

Altibase 6.5.1.9.3 Patch Notes

Table of Contents

- [New Features](#)

- [BUG-49776 IPC 및 IPCDA 채널을 생성하는 데 필요한 공유 메모리와 세마포어의 키를 사용자가 정의한 값으로 설정하는 기능을 추가합니다.](#)

- [Fixed Bugs](#)

- [BUG-48767 같은 파티션 키 값으로 여러 세션에서 동시에 파티션을 분할할 때 최솟값과 최댓값이 동일한 파티션이 생성될 수 있습니다.](#)
- [BUG-48882 파티션 테이블 대상으로 이중화 SYNC 수행 시 이중화 송신자 시작 이전 시점의 로그를 읽는 문제를 수정합니다.](#)
- [BUG-48907 이중화 SET 절에서 예외 발생 시 메모리 누수가 발생할 수 있는 문제를 개선합니다.](#)
- [BUG-49094 뷰 머징이 발생한 뷰의 컬럼을 HAVING 절의 CONNECT BY 절에서 사용할 때 발생하는 안정성 문제를 개선합니다.](#)
- [BUG-49524 ALTER TABLE table name MODIFY COLUMN 수행 시 테이블스페이스 공간 부족으로 에러 발생 시 예외 처리를 개선합니다.](#)
- [BUG-49773 PSM에서 EXECUTE IMMEDIATE 문에 INTO 절을 사용하지 않고 DEQUEUE 문을 수행할 때 ERR-4108A : Queue not found 에러가 발생할 수 있습니다.](#)
- [BUG-49778 디스크 인덱스가 선택되고 집계 함수를 사용한 서브쿼리가 Subquery Unnesting으로 쿼리 변환 시 결과 오류가 발생합니다.](#)
- [BUG-49779 라이브러리\(library\) 객체를 변경하면 해당 객체가 사용된 저장 패키지 바디를 컴파일이 필요한 상태로 변경해야 합니다.](#)
- [BUG-49786 디스크 인덱스 재구성 중 예외가 발생하여 트랜잭션 롤백 수행 시, 비활성화 상태의 인덱스 때문에 발생하는 안정성 문제를 개선합니다.](#)
- [BUG-49804 ANSI 문법의 INNER JOIN으로 구성된 SQL의 수행 성능이 저하되는 문제를 개선합니다.](#)
- [BUG-49827 \\$ALTIBASE_HOME/sample 아래의 APRE, CAPI, CHECKSERVER, SPATIAL, SQLCLI 샘플 코드를 ODBC 표준에 맞춰 수정합니다.](#)
- [BUG-49834 저장 프로시저\(또는 저장 함수 및 패키지\)에서 접근하는 시노ним의 정의가 변경될 때 저장 프로시저\(또는 저장 함수 및 패키지\)에 반영되지 않습니다.](#)
- [BUG-49850 패키지 스펙의 변수를 기본값으로 하는 인자를 사용한 서브 프로그램을 다른 저장 프로시저 또는 저장 함수에서 호출할 때 ERR-31455 에러가 발생합니다.](#)
- [BUG-49868 EXECUTE STMT MEMORY MAXIMUM 초과로 재컴파일이 실패한 뷰 사용 시 발생하는 안정성 문제를 개선합니다.](#)
- [BUG-49894 aexport 수행 시 TYPESET 객체가 추출되지 않습니다.](#)
- [BUG-49900 Altibase 6.5.1 Adapter for Oracle 지원 OS에 SunOS 10을 추가합니다.](#)

- [Changes](#)

- [Version Info](#)
- [호환성](#)
- [프로퍼티](#)
- [성능 뷰](#)

New Features

BUG-49776 IPC 및 IPCDA 채널을 생성하는 데 필요한 공유 메모리와 세마포어의 키를 사용자가 정의한 값으로 설정하는 기능을 추가합니다.

Module

cm-ipc

Category

Functionality

재현 빈도

Impossible

설명

IPC 및 IPCDA 채널을 생성하는 데 필요한 공유 메모리와 세마포어의 키를 사용자가 정의한 값으로 설정하는 기능을 추가합니다.

사용자는 다음 4가지 프로퍼티를 사용하여 공유 메모리와 세마포어의 키를 설정할 수 있습니다.

