

Real Alternative DBMS ALTIBASE, Since 1999

APRE*C/C++ New Features & 업그레이드 가이드

ALTIBASE 5.3.3

2010. 05



Copyright © 2000~2013 ALTIBASE Corporation. All Rights Reserved.

Document Control

Change Record

| Date | Author | Change Reference |
|------------|----------|------------------|
| 2010-05-04 | durusari | Created |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Reviews

| Date | Name (Position) |
|------------|--|
| 2010-01-20 | omegaman(SC), bluetheme(TS), hjkim(SC), fhan(SC) |
| 2010-05-04 | wlgml337(TC) |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Distribution

| Name | Location |
|------|----------|
| | |
| | |
| | |
| | |

목차

| | |
|---|----|
| 개요 | 4 |
| APRE*C/C++ | 5 |
| 용어 설명 | 5 |
| 도입 배경 | 5 |
| New Features 요약 | 5 |
| New Features 상세 | 6 |
| 유의사항 | 10 |
| 참고사항 | 11 |
| SES*C/C++ 환경에서 APRE*C/C++로의 업그레이드 | 12 |
| 변경사항 | 12 |
| 업그레이드 절차 | 12 |
| 유의사항 | 13 |

개요

본 문서는 SES*C/C++의 후속버전인 APRE*C/C++의 새로운 기능과 변경사항 및 업그레이드 절차에 대해 설명한다. 문서에 언급하지 않은 구체적인 사항은 [Precompiler 매뉴얼]을 참고하도록 한다.

문서의 기준은 APRE*C/C++가 제공되는 ALTIBASE 5.3.3이며 문서상의 실행결과는 5.3.3.7로 수행한 결과이다.

문서의 이해를 돕기 위해 아래 기술문서를 함께 참조하는 것을 권장한다.

- ALTIBASE APRE(SES)*C/C++ makefile
- ALTIBASE Precompiler 개발 가이드 (APRE*C/C++)

APRE*C/C++

ALTIBASE의 새로워진 Precompiler of Embedded SQL에 대하여 설명한다. 요약형태의 설명으로 기능에 대한 상세사용법과 같은 구체적인 사항은 [Precompiler 매뉴얼]을 참고하도록 한다.

용어설명

Precompiler of Embedded SQL

내장 SQL(Embedded SQL)이 포함된 소스코드를 입력 받아, 해당 내장 SQL을 실행시간 라이브러리 함수 호출로 변환한 소스코드를 생성하는 프로그램.

SES*C/C++

ALTIBASE 5.1.5 이하의 Precompiler of Embedded SQL의 약어로 소스코드로 C와 C++을 지원한다.

APRE*C/C++

ALTIBASE 5.3.3 이상의 Precompiler of Embedded SQL의 약어로 소스코드로 C와 C++을 지원한다. SES*C/C++와 비교하여 업그레이드 수준의 기능개선을 하였다.

도입배경

SES*C/C++는 내장 SQL만 처리가 가능하며 이 외에도 아래와 같은 다수의 제약을 가지고 있어 사용상에 불편함이 있었다.

- 매크로(macro) 처리 제약.
- 호스트변수 선언부(DECLARE SECTION) 외부에 호스트변수 선언 불가.
- 호스트변수 선언방식 및 사용 제약.
- 일부 표준 내장 SQL구문 미 지원.

ALTIBASE 사용자에게 Precompiler of Embedded SQL을 통한 개발편의성을 제공하기 위해서 이러한 제약은 우선적으로 개선되어야 할 사항이다.

New Features 요약

SES*C/C++에 존재하던 다수의 제약을 대폭 개선하였다. 기능 업데이트 수준을 넘어서는 업그레이드를 적용하면서 명칭 또한 APRE*C/C++(ALTIBASE C/C++ Precompiler of Embedded SQL)로 변경되었다.

APRE*C/C++에 새롭게 추가된 기능은 아래와 같다.

- 매크로 처리를 위한 Partial C Preprocessor 탑재.
- 호스트변수 선언부 외부의 호스트변수 선언을 위한 C Parser 탑재.
- 호스트변수 선언방식 및 사용제한 완화를 위한 라이브러리 재 작성.
- DECLARE STATEMENT 구문 추가지원.

