Altibase 7.1.0.8.0 Patch Notes

Table Of Contents

New Features

BUG-49904 Adds the ability to assign static IP addresses to duplicate senders.

module

`rp-sender`

Category

`Functionality`

Reproducibility

`Always`

Description

Adds the ability to assign static IP addresses to duplicate senders.

To use this function, you must set the Altibase server property REPLICATION_SENDER_IP.

Changes

- Performance view
- Property
  - `REPLICATION_SENDER_IP`

    This property sets the IP address of the replication sender. You can enter ANY or an IP address as the value.

    In the default value of ANY, all IP addresses of local servers that create replication objects can be used for replication communication, and the IP address assigned by the OS is used as the sender IP address. If you set the IP address as a value, only the IP address you set will be used when communicating with the remote server (receiver). Multiple IP addresses can be set by adding REPLICATION_SENDER_IP = value, which are used as sender IP addresses in order. The IP address can be entered in the form of IPv4, IPv6, or IPv6 extended address. This property is supported from Altibase 7.1.0.8.0 and is applied when the duplication communication method is TCP.
- Compile Option
- Error Code

Fixed Bugs

BUG-48624 Improve stability issues caused by type set regeneration and PSM concurrency issues.

module

`qp-trigger-execute`

Category

`Fatal`

Reproducibility

`Rare`

Description

Fixed an issue where, when a type set or package specification is referenced in PSM, the Altibase server abnormally terminates when the corresponding type set or package specification is recreated in one session and PSM is repeatedly executed in another session.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49094 Resolves a stability problem that occurs when using a column of a view where view merging has occurred in the CONNECT BY clause of the HAVING clause.

module

`qp-dml-pvo`

Category

`Assert`

Reproducibility

`Always`

Description

Resolves a stability problem that occurs when using a column of a view where view merging has occurred in the CONNECT BY clause of the HAVING clause.

How to reproduce this bug

- Reproduction conditions

```
CREATE TABLE T1 ( I1 INT, I2 INT);
SELECT I1
  FROM (SELECT * FROM T1 ) AS V1
 GROUP BY I1
HAVING I1 IN (SELECT I1 FROM T1 START WITH I1 = V1.I1 CONNECT BY PRIOR I1 =
I2);
```

- Actual Results

  The following message appears and the Altibase server abnormally terminates.

```
[ERR-31455 : Failed to work because an internal exception occurred from an
OS.[Contact Altibase's Support Center]]
```

- Expected Results

```
I1
--------------
No rows selected.
```

Workaround

Use the NO_MERGE hint.

```
SELECT /*+ NO_MERGE(V1) */ I1
  FROM (SELECT * FROM T1 ) AS V1
 GROUP BY I1
HAVING I1 IN (SELECT I1 FROM T1 START WITH I1 = V1.I1 CONNECT BY PRIOR I1 = I2);
```

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49712 In a replication object with a primary partition table, dropping a specific partition table from the replication table will break the replication sender.

module

`rp-control`

Category

`Functional Error`

Reproducibility

`Frequence`

Description

In a replication object including the default partition table, if a specific partition table is deleted from the replication table, the replication sender is stopped.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49737 Fix security vulnerability of JDBC driver detected by Veracode.

module

`mm-jdbc`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fixes security vulnerabilities in JDBC drivers detected by Veracode, a security vulnerability inspection tool.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49774 Referencing a performance view in a regular expression function when the value of the REGEXP_MODE property is 1 results in an error.

module

`qp`

Category

`Functional Error`

Reproducibility

`Always`

Description

Referencing a performance view in a regular expression function when the value of the REGEXP_MODE property is 1 results in an error.