- IPC_SEM_KEY
- IPC_SHM_KEY
- IPCDA_SEM_KEY
- IPCDA_SHM_KEY

IPC와 IPCDA 채널은 Altibase 서버 구동 시 생성되는데, 공유 메모리/세마포어 키가 사용 중이거나 다른 이유로 공유 메모리/세마포어를 생성하지 못하면 Altibase 서버 구동은 실패합니다. 이때, Altibase 서버 트레이스 로그 altibase_boot.log에서 시스템 에러(errno)를 확인하고 그에 따른 적절한 처리를 해야 합니다.

프로퍼티 설명은 [General Reference-1.Data Types & Altibase Properties](#) 매뉴얼에서 확인할 수 있습니다.

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

Workaround

변경사항

- Performance view
- Property
 - IPC_SEM_KEY

IPC 채널을 생성하는 데 필요한 세마포어 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다.

기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 세마포어 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPC_SEM_KEY 값을 기준으로 IPC_SEM_KEY부터 IPC_SEM_KEY + (IPC_CHANNEL_COUNT + 1)만큼의 연속된 세마포어 키를 사용하여 IPC 채널을 생성한다. +1은 SYS 사용자가 관리자 모드(sysdba)로 접속하기 위해 예약된 IPC 채널이다. 예를 들어 IPC_SEM_KEY 값이 10000이고 IPC_CHANNEL_COUNT 값이 1000이면 세마포어 키로 10000부터 11000까지 사용한다.

- **IPC_SHM_KEY**

IPC 채널을 생성하는 데 필요한 공유 메모리 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다. 기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 공유 메모리 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPC_SHM_KEY 값을 공유 메모리 키로 사용한다.

- **IPCDA_SEM_KEY**

IPCD 채널을 생성하는 데 필요한 세마포어 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다. 기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 세마포어 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPCDA_SEM_KEY 값을 기준으로 IPCDA_SEM_KEY부터 IPCDA_SEM_KEY + IPC_CHANNEL_COUNT만큼의 연속된 세마포어 키를 사용하여 IPCDA 채널을 생성한다. 예를 들어 IPCDA_SEM_KEY 값이 10000이고 IPC_CHANNEL_COUNT 값이 1000이면 세마포어 키로 10000부터 10999까지 사용한다.

- **IPCD_SHM_KEY**

IPCD 채널을 생성하는 데 필요한 공유 메모리 키(key)를 사용자가 정의한 값으로 설정하는 프로퍼티이다.

기본값은 0으로 Altibase 서버 프로세스의 프로세스 식별자(PID)를 기준으로 공유 메모리 키를 자동으로 생성한다. 0이 아닌 값을 설정하면 IPCDA_SHM_KEY 값을 기준으로 연속된 키 2개를 공유 메모리 키로 사용한다. 예를 들어 IPCDA_SHM_KEY=10000이면 10000, 10001을 공유 메모리 키 값으로 사용한다.

- Compile Option

- Error Code

에러 메시지 2가지가 추가되었습니다.

- IPC와 IPCDA 채널 생성 시 Altibase 서버 프로퍼티에 정의된 키로 공유 메모리를 생성할 수 없을 때

```
0x710C6 ( 463046) cmERR_ABORT_SHMGET_ERROR_WITH_KEY A system call error occurred
while creating shared memory for <0%>. [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.
```

- IPC와 IPCDA 채널 생성 시 Altibase 서버 프로퍼티에 정의된 키로 세마포어를 생성할 수 없을 때

```
0x710C7 ( 463047) cmERR_ABORT_SEMGET_ERROR_WITH_KEY A system call error occurred
while creating semaphore for <0%>. [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 같은 파티션 키 값으로 여러 세션에서 동시에 파티션을 분할할 때 최솟값과 최댓값이 동일한 파티션이 생성될 수 있습니다.

Module

qp-ddl-dcl-pvo

Category

Functional Error

재현 빈도

Always

설명

같은 파티션 키 값으로 여러 세션에서 동시에 파티션을 분할할 때, 동시성 문제로 최솟값과 최댓값 이 동일한 파티션이 생성되는 현상을 수정합니다. SPLIT PARTITION 구문 실행 시점에 최솟값과 최댓값이 같으면 에러가 발생하도록 수정하였습니다.

Workaround

없음

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-48882 파티션 테이블 대상으로 이중화 SYNC 수행 시 이중화 송신자 시작 이전 시점의 로그를 읽는 문제를 수정합니다.

Module

rp

Category

Functionality

재현 빈도

Always

설명

ALTER REPLICATION replication_name SYNC TABLE ~ PARTITION 수행 시 이중화 송신자 시작 이전 시점의 로그를 읽는 문제를 수정합니다. 이 버그 현상 발생 시 이중화 대상 서버 간 데이터가 일치하지 않을 수 있습니다.

Workaround

없음

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-48907 이중화 SET 절에서 예외 발생 시 메모리 누수가 발생할 수 있는 문제를 개선합니다.

Module

rp

Category

Memory Error

재현 빈도

Always

설명

다음과 같은 이중화 SET 절에서 예외 발생 시 메모리 누수가 발생할 수 있는 문제를 개선합니다.

- ALTER REPLICATION replication_name SET PARALLEL
- ALTER REPLICATION replication_name SET GROUPING ENABLE/DISABLE
- ALTER REPLICATION replication_name SET OFFLINE ENABLE WITH

Workaround

없음

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49094 뷰 머징이 발생한 뷰의 컬럼을 HAVING 절의 CONNECT BY 절에서 사용할 때 발생하는 안정성 문제를 개선합니다.

Module

qp-dml-pvo

Category

Assert

재현 빈도

Always

설명

뷰 머징이 발생한 뷰의 컬럼을 HAVING 절의 CONNECT BY 절에서 사용할 때 Altibase 서버가 비정상 종료하는 현상을 개선합니다.

Workaround

NO_MERGE 힌트를 사용합니다.

재현 방법

- 재현 절차

