Altibase 7.1.0.7.9 Patch Notes

Table of Contents

New Features

BUG-49776 Adds the ability to set the keys of the shared memory and semaphores required to create IPC and IPCDA channels to user-defined values.

Module

`cm-ipc`

Category

`Functionality`

Description

Adds the ability to set the keys of the shared memory and semaphores required to create IPC and IPCDA channels to user-defined values.

Users can set the keys of shared memory and semaphores using the following four properties.

- IPC_SEM_KEY
- IPC_SHM_KEY
- IPCDA_SEM_KEY
- IPCDA_SHM_KEY

IPC and IPCDA channels are created when the Altibase server is started. If the shared memory/semaphore key set as a property is in use or the shared memory/semaphore cannot be created for other reasons, the Altibase server will fail. At this time, you must check the system error (errno) in the altibase server trace log altibase_boot.log and take appropriate actions accordingly.

Changes

- Performance view
- Property
  - `IPC_SEM_KEY`

    This property sets the semaphore key required to create an IPC channel to a value defined by the user.

    The default value is 0, which automatically generates a semaphore key based on the process identifier (PID) of the Altibase server process. If a value other than 0 is set, an IPC channel is created using consecutive semaphore keys from IPC_SEM_KEY to IPC_SEM_KEY + (IPC_CHANNEL_COUNT + 1) based on the IPC_SEM_KEY value. +1 is an IPC channel reserved for the SYS user to access in administrator mode (sysdba). For example, if the value of IPC_SEM_KEY is 10000 and the value of IPC_CHANNEL_COUNT is 1000, semaphore keys from 10000 to 11000 are used.

  - `IPC_SHM_KEY`

    This is a property that sets the shared memory key required to create an IPC channel to a value defined by the user.

The default value is 0, which automatically creates a shared memory key based on the process identifier (PID) of the Altibase server process. If a value other than 0 is set, the IPC_SHM_KEY value is used as the shared memory key.

- `IPCDA_SEM_KEY`

   This property sets the semaphore key required to create an IPCDA channel to a value defined by the user.

   The default value is 0, which automatically generates a semaphore key based on the process identifier (PID) of the Altibase server process. If a value other than 0 is set, IPCDA channels are created using consecutive semaphore keys from IPCDA_SEM_KEY to IPCDA_SEM_KEY + IPC_CHANNEL_COUNT based on the IPCDA_SEM_KEY value. For example, if the value of IPCDA_SEM_KEY is 10000 and the value of IPC_CHANNEL_COUNT is 1000, semaphore keys from 10000 to 10999 are used.

- `IPCDA_SHM_KEY`

   This property sets the shared memory key required to create an IPCDA channel to a value defined by the user.

   The default value is 0, which automatically creates a shared memory key based on the process identifier (PID) of the Altibase server process. If a value other than 0 is set, two consecutive keys based on the value of IPCDA_SHM_KEY are used as shared memory keys. For example, if IPCDA_SHM_KEY=10000, 10000 and 10001 are used as shared memory key values.

- Compile Option
- Error Code

   Two error messages have been added.

   - When the shared memory cannot be created with the key defined in the Altibase server properties when creating IPC and IPCDA channels

      ```
      0x710C6 ( 463046) cmERR_ABORT_SHMGET_ERROR_WITH_KEY A system call error
      occurred while creating shared memory for <0%s>. [key : <1%u>]
      # *Cause: shmget() system call failed.
      # *Action: Check the errno and take an appropriate action. For example,
      if the errno is EEXIST, check the shared memory status. If there is a
      shared memory that has the same key value, remove the shared memory or
      retry with another key value.
      ```

   - When a semaphore cannot be created with the key defined in the Altibase server properties when creating IPC and IPCDA channels

      ```
      0x710C7 ( 463047) cmERR_ABORT_SEMGET_ERROR_WITH_KEY A system call error
      occurred while creating semaphore for <0%s>. [key : <1%u>]
      # *Cause: semget() system call failed.
      # *Action: Check the errno and take an appropriate action. For example,
      if the errno is EEXIST, check the semaphore status. If there is a
      semaphore that has the same key value, remove the semaphore or retry
      with another key value.
      ```

Fixed Bugs

BUG-48767 When partitioning simultaneously in multiple sessions with the same partition key value, partitions with the same minimum and maximum values may be created.

Module

`qp-ddl-dcl-pvo`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fixed a phenomenon in which partitions with the same minimum and maximum values were created due to concurrency issues when partitioning simultaneously in multiple sessions with the same partition key value. Fixed an error to occur when the minimum and maximum values are the same at the time of executing the SPLIT PARTITION statement.

Workaround

`none`

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-48882 Fixes an issue where logs from a point before the start of a replication sender are read when performing a replication SYNC to a partition table.

module

`rp`

Category

`Functionality`

Reproducibility

`Always`

Description

ALTER REPLICATION replication_name SYNC TABLE ~ PARTITION fixes the problem of reading the log from a point in time before the start of the replication sender. When this bug occurs, data between replication target servers may not match.