이와 더불어 아래와 같은 개선을 하였다.

- WHENEVER 구문 사용시 함수 호출 가능.
- APRE*C/C++ 실행파일(apre) 명령옵션 변경 및 추가.
- 에러메세지 출력형태 변경.

New Features 상세

Partial C Preprocessor 탑재

아래와 같은 대부분의 매크로가 선언영역에 대한 제한 없이 처리가 가능하게 되었다.

✓ #include, #define, #if, #ifdef, #ifndef, #endif, #else, #elif

```
#define ALTIBASE
...
EXEC SQL BEGIN DECLARE SECTION;
...
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO T1 VALUES
#ifdef ALTIBASE
( :altibase );
#else
( :other_dbms );
#endif
...

```

C Parser 탑재

소스코드를 C로 작성하는 경우에 한하여 호스트변수 선언부 외부(DECLARE SECTION)에서도 호스트변수 선언이 가능하게 되었다.

```
...
int i = 10;
EXEC SQL BEGIN DECLARE SECTION;
int j;
int k;
EXEC SQL END DECLARE SECTION;
varchar vc[LEN];
...
EXEC SQL INSERT INTO T1 VALUES (:i, :j, :k, :vc);
...

```

호스트변수 선언방식 및 사용제한 완화

호스트변수 사용과 관련된 다수의 제약이 제거되었다. 상세내용은 아래와 같다.

1. 호스트변수 선언과 동시에 초기값을 할당 가능.

```
...
EXEC SQL BEGIN DECLARE SECTION;
```

```

int i = 10*10;
char j[10] = "abcd";
EXEC SQL END DECLARE SECTION;
...

```

2. typedef 후 구조체 정의 가능. (역으로도 가능)

```

...
EXEC SQL BEGIN DECLARE SECTION;
typedef struct department department;
struct department
{
    short dno;
    char dname[30+1];
    char dep_location[9+1];
};
EXEC SQL END DECLARE SECTION;
...

```

3. 내장 SQL문에서 배열형의 호스트변수 사용 시 배열요소 지정 가능.

```

...
EXEC SQL BEGIN DECLARE SECTION;
struct tag1
{
    int i1;
    int i2;
} var1[10];
EXEC SQL END DECLARE SECTION;

EXEC SQL INSERT INTO T1 (I1, I2) VALUES (:var1[0].i1, :var1[0].i2);
...

```

4. char *, struct * 외의 다른 데이터형도 포인터형 호스트변수로 사용 가능.

```

...
struct {
    int i;
    char c;
} structvar;

int *i_ptr = &structvar.i;
char *c_ptr = &structvar.c;
EXEC SQL SELECT i, c INTO :i_ptr, :c_ptr FROM TMP;
...

```

5. SELECT 구문의 INTO절에 ":" 없이 출력용 호스트변수 사용 가능.

```

...
EXEC SQL BEGIN DECLARE SECTION;
char name[10];
EXEC SQL END DECLARE SECTION;
EXEC SQL SELECT DEP_NAME INTO name FROM DEPT;
...

```

6. FOR 절의 입력 호스트변수가 배열 타입이 아니더라도 사용 가능.

```

...
EXEC SQL BEGIN DECLARE SECTION;
int cnt = 1;
int var1 = 10;
EXEC SQL END DECLARE SECTION;
...
EXEC SQL FOR :cnt INSERT INTO T1 VALUES (:var1);
...

```

7. 유니온(Union)형의 호스트변수 사용 가능.

```

...