```
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);
```

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

- 예상 결과

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

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49524 ALTER TABLE table_name MODIFY COLUMN 수행 시 테이블스페이스 공간 부족으로 에러 발생 시 예외 처리를 개선합니다.

Module

qp

Category

Fatal

재현 빈도

Always

설명

ALTER TABLE MODIFY COLUMN 수행 시 테이블스페이스 공간 부족으로 에러 발생 시 Altibase 서버가 비정상 종료하는 현상을 수정합니다.

Workaround

없음

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

변경사항

- Performance view
- Property

BUG-49773 PSM에서 EXECUTE IMMEDIATE 문에 INTO 절을 사용하지 않고 DEQUEUE 문을 수행할 때 ERR-4108A : Queue not found 에러가 발생할 수 있습니다.

Module

qp-psm-trigger-execute

Category

Functional Error

재현 빈도

Always

설명

PSM에서 EXECUTE IMMEDIATE 문에 INTO 절을 사용하지 않고 DEQUEUE 문을 수행할 때 ERR-4108A : Queue not found 에러가 발생하는 현상을 수정합니다. 큐가 비어있는 상태에서 버그 발생 조건을 만족하면 세션에서 큐 정보를 삭제하는 문제를 수정하였습니다.

이 버그는 아래의 순서대로 큐와 PSM을 생성하고 수행할 때 발생합니다. 실제 수행 예시는 재현 절차를 참고하세요.

1. 큐 생성
2. PSM 생성
 - DEQUEUE 문을 동적 SQL로 수행
 - EXECUTE IMMEDIATE 문에 INTO 절 사용하지 않음
3. PSM 수행
4. 임의의 DDL 문 수행
5. PSM 수행

Workaround

없음

재현 방법

- 재현 절차

```
DROP QUEUE q1;
DROP TABLE t1;
CREATE QUEUE q1(1000);

CREATE OR REPLACE PROCEDURE dq_test ()
AS
    OUT1 VARCHAR(1000);
BEGIN
```

```
/

EXEC dq_test;

CREATE TABLE t1 (c1 INTEGER);

EXEC dq_test;
```

- 수행 결과

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

- 예상 결과

```
Execute success.
```

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49778 디스크 인덱스가 선택되고 집계 함수를 사용한 서브쿼리가 Subquery Unnesting으로 쿼리 변환 시 결과 오류가 발생합니다.

Module

qp

Category

Functional Error

재현 빈도

Always

설명

디스크 인덱스가 선택되고 집계 함수를 사용한 서브쿼리가 Subquery Unnesting으로 쿼리 변환 시 결과 오류가 발생하는 현상을 수정합니다. 결과 오류를 수정한 버그로 조건에 만족하는 질의 수행 시 결과 및 실행 계획이 변경됩니다. 버그 수정 전/후 실행 계획 차이는 Actual Results, Expected Results 항목을 참고하시기 바랍니다.

Workaround

NO_UNNEST 힌트를 사용합니다.

재현 방법

- 재현 절차

