비정상 종료된 서버가 재구동 될 때, 다른 쪽(즉, 원격) 서버의 데이터와 동기화된 후에야 서비스가 재개된다. 이 때, 다른 서버의 로그를 비정상 종료했던 서버에 반영하는 과정이 이 프로퍼티에 설정된 시간(초)보다 오래 걸린다면, 비정상 종료했던 서버는 동기화가 완료되는 것을 포기하게 된다.

**REPLICATION\_SQL\_APPLY\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

Lazy 모드로 이중화 수행 과정에서 Active 서버와 Standby 서버의 이중화 테이블 메타 정보가 다른 경우, 아래의 조건에 따라 XLog가 SQL 문으로 변환되어 반영된다.

- 0: XLog를 이용하여 이중화가 동작한다. 이때, 메타 정보가 다르면 Handshaking 에러가 발생한다.
- 1: 아래의 조건에서 이중화를 수행하면, XLog가 SQL 문으로 변환이 되어 이중화 테이블에 반영한다.
- 칼럼 정보  
데이터 타입이 다를 경우  
size, precision, scale 이 다를 경우
- 제약 조건  
check 제약 조건이 다를 경우  
Not Null 제약 조건이 다를 경우  
다른 메타 정보 중 LOB 칼럼이 포함된 경우
- 인덱스  
유니크 인덱스나 Function-based 인덱스가 이중화 대상 칼럼과 이중화 대상이 아닌 칼럼으로 구성되어 있을 경우  
유니크 인덱스의 구성 정보가 다를 경우  
Function-based 인덱스의 구성 정보가 다를 경우

**REPLICATION\_SYNC\_APPLY\_METHOD****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 수행 중에 지역 서버와 원격 서버간의 불일치한 데이터를 어떤 방식으로 동기화 할 것인지 설정한다.

- 0: Normal Insert
- 1: Direct-Path Insert

이중화 SYNC 과정에서 Direct-Path Insert 방식을 이용할 경우, 데이터 동기화 이후 Index를 재생성한다. 따라서, 이중화 SYNC 과정이 도중에 실패할 경우, Index inconsistent 가 발생할 수 있다.

Direct-Path Insert에 대한 자세한 설명은 *Administrator's Manual*을 참조하기 바란다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_SYNC\_LOCK\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

30

**속성**

변경 가능, 단일 값

**값의 범위**[0,  $2^{32}-1$ ]**설명**

이중화 송신 쓰레드가 동기화 대상 테이블의 잠금(Lock)을 획득하기 위해 대기하는 시간을 정하는 프로퍼티이다.

ALTER REPLICATION .. SYNC 구문으로 이중화 동기화를 실행하면, 이중화 송신 쓰레드는 데이터 동기화를 수행한 후 계속해서 이중화를 시작할 로그 위치를 결정한다. 이 때 동기화 대상 테이블이 다른 트랜잭션에 의해 데이터가 변경되지 않도록 송신 쓰레드는 공유 잠금(S Lock)을 획득한다. 그러나 이미 다른 트랜잭션에 의해 잠금이 되었다면, 이 프로퍼티에 지정한 시간만큼 대기한다.

예를 들어 이 프로퍼티의 값이 30인 경우, 30초 동안 잠금을 획득할 수 없으면 동기화 시도는 오류 처리된다. 만약 이 값을 0으로 설정하면 이중화 대상 테이블에 대한 잠금은 획득하지 않지만, 데이터의 충돌이 발생할 수 있다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_SYNC\_LOG****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 수행 시 송신 스레드가 보내는 로그는 디스크에 내려갔는지의 여부에 관계없이 메모리 상의 로그를 보내기 때문에, 시스템 또는 매체 장애(media failure)와 같은 상황에서 데이터 불일치 등의 문제가 발생할 소지가 있다.

이러한 문제를 해결하기 위해서 이 값을 1로 설정하면, 송신 스레드는 디스크에 내려간 로그만 보내게 된다.

**REPLICATION\_SYNC\_TUPLE\_COUNT****데이터 타입**

Unsigned long

**기본값**

500000

**속성**

변경 가능, 단일 값

**값의 범위**[0,  $2^{64}-1$ ]**설명**

병렬 동기화시 송신 스레드가 한번에 읽어서 처리할 수 있는 레코드의 최대 개수를 지정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_TIMESTAMP\_RESOLUTION****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값



## 값의 범위

[0, 1]

## 설명

Active-Active 이중화 환경에서 데이터 충돌이 발생하였을 때 이 프로퍼티의 값에 따라 해결방식이 다르게 적용된다.

이 프로퍼티의 값이 1로 설정되고, 이중화 대상 테이블에 TIMESTAMP 칼럼이 있는 경우에 Timestamp-based Scheme 방식으로 충돌이 해결된다.

그러나 이중화 대상 테이블에 TIMESTAMP 칼럼이 있어도, 이 프로퍼티의 값이 0이면 이미 설정되어 있는 Conflict Resolution Scheme 방식이 사용된다.

데이터 충돌에 대한 자세한 설명은 *Replication Manual*을 참고한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## REPLICATION\_TRANSACTION\_POOL\_SIZE

### 데이터 타입

Unsigned Integer

### 기본값

2

### 속성

변경 가능, 단일 값

## 값의 범위

[0,  $2^{32}-1$ ]

## 설명

이 프로퍼티는 트랜잭션 풀에 미리 생성해 두는 트랜잭션의 개수를 지정한다. 이중화 수신자는 데이터를 복제할 때 트랜잭션을 사용한다. 그리고 성능 향상을 위해 이중화 수신자 별로 트랜잭션 풀(pool)을 만들고, 이 곳에 트랜잭션들을 미리 생성해 두고 사용할 수 있다.

만약 풀에 존재하는 트랜잭션의 수가 부족하다면, 이중화 수신자는 새로운 트랜잭션을 생성하여 이를 사용할 것이다. 그리고 사용이 끝난 트랜잭션은 풀에 반환되는데, 이 때 존재하는 트랜잭션의 개수가 이 프로퍼티에 지정된 값보다 크면, 풀에 반환되지 않고 바로 해제된다.

이 값을 너무 크게 지정하면, 일반 트랜잭션의 개수에 제한을 줄 수 있으므로 적절한 값으로 설정해야 한다. 이 프로퍼티에 허용된 최댓값은  $2^{32}-1$ 이지만, 실제 최댓값은 TRANSACTION\_TABLE\_SIZE 프로퍼티에 지정된 값과 같다. 만약 사용자가 이 값을 TRANSACTION\_TABLE\_SIZE의 값보다 크게 지정하면, 내부적으로 이 프로퍼티의 값이 TRANSACTION\_TABLE\_SIZE의 값으로 설정된다.

Altibase 구동 중에 이 프로퍼티의 값을 변경할 수 있다. 하지만 이중화 수신 스레드 생성시 트랜잭션 풀이 초기화되므로, 이중화를 재시작해야 변경된 프로퍼티의 값이 적용된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_UPDATE\_REPLACE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 작업 중 변경작업 충돌(update conflict) 시 변경된 내용의 반영을 결정한다. 값이 0 이면 충돌이 있을 경우 반영하지 않고 오류 처리하며, 1일 경우 충돌을 무시하고 반영한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REPLICATION\_META\_ITEM\_COUNT\_DIFF\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

Lazy 모드로 이중화 수행 과정에서 SPLIT PARTITION과 MERGE PARTITION, DROP PARTITION을 수행하여 Active 서버와 Standby 서버의 이중화 테이블 파티션 메타 아이템 개수가 다른 경우에 이중화를 START 할 수 있는 프로퍼티이다.

이 값을 1로 설정하면 이중화 테이블 파티션 메타 아이템 개수가 다른 경우에도 이중화를 START 할 수 있다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## 네트워크 관련 프로퍼티

### IB\_CONCHKSPIN

#### 데이터 타입

Unsigned Integer

#### 기본값

0

#### 속성

읽기 전용, 단일 값

#### 값의 범위

[0, 2147483]

#### 설명

rsocket의 RDMA\_CONCHKSPIN 옵션 값을 설정한다. 인피니밴드의 연결 확인을 어느 주기로 할 것인지 설정하는 값이다. 이 값이 0이면 rsocket의 기본값을 사용한다.

이 프로퍼티는 Altibase rdma-core ( <https://github.com/ALTIBASE/rdma-core> )를 사용할 때에만 사용할 수 있다.

### IB\_ENABLE

#### 데이터 타입

Unsigned Integer

#### 기본값

0

#### 속성

읽기 전용, 단일 값

#### 값의 범위

[0, 1]

#### 설명

Altibase에서 인피니밴드(InfiniBand, IB)의 사용 여부를 설정한다. 이 기능은 Linux에서만 사용할 수 있다.