```

```
EXEC SQL BEGIN DECLARE SECTION;
union uni {
    int a;
    char s[10];
} u;
EXEC SQL END DECLARE SECTION;
...
EXEC SQL INSERT INTO T1 VALUES ( :u.s );
...
```

DECLARE STATEMENT 구문 추가지원

표준 Embedded SQL 구문으로 DECLARE STATEMENT 구문을 추가 지원한다. 다른 내장 SQL구문에 사용될 수 있도록, SQL 구문이나 PL/SQL 블록에 대한 식별자를 선언할 수 있다.

```
...
EXEC SQL DECLARE my_statement STATEMENT;
EXEC SQL DECLARE emp_cursor CURSOR FOR my_statement;
EXEC SQL PREPARE my_statement FROM :my_string;
...
```

WHENEVER 구문 사용시 함수 호출 가능

WHENEVER <condition> DO <function> 와 같은 형태로 WHENEVER 구문 사용시 특정 함수를 호출할 수 있게 개선되었다.

```
...
EXEC SQL WHENEVER SQLERROR DO sql_error_occur();
...
```

APRE*C/C++ 실행파일(apre) 명령옵션 변경

1. -I

기존 SES*C/C++에서 `-include` 옵션으로 제공되던 옵션으로써 `-I` 로 명칭이 변경되었다. `precompile`시 `include`할 소스코드 파일의 경로를 명시한다. 이 옵션은 코드 내의 EXEC SQL OPTION (`INCLUDE=library_path`)와 같은 기능을 한다.

```
$ apre -t cpp -I$APP_HOME/include/main tmp.sc
```

APRE*C/C++ 실행파일(apre) 명령옵션 추가

1. -D

전처리(Preprocess) 시 사용될 매크로를 선언한다. 이 옵션은 코드 내의 `#define`과 같은 기능을 한다.

```
$ apre -DALTIBASE -DOTHER_DBMS -t cpp tmp.sc
```

2. -keyword

예약된 키워드들을 보여준다.

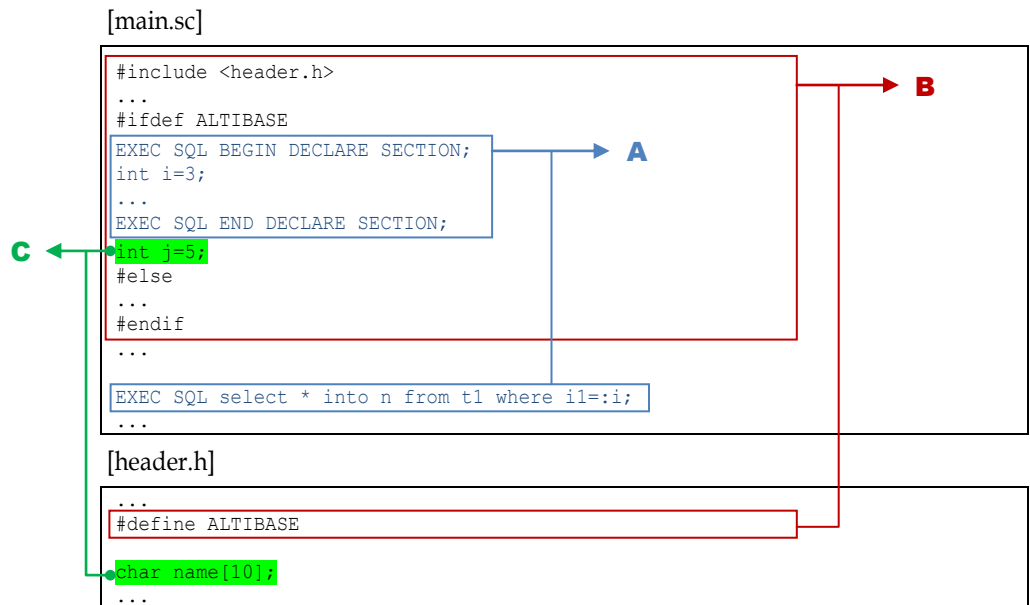
```
$ apre -keyword
```

3. -parse *parsing_mode*

파싱모드(parsing mode)를 지정하여 소스파일에 대한 precompile 범위를 결정한다. -parse 옵션의 파싱모드와 그에 따른 처리범위는 아래와 같다. -parse 옵션 자체를 생략할 경우는 기본적으로 파싱모드가 partial로 작동한다.