```

DROP TABLE T2;
CREATE TABLE T2
(
  "TRADE_DATE" CHAR(8) NOT NULL,
  "DATE_ID"    NUMERIC(10),
  "DAY_TYPE"   NUMERIC(4)
)TABLESPACE SYS_TBS_DISK_DATA;

ALTER TABLE T2 ADD CONSTRAINT "T2_SKSDATEI" PRIMARY KEY("TRADE_DATE");

INSERT INTO T2 VALUES('19801207', 1, 1);
INSERT INTO T2 VALUES('20220625', 2, 2);
INSERT INTO T2 VALUES('20220627', 3, 3);

ALTER SESSION SET OPTIMIZER_DISK_INDEX_COST_ADJ = 1;
ALTER SESSION SET EXPLAIN PLAN = ON;
ALTER SESSION SET TRCLOG_DETAIL_PREDICATE = 1;

SELECT MAX(TRADE_DATE) FROM T2 WHERE TRADE_DATE < (SELECT MAX(TRADE_DATE) FROM T2);

```

- 수행 결과

```

MAX(TRADE_DATE)
-----
19801207
1 row selected.

-----
PROJECT ( COLUMN_COUNT: 1, TUPLE_SIZE: 10, COST: 381.30 )
FILTER
[ FILTER ]
$VIEW1.COL1 IS NOT NULL
VIEW ( ACCESS: 1, COST: 379.98 )
PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 14, COST: 260.58 )
WINDOW SORT ( ITEM_SIZE: 24, ITEM_COUNT: 3, DISK_PAGE_COUNT: 0, ACCESS: 7,
SORT_COUNT: 0, COST: 251.34 )
[ ANALYTIC FUNCTION INFO ]
SORT_KEY[0]: ()
MAX(TRADE_DATE) OVER ()
SCAN ( TABLE: SYS.T2, INDEX: SYS.t2_SKSDATEI, RANGE SCAN, ACCESS: 3,
DISK_PAGE_COUNT: 64, COST: 117.82 )
-----

```

- 예상 결과

```

MAX(TRADE_DATE)

```

```

-----
PROJECT ( COLUMN_COUNT: 1, TUPLE_SIZE: 10, COST: 384.37 )
  GROUP-AGGREGATION ( ITEM_SIZE: 32, GROUP_COUNT: 1, BUCKET_COUNT: 1, ACCESS: 1, COST:
384.37 )
    FILTER
      [ FILTER ]
      $VIEW1.COL1 IS NOT NULL
    VIEW ( ACCESS: 3, COST: 379.98 )
      PROJECT ( COLUMN_COUNT: 2, TUPLE_SIZE: 14, COST: 260.58 )
        WINDOW SORT ( ITEM_SIZE: 24, ITEM_COUNT: 3, DISK_PAGE_COUNT: 0, ACCESS: 9,
SORT_COUNT: 0, COST: 251.34 )
          [ ANALYTIC FUNCTION INFO ]
          SORT_KEY[0]: ()
          MAX(TRADE_DATE) OVER ()
          SCAN ( TABLE: SYS.T2, INDEX: SYS.t2_SKSDATEI, FULL SCAN, ACCESS: 3,
DISK_PAGE_COUNT: 64, COST: 117.82 )
-----

```

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49779 라이브러리(library) 객체를 변경하면 해당 객체가 사용된 저장 패키지 바디를 컴파일 이 필요한 상태로 변경해야 합니다.

Module

qp-psm-trigger-execute

Category

Functional Error

재현 빈도

Always

설명

라이브러리(library) 객체를 CREATE OR REPLACE 문으로 변경하면 해당 객체가 사용된 저장 패키지 바디를 컴파일 이 필요한 상태(invalid)로 변경합니다.

Workaround

없음

재현 방법

- 재현 절차

