

# Altibase 7.1.0.7.2 Patch Notes

---

# Table of Contents

---

- [New Features](#)
  - [BUG-49567 Add LOB to the data types supported by Adapter for Oracle.](#)
  - [BUG-49616 In Altibase 7.1 Standard Edition, Enterprise Edition, MEM\\_MAX\\_DB\\_SIZE is added as a license issuance standard.](#)
- [Fixed Bugs](#)
  - [BUG-49546 Add a connection property getcolumns\\_return\\_jdbctype that defines the value of DATA\\_TYPE among the return results of the DatabaseMetaData.getColumns method.](#)
  - [BUG-49549 If an aggregation function used for a partitioned table is executed in parallel, the Altibase server may abnormally terminate.](#)
  - [BUG-49559 Improves SQL performance by eliminating redundant sort operations when the ROW\\_NUMBER function is used on tables with composite indexes.](#)
  - [BUG-49601 Change DatabaseMetaData.getUDTs\(\) to return empty ResultSet instead of SQLFeatureNotSupportedException when User Defined Type is not supported.](#)
  - [BUG-49602 The ResultSet.insertRow\(\) method should work even if the Updatable ResultSet object has zero results.](#)
  - [BUG-49603 If the database owner is specified in the SQL statement, the Updatable ResultSet object throws a \[java.sql.SQLException: Table or view was not found\] error.](#)
  - [BUG-49605 Fixes a phenomenon in which an Unsupported type conversion SQLException error occurs when data is input using an Updatable ResultSet.](#)
  - [BUG-49613 The Altibase server may abnormally terminate when the result of a subquery used in the = operator and the CAST operator is NULL.](#)
  - [BUG-49619 If the disk index uses the undo tablespace, the Altibase server may abnormally terminate as the undo space is reused by other transactions.](#)
  - [BUG-49620 In NVL\\_EQUAL\(expr1, expr2, expr3\), if an index exists on expr1 and the data type of a column in expr2 is different from expr1, the table scan method is changed to improve product stability.](#)
  - [BUG-49632 Improved product stability by changing exception handling when a stored function whose return data type is LOB is used in the PARTITION BY subclause of ORDER BY/GROUP BY/window functions.](#)
  - [BUG-49633 When executing SQL using the DISTINCT keyword in an aggregate function, SQL execution performance deteriorates due to excessive use of memory for DISTINCT processing.](#)
  - [BUG-49636 Improves the archive log thread behavior according to the Altibase server property ARCHIVE\\_FULL\\_ACTION setting value.](#)
  - [BUG-49647 When running altiMon on Ubuntu 16, Ubuntu 18, \[com.altibase.picl.LibLoader.OsNotSupportedException\] error occurs.](#)
  - [BUG-49655 Add TOTAL\\_USED\\_PAGE\\_CNT column to X\\$SEGMENT so that you can query the number of data pages in use in disk tables and disk indexes.](#)
- [Changes](#)
  - [Version Info](#)
  - [Altibase Server Properties](#)
  - [Performance Views](#)

# New Features

---

## BUG-49567 Add LOB to the data types supported by Adapter for Oracle.

- **module** : rp-oraAdapter
- **Category** : Enhancement
- **Reproducibility** : Always
- **Description** : Add LOB to the data types supported by Adapter for Oracle.

For the operation and restrictions of Adapter for Oracle related to LOB data type support, please refer to the Adapter for Oracle manual.

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

## BUG-49616 In Altibase 7.1 Standard Edition, Enterprise Edition, MEM\_MAX\_DB\_SIZE is added as a license issuance standard.

- **module** : id
- **Category** : Enhancement
- **Reproducibility** : Unknown
- **Description** : In Altibase 7.1 Standard Edition, Enterprise Edition, MEM\_MAX\_DB\_SIZE is added as a license issuance standard.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- **Workaround**
- **Changes**
  - Performance view
  - Property
  - Compile Option
  - Error Code

# Fixed Bugs

## BUG-49546 Add a connection property `getcolumns_return_jdbctype` that defines the value of `DATA_TYPE` among the return results of the `DatabaseMetaData.getColumns` method.

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Add a connection property [getcolumns\\_return\\_jdbctype](#) that defines the value of `DATA_TYPE` among the return results of the `DatabaseMetaData.getColumns` method.

