

Altibase 7.3.0.0.4 Patch Notes

- [New Features](#)
 - [BUG-50695 aku에서 6노드 지원](#)
 - [BUG-50703 Row Referencing 절을 사용하는 트리거에서 사용하지 않는 컬럼을 내부적으로 복사하지 않도록 개선합니다.](#)
 - [BUG-50713 잘못된 패킷으로 Protocol header error가 발생한 경우, 에러메시지에 헤더 내용 출력하도록 개선합니다.](#)
- [Fixed Bugs](#)
 - [BUG-50660 REFERENCING NEW ROW 절을 사용하는 트리거가 참조하 테이블의 테이블 스페이스를 변경 후 트리거 동작시, 서버가 비정상 종료하는 경우가 있어서 수정합니다.](#)
 - [BUG-50714 Package의 constant 또는 nocopy 옵션이 있는 변수를 직접 변경할 때 잘못 동작하는 문제를 수정합니다.](#)
 - [BUG-50729 종료된 쓰레드의 스택 크기가 v\\$memstat 에서 계속 출력됩니다.](#)
 - [BUG-50758 APRE에서 double quote 문자에 대한 escape 처리 누락](#)
- [Changes](#)
 - [Version Info](#)
 - [호환성](#)
 - [프로퍼티](#)
 - [성능 뷰](#)

New Features

BUG-50695 aku에서 6노드 지원

- **module** : aku
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : aku가 기존에는 4 노드까지 지원 가능하였는데, 6노드를 지원하도록 수정하였습니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-50703 Row Referencing 절을 사용하는 트리거에서 사용하지 않는 컬럼을 내부적으로 복사하지 않도록 개선합니다.

- **module** : qp-psm-trigger-execute
- **Category** : Enhancement
- **재현 빈도** : Always
- **설명** : Row Referencing 절을 사용하는 트리거가 실행될 때, 내부적으로 참조 레코드를 특정 변수에 복사하여 처리하는데, 이 과정에서 불필요한 컬럼도 복사 되었습니다. 이제는 실제로 사용되는 컬럼만 복사하도록 개선되었습니다.

이 패치의 적용으로 Referencing 절을 사용하는 트리거에서 LOB 컬럼을 사용하지 않는 경우, [ERR-21031 : Unable to convert the data type.] 오류가 발생하는 문제가 해결되었습니다.

또한, 트리거의 동작을 유발하는 DML의 실행 성능이 개선되었습니다.

주의: 이번 패치에서는 아래의 문제는 해결되지 않습니다.

- 100MB이상의 LOB 데이터가 있는 테이블에 Row Referencing 절을 사용하는 트리거에서 실제로 LOB 컬럼을 참조하는 경우, [ERR-21031 : Unable to convert the data type.] 오류가 발생할 수 있습니다.

- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과

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

BUG-50713 잘못된 패킷으로 Protocol header error가 발생한 경우, 에러메시지에 헤더 내용 출력하도록 개선합니다.

- **module** : cm
- **Category** : Functionality
- **재현 빈도** : Rare
- **설명** : 잘못된 패킷이 수신되어 Protocol header error가 발생 했을 때, 헤더 내용을 덤프해서 문제 분석에 도움을 주도록 수정하였습니다.
 - 수정전 : ERR-7101d(errno=0) Protocol header error.(TCP 127.0.0.1:41462)
 - 수정후 : ERR-710cc(errno=0) Protocol header error.(TCP 127.0.0.1:41462, 0, 3132333435363738393031323334353637383930)
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code
 - 수정전 : ERR-7101d(errno=0) Protocol header error.(TCP 127.0.0.1:41462)
 - 수정후 : ERR-710cc(errno=0) Protocol header error.(TCP 127.0.0.1:41462, 0, 3132333435363738393031323334353637383930)

Fixed Bugs

BUG-50660 REFERENCING NEW ROW절을 사용하는 트리거가 참조 하 테이블의 테이블 스페이스를 변경 후 트리거 동작시, 서버가 비정상 종료하는 경우가 있어서 수정합니다.

- **module** : qp-ddl-dcl-execute
- **Category** : Fatal
- **재현 빈도** : Always
- **설명** : 트리거가 동작할때, 아래의 조건을 모두 만족하는 경우 비정상 종료하는 문제를 수정하였습니다.
 1. REFERENCING NEW ROW절을 사용하는 트리거가 참조하는 테이블이 있고,
 2. 이 테이블의 테이블 스페이스를 아래와 같이 변경한 후
 - 메모리 테이블 스페이스라면, 디스크 테이블 스페이스로 변경
 - 디스크 테이블 스페이스라면, 메모리 테이블 스페이스로 변경
 3. 트리거가 동작하면서 trigger_event에 정의된 DML을 수행
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- **변경사항**
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-50714 Package의 constant 또는 nocopy 옵션이 있는 변수를 직접 변경할 때 잘못 동작하는 문제를 수정합니다.