```
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_;
```

- 수행 결과

USER_ID	PACKAGE_NAME	PACKAGE_TYPE	STATUS
2	PKG1	6	0
2	PKG1	7	0

2 rows selected.

- 예상 결과

USER_ID	PACKAGE_NAME	PACKAGE_TYPE	STATUS
2	PKG1	6	0
2	PKG1	7	1

2 rows selected.

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49786 디스크 인덱스 재구성 중 예외가 발생하여 트랜잭션 롤백 수행 시, 비활성화 상태의 인덱스 때문에 발생하는 안정성 문제를 개선합니다.

Module

sm

Category

Fatal

재현 빈도

Always

설명

디스크 인덱스 재구성 중 예외가 발생하여 트랜잭션 롤백 수행 시, 비활성화 상태의 인덱스 때문에 Altibase 서버가 비정상 종료하는 문제를 개선합니다.

Workaround

없음

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49804 ANSI 문법의 INNER JOIN으로 구성된 SQL의 수행 성능이 저하 되는 문제를 개선합니다.

Module

qp

Category

Enhancement

재현 빈도

Always

설명

ANSI INNER JOIN으로 구성된 SQL의 질의 최적화 과정에서 SQL 작성 순서를 고려하지 않아 SQL의 수행 성능이 저하되는 문제를 개선합니다.

질의 최적화 과정에서 ANSI INNER JOIN을 일반 INNER JOIN으로 변환할 때, SQL 작성 순서를 고려하여 변환하도록 수정하였습니다.

- 발생 조건

이 버그 발생 조건은 ANSI 문법의 INNER JOIN으로만 구성된 질의문입니다.

- 반영 버전

이 버그는 Altibase 6.5.1.9.3 이상에 적용됩니다.

- 적용 방법

이 버그의 변경 사항을 적용하려면 비공개 Altibase 서버 프로퍼티 `__OPTIMIZER_ANSI_INNER_JOIN_CONVERT`의 값을 2로 변경해야 합니다. 이 프로퍼티는 ALTER SYSTEM으로 변경할 수 있으면 영구 적용하려면 `altibase.properties` 파일에 `__OPTIMIZER_ANSI_INNER_JOIN_CONVERT = 2`를 추가하고 Altibase 서버를 재시작해야 합니다.

- 영향도

버그 조건에 해당하는 질의문의 실행 계획이 달라집니다. 변경 전/후 차이는 Actual Results, Expected Results를 참고하세요.

Workaround

이 버그가 반영되지 않은 Altibase 서버 버전에서 ANSI INNER JOIN으로 구성된 SQL의 성능 저하 현상을 조치하려면 `__OPTIMIZER_ANSI_INNER_JOIN_CONVERT = 0`으로 변경합니다.

```
ALTER SYSTEM SET __OPTIMIZER_ANSI_INNER_JOIN_CONVERT = 0;
```

재현 방법

- 재현 절차

```
DROP TABLE T1;
DROP TABLE T2;
DROP TABLE T3;
DROP TABLE T4;
DROP TABLE T5;
DROP TABLE T6;

CREATE TABLE T1 ( I1 INT , I2 INT , I3 INT );
CREATE TABLE T2 ( I1 INT , I2 INT , I3 INT );
CREATE TABLE T3 ( I1 INT , I2 INT , I3 INT );
```

```

INSERT INTO T1 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <=10;
INSERT INTO T2 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <=10;
INSERT INTO T3 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <=10;
INSERT INTO T4 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <=10;
INSERT INTO T5 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <=10;
INSERT INTO T6 SELECT LEVEL, LEVEL, LEVEL FROM DUAL CONNECT BY LEVEL <=10;

```

```

ALTER SESSION SET EXPLAIN PLAN = ON;

```

```

SELECT H1.I1
FROM T1 H1
INNER JOIN T2 H2 ON H1.I1 = H2.I1
INNER JOIN T3 H3 ON H2.I1 = H3.I1
INNER JOIN T4 H4 ON H2.I1 = H4.I1
INNER JOIN T5 H5 ON H3.I1 = H5.I1
INNER JOIN T6 H6 ON H3.I1 = H6.I1;

```

- 수행 결과

```

-----
PROJECT ( COLUMN_COUNT: 18, TUPLE_SIZE: 72, COST: 71358818392868040.00 )
  JOIN ( METHOD: HASH, COST: 44619453052271392.00 )
    SCAN ( TABLE: SYS.T5 H5, FULL SCAN, ACCESS: 10, COST: 116.76 )
    HASH ( ITEM_SIZE: 56, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
44619453052271392.00 )
      JOIN ( METHOD: HASH, COST: 43573684623425.38 )
        SCAN ( TABLE: SYS.T4 H4, FULL SCAN, ACCESS: 10, COST: 116.76 )
        HASH ( ITEM_SIZE: 48, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
43573684623425.38 )
          JOIN ( METHOD: HASH, COST: 42552427988.33 )
            SCAN ( TABLE: SYS.T1 H1, FULL SCAN, ACCESS: 10, COST: 116.76 )
            HASH ( ITEM_SIZE: 40, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
42552427988.33 )
              JOIN ( METHOD: HASH, COST: 41556235.38 )
                SCAN ( TABLE: SYS.T2 H2, FULL SCAN, ACCESS: 10, COST: 116.76 )
                HASH ( ITEM_SIZE: 32, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
41556235.38 )
                  JOIN ( METHOD: HASH, COST: 41243.85 )
                    SCAN ( TABLE: SYS.T3 H3, FULL SCAN, ACCESS: 10, COST: 116.76 )
                    HASH ( ITEM_SIZE: 24, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
41243.85 )
                      SCAN ( TABLE: SYS.T6 H6, FULL SCAN, ACCESS: 10, COST: 116.76 )

```

- 예상 결과

