

# Altibase 7.1.0.7.3 Patch Notes

---

# Table of Contents

---

- [New Features](#)
  - [BUG-49668 Increase the maximum number of concurrent disk transactions \(the maximum value of the TRANSACTION\\_SEGMENT\\_COUNT property\) from 512 to 16384.](#)
- [Fixed Bugs](#)
  - [BUG-48065 If a DROP SEQUENCE statement and a SELECT statement are executed at the same time, the Altibase server may abnormally terminate.](#)
  - [BUG-49505 Exception handling is added when an execution plan cannot be registered in the SQL Plan Cache due to exceeding PREPARE\\_STMT\\_MEMORY\\_MAXIMUM when executing a high-complexity SQL statement.](#)
  - [BUG-49569 Adds exception handling when a LOB, GEOMETRY, or binary data type is used in the select list clause of a SELECT statement using the UNION ALL operator.](#)
  - [BUG-49580 Improves excessive memory usage when entering LOB data into memory tables or volatile memory tables.](#)
  - [BUG-49649 If an \[Unable to find record in ~\] error occurs while performing replication transactions including LOB columns, the Receiver is stopped.](#)
  - [BUG-49654 \[ERR-31001 SQL syntax error\] error occurs when replication is started after creating a replication object for a table with CHECK constraints including strings.](#)
  - [BUG-49661 The \[ERR-11069: Internal server error in the storage manager \(\[FAILURE\]. ERR-110C1 \(error=11\) The data file containing page \[129832016\] does not exist.\] error occurs in certain situations when performing SQL statements using disk temporary tablespace.](#)
- [Changes](#)
  - [Version Info](#)
  - [Altibase Server Properties](#)
  - [Performance Views](#)

# New Features

---

## BUG-49668 Increase the maximum number of concurrent disk transactions (the maximum value of the TRANSACTION\_SEGMENT\_COUNT property) from 512 to 16384.

- **module** : sm\_recovery
- **Category** : Usability
- **Reproducibility** : Always
- **Description** : Increase the maximum number of concurrent disk transactions (the maximum value of the TRANSACTION\_SEGMENT\_COUNT property) from 512 to 16384.

The maximum value and properties of the associated property, TRANSACTION\_SEGMENT\_COUNT, have been changed.

- Max changed from 512 to 16384
- The attribute of the property has been changed from 'mutable' to 'read-only'.

If the setting value of the TRANSACTION\_SEGMENT\_COUNT property exceeds 900, a separate file (txSegEntry.hdr) is created that stores segment header information in the undo tablespace. The location of this file is the same as the datafiles in the undo tablespace and cannot be changed.

- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
    - TRANSACTION\_SEGMENT\_COUNT : Max increased from 900 to 16384
  - Compile Option
  - Error Code

# Fixed Bugs

## BUG-48065 If a DROP SEQUENCE statement and a SELECT statement are executed at the same time, the Altibase server may abnormally terminate.

- **module** : sm
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Fixes a phenomenon in which the Altibase server abnormally terminates when a DROP SEQUENCE statement and a SELECT statement are executed at the same time.

If any of the conditions below are true, you are exposed to this bug.

- Altibase 7.1.0.1.2 to 7.1.0.7.0
- If you add `__CATALOG_SLOT_REUSABLE = 1` to `altibase.properties`
- **How to reproduce this bug**
  - **Reproduction conditions**

```
CREATE SEQUENCE S1;
-- SESSION 1
declare
  cursor c1 is select s1.nextval from dual connect by level < 5;
  var1 integer;
begin
  open c1;
  loop
    fetch c1 into var1;
    exit when c1%notfound;
    println(var1);
    sleep(10);
  end loop;
end;
/
-- SESSION 2
DROP SEQUENCE S1;
```

- **Actual Results**

```
1234
Execute success.
```

- **Expected Results**

```
1
[ERR-1107C : The sequence has already been dropped.at
"SYS.ANONYMOUS_BLOCK", line 7]
```

- **Workaround**
- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## **BUG-49505 Exception handling is added when an execution plan cannot be registered in the SQL Plan Cache due to exceeding PREPARE\_STMT\_MEMORY\_MAXIMUM when executing a high-complexity SQL statement.**

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Fixes a bug in which the Altibase server abnormally terminates when an execution plan cannot be registered in the SQL Plan Cache due to exceeding PREPARE\_STMT\_MEMORY\_MAXIMUM when executing a high-complexity SQL statement.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
DROP TABLE BUG-49505;