- connection property name  
`getcolumns_return_jdbctype`

- description

Defines the value of `DATA_TYPE` among the return results of the `DatabaseMetaData.getColumns` method.

True is returned as the SQL data type of `java.sql.Types` defined in the JDBC API, and false is returned as the data type defined in `V$DATATYPE`.

- default  
false

To apply this bug, you need to patch the Altibase JDBC driver to Altibase 7.1.0.7.2 or later and add the connection property `getcolumns_return_jdbctype`.

- **How to reproduce this bug**
  - **Reproduction conditions**

```
iSQL> CREATE TABLE T1 (C1 CLOB, C2 BLOB);
```

```
Connection sConn = getConnection("20300");
DatabaseMetaData sMeta = sConn.getMetaData();
ResultSet sRs = sMeta.getColumns(null, "SYS", "T1", "%");
while (sRs.next())
{
    String sColumnName = sRs.getString("COLUMN_NAME");
    String sDataType = sRs.getString("DATA_TYPE");
    System.out.println("COLUMN_NAME==>" + sColumnName);
    System.out.println("DATA_TYPE==>" + sDataType);
}
```

- **Actual Results**

```
COLUMN_NAME==>C1  
DATA_TYPE==>40  
COLUMN_NAME==>C2  
DATA_TYPE==>30
```

- **Expected Results**

```
COLUMN_NAME==>C1  
DATA_TYPE==>2005  
COLUMN_NAME==>C2  
DATA_TYPE==>2004
```

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## **BUG-49549 If an aggregation function used for a partitioned table is executed in parallel, the Altibase server may abnormally terminate.**

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Frequence
- **Description** : Fixes a bug where the Altibase server abnormally terminates when an aggregation function used for a partitioned table is executed in parallel.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
DROP TABLE t1 cascade;  
CREATE TABLE T1( I1 INTEGER, I2 INTEGER ) PARTITION BY HASH(I1)  
  ( PARTITION P1, PARTITION P2 );  
ALTER TABLE T1 PARALLEL 2;  
VAR v1 integer;  
PREPARE SELECT /*+ no_plan_cache */ COUNT(*) FROM T1 WHERE (I1=:v1 OR  
I1=999) AND (I2=1 OR I2=2);
```

- **Actual Results**

```
Altibase server abnormally terminates
```

- **Expected Results**

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

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

## BUG-49559 Improves SQL performance by eliminating redundant sort operations when the ROW\_NUMBER function is used on tables with composite indexes.

- **module** : qp
- **Category** : Enhancement
- **Reproducibility** : Always
- **Description** : Improved SQL execution performance by adding the Preserved Order function to avoid redundant sort operations when using the ROW\_NUMBER function on a table with a composite index.

Note: Preserved Order function

Altibase uses the concept of Preserved Order to avoid redundant sorting operations when already sorted tables are equally sorted using the ORDER BY statement. For example, if the i1 column in table t1 has an Ascending sorted index, if the i1 column is ascending sorted using the ORDER BY statement, it is read as it is without performing a separate sorting operation.

You need to change the private property to apply this bug. If necessary, please contact the Altibase Technical Support Center.

- **How to reproduce this bug**
  - **Reproduction conditions**

```
DROP TABLE T1;
CREATE TABLE T1 ( I1 INT, I2 INT, I3 INT, I4 INT);
DROP INDEX IDX1;
CREATE INDEX IDX1 ON T1 ( I1, I2, I3 DESC );
ALTER SESSION SET EXPLAIN PLAN = ON;
SELECT ROW_NUMBER() OVER ( ORDER BY I3 DESC ), * FROM T1 WHERE I1 = 1
AND I2 = 2;
```

- **Actual Results**

```
-----
PROJECT ( COLUMN_COUNT: 5, TUPLE_SIZE: 24, COST: 0.26 )
  WINDOW SORT ( ITEM_SIZE: 24, ITEM_COUNT: 0, ACCESS: 0, SORT_COUNT: 1,
COST: 0.19 )
    SCAN ( TABLE: SYS.T1, INDEX: SYS.IDX1, RANGE SCAN, ACCESS: 0, COST:
0.01 )
-----
```

- **Expected Results**

SORT\_COUNT is decremented.