```

-----
PROJECT ( COLUMN_COUNT: 1, TUPLE_SIZE: 4, COST: 46104973348971200.00 )

```



```

HASH ( ITEM_SIZE: 32, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
44619453052271392.00 )
JOIN ( METHOD: HASH, COST: 43573684623425.38 )
SCAN ( TABLE: SYS.T5 H5, FULL SCAN, ACCESS: 10, COST: 116.76 )
HASH ( ITEM_SIZE: 32, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
43573684623425.38 )
JOIN ( METHOD: HASH, COST: 42552427988.33 )
SCAN ( TABLE: SYS.T4 H4, FULL SCAN, ACCESS: 10, COST: 116.76 )
HASH ( ITEM_SIZE: 40, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
42552427988.33 )
JOIN ( METHOD: HASH, COST: 41556235.38 )
SCAN ( TABLE: SYS.T3 H3, FULL SCAN, ACCESS: 10, COST: 116.76 )
HASH ( ITEM_SIZE: 32, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
41556235.38 )
JOIN ( METHOD: HASH, COST: 41243.85 )
SCAN ( TABLE: SYS.T1 H1, FULL SCAN, ACCESS: 10, COST: 116.76 )
HASH ( ITEM_SIZE: 24, ITEM_COUNT: 10, BUCKET_COUNT: 1024, ACCESS: 10, COST:
41243.85 )
SCAN ( TABLE: SYS.T2 H2, FULL SCAN, ACCESS: 10, COST: 116.76 )
-----

```

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49827 \$ALTIBASE_HOME/sample 아래의 APRE, CAPI, CHECKSERVER, SPATIAL, SQLCLI 샘플 코드를 ODBC 표준에 맞춰 수정합니다.

Module

mm

Category

other

재현 빈도

Always

설명

\$ALTIBASE_HOME/sample 아래의 APRE, CAPI, CHECKSERVER, SPATIAL, SQLCLI 샘플 코드를 ODBC 표준에 맞춰 수정합니다.

재현 방법

- 재현 절차
- 수행 결과
- 예상 결과

Workaround

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49834 저장 프로시저(또는 저장 함수 및 패키지)에서 접근하는 시노님의 정의가 변경될 때 저장 프로시저(또는 저장 함수 및 패키지)에 반영되지 않습니다.

Module

qp-ddl-dcl-execute

Category

Functional Error

재현 빈도

Always

설명

저장 프로시저나 저장 함수의 별칭으로 생성한 시노님을 또 다른 저장 프로시저(또는 저장 함수 및 패키지)에서 사용할 때 해당 시노님을 연관된 객체로 저장하지 않는 문제를 수정합니다.

이 버그는 아래 조건을 만족할 때 발생합니다.

- 저장 프로시저나 저장 함수의 별칭으로 생성한 시노님
- 저장 프로시저나 저장 함수의 별칭으로 생성한 시노님을 또 다른 저장 프로시저(또는 저장 함수 및 패키지)에서 사용
- 저장 프로시저나 저장 함수의 별칭으로 생성한 시노님의 정의를 변경하여 재생성

Workaround

없음

재현 방법

- 재현 절차
-

```

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;
/

SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
      , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3, 'TYPESET',
4, 'SYNONYM', 'ETC') OBJECT_TYPE
  FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
 WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;

EXEC PROC3;

CREATE OR REPLACE SYNONYM TABLE_SYM FOR T2;

SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
      , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3, 'TYPESET',
4, 'SYNONYM', 'ETC') OBJECT_TYPE
  FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B

```

```
EXEC PROC3;

CREATE OR REPLACE SYNONYM PROC_SYM FOR PROC2;

SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
      , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3, 'TYPESET',
4, 'SYNONYM', 'ETC') OBJECT_TYPE
  FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
 WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;