0 : IB사용하지 않음 (기본값)

1: IB사용

**IB\_LATENCY****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

rsocket의 RDMA\_LATENCY 옵션 값을 설정한다.

이 프로퍼티의 값이 1이면 CPU 자원을 더 소모하더라도, 대기 시간을 줄이기 위하여 튜닝 코드를 사용한다.

이 프로퍼티는 Altibase rdma-core ( <https://github.com/ALTIBASE/rdma-core> )를 사용할 때에만 사용할 수 있다.

**IB\_LISTENER\_DISABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

Altibase를 시작할 때 인피니밴드 리스너(InfiniBand Listener)의 시작 여부를 설정할 수 있다.

0: 인피니밴드 리스너 시작(기본값)

1: 인피니밴드 리스너 시작하지 않음

**IB\_MAX\_LISTEN****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1024]

**설명**

인피니밴드(InfiniBand, IB)로 접속하여 동시에 Altibase를 사용할 수 있는 클라이언트의 최대 숫자이다.

**IB\_PORT\_NO****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[1024, 65535]

**설명**

인피니밴드(InfiniBand, IB)를 이용하여 통신하기 위해 사용하는 포트 번호이다.

**SNMP\_ALARM\_QUERY\_TIMEOUT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

세션에서 쿼리 타임아웃이 발생했을 때 트랩을 보낼지 여부를 결정한다. 디폴트 값은 1이며 트랩을 보낸다. 자세한 내용은 SNMP Agent Guide의 altiPropertyAlarmQueryTimeout를 참고하기 바란다.

**SNMP\_ALARM\_FETCH\_TIMEOUT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

FETCH TIMEOUT이 발생했을 때 트랩을 보낼지 여부를 결정한다. 기본값은 1이며 트랩을 보낸다. 자세한 내용은 SNMP Agent Guide의 altiPropertyAlarmFetchTimeout를 참고하기 바란다.

**SNMP\_ALARM\_UTRANS\_TIMEOUT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

UTRANS TIMEOUT이 발생했을 때 트랩을 보낼지 여부를 결정한다.

0: 트랩을 발생시키지 않는다.

1: 트랩을 발생시킨다.

자세한 내용은 SNMP Agent Guide의 altiPropertyAlarmUtransTimeout를 참고하기 바란다.

**SNMP\_ALARM\_SESSION\_FAILURE\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

3

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

세션에서 연속으로 프로퍼티에 설정한 횟수만큼 오류가 발생했을 때 트랩을 보낸다. 기본값은 3이고, 0이면 트랩을 보내지 않는다. 자세한 내용은 SNMP Agent Guide의 altiPropertyAlarmSessionFailureCount를 참고하기 바란다.

**SNMP\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 1]$ **설명**

SNMP 서비스를 사용하기 위해서는 1 (Enable)로 설정해야 한다. 기본값은 0 (Disable)이다.

0 : SNMP 사용

1: SNMP 사용하지 않음

**SNMP\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

3

**속성**

변경 가능, 단일 값

**값의 범위**

[3, 12]

**설명**

로그 출력의 레벨(Level)을 설정한다. 레벨은 '1', '2', '4', '8' 4가지가 있으며 각각의 합으로 설정된다. 기본값은 3(1 + 2)이며, 레벨이 Level1, Level3가 되면 관련된 로그가 출력된다.

**SNMP\_PORT\_NO****데이터 타입**

Unsigned Integer

**기본값**

20400

**속성**

읽기 전용, 단일 값

**값의 범위**

[1024, 65535]

**설명**

Altibase와 altisnmpd(Altibase sub agent)간에 통신을 하기 위해 사용되는 UDP 포트이다.

**SNMP\_TRAP\_PORT\_NO****데이터 타입**

Unsigned Integer

**기본값**

20400

**속성**

읽기 전용, 단일 값

**값의 범위**

[1024, 65535]



**설명**

Altibase와 altisnmpd(Altibase sub agent)간에 트랩을 보내기 위해 통신을 하는 UDP 포트이다.

**SNMP\_RECV\_TIMEOUT (단위: 밀리초)****데이터 타입**

Unsigned Integer

**기본값**

1000

**속성**

읽기 전용, 단일 값

**값의 범위**

[1,  $2^{32}-1$ ]

**설명**

통신을 사용하기 위한 타임아웃 값이다.

Altibase와 altisnmpd(Altibase sub agent)간에 통신을 할 때 이 프로퍼티에 설정한 값만큼 대기한다.

**SNMP\_SEND\_TIMEOUT (단위: 밀리초)****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

읽기 전용, 단일 값

**값의 범위**

[1,  $2^{32}-1$ ]

**설명**

통신을 사용하기 위한 타임아웃 값이다.

Altibase와 altisnmpd(Altibase sub agent)간에 통신을 할 때 이 프로퍼티에 설정한 값만큼 대기한다.

**SSL\_CA****데이터 타입**

String

**기본값**

없음

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

CA 인증서의 소유권을 인정받기 위해서 CA 인증서를 저장하는 파일의 경로를 지정할 수 있다. CA 인증서 파일은 사용자 지정 경로나 X.509 형식의 디렉토리에 위치한다. 예를 들어 \$ALTIBASE\_HOME/cert/ ca-cert.pem으로 할 수 있다.

**SSL\_CAPATH****데이터 타입**

String

**기본값**

없음

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

CA 디렉토리 형식의 CAPATH를 지정할 수 있다. 예를 들어서, 지정할 경우 \$ALTIBASE\_HOME/etc/ssl/certs로 할 수 있다.

**SSL\_CERT****데이터 타입**

String

**기본값**

없음

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

Altibase 서버의 인증서 파일 경로를 지정한다. 예를 들어 \$ALTIBASE\_HOME/cert/server-cert.pem으로 지정할 수 있다.

**SSL\_CIPHER\_LIST****데이터 타입**

String

**기본값**

없음

**속성**

읽기 전용, 단일 값

**값의 범위**

255

**설명**

이 프로퍼티는 Altibase가 클라이언트와 협의하여 사용할 수 있는 암호 알고리즘의 이름 후보들이다. 암호 알고리즘은 사용자의 보안 정책에 따라 하나 또는 그 이상의 암호를 사용할 수 있다. 두 개 이상의 암호를 사용할 경우 콜론(:)으로 구분한다. 사용자가 사용할 수 있는 암호 목록은 OpenSSL( <http://www.openssl.org/> )에서 확인하거나 아래처럼 명령어를 사용하여 확인할 수 있다.

```
$ openssl ciphers
```

**SSL\_CLIENT\_AUTHENTICATION****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

서버와 클라이언트가 SSL로 통신을 할 때 클라이언트에게 인증서를 요청할 것인지를 설정할 수 있다. 기본값이 0일 때에는 클라이언트가 서버만 인증서를 통해 확인하다. 이 값을 1로 하면, 서버는 SSL 핸드셰이크중에 클라이언트에게 인증서를 요청한다.

0: 서버만 인증(기본값)

1: 서버와 클라이언트 동시 인증

**SSL\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

Altibase가 SSL/TLS를 이용할지 여부를 설정할 수 있다. 기본값은 0 (Disable)이다.

0 : SSL/TLS 사용하지 않음

1: SSL/TLS 사용

**SSL\_KEY****데이터 타입**

String

**기본값**

없음

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

Altibase 서버의 개인 키(private key)가 저장된 파일 경로를 지정한다.

예를 들어 \$ALTIBASE\_HOME/cert/server-key.pem으로 지정할 수 있다.

**SSL\_MAX\_LISTEN****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 16384]

**설명**

SSL/TLS로 접속하여 동시에 Altibase를 사용할 수 있는 대기 큐(listen queue)의 최대 크기를 지정한다. 이 값이 클수록 Altibase에 더 많은 메모리가 필요하다.

**SSL\_PORT\_NO****데이터 타입**

Unsigned Integer

**기본값**

20443

**속성**

변경 가능, 단일 값

**값의 범위**

[1024, 65535]

**설명**

서버와 클라이언트가 SSL/TLS를 이용하여 통신하기 위해 사용되는 포트번호를 설정한다.

**TCP\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

Altibase 서버가 통신 프로토콜로 TCP/IP를 선택하여 통신할 것인지를 설정할 수 있다.

0 : TCP/IP 사용하지 않음

1: TCP/IP 사용

**메시지 로그 관련 프로퍼티****ALL\_MSGLOG\_FLUSH****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 값이 1인 경우 데이터베이스의 모든 메시지가 기록 즉시 디스크에 반영되고 0인 경우에는 일정 주기에 한 번씩 디스크에 반영된다. 과도한 로깅으로 인한 성능저하를 예방하기 위해서는 0으로 설정하는 것이 적절하고, 데이터베이스 문제 진단 시에는 1로 설정하고 작업한다.

**COLLECT\_DUMP\_INFO****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

서버가 비정상 종료할 때 메시지 파일과 로그 파일 등의 정보를 수집할 것인지 여부를 지정한다.