CREATE TABLE BUG-49505
(
    I1  INTEGER,
    I2  INTEGER,
    I3  INTEGER,
    I4  INTEGER,
    I5  INTEGER,
    I6  INTEGER,
    I7  INTEGER,
    I8  INTEGER,
    I9  INTEGER,
    I10 INTEGER,
    I11 INTEGER,
    I12 GEOMETRY
)
PARTITION BY RANGE( I1 )
(
    PARTITION P1 VALUES LESS THAN ( 10 ),
    PARTITION P2 VALUES LESS THAN ( 20 ),
    PARTITION P3 VALUES LESS THAN ( 30 ),
    PARTITION P4 VALUES DEFAULT
);

ALTER TABLE BUG-49505 MERGE PARTITIONS P2, P3 INTO PARTITION P3 ;
ALTER TABLE BUG-49505 DROP PARTITION P1;
```

[illegible]

```
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 UNION ALL SELECT * FROM BUG-49505 UNION ALL SELECT * FROM BUG-
49505 );
```

#### ◦ Actual Results

Altibase server abnormally terminates

#### ◦ Expected Results

When performing the same query for the first time

```
ERR-01067 : The memory size allocated for the statement has exceeded the
maximum limit ( Name : Query_Prepere, Wanted Memory Size : 209719025,
Max size : 209715200 ).
```

When executing the same query again

```
COUNT(*)
-----
0
1 row selected.
```

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## **BUG-49569 Adds exception handling when a LOB, GEOMETRY, or binary data type is used in the select\_list clause of a SELECT statement using the UNION ALL operator.**

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : In a SELECT statement using the UNION ALL operator, when two of the LOB, GEOMETRY, NIBBLE, BYTE, and VARBYTE data types are used together in the select\_list clause, exception handling has been added in situations where the Altibase server abnormally terminates or waits indefinitely.

[ERR-91022 : LOB and GEOMETRY type data cannot be displayed] or [ERR-21043 : Unexpected errors may have occurred: mtvCalculate\_Blob2BlobLocator: not BLOB module] errors will occur in versions that reflect this bug.

- **How to reproduce this bug**

- **Reproduction conditions**

```
DROP TABLE t1 CASCADE;  
DROP TABLE t2 CASCADE;  
CREATE TABLE t1( i1 BLOB );  
CREATE TABLE t2 ( i1 BYTE(12) );  
INSERT INTO t2 VALUES (1);  
SELECT i1 FROM t1 UNION ALL SELECT i1 FROM t2;
```

- **Actual Results**

```
[ERR-91015 : Communication failure.]
```

- **Expected Results**

```
[ERR-91022 : LOB and GEOMETRY type data cannot be displayed]
```

- **Workaround**

- **Changes**

- Performance view
  - Property
  - Compile Option
  - Error Code

## **BUG-49580 Improves excessive memory usage when entering LOB data into memory tables or volatile memory tables.**

- **module** : sm-mem-resource
- **Category** : Other
- **Reproducibility** : Always
- **Description** : When LOB data is input to a memory table or volatile memory table, the runtime object that manages LOB information uses excessive memory.
  - Example of this bug phenomenon  
196G of memory is used when inputting 4G-sized LOB data. Excessive use of memory during INSERT and returns after transaction ends.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**



- Performance view
- Property
- Compile Option
- Error Code

## BUG-49649 If an [Unable to find record in ~] error occurs while performing replication transactions including LOB columns, the Receiver is stopped.

- **module** : rp-receiver
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : If an Unable to find record in ~ error occurs when performing replication transactions including LOB columns, modify the receiver so that it does not stop and handles data collisions normally.

This is a phenomenon that can be confirmed by the receiver and sender when this bug occurs.

### ○ receiver side

The following error remains in the trace log.

#### ■ altibase\_rp\_conflict.log

```
ERR-610f7(errno=62) [Receiver] Unable to find record in ...
```

#### ■ altibase\_rp.log

```
ERR-610f7(errno=62) [Receiver] Unable to find record in
openLOBCursor() function
ERR-61048(errno=62) [Receiver] replication_name receiver has
recvXLog error in run()
[Receiver] RepName:replication_name is processed at ~
[Receiver] RepName:replication_name is received at ~
ERR-6104b(errno=62) [Receiver] replication_name receiver is ended(by
thr_exit)
Error Stop!
[Receiver] Replication replication_name Stopped ...
```

### ○ sender side

On the sender side, the restart is repeated according to the REPLICATION\_SENDER\_SLEEP\_TIMEOUT property setting, but on the receiver side, the receiver does not start normally with an Unable to find record in openLOBCursor() function error. This causes a replication gap.

The following error remains in the trace log.

#### ■ altibase\_rp.log