| 파싱 모드 | 내장 SQL A | 매크로 B | 선언부외부 호스트변수 C | 비고 |
|----------------|-----------------------|-------------------------------------|-------------------------------------|--|
| <i>none</i> | <input type="radio"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> SES*C/C++와 동일하게 작동 <input type="checkbox"/> #include 형식의 헤더파일은 처리하지 않음 |
| <i>partial</i> | <input type="radio"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> Partial C Preprocessor 추가 작동 <input type="checkbox"/> #include 형식의 헤더파일까지 처리 <input type="checkbox"/> APRE*C/C++의 기본 파싱모드 |
| <i>full</i> | <input type="radio"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> C Parser가 추가 작동 <input type="checkbox"/> #include 형식의 헤더파일까지 처리 <input type="checkbox"/> 단, C++ 스타일로 작성된 소스코드는 선언부외부 호스트변수 인식불가 |

[#include 형식의 헤더파일까지 처리]의 경우, 해당 헤더파일이 #include 형식으로 선언한 또 다른 헤더파일까지도 모두 처리된다.



예를 들어, 위 예제에서의 쿼리를 아래와 같이 매크로를 사용한 선택적인 쿼리로 변경 후

```

...
EXEC SQL select * into n from t1
#ifdef ALTIBASE
where i1=:i and i2 > 'A';
#else
where i1=:i;
#endif
...

```

파싱옵션을 생략하여 기본 파싱모드인 partial로 precompile 할 경우

```

$ apre -t c main.sc

```

실제 생성코드는 아래와 같이 매크로 처리가 완료된 상태로만 남게 된다. (매크로 처리 후 불필요하여 제거된 부분은 공백으로 치환된다)

```

...
/* select *      from t1
where i1=:i and i1> 'A'; */
{
    struct ulpSqlstmt ulpSqlstmt;
    ulpHostVar ulpHostVar[2];
    ulpSqlstmt.hostvalue = ulpHostVar;
    ulpSqlstmt.stmttype = 4;
    ulpSqlstmt.stmtname = NULL;
    ulpSqlstmt.ismt = 0;
...

```

기존 SES*C/C++와 동일하게 precompile 하기 위해 -parse none을 지정한 경우에는 매크로 처리기능이 작동하지 않음으로 변수 중복선언 에러가 발생한다.

```

$ apre -t c -parse none main.sc
-----
Altibase C/C++ Precompiler.
Release Version 5.3.3.7
Copyright 2000, ALTIBASE Corporation or its subsidiaries.
All Rights Reserved.
-----
[ERR-51011 : redefinition of 'i'.]

[ERR-204E : The symbol name [i] can't be added on the symbol table.]
File : main.sc
Line : 13
Offset: 24-24
Error_token:;

```

유의사항

문법(syntax) 강화

기존 SES*C/C++는 코드에서 EXEC SQL BEGIN/END DECLARE/ARGUMENT SECTION을 기술할 경우, 종결자인 “;” 이 누락되어도 무관하였으나 APRE*C/C++는 반드시 “;” 가 있어야 한다. 이는 타 DBMS Precompiler와도 동일한 제약사항이다.

아래는 END DECLARE SECTION에 종결자를 기재하지 않은 경우의 에러이다.

```

[ERR-302L : EXEC SQL END DECLARE SECTION is not exist.]

```

C++ 스타일의 소스코드 precompile

호스트변수 선언부외부에 호스트변수를 사용하기 위해서는 parse 옵션의 모드를 full로 해야 한다. 하지만, 파싱모드를 full로 설정하면 C 파서가 동작하기 때문에 C++ 스타일의 소스코드는 전처리 중 각종 파싱에러를 발생시킬 수 있다.

즉, C++ 스타일로 작성된 소스코드는 SES*C/C++와 같이 호스트변수 선언부에만 호스트변수를 선언하고, -parse 옵션을 명시적으로 지정하여 precompile할 경우에는 파싱모드를 partial 또는 none으로 해야 한다.

-D, -I 옵션

C/C++ 컴파일 단계에서 -D, -I 옵션을 사용한다면 대부분의 경우 APRE*C/C++를 통한 precompile 단계에서도 동일한 옵션을 주어야 올바른 precompile이 가능하다.

참고사항

-I 옵션

기존처럼 -include로 사용하여도 무방하나 향후 유지보수를 고려하여 새로운 옵션으로 변경하여 사용하는 것을 권장한다.