How to reproduce this bug

- Reproduction conditions

  ```
  ALTER SESSION SET REGEXP_MODE=1;
  SELECT * FROM V$NLS_TERRITORY WHERE REGEXP_LIKE(CONVERT(NAME, 'UTF8'), 'C');
  ```

- Actual Results

```
NAME
--------------------------------------------
No rows selected.
```

- Expected Results

```
NAME
--------------------------------------------
AMERICA
CANADA
CATALONIA
CHILE
CHINA
CIS
COLOMBIA
COSTA RICA
CROATIA
CYPRUS
CZECH REPUBLIC
CZECHOSLOVAKIA
ECUADOR
FRANCE
FYR MACEDONIA
GREECE
ICELAND
MACEDONIA
MEXICO
MOROCCO
NICARAGUA
PUERTO RICO
SOUTH AFRICA
23 rows selected.
```

Workaround

```
ALTER SESSONI SET REGEXP_MODE = 0;
```

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49825 When the NORMALFORM_MAXIMUM property is 0, stability problems that occur when executing queries with tables and subqueries with PARALLEL specified are improved.

module

`qp`

Category

`Fatal`

Reproducibility

`Always`

Description

When the NORMALFORM_MAXIMUM property is 0, fixes the abnormal termination of the Altibase server when executing a query with a table and subquery for which PARALLEL is specified.

This bug occurs intermittently when executing a query that satisfies all of the following conditions.

- Set NORMALFORM_MAXIMUM property to 0
- Create a table by specifying PARALLEL
- Use tables created by specifying PARALLEL in the FROM clause
- Use a subquery in the WHERE clause
- Use USE_FULL_STORE_NL hint for subquery or use USE_FULL_STORE_NL hint and RESULT_CACHE hint together
- In addition to the join condition in the WHERE clause of the subquery, another condition using the columns of the second table in the FROM clause is used.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49834 When the definition of a synonym accessed from a stored procedure (or stored function and package) changes, it is not reflected in the stored procedure (or stored function and package).

module

`qp-ddl-dcl-execute`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fix an issue where a synonym created as an alias of a stored procedure or stored function would not be saved as an associated object when used in another stored procedure (or stored function or package).

This bug occurs when performing a stored procedure (or stored function and package) that satisfies all of the conditions below.

- Synonyms created as aliases for stored procedures or stored functions
- Use a synonym created as an alias of a stored procedure or stored function in another stored procedure (or stored function or package)
- Recreate by changing the definition of a synonym created as an alias of a stored procedure or stored function

patch impact

When there are synonyms and stored procedures (or stored functions and packages) that satisfy the bug conditions, the Actual Results of the stored procedures (or stored functions and packages) before and after the patch are different.

How to reproduce this bug

- Reproduction conditions

```sql
DROP TABLE T1;
DROP TABLE T2;
DROP SYNONYM TABLE_SYM;
DROP PROCEDURE PROC1;
DROP PROCEDURE PROC2;
DROP PROCEDURE PROC3;
DROP SYNONYM PROC_SYM;

CREATE TABLE T1 (C1 VARCHAR(10));
CREATE TABLE T2 (C1 VARCHAR(10));
CREATE SYNONYM TABLE_SYM FOR T1;

INSERT INTO T1 VALUES('T1');
```

```
INSERT INTO T2 VALUES('T2');

CREATE OR REPLACE PROCEDURE PROC1 AS
BEGIN
    PRINTLN('PROC1');
END;
/

CREATE OR REPLACE PROCEDURE PROC2 AS
BEGIN
    PRINTLN('PROC2');
END;
/

CREATE SYNONYM PROC_SYM FOR PROC1;

CREATE OR REPLACE PROCEDURE PROC3 AS
VAR1 VARCHAR(10);
BEGIN
    SELECT C1 INTO VAR1 FROM TABLE_SYM;
    PRINTLN(VAR1);
    PROC_SYM;
END;
/

EXEC PROC3;
CREATE OR REPLACE SYNONYM TABLE_SYM FOR T2;

EXEC PROC3;
CREATE OR REPLACE SYNONYM PROC_SYM FOR PROC2;

EXEC PROC3;
```

- Actual Results

```
iSQL> EXEC PROC3;
T2
PROC1
Execute success.
```

- Expected Results

```
iSQL> EXEC PROC3;
T2
PROC2
Execute success.
```

Workaround

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49845 [ERR-0109E: Internal server error.] error occurs when performing BEGIN BACKUP for a volatile tablespace.

module

`sm`

Category

`Assert`

Reproducibility

`Always`

Description

Fixes a phenomenon in which [ERR-0109E: Internal server error.] error occurs when ALTER TABLESPACE ... BEGIN BACKUP is executed for a volatile tablespace. Since volatile tablespace is not a target for database backup, change the [ERR-11101 : Unable to back up a vilatile tablespace.] error to occur.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49846 Hard parsing occurs when repeatedly executing a query statement using the package's synonym.

module

`qp-select-pvo`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fixed an issue where hard parsing occurred when repeatedly executing queries using synonyms in the package.