Workaround

`none`

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-48907 Improves an issue that can cause a memory leak when an exception occurs in the redundant SET clause.

module

`rp`

Category

`Memory Error`

Reproducibility

`Always`

Description

Improves the problem that memory leaks may occur when an exception occurs in the following redundant SET clause.

```
ALTER REPLICATION replication_name SET PARALLEL
ALTER REPLICATION replication_name SET GROUPING ENABLE/DISABLE
ALTER REPLICATION replication_name SET OFFLINE ENABLE WITH
ALTER REPLICATION replication_name SET GAPLESS ENABLE/DISABLE
```

Workaround

`none`

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49524 Improves exception handling when an error occurs due to insufficient tablespace space when executing ALTER TABLE table_name MODIFY COLUMN.

module

`qp`

Category

`Fatal`

Reproducibility

`Always`

Description

Fixes a phenomenon in which the Altibase server abnormally terminates when an error occurs due to insufficient tablespace space when executing ALTER TABLE MODIFY COLUMN.

Workaround

`none`

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49711 Improves stability issues that occur when unique key constraints are violated when performing multiple updates.

module

`qp-dml-execute`

Category

`Fatal`

Reproducibility

`Always`

Description

Fixes a phenomenon in which the Altibase server abnormally terminates when a unique key constraint is violated during multiple update.

This bug occurs when all of the conditions below are met.

- multiple update statement
- There are two or more indexes with unique key constraints on the table to be altered.
- When a change value violates a unique key constraint

Workaround

`none`

How to reproduce this bug

- Reproduction conditions

```
CREATE TABLE T1( I1 CHAR(12),
                 I2 NUMBER(9,3),
```

```
                    I3 CHAR DEFAULT 'C',
                    I4 NUMBER(9,3))
  PARTITION BY RANGE( I1 )
  ( PARTITION P1 VALUES DEFAULT )
  TABLESPACE SYS_TBS_DISK_DATA;

CREATE UNIQUE INDEX T1IDX ON T1 ( I2 DESC );
ALTER TABLE T1 ADD CONSTRAINT T1_UK2 UNIQUE(I2,I3,I4);

CREATE TABLE T2 ( I1 INTEGER, I2 INTEGER, I3 INTEGER ) TABLESPACE
SYS_TBS_DISK_DATA;

INSERT INTO T1(I2) VALUES (960);
INSERT INTO T1(I2) VALUES (742);
INSERT INTO T2 VALUES ( 2, 1, 1 );

UPDATE T1,T2 SET T1.I2 = 10;
```

- Actual Results

```
Altibase server abnormal termination
```

- Expected Results

```
[ERR-11058 : The row already exists in a unique index.]
```

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49773 ERR-4108A : Queue not found error may occur when executing DEQUEUE statement without using INTO clause in EXECUTE IMMEDIATE statement in PSM.

module

`qp-psm-trigger-execute`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fixes a phenomenon in which an ERR-4108A : Queue not found error occurs when executing a DEQUEUE statement without using the INTO clause in the EXECUTE IMMEDIATE statement in PSM. Fixed the problem of deleting queue information from the session when the bug condition is satisfied while the queue is empty.

This bug occurs when creating and running queues and PSMs in the order given below. See Reproduction conditions for an actual implementation example.

1. Create a queue
2. Create PSM

   - Execute DEQUEUE statement as dynamic SQL
   - Do not use INTO clause in EXECUTE IMMEDIATE statement

3. Execute PSM
4. Execute arbitrary DDL statements
5. Execute PSM

Workaround

Use the INTO clause in the EXECUTE IMMEDIATE statement.

```
EXECUTE IMMEDIATE('DEQUEUE MESSAGE INTO OUT1 FROM q1') INTO OUT1;
```

How to reproduce this bug

- Reproduction conditions

```
CREATE QUEUE q1(1000);

CREATE OR REPLACE PROCEDURE dq_test ()
AS
    OUT1 VARCHAR(1000);
BEGIN
    EXECUTE IMMEDIATE('DEQUEUE MESSAGE INTO OUT1 FROM q1');
    PRINTLN(OUT1);
END;
/

EXEC dq_test;

CREATE TABLE t1 (c1 INTEGER);

EXEC dq_test;
```

- Actual Results

```
[ERR-4108A : Queue not found
```

- Expected Results

```
Execute success.
```

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49779 When a library object is changed, the storage package body in which the object is used must be changed to a state that requires compilation.

module

`qp-psm-trigger-execute`

Category

`Functional Error`

Reproducibility

`Always`

Description

Changing a library object with the CREATE OR REPLACE statement changes the stored package body in which the object is used to a state that requires compilation (invalid).