0: 메시지 파일과 로그 파일의 정보를 수집하지 않음

1: 메시지 파일과 로그 파일의 정보를 수집함

**DK\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**[0,  $2^{32}-1$ ]**설명**

데이터베이스 링크를 할 때 연결 프로세스에 지정할 수 있는 메시지 파일의 최대 개수를 설정한다.

**DK\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_lk.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

데이터베이스 링크가 연결 프로세스 처리 과정에서 발생하는 메시지가 기록되는 파일이다.

**DK\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

6

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

데이터베이스 링크의 연결 프로세스 모듈에서 발생하는 경고 메시지나 추적 메시지를 LK\_MSGLOG\_FILE에 기록할 것인지 여부를 나타내는 플래그 값이다.

0이면 기록하지 않고, 0보다 큰값이면 기록한다.

**DK\_MSGLOG\_SIZE****데이터 타입**

Unsigned Integer

**기본값** $10 * 1024 * 1024$ **속성**

읽기 전용, 단일 값



**값의 범위**[0, 2<sup>32</sup>-1]**설명**

데이터베이스 링크에서 연결 프로세스의 메시지 파일 최대 크기를 지정한다.

**DUMP\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

서버 오류에 대한 디버깅 정보가 기록되는 메시지 파일의 최대 개수를 지정한다.

**DUMP\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_dump.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

서버 오류에 대한 디버깅 정보가 기록되는 메시지 파일명을 지정한다.

**DUMP\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

10 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

디버깅 정보가 기록되는 메시지 파일의 최대 크기를 지정한다.

**ERROR\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

서버 오류 메시지 파일의 최대 개수를 지정한다.

**ERROR\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_error.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

서버에서 발생하는 오류 메시지를 기록할 파일명을 지정한다. Altibase 서버 비정상 종료시 프로세스의 콜 스택도 이 파일에 기록된다.

**ERROR\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

10 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

서버 오류 메시지가 저장되는 파일의 최대 크기를 지정한다.

**LB\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

서비스 쓰레드 관련 메시지 파일의 최대 개수를 설정한다.

**LB\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_lb.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

서비스 쓰레드 관련 메시지를 기록하는 파일이다.

**LB\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

LB\_MSGLOG\_FILE에 기록될 서비스 쓰레드 관련 메시지 항목을 지정한다.

0: 기록하지 않음

1: 서비스 쓰레드의 생성 및 해제

2: Task 이동에 대한 로그

3: 서비스 쓰레드의 생성 및 해제, Task 이동에 대한 로그

**LB\_MSGLOG\_SIZE****데이터 타입**

Unsigned Integer

**기본값** $10 * 1024 * 1024$

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

서비스 스레드 관련 메시지가 기록되는 파일의 최대 크기를 지정한다.

**MM\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

메인 모듈을 위한 메시지 파일의 최대 개수를 지정한다.

**MM\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_mm.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

메일 모듈 처리 시에 발생하는 메시지가 기록되는 파일이다.

**MM\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

10 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

메인 모듈의 메시지 파일의 최대 크기를 지정한다.

**MM\_SESSION\_LOGGING****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위** $[0, 1]$ **설명**

데이터베이스 서버에 대한 모든 로그인, 로그오프 이벤트를 발생시키는 세션 정보를 MM\_MSGLOG\_FILE에 기록할 지 여부를 지정한다. 세션 정보는 세션 ID, 사용자 이름, 클라이언트 IP 주소, 클라이언트 프로그램의 PID, 그리고 클라이언트 프로그램에 대한 세부 정보를 포함한다.

**NETWORK\_ERROR\_LOG****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

서버 메시지 파일에 네트워크 관련 에러 메시지의 출력 여부를 지정한다.

네트워크 환경이 불안정하여 에러 메시지의 출력이 많을 때 0으로 설정하면, 네트워크 관련 에러 메시지의 출력을 막을 수 있다.

**QP\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,  $2^{32}-1$ ]

**설명**

질의 처리기 메시지 파일의 최대 개수를 지정한다.

**QP\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_qp.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

질의 연산 (Query Processing) 처리 시에 발생하는 메시지가 기록되는 파일이다.

**QP\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

변경가능, 단일 값

**값의 범위**

$[0, 2^{32} - 1]$

**설명**

쿼리 프로세싱 모듈에서 발생하는 경고 메시지나 트레이스 메시지를 QP\_MSGLOG\_FILE에 기록 할지 여부를 나타내는 플래그 값이다.

0이면 기록하지 않고, 0 보다 큰값이면 기록한다.

**QP\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

$10 * 1024 * 1024$

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32} - 1]$

**설명**

질의 처리기 메시지 파일의 최대 크기를 지정한다.

**QUERY\_PROF\_FLAG****데이터 타입**

Integer



**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**[0, 2<sup>6</sup>-1]**설명**

서버 내에서 수행되는 작업과 서버의 상태 정보를 파일로 기록하여 분석할 수 있도록 한다. 사용자는 값을 조합하여 원하는 정보를 기록하도록 설정할 수 있으며, 값에 따라서 기록되는 정보는 다음과 같다.

- 0 : 기록하지 않음
- 1 : SQL 문이 실행될 때마다 실행된 SQL문, 실행시간, 실행정보, 색인 및 디스크 접근 정보 출력. 단, 실행시간이 제대로 출력되게 하려면 TIMED\_STATISTICS 프로퍼티를 1로 설정해야 한다.
- 2 : SQL 문이 실행될 때마다 BIND 파라미터 출력
- 4 : SQL 문이 실행될 때마다 실행계획 출력
- 8 : 3초마다 세션 정보 출력 (V\$SESSTAT 의 데이터)
- 16 : 3초마다 시스템 정보 출력 (V\$SYSSTAT 의 데이터)
- 32 : 3초마다 메모리 정보 출력 (V\$MEMSTAT 의 데이터)

예를 들어 프로퍼티를 1+4+32=37로 설정하면, SQL 문이 실행될 때마다 SQL 문의 실행 정보와 실행계획을 출력하고 3초마다 메모리 정보를 출력한다.

파일에 대한 분석은 altiProfile 유틸리티로 가능하며, 자세한 설명은 *Utilities Manual*의 *유틸리티*를 참조한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**QUERY\_PROF\_LOG\_DIR****데이터 타입**

String

**기본값**

\$ALTIBASE\_HOME/trc

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 1024]

**설명**

Altibase에서 수행된 작업과 서버의 상태 정보가 기록되는 파일이 위치할 디렉토리 경로를 지정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**JOB\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

JOB 관련 메시지 파일의 최대 개수를 지정한다.

**JOB\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_job.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

JOB 처리 시에 발생하는 메시지가 기록되는 파일이다.

**JOB\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

2

**속성**

변경가능, 단일 값

**값의 범위** $[0, 2^{32} - 1]$ **설명**

JOB 에서 발생하는 경고 메시지나 트레이스 메시지를 JOB\_MSGLOG\_FILE에 기록할지를 나타내는 플래그 값이다.

0이면 기록하지 않고, 0 보다 큰 값이면 기록한다.

**JOB\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값** $10 * 1024 * 1024$ **속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32} - 1]$ **설명**

JOB 관련 메시지 파일의 최대 크기를 지정한다.

**RP\_CONFLICT\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**[0, 2<sup>32</sup>-1]**설명**

이중화 충돌시 트레이스 로그가 기록되는 파일의 최대 개수를 지정한다.

**RP\_CONFLICT\_MSGLOG\_DIR****데이터 타입**

String

**기본값**

\$ALTIHOME/trc

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

이중화 충돌이 발생할 때 트레이스 로그가 기록되는 파일이 위치할 디렉토리를 지정한다.

**RP\_CONFLICT\_MSGLOG\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

이중화 충돌에 대한 트레이스 로그를 기록할 것인지 여부를 지정한다.

**RP\_CONFLICT\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_rp\_conflict.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

이중화 충돌에 대한 트레이스 로그가 기록되는 파일이다.

**RP\_CONFLICT\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

6

**속성**

변경가능, 단일 값

**값의 범위** $[0, 2^{32} - 1]$ **설명**

이중화 충돌과 관련된 트레이스 로그의 로깅 레벨을 지정한다.

설정 가능한 값은 아래와 같다:

2: 충돌 메시지 기록

4: 충돌을 유발한 SQL문 기록

6: 충돌 메시지와 충돌을 유발한 SQL문을 모두 기록

**RP\_CONFLICT\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

10 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

이중화 충돌에 대한 트레이스 로그 파일의 최대 크기를 지정한다.

**RP\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

이중화 메시지 파일의 최대 개수를 지정한다.

**RP\_MSGLOG\_FILE****데이터 타입**

String

**기본값**

altibase\_rp.log

**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

복제 관리자 (Replication) 처리 과정에서 발생하는 메시지가 기록되는 파일이다.

