

# Altibase 7.1.0.9.8 Patch Notes

---

## Table of Contents

---

- [New Features](#)
  - [BUG-50912 TCP 소켓 옵션 SO\\_LINGER를 설정하는 연결 속성 socket\\_immediate\\_close를 추가합니다.](#)
- [Fixed Bugs](#)
  - [BUG-50948 서버에 fetch할 데이터가 남아 있는 경우 ResultSet.close\(\)를 해도 커서가 바로 닫히지 않을 수 있습니다.](#)
  - [BUG-50969 altibase stmt bind\\_param 함수의 두 번째 인자의 메모리 주소가 변경되면 Some parameters were not bound. 에러가 발생합니다.](#)
  - [BUG-50975 SIMPLE QUERY 최적화 기능을 활성화하고 JDBC 연결 속성에 remove\\_redundant\\_transmission을 사용할 때, SQL 문 수행 중 메모리 오류가 발생할 수 있습니다.](#)
- [Changes](#)
  - [Version Info](#)
  - [호환성](#)
  - [프로퍼티](#)
  - [성능 분](#)

## New Features

---

**BUG-50912 TCP 소켓 옵션 SO\_LINGER를 설정하는 연결 속성 socket\_immediate\_close를 추가합니다.**

**module**

mm-jdbc

**Category**

Functionality

**재현 빈도**

Always

## 설명

TCP 소켓 옵션인 SO\_LINGER의 활성화 여부를 설정하는 연결 속성 `socket_immediate_close`를 추가합니다.

- `true`: SO\_LINGER 값을 0으로 설정합니다. 소켓을 닫는 즉시 연결이 종료되고 남아있는 데이터는 전송되지 않습니다.
- `false`: SO\_LINGER를 비활성화합니다. 소켓은 바로 닫히지만, 소켓 버퍼에 남아있는 데이터가 있다면 커널은 일정 시간 동안 데이터를 보내려고 시도합니다.

기본값은 `false`이며, 연결 속성에 관한 설명은 [JDBC User's Manual](#)에서 확인할 수 있습니다.

이 연결 속성은 Altibase JDBC 드라이버 7.1.0.9.8 이상부터 지원합니다.

## 재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

## Workaround

### 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## Fixed Bugs

---

**BUG-50948 서버에 fetch할 데이터가 남아 있는 경우  
ResultSet.close()를 해도 커서가 바로 닫히지 않을 수 있습니다.**

### module

mm-jdbc

### Category

Functionality

### 재현 빈도

Rare

## 설명

ResultSet 객체에서 조회할 데이터가 Altibase 서버에 남아 있을 때, ResultSet.close()를 호출해도 커서가 닫히지 않는 문제를 수정하였습니다. 이 문제가 발생하면 Altibase 서버 설정에 의해 FETCH\_TIMEOUT 오류가 발생할 수 있습니다.

이 버그가 수정된 버전을 반영하려면 Altibase JDBC 드라이버를 7.1.0.7.8 이상 버전으로 패치해야 합니다.

## 재현 방법

- 재현 절차

```
isQL> create table t1 (c1 int);
isQL> insert into t1 select level from dual connect by level <= 1000;

[altibase.properties]
FETCH_TIMEOUT = 5

[Altibase 클라이언트]
sStmt.setFetchSize(1);
ResultSet sRs = sStmt.executeQuery("SELECT * FROM t1");
if (sRs.next())
{
    System.out.println("col1 ==>" + sRs.getInt(1));
}
sRs.close();
Thread.sleep(7000);
sRs = sStmt.executeQuery("SELECT 1 FROM dual");
if (sRs.next())
{
    System.out.println(sRs.getString(1));
}
```

- 수행 결과

```
[Altibase 클라이언트]
Exception in thread "main" java.sql.SQLException: Communication link failure:
There was no response from the server, and the channel has reached end-of-
stream.

[Altibase 서버]
[2024/06/03 09:51:01 4EF][PID:26603][Thread-139699788171328][LWP-26592]
[Notify : Fetch Timeout] Session Closed by Server : Session ID = 2
CLIENT_INFO          => TCP 192.168.1.48:54096(PID : 2065530879)
Time Limit            => 5
Running Time          => 6
Transaction ID        => 5152
Caused by Query       => SELECT * FROM t1
```

- 예상 결과

```
1
```

## Workaround

### 변경사항

- Performance view
- Property
- Compile Option
- Error Code

# BUG-50969 altibase\_stmt\_bind\_param 함수의 두 번째 인자의 메모리 주소가 변경되면 Some parameters were not bound. 에러가 발생합니다.

## module

ux-cdbc

## Category

Functional Error

## 재현 빈도

Always

## 설명

altibase\_stmt\_bind\_param 함수의 두 번째 인자의 메모리 주소가 변경되면 Some parameters were not bound. 에러가 발생하거나

클라이언트가 비정상 종료하는 문제를 수정합니다.

