

# Altibase 7.1.0.6.9 Patch Notes

---

# Table of Contents

---

- [Fixed Bugs](#)
  - [BUG-47420 Fixed the issue where the Altibase server abnormally terminates when using LENGTH\(\) to output the length of LOB data and the LOB data type exceeds the maximum size when input into a disk table.](#)
  - [BUG-49133 Setting SORT\\_AREA\\_SIZE to its minimum value of 524288 and setting TEMP\\_HASH\\_BUCKET\\_DENSITY to 65 or higher may cause the Altibase server to abnormally terminate.](#)
  - [BUG-49439 In an AUDIT disconnect setting, the Altibase server may abnormally terminate when a database session is terminated.](#)
  - [BUG-49473 When a sequence object is specified in the GET\\_DDL function of the DBMS\\_METADATA package, statements with MIN and MAX values different from those of the sequence properties may be extracted.](#)
  - [BUG-49543 Altibase server may abnormally terminate due to a bug that deletes a deleted disk index key again during a disk table change transaction.](#)
  - [BUG-49544 Add a debugging log to analyze the cause of the \[The tablespace does not have enough free space \(TBS Name : \).\] error when there is free space when expanding files when multiple data files are used in the undo tablespace.](#)
  - [BUG-49553 Altibase server may abnormally terminate due to a bug that frees memory released during the process of replacing SQL Plan Cache's PCO \(Plan Cache Object\).](#)
  - [BUG-49565 Supplements the error message displayed in altibase\\_error.log when a failure occurs in the Refine Memory Table step while running the Altibase server.](#)
  - [BUG-49576 Add LOB to data types supported by Adapter for JDBC.](#)
- [Changes](#)
  - [Version Info](#)
  - [Altibase Server Properties](#)
  - [Performance Views](#)

# Fixed Bugs

---

**BUG-47420 Fixed the issue where the Altibase server abnormally terminates when using LENGTH() to output the length of LOB data and the LOB data type exceeds the maximum size when input into a disk table.**

- **module** : sm
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixed the issue related to inputting LOB data and outputting with LENGTH().
  1. Problem with inputting LOB data exceeding the maximum size (4G-1bytes) in disk tables
  2. Problem with inputting LOB data exceeding the maximum size in memory/volatile tables, resulting in failure, but some data remains after rollback.
  3. Abnormal termination of the Altibase server when using LENGTH() to output the length of LOB data in disk/memory/volatile memory tables.

After this bug fix, inputting LOB data exceeding 4GB-1byte in disk tables will result in error [0x110D0, 'The lob size is bigger than the maximum lob size].

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

**BUG-49133 Setting SORT\_AREA\_SIZE to its minimum value of 524288 and setting TEMP\_HASH\_BUCKET\_DENSITY to 65 or higher may cause the Altibase server to abnormally terminate.**

- **module** : sm
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Setting SORT\_AREA\_SIZE to its minimum value of 524288 and setting TEMP\_HASH\_BUCKET\_DENSITY to 65 or higher may cause the Altibase server to abnormally terminate. This issue has been fixed.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**

- Expected Results
- Workaround
- Changes
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-49439 In an AUDIT disconnect setting, the Altibase server may abnormally terminate when a database session is terminated.

- **module** : mm-altiaudit
- **Category** : Maintainability
- **Reproducibility** : Rare
- **Description** : Fix the issue where the Altibase server abnormally terminates when a database session is terminated in an AUDIT disconnect setting.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**

The Altibase server abnormally terminates and leaves the following call stack in altibase\_error.log.

```
BEGIN-STACK [CRASH] =====
Caller[0] 0000000000417208 => mmmSignalHandler
Caller[1] 00007F004DB5D5E0 => not found
Caller[2] 0000000000442C3F =>
mmtAuditManager::auditBySession(mmcStatement*)
Caller[3] 000000000046925F => mmcStatement::finalize()
Caller[4] 0000000000470E6B =>
mmcStatementManager::freeStatement(mmcStatement*)
Caller[5] 0000000000462669 => mmcSession::finalize()
Caller[6] 000000000041D62F =>
mmtSessionManager::freeSession(mmcTask*)
Caller[7] 0000000000413558 => mmcTask::finalize()
Caller[8] 000000000041C9ED => mmtSessionManager::freeTask(mmcTask*)
Caller[9] 000000000042A485 =>
mmtServiceThread::terminateTask(mmcTask*)
Caller[10] 000000000042AD82 => mmtServiceThread::executeTask()
Caller[11] 000000000042B75C => mmtServiceThread::run()
Caller[12] 0000000000EA4A1B => idtContainer::staticRunner(void*)
Caller[13] 00007F004DB55E25 => not found
Caller[14] 00007F004CC2434D => not found
END-STACK =====
```

- Expected Results
 

Disable the AUDIT disconnect setting.
- Workaround

- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-49473 When a sequence object is specified in the GET\_DDL function of the DBMS\_METADATA package, statements with MIN and MAX values different from those of the sequence properties may be extracted.