**RP\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

복제 관리자(Replication Manager) 모듈에서 발생하는 경고 메시지나 트레이스 메시지를 RP\_MSGLOG\_FILE에 기록할지 여부를 나타내는 플래그 값이다.

0이면 기록하지 않고, 0보다 큰 값이면 기록한다.

**RP\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

$10 * 1024 * 1024$

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

이중화 메시지 파일의 최대 크기를 지정한다.

**SERVER\_MSGLOG\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

10

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

서버 메시지 파일의 최대 개수를 지정한다.

**SERVER\_MSGLOG\_DIR****데이터 타입**

String

**기본값**`$ALTIBASE_HOME/trc`**속성**

읽기 전용, 단일 값

**값의 범위**

없음

**설명**

서버 구동 및 종료 등에 대한 시스템 정보가 기록되어 있는 서버 모듈의 메시지 파일인 `SERVER_MSGLOG_FILE`과 서버 관리 프로그램에서 사용하는 내부 파일인 `altibase.lock`이 위치할 경로이다.

또한, `SM_MSGLOG_FILE`, `QP_MSGLOG_FILE`, `RP_MSGLOG_FILE` 등과 같은 서버 모듈의 메시지 파일을 위한 기본 경로로 사용된다.

**SERVER\_MSGLOG\_FILE****데이터 타입**

String

**기본값**`altibase_boot.log`**속성**

읽기 전용, 단일 값



**값의 범위**

없음

**설명**

서버 모듈에 대한 메시지를 남기는 파일명을 지정한다.

Altibase의 구동 및 경고, 비정상 종료 시에 출력되는 메시지를 기록하는 파일이다.

**SERVER\_MSGLOG\_FLAG****데이터 타입**

Unsigned Integer

**기본값**

7

**속성**

변경가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

서버 모듈에서 발생하는 경고 메시지나 트레이스 메시지를 SERVER\_MSGLOG\_FILE에 기록 할지 여부를 나타내는 플래그 값이다.

0이면 기록하지 않고, 0 보다 큰 값이면 기록한다.

**SERVER\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

$10 * 1024 * 1024$

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

서버 메시지 파일의 최대 크기를 지정한다.

## SM\_MSGLOG\_COUNT

### 데이터 타입

Unsigned Integer

### 기본값

10

### 속성

읽기 전용, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

저장 관리자 메시지 파일의 최대 개수를 지정한다.

## SM\_MSGLOG\_FILE

### 데이터 타입

String

### 기본값

altibase\_sm.log

### 속성

읽기 전용, 단일 값

### 값의 범위

없음

### 설명

저장 관리자(Storage Manager) 처리 과정에서 발생하는 메시지가 기록되는 파일이다.

## SM\_MSGLOG\_FLAG

### 데이터 타입

Unsigned Integer

### 기본값

2147483647

**속성**

변경가능, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

저장 관리자(Storage Manager)모듈에서 발생하는 경고 메시지나 트레이스 메시지를 SM\_MSGLOG\_FILE에 기록 할지 여부를 나타내는 플래그 값이다. 0이면 기록하지 않고, 0보다 큰값이면 기록한다.

**SM\_MSGLOG\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

$10 * 1024 * 1024$

**속성**

읽기 전용, 단일 값

**값의 범위**

$[0, 2^{32}-1]$

**설명**

저장 관리자 메시지 파일의 최대 크기를 지정한다.

**TRCLOG\_DETAIL\_PREDICATE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

$[0, 1]$

**설명**

iSQL에서 Explain plan 기능 사용 시 where 절의 predicate 분류 상태를 나타낸다. 이 trace log를 사용하기 위해 1을 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## XA\_MSGLOG\_COUNT

### 데이터 타입

Unsigned Integer

### 기본값

10

### 속성

읽기 전용, 단일 값

### 값의 범위

$[0, 2^{32}-1]$

### 설명

서버용 XA 메시지 파일의 최대 개수를 지정한다.

## XA\_MSGLOG\_FILE

### 데이터 타입

String

### 기본값

altibase\_xa.log

### 속성

읽기 전용, 단일 값

### 값의 범위

없음

### 설명

서버용 XA 메시지 로그가 기록되는 파일이다.

## XA\_MSGLOG\_FLAG

### 데이터 타입

Unsigned Integer

### 기본값

0

**속성**

변경가능, 단일 값

**값의 범위**

[0, 3]

**설명**

서버용 XA 메시지 로그의 기록 단계를 설정하는 속성으로 설정값은 다음과 같다.

0: XA 관련 최소 필수 메시지만 기록

1: XA 연산 호출을 기록

2: XID 할당, 해제 등을 기록함

3: XA 관련 모든 메시지 로그를 기록함

**XA\_MSGLOG\_SIZE****데이터 타입**

Unsigned Integer

**기본값**

10 \* 1024 \* 1024

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,  $2^{32}-1$ ]

**설명**

서버용 XA 메시지 파일의 최대 크기를 지정한다.

**데이터베이스 링크 관련 프로퍼티****DBLINK\_ALILINKER\_CONNECT\_TIMEOUT (단위: 초)****데이터 타입**

Unsigned Integer

**기본값**

100

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{32}-1]$ **설명**

Altibase 서버에서 AltiLinker로 접속할 때, 접속 대기 최대 시간을 지정한다.

**DBLINK\_DATA\_BUFFER\_ALLOC\_RATIO****데이터 타입**

Unsigned Integer

**기본값**

50

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 1]$ **설명**

데이터베이스 링크로 원격 서버에서 수행한 질의의 결과 집합을 로컬 서버가 가져와서 저장하는 공간을 데이터베이스 링크 전용 데이터 버퍼라고 한다. 이 프로퍼티는 데이터베이스 링크 전용 데이터 버퍼의 남은 공간에서 원격 질의별로 할당 받을 record buffer의 비율을 지정한다.

**DBLINK\_DATA\_BUFFER\_BLOCK\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

128

**속성**

읽기 전용, 단일 값

**값의 범위** $[0, 2^{12}-1]$

**설명**

데이터베이스 링크를 위해 원격 서버에서 수행한 질의의 결과 집합을 로컬 서버에서 받아와서 저장하는 공간을 데이터 버퍼라고 한다. 이 프로퍼티는 데이터 버퍼를 구성하는 메모리 블록(record buffer block)의 초기 할당 개수를 지정한다. 데이터 버퍼의 크기는  $DBLINK\_DATA\_BUFFER\_BLOCK\_COUNT * DBLINK\_DATA\_BUFFER\_BLOCK\_SIZE$ 가 될 것이다.

**DBLINK\_DATA\_BUFFER\_BLOCK\_SIZE (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

2 MBytes

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 29]

**설명**

데이터베이스 링크를 위해 원격 서버에서 수행한 질의의 결과 집합을 로컬 서버에서 받아와서 저장하는 공간을 데이터 버퍼라고 한다. 이 프로퍼티는 데이터 버퍼를 구성하는 메모리 블록(record buffer block)의 크기를 지정한다.

**DBLINK\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 1]

**설명**

데이터베이스 링크 사용 여부를 결정한다. 데이터베이스 링크를 사용하고자 할 때는 이 값을 1로 설정한다. 값이 0이면 데이터베이스 링크를 사용할 수 없다.

**DBLINK\_GLOBAL\_TRANSACTION\_LEVEL****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 2]

**설명**

글로벌 트랜잭션 수행 레벨을 지정한다. 이 프로퍼티를 0으로 설정할 경우, DBLINK\_REMOTE\_STATEMENT\_AUTOCOMMIT 프로퍼티를 원격 데이터베이스의 AUTOCOMMIT 모드와 동일하게 지정해야 한다.

0: remote statement execution level. 이 레벨에서는 하나의 글로벌 트랜잭션에 참여하는 서버들(로컬 및 원격 서버)의 각 트랜잭션이 별개의 트랜잭션으로 인식되므로, 글로벌 트랜잭션을 커밋하기 위해서는 각 서버의 트랜잭션을 별도로 commit해 주어야 한다. 즉, 이 레벨에서는 글로벌 트랜잭션이라 하더라도 로컬 서버에서 수행하는 commit이 원격 서버에 영향을 미치지 않는다.

1: simple transaction commit level. 이 레벨에서는 하나의 글로벌 트랜잭션에 참여하는 서버들(로컬 및 원격 서버)의 모든 트랜잭션이 하나의 트랜잭션으로 인식된다. 즉, 로컬 서버에서 글로벌 트랜잭션을 커밋하면 해당 트랜잭션에 참여하는 모든 트랜잭션들이 커밋된다.

2: Two-Phase Commit (2PC) Level. 이 레벨에서는 하나의 글로벌 트랜잭션에 참여하는 데이터베이스 시스템의 트랜잭션 정합성을 보장하기 위해 2PC 프로토콜을 지원한다. 글로벌 트랜잭션이 시작되면 트랜잭션이 종료할 때까지 이 프로퍼티 값을 변경 할 수 없다.