Hard parsing occurs because execution plans registered in the SQL Plan Cache are not shared due to missing synonym validation of the package during query processing.

After reflecting this bug, soft parsing is performed if the package has not changed since the creation of the execution plan when the query statement using the synonym of the package is repeatedly executed.

How to reproduce this bug

- Reproduction conditions

```
CREATE OR REPLACE PACKAGE PKG1 AS
FUNCTION FUNC1 RETURN INTEGER;
END;
/
CREATE OR REPLACE PACKAGE BODY PKG1 AS
FUNCTION FUNC1 RETURN INTEGER AS
BEGIN
  RETURN 1;
END;
END;
/
CREATE SYNONYM SYN2 FOR SYS.PKG1;
SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
SELECT SYN2.FUNC1 FROM DUAL;
SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
SELECT SYN2.FUNC1 FROM DUAL;
SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
```

- Actual Results

```
iSQL> SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
No rows selected.
iSQL> SELECT SYN2.FUNC1 FROM DUAL;
SYN2.FUNC1 : 1
```

```
1 row selected.
iSQL> SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
SQL_TEXT_ID           : 00500
SQL_TEXT              : SELECT SYN2.FUNC1 FROM DUAL
CHILD_PCO_COUNT       : 1
CHILD_PCO_CREATE_COUNT : 1
PLAN_CACHE_KEEP       : UNKEEP
1 row selected.
iSQL> SELECT SYN2.FUNC1 FROM DUAL;
SYN2.FUNC1 : 1
1 row selected.
iSQL> SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
SQL_TEXT_ID           : 00500
SQL_TEXT              : SELECT SYN2.FUNC1 FROM DUAL
CHILD_PCO_COUNT       : 2
CHILD_PCO_CREATE_COUNT : 2
PLAN_CACHE_KEEP       : UNKEEP
```

- Expected Results

```
iSQL> SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
No rows selected.
iSQL> SELECT SYN2.FUNC1 FROM DUAL;
SYN2.FUNC1 : 1
1 row selected.
iSQL> SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
SQL_TEXT_ID           : 00500
SQL_TEXT              : SELECT SYN2.FUNC1 FROM DUAL
CHILD_PCO_COUNT       : 1
CHILD_PCO_CREATE_COUNT : 1
PLAN_CACHE_KEEP       : UNKEEP
1 row selected.
iSQL> SELECT SYN2.FUNC1 FROM DUAL;
SYN2.FUNC1 : 1
1 row selected.
iSQL> SELECT * FROM V$SQL_PLAN_CACHE_SQLTEXT WHERE SQL_TEXT LIKE
'SELECT%SYN2%DUAL';
SQL_TEXT_ID           : 00500
SQL_TEXT              : SELECT SYN2.FUNC1 FROM DUAL
CHILD_PCO_COUNT       : 1
CHILD_PCO_CREATE_COUNT : 1
PLAN_CACHE_KEEP       : UNKEEP
1 row selected.
```

Workaround

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49849 In a non-autocommit environment, if PSM execution fails after an explicit or implicit commit occurs during PSM execution, the Altibase server abnormally terminates.

module

`sm`

Category

`Fatal`

Reproducibility

`Always`

Description

Fixes a phenomenon in which the Altibase server abnormally terminates when PSM execution fails in a non-autocommit environment.

When PSM starts in a transaction, the Altibase server sets a PSM savepoint, which is a rollback point if PSM execution fails. This bug is a problem in which the PSM savepoint is initialized during PSM execution. If PSM execution fails, the Altibase server abnormally terminates with a memory reference error.