```
-----
PROJECT ( COLUMN_COUNT: 5, TUPLE_SIZE: 24, COST: 0.26 )
  WINDOW SORT ( ITEM_SIZE: 24, ITEM_COUNT: 0, ACCESS: 0, SORT_COUNT: 0,
COST: 0.19 )
    SCAN ( TABLE: SYS.T1, INDEX: SYS.IDX1, RANGE SCAN, ACCESS: 0, COST:
0.01 )
-----
```

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-49601 Change DatabaseMetaData.getUDTs() to return empty ResultSet instead of SQLFeatureNotSupportedException when User Defined Type is not supported.

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Change DatabaseMetaData.getUDTs() to return empty ResultSet instead of SQLFeatureNotSupportedException when User Defined Type is not supported.
  - 참고 : <https://docs.oracle.com/javame/config/cdc/opt-pkgs/api/jsr169/java/sql/DatabaseMetaData.html>

To apply this bug, the Altibase JDBC driver must be patched to 7.1.0.7.2 or higher.

- **How to reproduce this bug**

- **Reproduction conditions**

```
Connection sConn = getConnection("20300");
DatabaseMetaData sMeta = sConn.getMetaData();
ResultSet sRs = sMeta.getUDTs(null, null, null, null);
System.out.println("sRs.next()==>" + sRs.next());
```

- **Actual Results**

```
Exception in thread "main" java.sql.SQLException:
User defined type
```

- **Expected Results**

```
sRs.next() ==> false
```

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## BUG-49602 The ResultSet.insertRow() method should work even if the Updatable ResultSet object has zero results.

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Improve the phenomenon that the ResultSet.insertRow() method does not work even when the result of the Updatable ResultSet object is 0.

The options for allowing scrolling and updating of ResultSet objects affected by this bug are listed below.

- scroll options
  - TYPE\_FORWARD\_ONLY
  - TYPE\_SCROLL\_INSENSITIVE
- Allow updates option
  - CONCUR\_UPDATABLE

To apply this bug, the Altibase JDBC driver must be patched to 7.1.0.7.2 or higher.

- **How to reproduce this bug**

- **Reproduction conditions**

```
iSQL> CREATE TABLE T1 (C1 INT, C2 VARCHAR(10));
```

```
Connection sConn = getConnection("20300");
Statement sStmt = sConn.createStatement(ResultSet.TYPE_FORWARD_ONLY,
                                         ResultSet.CONCUR_UPDATABLE);
ResultSet sRs = sStmt.executeQuery("SELECT c1, c2 FROM t1");
sRs.moveToInsertRow();
sRs.updateObject(1, 1);
sRs.updateObject(2, "test..");
sRs.insertRow();
```

```
iSQL> SELECT * FROM t1;
```



- **Actual Results**

```
isQL> SELECT * FROM t1;
c1          c2
-----
No rows selected.
```

- **Expected Results**

```
isQL> SELECT * FROM t1;
c1          c2
-----
1          test..
1 row selected.
```

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

## **BUG-49603 If the database owner is specified in the SQL statement, the Updatable ResultSet object throws a [java.sql.SQLException: Table or view was not found] error.**

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixes an issue where an error [java.sql.SQLException: Table or view was not found] occurs in an Updatable ResultSet object when the database owner is specified in the SQL statement.

To apply this bug, the Altibase JDBC driver must be patched to 7.1.0.7.2 or higher.

- **How to reproduce this bug**

- **Reproduction conditions**

```
isQL> CREATE USER altitest IDENTIFIED BY altitest;
Create success.
isQL> CONNECT altitest;
Write Password :
Connect success.
isQL> CREATE TABLE T2 (C1 INT, C2 VARCHAR(10));
Create success.
```

```

Connection sConn = getConnection("20300"); // connect with SYS
Statement sStmt = sConn.createStatement(ResultSet.TYPE_FORWARD_ONLY,
                                         ResultSet.CONCUR_UPDATABLE);
ResultSet sRs = sStmt.executeQuery("SELECT c1, c2 FROM altitest.t2");
sRs.moveToInsertRow();
sRs.updateObject(1, 1);
sRs.updateObject(2, "test..");
sRs.insertRow();

```

#### ◦ Actual Results