EXEC PROC3;
```

- 수행 결과

```
iSQL> SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
      , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3,
'TYPESET', 4, 'SYNONYM', 'ETC') OBJECT_TYPE
  FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
 WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;
PROC_NAME          STATUS      RELATED_OBJECT_NAME
OBJECT_TYPE
-----
PROC3              0          PROC1              PROC
PROC3              0          PRINTLN            PROC
PROC3              0          T1                TABLE
PROC3              0          TABLE_SYM        SYNONYM
4 rows selected.
```

```
iSQL> EXEC PROC3;
T1
PROC1
Execute success.
```

```
iSQL> CREATE OR REPLACE SYNONYM TABLE_SYM FOR T2;
Create success.
```

```
iSQL> SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
      , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3,
'TYPESET', 4, 'SYNONYM', 'ETC') OBJECT_TYPE
  FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
 WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;
PROC_NAME          STATUS      RELATED_OBJECT_NAME
OBJECT_TYPE
-----
PROC3              1          PROC1              PROC
PROC3              1          PRINTLN            PROC
PROC3              1          T1                TABLE
```

```

iSQL> EXEC PROC3;
T2
PROC1
Execute success.

iSQL> CREATE OR REPLACE SYNONYM PROC_SYM FOR PROC2;
Create success.

iSQL> SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
           , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3,
'TYPESET', 4, 'SYNONYM', 'ETC') OBJECT_TYPE
           FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
           WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;
PROC_NAME                STATUS      RELATED_OBJECT_NAME
OBJECT_TYPE
-----
-----
PROC3                     0          PROC1                PROC
PROC3                     0          PRINTLN             PROC
PROC3                     0          T1                  TABLE
PROC3                     0          TABLE_SYM          SYNONYM
4 rows selected.

iSQL> EXEC PROC3;
T2
PROC1
Execute success.

```

- 예상 결과

```

iSQL> SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
           , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3,
'TYPESET', 4, 'SYNONYM', 'ETC') OBJECT_TYPE
           FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
           WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;
PROC_NAME                STATUS      RELATED_OBJECT_NAME
OBJECT_TYPE
-----
-----
PROC3                     0          PROC1                PROC
PROC3                     0          PROC_SYM             SYNONYM
PROC3                     0          PRINTLN             PROC
PROC3                     0          T1                  TABLE
PROC3                     0          TABLE_SYM          SYNONYM
5 rows selected.

iSQL> EXEC PROC3;
--1

```

```
isql> CREATE OR REPLACE SYNONYM TABLE_SYM FOR T2;
```

Create success.

```
isql> SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
          , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3,
'TYPESET', 4, 'SYNONYM', 'ETC') OBJECT_TYPE
```

```
      FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
```

```
      WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;
```

PROC_NAME

STATUS

RELATED_OBJECT_NAME

OBJECT_TYPE

```
-----
-----
PROC3          1          PROC1          PROC
PROC3          1          PROC_SYM        SYNONYM
PROC3          1          PRINTLN         PROC
PROC3          1          T1              TABLE
PROC3          1          TABLE_SYM      SYNONYM
```

5 rows selected.

```
isql> EXEC PROC3;
```

T2

PROC1

Execute success.

```
isql> CREATE OR REPLACE SYNONYM PROC_SYM FOR PROC2;
```

Create success.

```
isql> SELECT B.PROC_NAME, B.STATUS, A.RELATED_OBJECT_NAME
          , DECODE(A.RELATED_OBJECT_TYPE, 0, 'PROC', 1, 'FUNC', 2, 'TABLE', 3,
'TYPESET', 4, 'SYNONYM', 'ETC') OBJECT_TYPE
```

```
      FROM SYSTEM.SYS_PROC_RELATED A, SYSTEM.SYS_PROCEDURES B
```

```
      WHERE B.PROC_NAME='PROC3' AND A.PROC_OID=B.PROC_OID;
```

PROC_NAME

STATUS

RELATED_OBJECT_NAME

OBJECT_TYPE

```
-----
-----
PROC3          1          PROC1          PROC
PROC3          1          PROC_SYM        SYNONYM
PROC3          1          PRINTLN         PROC
PROC3          1          T1              TABLE
PROC3          1          TABLE_SYM      SYNONYM
```

5 rows selected.