The conditions under which this bug occurs are:

- Altibase client version 5.3.3 ~ 6.1.1

- Altibase server version 6.5.1, 7.1

- non autocommit mode

- Performed in the following order within one transaction

    - Execute PSM
    - Explicit or implicit commit occurs
    - Error during PSM execution

When the Altibase server abnormally terminates, the following message is recorded in the trace log.

altibase_sm.log

```
[2022/08/03 17:18:36 14C7F79][PID:7667][Thread-139792713553664][LWP-7670]
Invalid PSMSavepoint
```

altibase_error.log

```
[2022/08/03 17:18:36 14C7F7A][PID:7667][Thread-139792713553664][LWP-7670]
IDE_Assert( 0 ), [smxSavepointMgr.cpp:664], errno=[11]
```

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49850 An ERR-31455 error occurs when calling a subprogram from another stored procedure or stored function that uses arguments that default to variables in the package specification.

module

`qp-PSM-trigger-pvo`

Category

`Memory Error`

Reproducibility

`Always`

Description

Fixes a phenomenon in which an ERR-31455 error occurs when calling a subprogram that uses a package specification variable as a default value from another stored procedure or stored function.

How to reproduce this bug

- Reproduction conditions

```
DROP PACKAGE PGK1;
DROP PROCEDURE PROC1;

CREATE OR REPLACE PACKAGE PKG1 AS
V2 INTEGER := 1;
PROCEDURE SUB1( P1 IN INTEGER DEFAULT V2);
END;
```

```
     /

CREATE OR REPLACE PROCEDURE PROC1 AS
BEGIN
PKG1.SUB1;
END;
     /
```

- Actual Results

```
[ERR-31455 : Failed to work because an internal exception occurred from an
OS.[Contact Altibase's Support Center]]
```

- Expected Results

```
Create success.
```

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49857 Change the maximum value of the TRANSACTION_SEGMENT_COUNT property in altibase.properties to 16384.

module

`sm`

Category

`Other`

Reproducibility

`Always`

Description

Change the maximum value of the TRANSACTION_SEGMENT_COUNT property in altibase.properties to 16384. The maximum value of the TRANSACTION_SEGMENT_COUNT property has been changed to 16384 from Altibase 7.1.0.7.3. (related bug : BUG-49668)

Altibase 7.1.0.7.9 or earlier

```
$ grep TRANSACTION_SEGMENT_COUNT $ALTIBASE_HOME/conf/altibase.properties
TRANSACTION_SEGMENT_COUNT = 256 # ( 1 ~ 512 )
```

Altibase 7.1.0.8.0 or higher

```
$ grep TRANSACTION_SEGMENT_COUNT $ALTIBASE_HOME/conf/altibase.properties
TRANSACTION_SEGMENT_COUNT = 256 # ( 1 ~ 16384 )
```

BUG-49867 When creating a disk index, the message [[IDX_CRE] 2. Merge] is output to altibase_sm.log, and then the system waits infinitely.

module

`sm-disk-page`

Category

`Hang`

Reproducibility

`Always`

Description

Fixes a phenomenon in which the available space of the disk buffer pool cannot be used and waits indefinitely when creating a disk index. Improved the condition under which dirty pages in the delayed flush list are reflected to disk.

How to reproduce this bug

- Reproduction conditions
- Actual Results
- Expected Results

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49868 Addresses a stability issue that occurs when using views that fail to recompile due to exceeding EXECUTE_STMT_MEMORY_MAXIMUM.

module

`qp-ddl-dcl-execute`

Category

`Fatal`

Reproducibility

`Always`

Description

Fixes an issue where, when DDL is executed on an object used in a view subquery, if recompilation of the view fails due to exceeding EXECUTE_STMT_MEMORY_MAXIMUM, the view state remains in an abnormal state, causing an abnormal termination of the Altibase server when using the view.

When recompilation of a view fails due to memory constraints, DDL is handled as a failure to prevent abnormal termination of the Altibase server.