```

Exception in thread "main" java.sql.SQLException: Table or view was not
found :
0001 : DELETE FROM T2 WHERE _PROVID=?
      ^ ^
      at Altibase.jdbc.driver.ex.Error.processServerError(Error.java:397)
      at
Altibase.jdbc.driver.AltibasePreparedStatement.prepare(AltibasePreparedS
tatement.java:139)
      at Altibase.jdbc.driver.AltibasePreparedStatement.
(AltibasePreparedStatement.java:94)
      at
Altibase.jdbc.driver.AltibaseConnection.prepareStatement(AltibaseConnect
ion.java:1346)
      at Altibase.jdbc.driver.AltibaseUpdatableResultSet.
(AltibaseUpdatableResultSet.java:66)
      at
Altibase.jdbc.driver.AltibaseResultSet.createResultSet(AltibaseResultSet
.java:90)
      at
Altibase.jdbc.driver.AltibaseStatement.processExecutionQueryResult(Altib
aseStatement.java:414)
      at
Altibase.jdbc.driver.AltibaseStatement.executeQuery(AltibaseStatement.ja
va:754)

```

#### ◦ Expected Results

```
insert succeeded
```

#### • Workaround

#### • Changes

- Performance view
- Property
- Compile Option
- Error Code

## BUG-49605 Fixes a phenomenon in which an Unsupported type conversion SQLException error occurs when data is input using an Updatable ResultSet.

- **module** : mm-jdbc
- **Category** : Functional Error
- **Reproducibility** : Always
- **Description** : Fixes a phenomenon in which an Unsupported type conversion SQLException error occurs when data is input using an Updatable ResultSet.

To apply this bug, the Altibase JDBC driver must be patched to 7.1.0.7.2 or higher.

- **How to reproduce this bug**

- **Reproduction conditions**

```
CommonBinaryColumn.setValueSub(Duration);
CommonBinaryColumn.setValueSub(ByteArrayInputStream);
CommonCharVarcharColumn.setValueSub(Clob);
CommonCharVarcharColumn.setValueSub(StringReader);
CommonCharVarcharColumn.setValueSub(ByteArrayInputStream)FloatColumn.getObjectSub();
TimestampColumn.setValueSub(Time);
```

- **Actual Results**

Throws Unsupported type conversion SQLException

- **Expected Results**

normal processing

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

## BUG-49613 The Altibase server may abnormally terminate when the result of a subquery used in the = operator and the CAST operator is NULL.

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Fixes an issue where the Altibase server abnormally terminates when the result of a subquery used in the = operator and the CAST operator is NULL.

- **How to reproduce this bug**

- **Reproduction conditions**

```
CREATE TABLE T1 ( I1 INTEGER, I2 INTEGER, I3 INTEGER ) TABLESPACE
SYS_TBS_DISK_DATA;
INSERT INTO T1 VALUES (1, 1, NULL);
CREATE TABLE T2 ( I1 INTEGER ) TABLESPACE SYS_TBS_DISK_DATA;
SELECT * FROM T1 WHERE I1 = (SELECT DISTINCT I1 || I2 FROM T2);
```

- **Actual Results**

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

- **Expected Results**

```
No rows selected.
```

- **Workaround**

- **Changes**

- Performance view
  - Property
  - Compile Option
  - Error Code

## **BUG-49619 If the disk index uses the undo tablespace, the Altibase server may abnormally terminate as the undo space is reused by other transactions.**

- **module** : sm-disk-index

- **Category** : Fatal

- **Reproducibility** : Rare

- **Description** : Fixes a bug in which the Altibase server abnormally terminates due to reuse of the undo tablespace by other transactions when disk indexes use the undo tablespace.

When an Altibase server abnormally terminates due to this bug, the following log is recorded in the trace log.

- altibase\_error.log

```
[ASSERT] ERROR LINE => [sdnIndexCTL.cpp:2208]
```

- Record index and undo page information in the form of the following in altibase\_boot.log

```
[2022/02/09 16:42:10] [Thread-28749] [Level-0]
Undo PageID          : 56023
Undo Slot Number     : 91
UndoRec Hdr Type     : 2
Chained CTS's Commit SCN : 61801506952
Min DskView SCN      : 61801348159
```

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

**BUG-49620 In NVL\_EQUAL(expr1, exp2, expr3), if an index exists on expr1 and the data type of a column in expr2 is different from expr1, the table scan method is changed to improve product stability.**

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : In NVL\_EQUAL(expr1, exp2, expr3), if an index exists on expr1 and the data type of a column in expr2 is different from expr1, the table scan method is changed to improve product stability.
- **How to reproduce this bug**
  - **Reproduction conditions**