Workaround

`none`

How to reproduce this bug

- Reproduction conditions

```
CREATE OR REPLACE LIBRARY lib1 AS 'normal.so';

CREATE OR REPLACE PACKAGE pkg1 AS
PROCEDURE proc1( a1 IN VARCHAR(30), a2 OUT VARCHAR(30) );
END;
/

CREATE OR REPLACE PACKAGE BODY pkg1 AS
PROCEDURE proc1( a1 IN VARCHAR(30), A2 OUT VARCHAR(30) )
AS
LANGUAGE C
LIBRARY lib1
NAME "andy_upper";
END;
/

CREATE OR REPLACE LIBRARY lib1 AS 'normal.so';
```

```
SELECT USER_ID, PACKAGE_NAME, PACKAGE_TYPE, STATUS FROM
SYSTEM_.SYS_PACKAGES_;
```

- Actual Results

```
USER_ID       PACKAGE_NAME                       PACKAGE_TYPE STATUS
--------------------------------------------------------------------------
2             PKG1                               6            0
2             PKG1                               7            0
2 rows selected.
```

- Expected Results

```
USER_ID       PACKAGE_NAME                       PACKAGE_TYPE STATUS
--------------------------------------------------------------------------
2             PKG1                               6            0
2             PKG1                               7            1
2 rows selected.
```

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49786 Improves stability issues caused by inactive indexes when transaction rollback is performed when an exception occurs during disk index reorganization.

module

`sm`

Category

`Fatal`

Reproducibility

`Always`

Description

Fixes an abnormal shutdown of the Altibase server due to an inactive index when performing a transaction rollback due to an exception occurring during disk index reorganization.

Workaround

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49796 [That had return update result] error may occur when executing a performance view search query of Altibase 7.1 consisting of x$ and v$ in an Altibase 6.5.1 or earlier client.

module

`qp`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fixes a phenomenon in which the [That had return update result] error occurs when querying the performance view of Altibase 7.1 in the JDBC driver of Altibase 6.5.1 or earlier.

Workaround

Set the OPTIMIZER_PERFORMANCE_VIEW property value to 0 on the Altibase 7.1 server. This can be changed with the ALTER SYSTEM statement. Changing the value of the OPTIMIZER_PERFORMANCE_VIEW property can degrade performance view query performance.

How to reproduce this bug

- Reproduction conditions

```
Connection sConn = getConnection(aIp, aPort);
try
{
    Statement sStmt = sConn.createStatement();
    ResultSet sRs = sStmt.executeQuery("SELECT 1 FROM dual");
    if (sRs.next())
```

```
    {
        System.out.println("[SUCCESS] SELECT 1 FROM dual ==> " +
sRs.getString(1));
    }
    sRs.close();
    sStmt.close();
}
catch (SQLException aEx)
{
    System.out.println("[FAIL] " + aEx.getMessage());
}
sConn.close();
```

- Actual Results

```
[FAIL] That had return update result
```

- Expected Results

```
[SUCCESS] SELECT 1 FROM dual ==> 1
```

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49807 aexport may crash when extracting range partitioned table objects (RANGE_USING_HASH) using hash.

module

`ux-aexport`

Category

`Memory Error`

Reproducibility

`Rare`

Description

Fixed a bug where aexport crashes when extracting a range partitioned table object (RANGE_USING_HASH) using hash.

Workaround

`none`

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49813 Fixes a phenomenon in which unnecessary text appears when executing altierr.

module

`sm`

Category

`Message Error`

Reproducibility

`Always`

Description

Fixes a phenomenon in which unnecessary text appears when executing altierr.

Workaround

`none`

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49841 [The primary key column count of the replicated table does not match] error occurs when executing the BUILD OFFLINE META statement for a replication object with a different name in the replication pair.

module

`rp-jdbcAdapter`

Category

`Functional Error`

Reproducibility

`Always`

Description

This is a bug that occurs in the process of using the offline option to reflect untransmitted logs to the target database when an Altibase server failure occurs in an Adapter for JDBC or Adapter for Oracle environment.

When executing the BUILD OFFLINE META statement, the criteria for sorting the information in the sender metafile (replication_name_META_NEW.bin, replication_name_META_OLD.bin) Fix the problem caused by the difference in sorting criteria of the replication target table information of the server performing the offline option.

This bug occurs when performing the BUILD OFFLINE META statement for a replication object with a different name for the replication pair. A replication object with a different name in a replication pair means that the user_name.table_name of the FROM clause that sets the replication target table is different from the user_name.table_name of the TO clause, as shown in the example below.

```
CREATE REPLICATION ala FOR ANALYSIS OPTIONS META_LOGGING WITH '127.0.0.1', 30300
  FROM test.tablea TO scott.tablea,
  FROM sys.tableb  TO scott.tableb,
  FROM sys.tablec  TO scott.tablec,
  FROM sys.tabled  TO scott.tabled,
  FROM sys.tablee  TO scott.tablee;
```

Workaround

Set the sort result of user_name.table_name in the FROM clause and user_name.table_name in the TO clause to be the same.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Changes

- Performance view
- Property
- Compile Option
- Error Code

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.1.0.7.9	6.5.1	8.10.1	7.1.7	7.4.7

> You can check the module version change history in [Version Histories](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](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

- IPC_SEM_KEY
- IPC_SHM_KEY
- IPCDA_SEM_KEY
- IPCDA_SHM_KEY

Changed Properties

Deleted Properties

Performance Views

Added Performance Views

Changed Performance Views

Deleted Performance Views