이 버그는 altibase\_stmt\_bind\_param 함수와 altibase\_stmt\_execute 함수를 반복적으로 수행할 때 발생합니다.

이 버그를 반영하려면 ACI 라이브러리(libalticapi.a)를 패치해야 합니다.

## 재현 방법

- 재현 절차

```
src = altibase_stmt_prepare(sStmt, "INSERT INTO bug50969 VALUES(?, ?)");
...중략...
memset(sBind1, 0, sizeof(sBind1));
...중략....
src = altibase_stmt_bind_param(sStmt, sBind1);
...중략....
sCount++;
sBind1Count = 11;
if (altibase_stmt_execute(sStmt) != ALTIBASE_SUCCESS)
{
    PRINT_STMT_ERROR(sStmt);
    goto end_stmt;
}
altibase_stmt_free_result(sStmt);
memset(sBind2, 0, sizeof(sBind2));
...중략....
src = altibase_stmt_bind_param(sStmt, sBind2);
...중략....
sCount++;
sBind2Count = 12;
if (altibase_stmt_execute(sStmt) != ALTIBASE_SUCCESS)
{
    PRINT_STMT_ERROR(sStmt);
    goto end_stmt;
```

```
}
```

- 수행 결과

```
[51051] HY000 Some parameters were not bound.
```

- 예상 결과

```
SELECT * FROM bug50969;  
C1      C2  
-----  
11      1  
12      2  
2 rows selected.
```

## Workaround

### 변경사항

- Performance view
- Property
- Compile Option
- Error Code

**BUG-50975 SIMPLE QUERY 최적화 기능을 활성화하고 JDBC 연결 속성에 `remove_redundant_transmission`을 사용할 때, SQL 문 수행 중 메모리 오류가 발생할 수 있습니다.**

### module

mm

### Category

Memory Error

### 재현 빈도

Always

### 설명

"SIMPLE QUERY 최적화" 기능을 활성화하고 JDBC 연결 속성에 `remove_redundant_transmission`을 사용할 때 SQL 문 수행 중 발생하는 메모리 오류를 수정하였습니다.

이 버그는 아래 조건을 만족할 때 Altibase 서버가 비정상 종료할 수 있습니다.

- Altibase 서버 프로퍼티 `EXECUTOR_FAST_SIMPLE_QUERY = 1` 설정
- JDBC 연결 속성 `remove_redundant_transmission`을 사용
- SQL 문이 SIMPLE QUERY 최적화 기능이 적용되어 실행

## 재현 방법

- 재현 절차

```
[Altibase 서버]
ALTER SYSTEM SET EXECUTOR_FAST_SIMPLE_QUERY = 1;
CREATE TABLE BUG
(
  C1          INTEGER          NOT NULL,
  C2          INTEGER          NOT NULL,
  C3          DECIMAL(12,2),
  C4          DECIMAL(4,4),
  C5          INTEGER,
  C6          VARCHAR(10),
  C7          VARCHAR(20),
  C8          VARCHAR(20),
  C9          VARCHAR(20),
  C10         CHAR(2),
  C11         CHAR(9)
);

[Altibase 클라이언트]
String sURL      = "jdbc:Altibase://127.0.0.1:" + sPort + "/mydb" +
                  "?remove_redundant_transmission=1";

...중략...
/* Initialize environment */
try
{
  sCon = DriverManager.getConnection( sURL, sProps );
  sStmt = sCon.createStatement();
  sRS = sStmt.executeQuery( "SELECT C6, C7, C8, C9, C10, C11 FROM
BUG50975");
  sRowCount = 0;
  /* Fetch all data */
  while( sRS.next() )
  {
    sRowCount++;
  }
  System.out.println(sRowCount + " rows selected");
  /* Finalize process */
  sStmt.close();
  sCon.close();
}
}
```

- 수행 결과

Altibase 서버 비정상 종료

- 예상 결과

SQL 문 정상 수행

# Workaround

## 변경사항

- Performance view
- Property
- Compile Option
- Error Code

## Changes

### Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.9.8	6.5.1	8.11.1	7.1.7	7.4.7

Altibase 7.1 패치 버전별 히스토리는 [Version Histories](#) 에서 확인할 수 있다.

## 호환성

### Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그파일의 호환성을 나타낸다. 이 버전이 다른 경우의 패치(업그레이드 포함)는 데이터베이스를 재구성해야 한다.

### Meta Version

메타 버전은 변경되지 않았다.

패치를 롤백하려는 경우, [메타다운그레이드](#)를 참고한다.

### CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

### Replication protocol Version

Replication 프로토콜 버전은 변경되지 않았다.

## 프로퍼티

### 추가된 프로퍼티

### 변경된 프로퍼티

### 삭제된 프로퍼티

## 성능 뷰

추가된 성능 부

변경된 성능 부

삭제된 성능 부