**DBLINK\_RECOVERY\_MAX\_LOGFILE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**[1,  $2^{32}-1$ ]



**설명**

이 프로퍼티는 트랜잭션 수행 중에 이기종 데이터베이스 시스템에 장애가 발생할 경우, Altibase가 분산 트랜잭션 복구를 위해 유지하는 로그 파일의 최대 개수이다. 이 프로퍼티 값이 0이면 분산 트랜잭션 복구를 위한 로그를 삭제하지 않아 정합성이 보장된다. 그러나 프로퍼티 값을 1보다 큰 값으로 설정하면, 로그 파일의 개수가 초과할 경우에 분산 트랜잭션이 완료되지 않더라도 체크포인트에 의해 로그가 삭제되므로 정합성을 보장할 수 없다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**DBLINK\_REMOTE\_STATEMENT\_AUTOCOMMIT****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

원격 데이터베이스의 AUTOCOMMIT 모드를 지정한다.

DBLINK\_GLOBAL\_TRANSACTION\_LEVEL을 0으로 설정한 경우에만 이 프로퍼티가 적용된다.

0: autocommit-off

1: autocommit-on

**DBLINK\_REMOTE\_TABLE\_BUFFER\_SIZE (단위: 메가바이트)****데이터 타입**

Unsigned Integer

**기본값**

50

**속성**

변경 가능, 단일 값

**값의 범위**

[0,  $2^{32}-1$ ]

## 설명

REMOTE\_TABLE 키워드를 사용하여 원격 서버에서 질의를 수행하면, 질의 결과를 임시로 저장할 수 있는 메모리 버퍼이다.

저장된 질의 결과는 질의 처리기에 전달된 후 삭제된다. 그러나 결과 집합중의 하나의 레코드가 이 프로퍼티에 지정한 크기보다 크면 저장할 수 없으므로, 프로퍼티의 값을 조정해야 한다.

## 감사 관련 프로퍼티

### AUDIT\_FILE\_SIZE (단위: 바이트)

#### 데이터 타입

Unsigned Int

#### 기본값

100M

#### 속성

변경 가능, 단일 값

#### 값의 범위

[0, 2<sup>32</sup>-1]

## 설명

감사된 구문의 정보가 기록되는 바이너리 파일의 크기를 바이트 단위로 지정한다. 이 프로퍼티는 AUDIT\_OUTPUT\_METHOD 프로퍼티의 값이 0일 때 적용된다.

Altibase 운영 중에는 아래와 같이 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

```
ALTER SYSTEM SET AUDIT_FILE_SIZE=10000;
```

단, 서버가 auditing을 수행 중이라면 다음 생성되는 파일부터 변경된 크기가 적용된다. 만약 프로퍼티의 변경을 바로 적용하려면 ALTER SYSTEM STOP AUDIT과 ALTER SYSTEM START AUDIT 구문으로 auditing을 재시작하라.

### AUDIT\_LOG\_DIR

#### 데이터 타입

String

#### 기본값

\$ALTIBASE\_HOME/trc

#### 속성

변경 가능, 다중값

**값의 범위**

없음

**설명**

감사된 구문 정보가 기록되는 바이너리 파일이 생성될 디렉토리를 지정한다. 이 프로퍼티는 AUDIT\_OUTPUT\_METHOD 프로퍼티의 값이 0일 때 적용된다.

Altibase 운영 중에는 아래와 같이 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

```
ALTER SYSTEM SET AUDIT_LOG_DIR='/tmp';
```

단, 서버가 auditing을 수행 중이라면 다음 생성되는 파일부터 변경된 디렉토리가 적용된다. 만약 프로퍼티의 변경을 바로 적용하려면 ALTER SYSTEM STOP AUDIT과 ALTER SYSTEM START AUDIT 구문으로 auditing을 재시작하라.

**AUDIT\_OUTPUT\_METHOD****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 불가, 단일값

**값의 범위**

[0, 9]

**설명**

감사된 구문 정보가 기록되는 파일의 형태를 지정한다. Altibase는 감사 정보를 바이너리 또는 시스로그(syslog)로 저장할 수 있다. 단 syslog는 리눅스 운영체제에서만 지원한다.

기본값을 설정한 경우 감사된 정보가 바이너리 형태로 AUDIT\_LOG\_DIR 프로퍼티에서 지정한 디렉토리에 저장된다. 1~9의 값을 설정한 경우, syslog를 사용해서 syslog.conf(또는 rsyslog.conf)에 로그가 쌓이게 된다. syslog의 순서는 syslog.conf 파일의 LOG\_INFO를 사용한다.

0: 바이너리 파일로 AUDIT\_LOG\_DIR 프로퍼티에서 지정한 디렉토리에 저장

1: user 퍼실리티를 /var/log/messages에 저장. 파일명은 시스템 설정시 변경 가능.

2: 사용자 정의에 따라 local0 facility에 저장

3: 사용자 정의에 따라 local1 facility에 저장

4: 사용자 정의에 따라 local2 facility에 저장

5: 사용자 정의에 따라 local3 facility에 저장

6: 사용자 정의에 따라 local4 facility에 저장

7: 사용자 정의에 따라 local5 facility에 저장

8: 사용자 정의에 따라 local6 facility에 저장

9: 사용자 정의에 따라 local7 facility에 저장

## AUDIT\_TAG\_NAME\_IN\_SYSLOG

### 데이터 타입

String

### 기본값

AUDIT

### 속성

변경 불가, 단일값

### 값의 범위

없음

### 설명

시스로그(syslog)를 사용해서 감사된 구문의 정보를 기록할 때 구분자로 "AUDIT" 태그가 사용된다.

## 에이전트 관련 프로퍼티

### EXTPROC\_AGENT\_CONNECT\_TIMEOUT

#### 데이터 타입

Unsigned Integer

#### 기본값

60 초

#### 속성

읽기 전용, 단일 값

#### 값의 범위

[5,  $2^{32}-1$ ]

#### 설명

사용자가 외부 프로시저를 호출할 때, Altibase가 외부 프로시저용 에이전트에 연결을 시도하는 최대 시간을 초 단위로 설정한다. 이 프로퍼티에 설정한 시간 동안 연결을 할 수 없다면 외부 프로시저 호출은 연결을 실패했다는 오류 메시지와 함께 종료된다. 연결이 맺어진 후의 외부 프로시저 호출은 이 프로퍼티 값의 영향을 받지 않는다.

**EXTPROC\_AGENT\_CALL\_RETRY\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[1, 10]

**설명**

사용자가 외부 프로시저를 호출할 때, Altibase가 외부 프로시저용 에이전트 연결을 재시도하는 횟수를 설정한다.

Altibase는 에이전트 프로세스에 외부 에이전트가 있는지 확인한 후 사용할 수 있다. 그러나 사용하려는 외부 에이전트가 EXTPROC\_AGENT\_CONNECT\_TIMEOUT 프로퍼티 값에 의해 타임아웃 되어 종료될 수 있다. 그러면 이 프로퍼티에서 설정한 횟수만큼 외부 에이전트에 연결을 재시도한다.

Altibase 운영 중에 ALTER SYSTEM 구문으로 이 프로퍼티 값을 변경할 수 있다.

**EXTPROC\_AGENT\_IDLE\_TIMEOUT****데이터 타입**

Unsigned Integer

**기본값**

300 초

**속성**

읽기 전용, 단일 값

**값의 범위**[5,  $2^{32}-1$ ]**설명**

외부 프로시저용 에이전트 프로세스가 idle 상태로 대기하는 최대 시간을 초 단위로 설정한다. 사용자가 외부 프로시저를 호출했을 때 에이전트 프로세스가 idle 상태로 대기 중이라면, 에이전트 프로세스로의 연결 과정 없이 프로시저를 수행할 수 있어 응답 시간이 적게 걸릴 것이다. 그러나, 사용자의 외부 프로시저 호출이 없는데도 에이전트 프로세스가 종료하지 않고 대기하는 것은 자원 낭비가 될 수 있으므로 외부 프로시저의 호출 빈도가 높은 경우에만 대기 시간을 늘릴 것을 권장한다.

## EXTPROC\_AGENT\_SOCKET\_FILEPATH

### 데이터 타입

String

### 기본값

\$ALTIBASE\_HOME/trc

### 속성

읽기 전용, 단일 값

### 값의 범위

없음

### 설명

Altibase 서버가 외부 프로시저 에이전트(external procedure agent)와 통신하기 위해 생성하는 소켓 파일의 경로이다.

외부 프로시저를 사용하면, 세션이 외부 프로시저 에이전트를 생성하면서 socket\_sessionID로 소켓 파일을 생성한다. 임의로 이 파일을 삭제하지 않도록 주의해야 한다.