- **module** : ux-dbms\_metadata
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixes a bug where statements with MIN and MAX values different from sequence properties are extracted when a sequence object is specified in the GET\_DDL function of the DBMS\_METADATA package.

This bug is possible if the following two conditions are met when creating a sequence object:

- When the INCREMENT BY value is less than 0 and the MINVALUE is specified as 1
- When the INCREMENT BY value is less than 0 and the MAXVALUE is specified as 9223372036854775806
- **How to reproduce this bug**
  - **Reproduction conditions**

```
CREATE SEQUENCE S1 INCREMENT BY -1 MINVALUE 1 MAXVALUE 10000;
CREATE SEQUENCE S2 INCREMENT BY 1 MINVALUE 10 MAXVALUE
9223372036854775806;
SELECT DBMS_METADATA.GET_DDL('SEQUENCE', 'S1', 'SYS') FROM DUAL;
SELECT DBMS_METADATA.GET_DDL('SEQUENCE', 'S2', 'SYS') FROM DUAL;
```

- **Actual Results**

```
iSQL> SELECT DBMS_METADATA.GET_DDL('SEQUENCE', 'S1', 'SYS') FROM DUAL;
DBMS_METADATA.GET_DDL('SEQUENCE', 'S1', 'SYS
-----
-----
CREATE SEQUENCE "SYS"."S1" START WITH 10000 INCREMENT BY -1 MAXVALUE
10000
1 row selected.
iSQL> SELECT DBMS_METADATA.GET_DDL('SEQUENCE', 'S2', 'SYS') FROM DUAL;
DBMS_METADATA.GET_DDL('SEQUENCE', 'S2', 'SYS
-----
-----
CREATE SEQUENCE "SYS"."S2" START WITH 9223372036854775806 INCREMENT BY
-1
1 row selected.
```

- **Expected Results**

```

iSQL> SELECT DBMS_METADATA.GET_DDL('SEQUENCE', 'S1', 'SYS') FROM DUAL;
DBMS_METADATA.GET_DDL('SEQUENCE', 'S1', 'SYS
-----
-----

CREATE SEQUENCE "SYS"."S1" START WITH 10000 INCREMENT BY -1 MINVALUE 1
MAXVALUE 10000
1 row selected.
iSQL> SELECT DBMS_METADATA.GET_DDL('SEQUENCE', 'S2', 'SYS') FROM DUAL;
DBMS_METADATA.GET_DDL('SEQUENCE', 'S2', 'SYS
-----
-----

CREATE SEQUENCE "SYS"."S2" START WITH 9223372036854775806 INCREMENT BY
-1 MINVALUE 1 MAXVALUE 922337
2036854775806
1 row selected.

```

- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-49543 Altibase server may abnormally terminate due to a bug that deletes a deleted disk index key again during a disk table change transaction.

- **module** : sm-disk-index
- **Category** : Fatal
- **Reproducibility** : Rare
- **Description** : Avoid abnormal termination of the Altibase server when attempting to delete a deleted disk index key again during a disk table change transaction, and add a log to analyze the cause. After reflecting this bug, when a phenomenon occurs, the following error message is displayed.

```

ERR-11069 : Internal server error in the storage manager (fail to delete key
- invalid key state)

```

In addition, to analyze the cause of attempting to delete the disk index key again, the following error message is additionally recorded in the trace log, altibase\_error.log.

```

deleteKeyFromLeafNode() INVALID KEY STATE ( IndexID: IndexID, LeafKey:
LeafKey, key state : key state, sequence number : sequence number )

Delete Key by same TX (TID:TID),((smxTrans*)(aMtx->mTrans))->mTransID );

```

The result of dumping the page in question is also recorded in altibase\_dump.log.