The changes before/after the reflection of this bug are as follows.

before bug fix

- Even if view recompilation fails due to memory constraints, DDL execution succeeds.
- Views are left in an abnormal state that does not complete recompilation.

after bug fix

- If view recompilation fails due to memory constraints, DDL fails.
- The user must adjust the EXECUTE_STMT_MEMORY_MAXIMUM property and execute DDL again.

How to reproduce this bug

- Reproduction conditions

```
ALTER SYSTEM SET EXECUTE_STMT_MEMORY_MAXIMUM = 1048576;

CREATE TABLE T1
( I1 INTEGER,
  I2 INTEGER
)
PARTITION BY RANGE(I1)
(
  PARTITION P1 VALUES DEFAULT
);

CREATE VIEW V1 AS SELECT * FROM T1;
```

```
DROP TABLE T1;

CREATE TABLE T1
( I1 INTEGER,
  I2 INTEGER,
  I3 INTEGER
)
PARTITION BY RANGE(I1)
(
  PARTITION P1 VALUES DEFAULT
);


SELECT * FROM V1;
```

- Actual Results

```
iSQL> CREATE TABLE T1
( I1 INTEGER,
  I2 INTEGER,
  I3 INTEGER
)
PARTITION BY RANGE(I1)
(
  PARTITION P1 VALUES DEFAULT
);
Create success.

iSQL> SELECT * FROM V1;
[ERR-31455 : Failed to work because an internal exception occurred from an
OS.[Contact Altibase's Support Center]]
```

- Expected Results

```
iSQL> CREATE TABLE T1
( I1 INTEGER,
  I2 INTEGER,
  I3 INTEGER
)
PARTITION BY RANGE(I1)
(
  PARTITION P1 VALUES DEFAULT
);
[ERR-01067 : The memory size allocated for the statement has exceeded the
maximum limit ( Name : Query_Execute, Wanted Memory Size : 1114112, Max size
: 1048576 ).]

iSQL> ALTER SYSTEM SET EXECUTE_STMT_MEMORY_MAXIMUM = 10485760;
Alter success.

iSQL> CREATE TABLE T1 ( I1 INTEGER, I2 INTEGER, I3 INTEGER ) PARTITION BY
RANGE(I1) ( PARTITION P1 VALUES DEFAULT  );
Create success.

iSQL> SELECT * FROM V1;
```

```
I1          I2          I3
----------------------------------------
No rows selected.
```

Workaround

`none`

Changes

- Performance view
- Property
- Compile Option
- Error Code

BUG-49894 TYPESET objects are not extracted when aexport is performed.

module

`ux-aexport`

Category

`Functional Error`

Reproducibility

`Always`

Description

Fixes an issue where TYPESET objects are not extracted when performing aexport.

How to reproduce this bug

- Reproduction conditions

```
CREATE TYPESET type1
AS
  TYPE rec1 IS RECORD (c1 INTEGER, c2 INTEGER);
  TYPE arr1 IS TABLE OF rec1 INDEX BY INTEGER;
END;
/
CREATE FUNCTION func1(i1 INTEGER)
RETURN type1.arr1
AS
  v1 type1.arr1;
BEGIN
  for i in 1 .. i1 loop
    v1[i].c1 := i;
    v1[i].c2 := i * i;
  END LOOP;
  RETURN v1;
END;
```

```
/
```

```
$ aexport -u sys -p manager -s localhost
$ grep TYPE1 *.sql
```

- Actual Results

```
$ grep TYPE1 *.sql
ALL_OBJECT.sql:RETURN TYPE1.ARR1
ALL_OBJECT.sql:  V1 TYPE1.ARR1;
```

- Expected Results

```
$ grep TYPE1 *.sql
ALL_CRT_VIEW_PROC.sql:CREATE TYPESET TYPE1
ALL_CRT_VIEW_PROC.sql:RETURN TYPE1.ARR1
ALL_CRT_VIEW_PROC.sql:  V1 TYPE1.ARR1;
```

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
7.1.0.8.0	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

- REPLICATION_SENDER_IP

Changed Properties

Deleted Properties

Performance Views

Added Performance Views

Changed Performance Views

Deleted Performance Views