생성된 소켓 파일은 세션이 정상적으로 종료할 때 자동으로 삭제된다.

## 사용자 계정 보안 관련 프로퍼티

### CASE\_SENSITIVE\_PASSWORD

#### 데이터 타입

Unsigned Integer

#### 기본값

0

#### 속성

변경 가능, 단일 값

#### 값의 범위

[0, 1]

#### 설명

데이터베이스에서 사용자 암호의 대소문자를 구분할지 여부를 결정한다. 기본값은 대소문자를 구분하지 않는 0이며, 데이터베이스 내에서 사용자 암호가 대문자로 다뤄진다.

- 0: 대소문자 구분을 하지 않는다.
- 1: 대소문자 구분을 한다. 단, 사용자 생성 구문에서 암호를 큰따옴표(")로 묶은 경우에만 대소문자가 구분된다. 큰따옴표를 쓰지않은 암호는 대문자로 인식된다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

## FAILED\_LOGIN\_ATTEMPTS

### 데이터 타입

Unsigned Int

### 기본값

0

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 1000]

### 설명

로그인을 시도할 때 이 프로퍼티에 설정된 횟수 이상 실패하면, 해당 계정은 로그인에 불가능하다.

## PASSWORD\_LOCK\_TIME

### 데이터 타입

Unsigned Int

### 기본값

0

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 3650]

### 설명

계정이 한 번 잠긴 후 다시 풀리기 위해 경과되어야 하는 날짜(단위: 일)를 지정한다.

## PASSWORD\_LIFE\_TIME

### 데이터 타입

Unsigned Int

### 기본값

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 3650]

**설명**

계정의 비밀번호가 유효한 기간(단위: 일)을 지정한다.

**PASSWORD\_GRACE\_TIME****데이터 타입**

Unsigned Int

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 3650]

**설명**

계정의 비밀번호가 만료되는 시점 이후의 유예 기간(단위: 일)을 지정한다.

**PASSWORD\_REUSE\_TIME****데이터 타입**

Unsigned Int

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 3650]

**설명**

동일한 비밀번호를 재사용하기 위해 경과해야 하는 기간(단위:일)을 지정한다. 즉, 여기에 설정한 기간이 지난 후에 동일한 비밀번호를 재사용할 수 있다.



## PASSWORD\_REUSE\_MAX

### 데이터 타입

Unsigned Int

### 기본값

0

### 속성

읽기 전용, 단일 값

### 값의 범위

[0, 1000]

### 설명

동일한 패스워드를 재사용하기 위한 패스워드 변경 횟수를 지정한다. 즉, 여기에 설정한 횟수만큼 패스워드를 변경한 후에 동일한 패스워드를 재사용할 수 있다.

## PASSWORD\_VERIFY\_FUNCTION

### 데이터 타입

String

### 기본값

""

### 속성

읽기 전용, 단일 값

### 값의 범위

최대 길이 40바이트

### 설명

패스워드를 검증할 사용자 정의 콜백 함수(CALLBACK function)를 지정한다.

## 기타 프로퍼티

## ACCESS\_LIST

### 형식

ACCESS\_LIST = operation, address, mask, [limit]

### 값의 범위

- operation ::= [PERMIT|DENY]  
검사 규칙과 일치하는 IP 패킷의 접근을 허용할 것인지 제한할 것인지 기술한다.

- address

검사할 패킷의 IP 주소를 기술한다.

- mask

명시한 주소값이 IPv4 주소 형식이면, 이것은 서브넷 마스크로, 패킷의 IP 주소 중, 특정 부분만 검사하도록 설정한다.

명시한 주소값이 IPv6 주소 형식이면, 이것은 등록된 IPv6 주소들과 비교할 prefix비트의 길이를 나타낸다. 즉, 등록된 주소의 마스크 비트 길이에 해당하는 값이 접근하는 패킷의 IP 주소의 마스크 비트 길이에 해당하는 값과 일치한다면 접근이 허용된다.

- limit

선택 입력 항목으로 ACCESS\_LIST에 명시된 접속 가능한 IP 주소 영역에서 허용되는 최대 접속 세션 개수를 지정한다.

limit 값을 입력하면 모든 접속 요청에 대해 limit 조건 검사를 수행한다. 따라서, 허용된 IP의 접속 요청도 limit 개수를 초과하면 접속을 허용하지 않는다. 값을 입력하지 않으면 limit 조건을 검사하지 않는다.

운영 중 RELOAD ACCESS LIST로 ACCESS\_LIST를 추가하면, 기존에 연결된 세션은 영향을 받지 않으며, 변경 이후 새로운 연결 요청에 대해서만 ACCESS\_LIST 조건이 적용된다. 예를 들어 ACCESS\_LIST에 limit값을 설정 후 RELOAD ACCESS LIST 수행하면, 적용 이후 새로운 연결에 대해서만 limit 값이 적용된다. 이런 경우, V\$ACCESS\_LIST 조회시 Limit 값보다 CONNECTED 값이 더 클 수도 있다.

## 검사 규칙

```
IF
BITXOR(BITAND(IP_패킷,mask), BITAND(address,mask)) = 0
THEN 일치
ELSE 불일치
```

## 설명

Altibase에 접근하고자 하는 IP 패킷을 주소에 따라 접근을 제한하거나 허용할 수 있다. IP 패킷의 주소를 검사 규칙에 따라 검사하여, 일치하면 operation에 기술된 대로 허용 또는 제한하며 불일치하면 무시하고 다음 리스트를 검사한다.

IP 패킷의 주소를 여러 개가 지정될 경우 기술된 순서대로 검사한다. 일치하는 조건이 없을 경우, 접근이 허용된다.

만약 ACCESS\_LIST\_FILE 프로퍼티에 값을 기술하면 ACCESS\_LIST 프로퍼티에 정의된 목록 대신 외부 파일의 목록을 사용할 수 있다. 외부 파일은 최대 1024개의 항목까지 사용할 수 있으며, 'ACCESS\_LIST='는 생략하고 내용만 작성해야 한다.

## 예제

IP 주소가 192.168.1.55인 패킷만 접근을 제한하고 나머지는 허용한다.

```
ACCESS_LIST = deny, 192.168.1.55, 255.255.255.255
```

192.168.3.\*과 219.211.253.\* 주소들은 접근을 허용하고 나머지는 모두 제한한다.

```
ACCESS_LIST = permit, 192.168.3.0, 255.255.255.0
ACCESS_LIST = permit, 219.211.253.0, 255.255.255.0
ACCESS_LIST = deny ,0.0.0.0, 0.0.0.0
```

로컬 호스트를 제외한 모든 IPv4, IPv6 주소들의 접근을 제한한다.

```
ACCESS_LIST = deny, 0.0.0.0, 0.0.0.0
ACCESS_LIST = deny, ::1, 1
ACCESS_LIST = deny, fe80::, 1
```

외부 파일에 검사할 IP 주소를 기술한다. 192.168.3.\*과 fe80으로 시작하는 IPv6 주소를 제외한 모든 IP 주소의 접근을 제한한다.

```
permit, 192.168.3.0, 255.255.255.0
permit, fe80::, 16
deny, 0.0.0.0, 0.0.0.0
deny, ::1, 1
deny, fe80::, 1
```

192.168.3.17 주소의 세션 접근 허용 개수를 5개로 제한한다.

```
ACCESS_LIST = permit, 192.168.3.17, 255.255.255.255, 5
```

## ACCESS\_LIST\_FILE

### 데이터 타입

String

### 기본값

없음

### 속성

읽기 전용, 단일 값

### 값의 범위

없음

### 설명

ACCESS\_LIST가 외부 파일 경로를 참조할 때 설정한다. 파일명과 경로를 정확히 설정하지 않은 경우 서버를 구동할 수 없다. 외부 파일 경로는 절대경로를 지정해야 한다. 파일에 ACCESS\_LIST를 기술할 때의 형식은 ACCESS\_LIST 프로퍼티의 설명을 참조한다.

이 프로퍼티를 사용하지 않을 때에는 ACCESS\_LIST 프로퍼티에 기술된 목록을 사용한다.

**ADMIN\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

이 프로퍼티는 관리자 모드로 접근하는 것만 허용한다.

0: OFF

1: ON

이 값을 1로 설정하면 관리자 모드로 활성화되어 SYS 또는 SYSTEM\_ 사용자가 SYSDBA 옵션으로 서버와 연결을 맺어 작업을 할 수 있고 그 외 일반 사용자들은 연결 자체가 실패한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**ARITHMETIC\_OPERATION\_MODE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

Altibase 서버의 산술 연산 모드를 설정하는 프로퍼티이다.