```
CREATE TABLE T1( I1 INTEGER, I2 VARCHAR(10) ) PARTITION BY HASH( I1 ) (
PARTITION P1, PARTITION P2 TABLESPACE SYS_TBS_MEM_DATA, PARTITION P3 )
TABLESPACE SYS_TBS_DISK_DATA;
CREATE INDEX IDX1 ON T1(I1) LOCAL;
SELECT * FROM T1 WHERE NVL_EQUAL(I1,I2,2);
```

- **Actual Results**

Altibase **server** abnormally terminates.

**ALTER SESSION SET EXPLAIN PLAN = ONLY;** Execution plan when **SELECT** is executed **after** execution

```
-----  
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 58.66 )  
  PARTITION-COORDINATOR ( TABLE: SYS.T1, PARTITION: 3/3, ACCESS: ??,  
COST: 57.87 )  
    SCAN ( PARTITION: P3, INDEX: SYS.IDX1, RANGE SCAN, ACCESS: ??, COST:  
23.09 )  
    SCAN ( PARTITION: P2, INDEX: SYS.IDX1, RANGE SCAN, ACCESS: ??, COST:  
11.69 )  
    SCAN ( PARTITION: P1, INDEX: SYS.IDX1, RANGE SCAN, ACCESS: ??, COST:  
23.09 )  
-----
```

- **Expected Results**

I1            I2

-----  
**No** rows selected.

**ALTER SESSION SET EXPLAIN PLAN = ONLY;** Execution plan when **SELECT** is executed **after** execution

```
-----  
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 16, COST: 386.68 )  
  PARTITION-COORDINATOR ( TABLE: SYS.T1, PARTITION: 3/3, FULL SCAN,  
ACCESS: 0, COST: 385.88 )  
    SCAN ( PARTITION: P3, FULL SCAN, ACCESS: 0, DISK_PAGE_COUNT: 64, COST:  
129.28 )  
    SCAN ( PARTITION: P2, FULL SCAN, ACCESS: 0, COST: 127.32 )  
    SCAN ( PARTITION: P1, FULL SCAN, ACCESS: 0, DISK_PAGE_COUNT: 64, COST:  
129.28 )  
-----
```

- **Workaround**

- **Changes**

- Performance view
- Property
- Compile Option
- Error Code

# BUG-49632 Improved product stability by changing exception handling when a stored function whose return data type is LOB is used in the PARTITION BY subclause of ORDER BY/GROUP BY/window functions.

- **module** : qp
- **Category** : Fatal
- **Reproducibility** : Always
- **Description** : Improved the abnormal termination of the Altibase server when a stored function with a return data type of LOB is used in the PARTITION BY subclause of ORDER BY/GROUP BY/window functions.

- **How to reproduce this bug**

- **Reproduction conditions**

```
CREATE TABLE T1( I1 CHAR(10), I2 CHAR(10), I3 CHAR(10), I4 CHAR(10) )
TABLESPACE SYS_TBS_MEM_DATA;
CREATE FUNCTION FUNC1
(V1 CLOB)
RETURN CLOB
AS
    V2 CLOB;
BEGIN
    V2 := V1;
    RETURN V2;
END;
/
INSERT INTO T1(I2) VALUES (637);
SELECT GROUPING_ID(FUNC1(I1))
FROM T1
GROUP BY GROUPING SETS(1), FUNC1(I1);
```

- **Actual Results**

Altibase 서버 비정상 종료

- **Expected Results**

[ERR-3123E : An incomparable data type (GEOMETRY,LOB) cannot be used in a SELECT statement that has a DISTINCT clause.]

- **Workaround**

- **Changes**

- Performance view
  - Property
  - Compile Option
  - Error Code

# BUG-49633 When executing SQL using the DISTINCT keyword in an aggregate function, SQL execution performance deteriorates due to excessive use of memory for DISTINCT processing.

- **module** : qp-select
- **Category** : Enhancement
- **Reproducibility** : Always
- **Description** :

When executing SQL using the DISTINCT keyword in an aggregate function, memory usage increases due to excessive use of the number of hash buckets for DISTINCT processing, which reduces SQL execution performance. Provides a property that allows you to set the number of hash buckets appropriate for your operating environment if this bug occurs. It does not add a Description to the manual as a hidden property.

- Add property

- property name

\_\_AGGREGATION\_DISTINCT\_BUCKET\_COUNT\_MAX

- Description

This property specifies the maximum number of hash buckets for DISTINCT processing in an aggregate function.

- Value

1 ~ 102400000

- Default

102400000

- attribute

Read-Write, Internal

It also displays the number of hash buckets used by the SQL in the execution plan. You can set \_\_AGGREGATION\_DISTINCT\_BUCKET\_COUNT\_MAX by referring to this value. This can be seen in the execution plan of SQL where the DISTINCT keyword is used with aggregate functions, OVER, GROUP BY ROLLUP, and GROUP BY CUBE.

- Affected SQL Examples