- **module** : qp-psm-trigger-execute
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : Package의 constant 또는 nocopy 변수를 직접 변경할 때 잘못 동작하는 문제를 수정합니다.
 1. constant 옵션이 있는 변수의 경우 값이 변경되면 안되지만, 직접 변경하는 경우 값이 변경되는 문제를 수정했습니다.
 2. nocopy 옵션이 있는 변수의 경우 직접 값을 변경하는 경우 초기화 문제로 변경이 안되는 문제를 수정했습니다.
- **재현 방법**
 - 재현 절차

```

CREATE OR REPLACE PACKAGE PKG1 AS
VAR1 CONSTANT INTEGER := 1;
end;
/
EXEC PKG1.VAR1 := 100;
EXEC PRINTLN(PKG1.VAR1);

CREATE OR REPLACE PACKAGE PKG2 AS
TYPE ARR_TYPE IS TABLE OF CHAR(32000) INDEX BY INTEGER;
VAR1 NOCOPY ARR_TYPE;
VAR2 ARR_TYPE;
RET INTEGER;
END;
/
BEGIN
  FOR I IN 1 .. 10000 LOOP
    PKG2.VAR2(I) := I;
  END LOOP;
END;
/
EXEC PKG2.VAR1 := PKG2.VAR2;

```

수행 결과

```

iSQL> EXEC PKG1.VAR1 := 100;
Execute success.
iSQL> EXEC PRINTLN(PKG1.VAR1);
100
Execute success.
iSQL> EXEC PKG2.VAR1 := PKG2.VAR2;
[ERR-31439 : An array (element) is not initialized.
at line 1]

```

예상 결과

```

iSQL> EXEC PKG1.VAR1 := 100;
[ERR-3113F : The L-value of the ASSIGN statement is either a CONSTANT
variable or an IN parameter.
0001 : EXEC PKG1.VAR1 := 100
          ^      ^
]
iSQL> EXEC PRINTLN(PKG1.VAR1);
1
Execute success.
iSQL> EXEC PKG2.VAR1 := PKG2.VAR2;
Execute success.

```

• Workaround

```

BEGIN
  PKG1.VAR1 := 100;
END;
/
BEGIN
  PKG2.VAR1 := PKG2.VAR2;
END;
/

```

- 변경사항
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-50729 종료된 쓰레드의 스택 크기가 v\$memstat 에서 계속 출력됩니다.

- **module** : id
- **Category** : Functional Error
- **재현 빈도** : Always
- **설명** : 쓰레드 종료 시 스택이 해제되어도 V\$MEMSTAT 조회에 반영되지 않는 문제를 수정합니다.
이로 인해, 실제 메모리 사용량보다 더 많은 사용량으로 표시되는 문제가 해결되었습니다.
- **재현 방법**
 - 재현 절차
 - 수행 결과
 - 예상 결과
- **Workaround**
- 변경사항
 - Performance view
 - Property
 - Compile Option
 - Error Code

BUG-50758 APRE에서 double quote 문자에 대한 escape 처리 누락

- **module** : mm-apre
 - **Category** : Functional Error
 - **재현 빈도** : Always
 - **설명** : EXEC SQL 구문에 "(double quote)문자가 포함된 경우,
apre precompile후에 생성되는 .c, .cpp 파일에 escape 문자가 자동으로
추가되도록 수정하였습니다.
- 예)

.sc 파일 : EXEC SQL SELECT ' ' ' FROM DUAL;

==>

.c 파일 : ulpSqlstmt.stmt = (char *)"SELECT ' \" ' FROM DUAL";

- 재현 방법

- 재현 절차

.sc 파일에서 아래와 같이 EXEC SQL 문을 작성한다.

```
int main()
{
    EXEC SQL SELECT ' ' ' FROM DUAL;
}
```

- 수행 결과

.c 파일에서 아래와 같이 변환된다.

```
int main()
{
    /* SELECT ' \" ' FROM DUAL; */
    {
        struct ulpSqlstmt ulpSqlstmt;
        memset(&ulpSqlstmt, 0, sizeof(ulpSqlstmt));
        ulpSqlstmt.stmttype = 4;
        ulpSqlstmt.stmtname = NULL;
        ulpSqlstmt.ismt = 0;
        ulpSqlstmt.numofhostvar = 0;
        ulpSqlstmt.statusptr = NULL;
        ulpSqlstmt.errcodeptr = NULL;
        ulpSqlstmt.isatomic = 0;
        ulpSqlstmt.stmt = (char *)"SELECT ' \" ' FROM DUAL";
        ...
    }
}
```

- 예상 결과

.c 파일에서 아래와 같이 " 앞에 \이 추가된 것을 확인할 수 있다.

```
int main()
{
    /* SELECT ' \" ' FROM DUAL; */
    {
        struct ulpSqlstmt ulpSqlstmt;
        memset(&ulpSqlstmt, 0, sizeof(ulpSqlstmt));
        ulpSqlstmt.stmttype = 4;
        ulpSqlstmt.stmtname = NULL;
        ulpSqlstmt.ismt = 0;
        ulpSqlstmt.numofhostvar = 0;
        ulpSqlstmt.statusptr = NULL;
        ulpSqlstmt.errcodeptr = NULL;
        ulpSqlstmt.isatomic = 0;
        ulpSqlstmt.stmt = (char *)"SELECT ' \" ' FROM DUAL";
        ...
    }
}
```

```
}  
}
```

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

Changes

Version Info

altibase version	database binary version	meta version	cm protocol version	replication protocol version
7.3.0.0.4	7.3.0	9.3.1	7.1.8	7.4.9

Altibase 7.3 패치 버전별 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.

패치를 롤백하려는 경우, [메타다운그레이드](#)를 참고한다.

CM protocol Version

통신 프로토콜 버전은 변경되지 않았다.

Replication protocol Version

Replication 프로토콜 버전은 변경되지 않았다.

프로퍼티

추가된 프로퍼티

변경된 프로퍼티

삭제된 프로퍼티

성능 뷰

추가된 성능 뷰

변경된 성능 뷰

삭제된 성능 뷰