0: 서버가 정밀도 우선 산술 연산 모드로 동작한다. 이 모드에서는 서버가 FLOAT 또는 NUMERIC 타입을 주로 사용하여 산술 연산의 오차를 줄인다. 단, "성능 우선 산술 연산 모드"에 비해 처리 속도가 떨어질 수 있다.

1: 서버가 성능 우선 산술 연산 모드로 동작한다. 이 모드에서는 서버가 산술 연산 시 DOUBLE 타입을 주로 사용하여 성능을 높이지만, 상대적으로 오차가 발생할 수 있다.

**CHECK\_MUTEX\_DURATION\_TIME\_ENABLE****데이터 타입**

Unsigned Integer

**기본 값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

MUTEX\_DURATION\_TIME을 확인할 것인지 여부를 설정한다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: check disable

1: check enable

**COERCE\_HOST\_VAR\_IN\_SELECT\_LIST\_TO\_VARCHAR****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 32000]

**설명**

이 프로퍼티는 select target list 절에 CAST 연산자 없이 호스트 변수를 사용할 수 있도록 해 준다.

이 프로퍼티를 1 이상의 값으로 설정하면, CAST 연산자 없이 사용된 호스트 변수에 대해서 Altibase 서버가 임의로 VARCHAR 타입으로 처리한다. 또한 설정된 값이 VARCHAR 타입의 크기(precision)가 된다.

이 프로퍼티의 값을 0으로 설정하면, CAST 연산자 없이 호스트 변수를 사용할 경우 에러가 발생한다.

## DEFAULT\_DATE\_FORMAT

### 데이터 타입

String

### 기본값

DD-MON-RRRR

### 속성

읽기 전용, 단일 값

### 값의 범위

없음

### 설명

테이블의 컬럼 도메인 중 DATE 타입 데이터의 기본 형식을 지정한다. 이 타입은 날짜, 시간을 저장할 수 있는 형식으로 제공되어야 한다. 형식은 “DD MON RRRR” 과 같이 따옴표 내에 공백도 사용할 수 있다.

```
DEFAULT_DATE_FORMAT = YYYY/MM/DD
iSQL> select sysdate from dual;
SYSDATE
-----
2008/06/16
1 row selected.
```

## EXEC\_DDL\_DISABLE

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

일반적으로 초기 데이터베이스를 구축한 이후에는 DML구문을 훨씬 더 빈번하게 수행하며 DDL 구문의 수행은 기존 데이터베이스 스키마를 변경시키는 작업이므로 상당한 주의를 요한다.

따라서 Altibase 운영 중 DDL구문을 수행하지 못하도록 운영자가 설정할 수 있으며 이 프로퍼티의 값을 1로 설정하면 Altibase 운영 중 DDL구문을 수행할 수 없으며 0인 경우 DDL구문을 수행할 수 있다.

Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**GROUP\_CONCAT\_PRECISION****데이터 타입**

Unsigned Integer

**기본값**

4000

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 32000]

**설명**

이 프로퍼티는 GROUP\_CONCAT 함수가 반환하는 VARCHAR 타입의 크기를 지정한다.

Altibase 운영 중에 ALTER SYSTEM 구문으로 이 프로퍼티의 값을 변경할 수 있다.

**JOB\_SCHEDULER\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

작업 스케줄러의 동작을 제어한다. 이 프로퍼티의 값을 1로 설정하여도 JOB\_THREAD\_COUNT 프로퍼티가 0이면 작업 스케줄러가 동작하지 않는다.

0: JOB스케줄러가 JOB 실행을 종료한다.

1: JOB스케줄러가 JOB 실행을 시작한다.

Altibase 운영 중에 ALTER SYSTEM 구문으로 이 프로퍼티의 값을 변경할 수 있다.

**JOB\_THREAD\_COUNT****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

읽기 전용, 단일 값

**값의 범위**

[0, 128]

**설명**

이 프로퍼티는 JOB을 실행하기 위한 스레드를 서버 구동 시 몇 개 생성할 것인지를 지정한다. 이 값이 0일 경우 작업 스케줄러(JOB Scheduler)를 위한 스레드가 생성되지 않는다.

**JOB\_THREAD\_QUEUE\_SIZE****데이터 타입**

Unsigned Integer

**기본값**

64

**속성**

읽기 전용, 단일 값

**값의 범위**

[64, 1024]

**설명**

이 프로퍼티는 JOB을 실행하기 위한 큐(Queue)를 서버 구동 시에 얼마나 생성할 것인지를 지정한다. 이 값이 크면, 동일 시간에 더 많은 JOB을 실행할 수 있다.

**LISTAGG\_PRECISION (단위: 바이트)****데이터 타입**

Unsigned Integer

**기본값**

4000



**속성**

변경 가능, 단일 값

**값의 범위**

[0, 32000]

**설명**

LISTAGG 함수가 반환하는 VARCHAR 타입의 크기를 지정한다. Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**MSG\_QUEUE\_PERMISSION****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

메시지 큐의 접근 권한(permission type)을 설정한다. Altibase 운영 중 ALTER SYSTEM 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

0: (rw-rw-rw 0666) - 모든 사용자는 읽기와 쓰기만 할 수 있으며 실행하기는 할 수 없다.

1: (rw-r-r 0644) - 소유자는 읽기와 쓰기만 할 수 있으며, 그 외의 사용자는 읽기만 할 수 있다.

**PARALLEL\_QUERY\_QUEUE\_SIZE****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

변경 가능, 단일 값

**값의 범위**

[4, 1048576]

**설명**

병렬 큐(Parallel Queue)가 담을 수 있는 최대 로우(row) 수를 지정한다.

**PARALLEL\_QUERY\_THREAD\_MAX****데이터 타입**

Unsigned Integer

**기본값**

논리 코어 수

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1024]

**설명**

병렬 쿼리에서 사용하는 최대 스레드 개수를 지정한다.

**PSM\_CASE\_SENSITIVE\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

PSM 내에서 RECORD 타입과 ROWTYPE 타입의 칼럼 이름을 참조하거나 LABEL의 이름을 참조할 때, 대소문자를 구별할지 여부를 설정하는 프로퍼티 이다. 0으로 설정한 경우 대소문자를 구별하지 않고 동작하고, 1로 설정한 경우 대소문자를 구별하여 동작한다.

Altibase 운영 중에 ALTER SYSTEM 구문으로 이 프로퍼티의 값을 변경할 수 있다.

## PSM\_CHAR\_DEFAULT\_PRECISION

### 데이터 타입

Unsigned Integer

### 기본값

32767

### 속성

읽기 전용, 단일 값

### 값의 범위

[1, 65534]

### 설명

아래와 같은 경우에 CHAR 타입의 크기를 명시하지 않으면 Altibase는 PSM\_CHAR\_DEFAULT\_PRECISION 프로퍼티에 설정된 값을 CHAR크기로 결정한다.

- 저장 프로시저 생성시 데이터 타입이 CHAR인 파라미터의 크기를 명시하지 않을 때
- 저장 함수 생성시 데이터 타입이 CHAR인 파라미터 또는 반환 값의 크기를 명시하지 않을 때

## PSM\_IGNORE\_NO\_DATA\_FOUND\_ERROR

### 데이터 타입

Unsigned Integer

### 기본값

0

### 속성

변경 가능, 단일 값

### 값의 범위

[0, 1]

### 설명

Altibase의 PSM에서 제공하는 시스템-정의 예외 중 NO\_DATA\_FOUND가 있다. 이 예외는 PSM(저장 프로시저 및 함수, 트리거) 내에 포함된 "SELECT ~ INTO" 구문이 실행될 때 결과 집합이 없으면 발생한다.

PSM\_IGNORE\_NO\_DATA\_FOUND\_ERROR 프로퍼티를 사용하면 저장 함수에 대해서는 NO\_DATA\_FOUND 예외가 발생하지 않도록 할 수 있다.

- 0: 결과 집합이 없는 경우에 NO\_DATA\_FOUND 예외가 발생한다.
- 1: 결과 집합이 없는 경우에 NO\_DATA\_FOUND 예외가 발생하지 않는다.

Altibase 운영 중에 ALTER SYSTEM 구문으로 이 프로퍼티의 값을 변경할 수 있다.

## PSM\_NCHAR\_UTF16\_DEFAULT\_PRECISION

### 데이터 타입

Unsigned Integer

### 기본값

16383

### 속성

읽기 전용, 단일 값

### 값의 범위

[1, 32766]

### 설명

Altibase 캐릭터 셋(character set)이 UTF16이고 아래와 같은 경우에 NCHAR타입의 크기를 명시하지 않으면 Altibase는 PSM\_NCHAR\_UTF16\_DEFAULT\_PRECISION 프로퍼티에 설정된 값을 NCHAR크기로 결정한다.