- **How to reproduce this bug**

- **Reproduction conditions**

- **Actual Results**

When the Altibase server abnormally terminates due to this bug, the following call stack is left in the altibase\_error.log.

```
[2021/12/13 21:58:29 32E85D][PID:6885][Thread-140305053693696][LWP-6948]
```

```
leaf sequence number : 130
```

```
[2021/12/13 21:58:29 32E861] Dump of stack
```

```
BEGIN-STACK [NORMAL] =====
```

```
Caller[0] 0000000000EC4DD8 =>
```

```
_ZN6ideLog23writeErrorTraceInternalEj12ideLogModuleejPKCS2_js2_P13__va_list_tag.constprop.2
```

```
Caller[1] 0000000000EC8075 => ideLog::writeErrorTrace(char const*, idBool, char const*, unsigned int, char const*, __va_list_tag*)
```

```
Caller[2] 0000000000EC3B2F => ideLogError
```

```
Caller[3] 0000000000D42CEF =>
```

```
_ZN9sdnbBTree21deleteKeyFromLeafNodeEP6idvSQLP13sdnbStatisticP6sdrMtxP10sdnbHeaderP13sdpPhyPageHdrPs6idBoolPSB_.constprop.42
```

```
Caller[4] 0000000000D62FC3 => sdnbBTree::deleteKey(idvSQL*, void*, void*, char*, smiIterator*, unsigned long)
```

```
Caller[5] 0000000000DDEC3E =>
```

```
smiTableCursor::deleteKeys(smiTableCursor*, scGRID, idBool)
```

```
Caller[6] 0000000000DDEEA8 =>
```

```
smiTableCursor::deleteRowDRDB(smiTableCursor*, smiDMLRetryInfo const*)
```

```
Caller[7] 0000000000DDAF45 =>
```

```
smiTableCursor::deleteRow(smiDMLRetryInfo const*)
```

```
...종락...
```

```
END-STACK =====
```

```
[2021/12/13 21:58:29 32E860][PID:6885][Thread-140305053693696][LWP-6948]
```

```
IDE_ASSERT( 0 ), [sdnbModule.cpp:5950], errno=[16]
```

```
errno=[16]
```

```
[2021/12/13 21:58:29 32E862][PID:6885][Thread-140305053693696][LWP-6948]
```

```
[ASSERT] ERROR LINE => [sdnbModule.cpp:5950]
```

```
ERR-0109e(errno=16) Internal server error.
```

```
altibase:
```

```
/home/hdb651_p/work/altidev4/src/sm/sdn/sdnb/sdnbModule.cpp:5950: static  
IDE_RC sdnbBTree::deleteKeyFromLeafNode(idvSQL*, sdnbStatistic*,  
sdrMtx*, sdnbHeader*, sdpPhyPageHdr*, SShort*, idBool, idBool*):  
Assertion `0' failed.
```

- **Expected Results**

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-49544 Add a debugging log to analyze the cause of the [The tablespace does not have enough free space ( TBS Name : ).] error when there is free space when expanding files when multiple data files are used in the undo tablespace.

- **module** : sm-disk-page
- **Category** : Functional Error
- **Reproducibility** : Frequence
- **Description** : If you encounter an error where the tablespace does not have enough free space despite having available space when using multiple data files in an undo tablespace, this log will be added to assist in debugging the cause of the error. Note that this log is only written when activating a private property, so please contact Altibase Technical Support Center if necessary.

```
Failed to expand UndoTBS File.  
UndoTBS FileName : file_name, FileID : file_id, CurrSize = curr_size,  
NextSize = next_size, MaxSize = max_size
```

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

## BUG-49553 Altibase server may abnormally terminate due to a bug that frees memory released during the process of replacing SQL Plan Cache's PCO (Plan Cache Object).

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Impossible
- **Description** : Fixed an issue where the Altibase server abnormally terminates due to a bug that frees memory released during the process of replacing the SQL Plan Cache PCO (Plan Cache Object) again.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**



- Performance view
- Property
- Compile Option
- Error Code

## BUG-49565 Supplements the error message displayed in altibase\_error.log when a failure occurs in the Refine Memory Table step while running the Altibase server.

- **module** : sm
- **Category** : Functional Error
- **Reproducibility** : Unknown
- **Description** : Supplements the error message displayed in altibase\_error.log when a failure occurs in the Refine Memory Table step while running the Altibase server.

After reflecting this bug, a more accurate error code and error message such as [ERR-11088 Invalid row SCN] are output in the ERR-00000 (errno=16) part.

```
[2022/01/11 00:14:56 416] [PID:2150] [Thread-47881693447936] [LWP-2279]
[FATAL] ERROR LINE => [smtPJMgr.cpp:154] MSG[error]
===== CURRENT IDE ERROR STACK =====
ERR-00000(errno=16)
[===== FATAL Terminated =====]
```

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

## BUG-49576 Add LOB to data types supported by Adapter for JDBC.

- **module** : rp-jdbcAdapter
- **Category** : Efficiency
- **Reproducibility** : Always
- **Description** : Add LOB to data types supported by Adapter for JDBC.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**

ALA Log converter error: Unsupported data type.

- **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

# Changes

---

## Version Info

| altibase<br>version | database binary<br>version | meta<br>version | cm protocol<br>version | replication protocol<br>version |
|---------------------|----------------------------|-----------------|------------------------|---------------------------------|
| 7.1.0.6.9           | 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.

### Meta Version

The meta version has not changed.

### 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