```
isql> EXEC PROC3;
```

T2

PROC2

Execute success.

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49850 패키지 스펙의 변수를 기본값으로 하는 인자를 사용한 서브 프로그램을 다른 저장 프로시저 또는 저장 함수에서 호출할 때 ERR-31455 에러가 발생합니다.

Module

qp-psm-trigger-pvo

Category

Memory Error

재현 빈도

Always

Workaround

없음

설명

패키지 스펙의 변수를 기본값으로 하는 인자를 사용한 서브 프로그램을 다른 저장 프로시저 또는 저장 함수에서 호출할 때 ERR-31455 : Failed to work because an internal exception occurred from an OS.[Contact Altibase's Support Center] 에러가 발생하는 현상을 수정합니다.

재현 방법

- 재현 절차

```

DROP PACKAGE PGK1;
DROP PROCEDURE PROC1;

CREATE OR REPLACE PACKAGE PGK1 AS
V2 INTEGER := 1;
PROCEDURE SUB1( P1 IN INTEGER DEFAULT V2);
END;
/

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

- 수행 결과

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

- 예상 결과

```
Create success.
```

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49868 EXECUTE_STMT_MEMORY_MAXIMUM 초과로 재컴파일이 실패한 뷰 사용 시 발생하는 안정성 문제를 개선합니다.

Module

qp-ddl-dcl-execute

Category

Fatal

재현 빈도

Always

설명

뷰의 부질의에 사용된 객체에 DDL이 수행될 때 EXECUTE_STMT_MEMORY_MAXIMUM 초과로 뷰의 재컴파일이 실패하면, 뷰의 상태가 비정상적인 상태로 남아 해당 뷰 사용 시 Altibase 서버의 비정상 종료를 유발하는 문제를 수정합니다.

메모리 제약으로 뷰의 재컴파일이 실패하면 DDL을 실패 처리하여 Altibase 서버가 비정상 종료하는 현상이 발생하지 않도록 개선하였습니다.

이 버그의 반영 전/후 변경 사항은 아래와 같습니다.

- 버그 반영 전**

메모리 제약으로 뷰의 재컴파일이 실패하더라도 DDL 수행은 성공합니다.

뷰는 재컴파일을 완료하지 않은 비정상적인 상태로 남습니다.

- 버그 반영 후**

메모리 제약으로 뷰의 재컴파일이 실패하면 DDL을 실패 처리합니다.

사용자는 EXECUTE_STMT_MEMORY_MAXIMUM 프로퍼티를 조정 후 DDL을 다시 수행해야 합니다.

Workaround

없음

재현 방법

- 재현 절차

```
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;
```

- 수행 결과

```
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]]
```

- 예상 결과

```
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
```

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49894 aexport 수행 시 TYPESET 객체가 추출되지 않습니다.

Module

ux-aexport

Category

Functional Error

재현 빈도

Always

설명

aexport 프로퍼티 DROP 의 설정이 ON 일 때 drop typeset 구문이 추출되지 않는 문제를 수정합니다.

Workaround

없음

재현 방법

- 재현 절차

```
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;
/
```

- 수행 결과

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

- 예상 결과

```
$ grep TYPE1 ALL_OBJECT.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;
```

변경사항

- Performance view
- Property
- Compile Option
- Error Code

BUG-49900 Altibase 6.5.1 Adapter for Oracle 지원 OS에 SunOS 10을 추가합니다.

Module

rp-oraAdapter

Category

Portability

재현 빈도

Always

설명

Altibase 6.5.1 Adapter for Oracle 지원 OS에 SunOS 10을 추가합니다. Adapter for Oracle 6.5.1.9.3 부터 지원합니다.

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
6.5.1.9.3	6.3.1	8.1.1	7.1.3	7.4.5

Altibase 6.5.1 패치 버전 별 히스토리는 [Version Histories](#)에서 확인할 수 있다.

호환성

Database binary version

데이터베이스 바이너리 버전은 변경되지 않았다.

데이터베이스 바이너리 버전은 데이터베이스 이미지 파일과 로그파일의 호환성을 나타낸다. 이 버전이 다른 경우의 패치(업그레이드 포함)는 데이터베이스를 재구성해야 한다.

Meta Version

메타 버전은 변경되지 않았다.

CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

Replication protocol Version

Replication 프로토콜 버전은 변경되지 않았다.

프로퍼티

추가된 프로퍼티

- [IPC SEM KEY](#)
- [IPC SHM KEY](#)
- [IPCD A SEM KEY](#)
- [IPCD A SHM KEY](#)

변경된 프로퍼티

삭제된 프로퍼티

성능 뷰

추가된 성능 뷰

변경된 성능 뷰

삭제된 성능 뷰