이진 데이터 타입 변경

기존에 사용되었던 SES_CLOB, SES_BLOB, SES_BINARY, SES_BYTES, SES_NIBBLE 타입의 이름이 APRE_CLOB, APRE_BLOB, APRE_BINARY, APRE_BYTES, APRE_NIBBLE로 변경됐다. 하위 호환성이 고려되었기에 기존 이름을 사용해도 무방하다.

precompile 수행시간

SES*C/C++와 비교하여 precompile 수행시간이 다소 느려졌다.

아래는 ALTIBASE 설치 시 제공되는 sample 예제 arrays1.sc(450 lines)를 SES*C/C++, APRE*C/C++로 각각 precompile을 수행한 시간을 측정한 결과이다.

| SES C/C++ Precompiler 3 Ver 5.1.5.53 | APRE C/C++ Precompiler Ver 5.3.3.7 |
|---|---|
| <pre>\$ time sesc -t cpp arrays1.sc -silent</pre> | <pre>\$ time apre -t cpp arrays1.sc -silent</pre> |
| <pre>real 0m0.203s user 0m0.168s sys 0m0.006s</pre> | <pre>real 0m0.278s user 0m0.245s sys 0m0.004s</pre> |

위 측정결과는 IBM AIX5.3 (Power5 1898 MHz 1EA)에서 precompile한 결과로 실제수행속도는 OS 환경에 따라 달라질 수 있다.

SES*C/C++환경에서 APRE*C/C++로의 업그레이드

SES*C/C++환경에서 APRE*C/C++로의 업그레이드 시 고려사항과 절차에 대해 설명한다.

변경사항

SES*C/C++에서 APRE*C/C++로 명칭이 변경되면서 아래와 같이 실행파일, 헤더파일, 라이브러리파일, 링크옵션, 실행파일 명령옵션 명칭이 일부분 변경되었다.

| 구분 | SES*C/C++ | APRE*C/C++ | 관련파일 경로 | 비고 |
|-----------|---------------|--------------------------|-------------------------|----|
| 실행파일 | sesc | apre | \$ALTIBASE_HOME/bin | 변경 |
| 헤더파일 | ses.h | ulpLibInterface.h | \$ALTIBASE_HOME/include | 변경 |
| 라이브러리파일 | libsesc.a | libapre.a | \$ALTIBASE_HOME/lib | 변경 |
| | libsesc_sl.so | libapre_sl.so | \$ALTIBASE_HOME/lib | 변경 |
| 링크옵션 | -lsesc | -lapre | - | 변경 |
| 실행파일 명령옵션 | -include | -include or -I | - | 추가 |

이 중 실행파일, 헤더파일 및 라이브러리파일은 하위 호환성이 고려되어 있어 별도로 조치할 사항이 없으나 실행파일명과 링크옵션은 반드시 변경되어야 한다.

또한, 필수적인 요소는 아니나 **-include** 옵션의 경우도 **-I** 로 변경하는 것을 권장한다.

업그레이드 절차

SES*C/C++ 개발환경에서 APRE*C/C++로 업그레이드시의 절차는 아래와 같다.

1. APRE*C/C++ 버전 확인

```
$ apre -v
```

기본적으로 최신버전의 APRE*C/C++를 사용하는 것을 권장하며 최소 5.3.3.15 이상을 사용하여야 한다.

- ✓ BUG-28392 Direct Execute 실패 시 메모리 증가 (fixed at 5.3.3.5)
- ✓ BUG-28588 null fetch시 statement 해제하지 않음 (fixed at 5.3.3.6)
- ✓ BUG-29745 SELECT 컬럼 절에 숫자가 있을 때 커서사용 시 접속해제가 발생
BUG-29903 prepare 실패를 반복 시 메모리 증가 (fixed at 5.3.3.15)

2. 컴파일 관련 옵션 수정

[변경사항] 섹션의 표를 참조, 아래와 같은 명령문 수행, 링크옵션, makefile과 같은 컴파일 관련파일의 옵션을 수정한다.