```
SELECT COUNT(DISTINCT i1) FROM t1;
SELECT SUM( DISTINCT I2 ) OVER ( PARTITION BY I1 ) FROM T1;
SELECT SUM( DISTINCT I2 ) FROM T1 GROUP BY ROLLUP ( I1, I3, I4 );
SELECT SUM( DISTINCT I2 ) FROM T1 GROUP BY CUBE ( I1, I3, I4 );
```

The default value of the \_\_AGGREGATION\_DISTINCT\_BUCKET\_COUNT\_MAX property is 102400000. If you want to set less than this during operation, you can change it with ALTER SYSTEM. To permanently set a small value, restart the Altibase server after adding it to altibase.properties.

- **How to reproduce this bug**
  - **Reproduction conditions**



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

## BUG-49636 Improves the archive log thread behavior according to the Altibase server property ARCHIVE\_FULL\_ACTION setting value.

- **module** : sm
- **Category** : Functionality
- **Reproducibility** : Always
- **Description** : Improves the behavior of the archivelog thread according to the ARCHIVE\_FULL\_ACTION setting value when log file backup fails due to lack of disk space.

### Action difference according to ARCHIVE\_FULL\_ACTION setting value

- ARCHIVE\_FULL\_ACTION = 0
  - Before change: If log file backup fails due to insufficient disk space, wait until the backup succeeds.
  - After change: If log file backup fails due to insufficient disk space, record in the trace log (altibase\_sm.log) and try the next log file backup.
- ARCHIVE\_FULL\_ACTION = 1
  - No change.
- ARCHIVE\_FULL\_ACTION = 2
  - Setting value 2 has been added. If log file backup fails for any reason other than insufficient disk space failure, it prints an error message to the trace log (altibase\_sm.log) and tries to back up the next log file.

### ARCHIVE\_FULL\_ACTION property change

Change from read-only to mutable.

- Before change

```
iSQL> ALTER SYSTEM SET ARCHIVE_FULL_ACTION = 2;
[ERR-0104E : The property [ARCHIVE_FULL_ACTION] is read-only.]
```

- After change

```
iSQL> ALTER SYSTEM SET ARCHIVE_FULL_ACTION = 2;
Alter success
```

- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**

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

## BUG-49647 When running altiMon on Ubuntu 16, Ubuntu 18, [com.altibase.picl.LibLoader.OsNotSupportedException] error occurs.

- **module** : ux-altiMon
- **Category** : Portability
- **Reproducibility** : Always
- **Description** : [com.altibase.picl.LibLoader.OsNotSupportedException] error occurs when running altiMon on Ubuntu 16 or Ubuntu 18 without Linux kernel 4 support.  
Add support for Linux kernel 4.x in altiMon.
- **How to reproduce this bug**
  - **Reproduction conditions**
  - **Actual Results**
  - **Expected Results**
- Workaround
- Changes
  - Performance view
  - Property
  - Compile Option
  - Error Code

## BUG-49655 Add TOTAL\_USED\_PAGE\_CNT column to X\$SEGMENT so that you can query the number of data pages in use in disk tables and disk indexes.

- **module** : sm
- **Category** : Other
- **Reproducibility** : Always
- **Description** : TOTAL\_USED\_PAGE\_CNT in X\$SEGMENT to query the number of data pages in use in disk tables and disk indexes Add a column. The TOTAL\_USED\_PAGE\_CNT column means the number of data-only pages excluding meta pages and FREE pages.
- **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
7.1.0.7.2	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**