- 저장 프로시저 생성시 데이터 타입이 NCHAR인 파라미터의 크기를 명시하지 않을 때
- 저장 함수 생성시 데이터 타입이 NCHAR인 파라미터 또는 반환 값의 크기를 명시하지 않을 때

## PSM\_NCHAR\_UTF8\_DEFAULT\_PRECISION

### 데이터 타입

Unsigned Integer

### 기본값

10921

### 속성

읽기 전용, 단일 값

### 값의 범위

[1, 21843]

### 설명

Altibase 캐릭터 셋(character set)이 UTF8이고 아래와 같은 경우에 NCHAR타입의 크기를 명시하지 않으면 Altibase는 PSM\_NCHAR\_UTF8\_DEFAULT\_PRECISION 프로퍼티에 설정된 값을 NCHAR크기로 결정한다.

- 저장 프로시저 생성시 데이터 타입이 NCHAR인 파라미터의 크기를 명시하지 않을 때
- 저장 함수 생성시 데이터 타입이 NCHAR인 파라미터 또는 반환 값의 크기를 명시하지 않을 때

## PSM\_NVARCHAR\_UTF16\_DEFAULT\_PRECISION

### 데이터 타입

Unsigned Integer

### 기본값

16383

### 속성

읽기 전용, 단일 값

### 값의 범위

[1, 32766]

### 설명

Altibase 캐릭터 셋(character set)이 UTF16이고 아래와 같은 경우에 NVARCHAR타입의 크기를 명시하지 않으면 Altibase는 PSM\_NVARCHAR\_UTF16\_DEFAULT\_PRECISION 프로퍼티에 설정된 값을 NVARCHAR크기로 결정한다.

- 저장 프로시저 생성시 데이터 타입이 NVARCHAR인 파라미터의 크기를 명시하지 않을 때
- 저장 함수 생성시 데이터 타입이 NVARCHAR인 파라미터 또는 반환 값의 크기를 명시하지 않을 때

## PSM\_NVARCHAR\_UTF8\_DEFAULT\_PRECISION

### 데이터 타입

Unsigned Integer

### 기본값

10921

### 속성

읽기 전용, 단일 값

### 값의 범위

[1, 21843]

### 설명

Altibase 캐릭터 셋(character set)이 UTF8이고 아래와 같은 경우에 NVARCHAR타입의 크기를 명시하지 않으면 Altibase는 PSM\_NVARCHAR\_UTF8\_DEFAULT\_PRECISION 프로퍼티에 설정된 값을 NVARCHAR크기로 결정한다.

- 저장 프로시저 생성시 데이터 타입이 NVARCHAR인 파라미터의 크기를 명시하지 않을 때
- 저장 함수 생성시 데이터 타입이 NVARCHAR인 파라미터 또는 반환 값의 크기를 명시하지 않을 때

**PSM\_PARAM\_AND\_RETURN\_WITHOUT\_PRECISION\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

읽기 전용, 단일 값

**값의 범위**

[0,1]

**설명**

저장 프로시저를 생성할 때 파라미터의 크기를 정의하지 않았거나 저장 함수를 생성할 때 파라미터 및 반환 값의 데이터 크기를 정의하지 않았다면, 이 프로퍼티 값에 따라 사용할 수 있는 데이터 크기가 달라진다.

파라미터 또는 반환 값의 타입이 CHAR, NCHAR, NVARCHAR, VARCHAR이고 이 프로퍼티의 값이 1이면, 데이터 타입의 크기는 아래의 프로퍼티에 설정한 값으로 결정된다. 그러나 이 값이 0으로 설정하면 데이터 타입의 크기는 1이다.

- PSM\_CHAR\_DEFAULT\_PRECISION
- PSM\_NCHAR\_UTF8\_DEFAULT\_PRECISION
- PSM\_NCHAR\_UTF16\_DEFAULT\_PRECISION
- PSM\_NVARCHAR\_UTF8\_DEFAULT\_PRECISION
- PSM\_NVARCHAR\_UTF16\_DEFAULT\_PRECISION
- PSM\_PARAM\_AND\_RETURN\_WITHOUT\_PRECISION\_ENABLE
- PSM\_VARCHAR\_DEFAULT\_PRECISION

**PSM\_VARCHAR\_DEFAULT\_PRECISION****데이터 타입**

Unsigned Integer

**기본값**

32767

**속성**

읽기 전용, 단일 값

**값의 범위**

[1, 65534]

**설명**

아래와 같은 경우에 VARCHAR 타입의 크기를 명시하지 않으면 Altibase는 PSM\_VARCHAR\_DEFAULT\_PRECISION 프로퍼티에 설정된 값을 VARCHAR의 크기로 결정한다.

- 저장 프로시저 생성시 데이터 타입이 VARCHAR인 파라미터의 크기를 명시하지 않을 때
- 저장 함수 생성시 데이터 타입이 VARCHAR인 파라미터 또는 반환 값의 크기를 명시하지 않을 때

**QUERY\_STACK\_SIZE (단위: 개수)****데이터 타입**

Unsigned Integer

**기본값**

1024

**속성**

변경 가능, 단일 값

**값의 범위**

[8, 65536]

**설명**

질의 수행 시 연산 및 비교 등의 연산자를 처리하기 위해 시스템 내부적으로 사용하는 스택의 크기를 설정하는 시스템 프로퍼티이다.

복잡한 연산식 또는 저장프로시저와 같이 많은 구문이 사용될 경우 stack overflow 오류가 날 수 있고 이 때 프로퍼티 값을 큰 값으로 변경해야 한다. 응용 프로그램 환경에 따라 적절한 값을 설정해야 하며 필요 이상 큰 값으로 설정할 경우 불필요한 메모리 공간 낭비가 될 수 있으므로 유의해야 한다.

이 프로퍼티는 altibase.properties 파일내에 명시할 수 있으며 ALTER SYSTEM 또는 ALTER SESSION 명령문으로 변경할 수 있다.

ALTER SESSION문으로 변경하는 경우 다음과 같이 값을 변경할 수 있다.

ALTER SESSION SET STACK SIZE = *n*;

**RECURSION\_LEVEL\_MAXIMUM****데이터 타입**

Unsigned Integer

**기본값**

1000

**속성**

변경 가능, 단일 값

**값의 범위**[5, 2<sup>32</sup>-1]**설명**

이 프로퍼티에 설정한 횟수(level) 만큼 재귀 질의를 반복하여 수행한다.

Altibase 운영 중 ALTER SESSION 문을 이용하여 이 프로퍼티의 값을 변경할 수 있다.

**REGEXP\_MODE****데이터 타입**

Unsigned Integer

**기본값**

0

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

정규 표현식 모드를 설정하는 프로퍼티로, 설정 값의 의미는 아래와 같다.

**0**

Altibase 정규 표현식 모드.

POSIX Basic Regular Expression (BRE)과 Extended Regular Expression(ERE)을 일부 지원한다.

**1**

PCRE2 호환 모드.

펄 호환 정규 표현식 (Perl Compatible Regular Expressions, PCRE2) 라이브러리의 정규 표현식 문법을 지원한다.

이 모드는 Altibase 서버 캐릭터셋이 US7ASCII 또는 UTF-8일 때 사용할 수 있으며 Altibase 정규 표현식 모드와 문법 차이가 있다. 관련 설명은 [SQL Reference-A.부록: 정규 표현식](#) 매뉴얼을 참고한다.

**REMOTE\_SYSDBA\_ENABLE****데이터 타입**

Unsigned Integer

**기본값**

1



**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

SYS 사용자가 원격에서 SYSDBA 모드로 접속할 수 있는지 여부를 설정한다. ALTER SYSTEM 명령문으로 값을 변경할 수 있다.

0: 원격에서 SYSDBA 모드로 접속 불가

1: 원격에서 SYSDBA 모드로 접속 가능 (기본값)

**SELECT\_HEADER\_DISPLAY****데이터 타입**

Unsigned Integer

**기본값**

1

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 1]

**설명**

SELECT 쿼리의 결과 출력 시 iSQL 상에 칼럼 이름만 출력할 것인지, 테이블 이름과 함께 칼럼 이름을 출력할 것인지를 설정하는 시스템 프로퍼티이다.

이 프로퍼티는 altibase.properties 파일에 명시할 수 있으며 ALTER SYSTEM 또는 ALTER SESSION 명령문으로 변경할 수 있다. 값이 0인 경우 iSQL 상에서 결과 출력 시 테이블 이름과 칼럼 이름이 함께 출력된다.

**SYS\_CONNECT\_BY\_PATH\_PRECISION****데이터 타입**

Unsigned Integer

**기본값**

4000

**속성**

변경 가능, 단일 값

**값의 범위**

[0, 32000]

**설명**

이 프로퍼티는 SYS\_CONNECT\_BY\_PATH\_PRECISION 함수가 반환하는 VARCHAR 타입의 크기를 지정한다.

Altibase 운영 중에 ALTER SYSTEM 구문으로 이 프로퍼티의 값을 변경할 수 있다.