- 실행파일명 변경

```
$ apre -t cpp connect1.sc
```

- 링크옵션 변경

```
$ g++ -Wl,-relax -L. -O3 -L/home/wlgml337/altibase_home2/lib -o connect1 connect1.o -lapre -lodbccli -ldl -lpthread -lcrypt -lrt
```

- makefile

```
...  
%.cpp : %.sc  
    apre -t cpp $<  
...  
connect1: connect1.$(OBJEXT) connect1.cpp  
    $(LD)    $(LFLAGS)    $(LDOUT)connect1$(BINEXT)    connect1.$(OBJEXT)  
    $(LIBOPT) apre$(LIBAFT) $(LIBOPT)odbccli$(LIBAFT) $(LIBS)  
...  
...
```

임시조치

변경 관련파일이 쉽게 파악되지 않거나 양이 많아 일일이 수정할 수 없는 경우는 [절차 2]를 생략하고 아래와 같이 APRE*C/C++ 실행파일에 대해 SES*C/C++ 실행파일 명칭으로 심볼릭(symbolic) 링크를 생성하여 조치가 가능하다. (5.3.3.9 이후부터는 ALTIBASE 패키지 자체에 반영되어 있으므로 아래 명령어를 수행하지 않아도 된다.)

```
$ ln -s $ALTIBASE_HOME/bin/apre $ALTIBASE_HOME/bin/sesc
```

하지만, 이러한 조치는 어디까지나 임시적인 조치일 뿐이므로 반드시 관련명칭을 변경하는 정상적인 업그레이드를 수행해야 한다.

유의사항

-parse none 옵션 추가

[제약사항] 섹션에서 언급한 문법강화로 인한 사항이 아닌데도 SES*C/C++에서는 이상 없이 precompile되던 소스코드가 APRE*C/C++에서는 에러가 발생할 수 있는데, 이 경우는 APRE*C/C++의 파싱모드가 기본적으로 partial 이기 때문에 #include 방식으로 포함된 헤더파일까지 매크로처리를 하면서 발생하는 오류일 가능성이 높다.

따라서, 에러발생 원인을 쉽게 알 수 없다면 SES*C/C++와 동일하게 precompile을 하기 위한 -parse none 옵션을 추가하여 일차적인 확인을 하는 과정이 필요하다.

Precompiler 라이브러리 사용으로 인한 오류

기존의 소스코드가 SES*C/C++ 라이브러리를 직접적으로 사용하도록 작성되었다면 Precompiler 라이브러리 interface 변경으로 인해 컴파일이 되지 않을 수 있다. 이 경우는 기존 소스코드에서 관련코드를 모두 제거하여야만 한다.

Precompiler의 라이브러리 interface는 변경이 잦은 내부적인 요소로써 원칙적으로 사용자의 직접적인 사용을 금한다.

따라서, precompile된 소스코드를 분석, 사용자 임의로 소스코드에 관련 매크로, 구조체 및 함수를 사용하면서 추후 발생하는 오류는 책임지지 않는다.



알티베이스㈜

서울특별시 구로구 구로 3 동 182-13
대림포스트 2 차 1008 호
02-2082-1000
<http://www.altibase.com>

대전사무소

대전광역시 서구 둔산동 921
주은리더스텔 901 호
042-489-0330

기술지원본부

서울특별시 구로구 구로 3 동 182-13
대림포스트 2 차 908 호
02-2082-1000

기술지원센터

02-2082-1114
<http://support.altibase.com>

Copyright © 2000~2013 ALTIBASE Corporation. All Rights Reserved.

이 문서는 정보 제공을 목적으로 제공되며, 사전에 예고 없이 변경될 수 있습니다. 이 문서는 오류가 있을 수 있으며, 상업적 또는 특정 목적에 부합하는 명시적, 묵시적인 책임이 일체 없습니다. 이 문서에 포함된 ALTIBASE 제품의 특징이나 기능의 개발, 발표 등의 시기는 ALTIBASE 재량입니다. ALTIBASE는 이 문서에 대하여 관련된 특허권, 상표권, 저작권 또는 기타 지적 재산권을 보유할 수 있습니다.