```
[SenderApply] Replication replication_name Start
[Sender] Replication replication_name:0 Start... at[93025988]
[SenderApply] Replication replication_name Got Error
ERR-620f0(errno=11) [Sender] Stop sender thread replication_name:0
at [93027296], Restart SN[93025988]
ERR-7101a(errno=16) Connection closed
ERR-61022(errno=11) [Sender] Sender Sleep : 60 seconds
```

In the version that reflects this bug, if a non-existent primary key value is executed when a replication transaction including LOB columns is performed, the following log is left in the trace log altibase\_rp\_conflict.log on the receiver side.

```
ERR-610f7(errno=16) [Receiver] Unable to find record in openLOBCursor()
function
A failure occurred while opening LOB Cursor (Table : ) (PK : ) ; (TID : )
COMMIT (TID : )
```

- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## **BUG-49654 [ERR-31001 SQL syntax error] error occurs when replication is started after creating a replication object for a table with CHECK constraints including strings.**

- **module** : rp-sender
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixes a phenomenon in which an [ERR-31001 SQL syntax error] error occurs when replication is started after creating a replication object for a table with CHECK constraints including strings.

The [ERR-31001 SQL syntax error] error caused by this bug can occur in the following three situations.

- When replication is started after creating a replication object for a table with CHECK constraints including strings

When executing ALTER REPLICATION replication\_name START, [ERR-6104F : [Sender] Failed to initialize] error occurs and [ERR-31001 (errno=11) SQL syntax error] occurs in altibase\_rp.log

- When adding a table with a CHECK constraint including a string to a replication object

[ERR-31001: SQL syntax error] error occurs when ALTER REPLICATION replication\_name ADD TABLE is executed

- When adding a CHECK constraint including a string to a replication target table

When ALTER TABLE table\_name ADD CONSTRAINT constraint\_name CHECK is executed, [ERR-31001 (errno=62) SQL syntax error] occurs in altibase\_rp.log and the replication sender stops.

When this situation occurs, the replication object must be deleted and recreated.

- **How to reproduce this bug**

- **Reproduction conditions**

```
CREATE TABLE RCSC_AUID
(
    TS TIMESTAMP,
    AUID VARCHAR(96) not null,
    AUID_USE SMALLINT default 1,
    XDMS VARCHAR(8),
    XDMS_USE SMALLINT default 1,
    PRIMARY KEY (AUID)
);

ALTER TABLE RCSC_AUID
ADD CONSTRAINT RCSC_RCSC_AUID_CHECK_01
CHECK( XDMS in ('GMS', 'CMS', 'IdMS', 'KMS', 'PROXY') );

ALTER REPLICATION REPl START;
```

- **Actual Results**

```
[ERR-6104F: [Sender] Failed to initialize] error occurred and [ERR-31001
(errno=11) SQL syntax error] occurred in altibase_rp.log
```

- **Expected Results**

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

**BUG-49661 The [ERR-11069: Internal server error in the storage manager ([FAILURE] ERR-110C1 (error=11) The data file containing page [129832016] does not exist.) error occurs in certain situations when performing SQL statements using disk temporary tablespace.**

- **module** : sm
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : The [ERR-11069: Internal server error in the storage manager ([FAILURE] ERR-110C1 (error=11) The data file containing page [129832016] does not exist.) error occurs in certain situations when performing SQL statements using disk temporary tablespace. This bug has been fixed.

This bug will occur with a very low probability if all of the conditions below are met.

- Altibase 7.1.0.5.7 or higher
- HASH operation performed during disk temp table operation
- Lack of HASH\_AREA\_SIZE to use disk temporary tablespace
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

# Changes

---

## Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.7.3	6.5.1	8.10.1	7.1.7	7.4.7

You can check the module version change history in [Version Histories](#).

## Compatibility

### Database binary version

The database binary version has not changed.

The database binary version indicates the compatibility of database image files and log files.  
If this version needs to be patched to a different version, the database must be reorganized.

### Meta Version

The meta version has not changed.

If you want to roll back the patch after patching to a version with a changed meta version,  
see [Meta Downgrade](#).

### CM protocol Version

The cm protocol version has not changed.

### Replication protocol Version

The replication protocol version has not changed.

## Altibase Server Properties

### Added Properties

### Changed Properties

### Deleted Properties

### Performance Views

### Added Performance Views

### Changed Performance Views

**Deleted Performance Views**