Pro*C/C++

Release Notes

12*c* Release 2 (12.2)

E54618-05

April 2017

About these Release Notes

This document contains important information about Pro*C/C++ 12*c* Release 2 (12.2).

It contains the following topics:

- Documentation Accessibility
- Compatibility and Migration Issues
- New Features in Pro*C/C++ 12.2 Production
- New Features in Previous Releases
- Known Bugs
- Bugs Fixed
- Support

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Compatibility and Migration Issues

This section describes compatibility issues when migrating from earlier releases of Pro*C/C++.

Compatibility Between 32 Bit and 64 Bit Implementations

On platforms which support both 32 bit and 64 bit implementations, you must re-precompile your applications which include sqlca via an EXEC SQL INCLUDE statement before linking with the 64 bit binaries. For applications which include sqlca.h via the #include preprocessor statement, you must recompile to include the 64 bit sqlca.h before relinking with the 64 bit binaries.

In the future, to support generated code compatibility across implementations, only one version, the 64 bit version, of sqlca.h may be supplied on ports which support both 32 bit and 64 bit binaries.

Pro*C/C++ Configuration File

The Pro*C/C++ configuration file precomp/admin/pcscfg.cfg needs to be updated with the appropriate path for `sys_include` option. Environment variables can be used to specify the path, e.g, `$ORACLE_HOME` for Unix systems and `%ORACLE_HOME%` for Windows. Your include option should also be updated appropriately.

```
sys_include=($ORACLE_HOME/precomp/public,/usr/include)
<pre>sys_include=/usr/lib/gcc/i386-redhat-linux/4.1.1/include
<pre>include=($ORACLE_HOME/precomp/public)
<pre>include=$ORACLE_HOME/precomp/hdrs
```

LTYPE=SHORT

Setting the LTYPE=SHORT option causes `.lis` files to be generated using the verbose form rather than the expository form in which the entire program is listed.

New Features in Pro*C/C++ 12.2 Production

This section briefly describes new features introduced in Pro*C/C++ 12*c* Release 2 (12.2).

Support for long Identifiers

Pro*C/C++ now supports object lengths of 128 bytes. In previous releases, the object length limit was 30 bytes.

Support for Oracle Instant Client - Basic Light version

Pro*C/C++ now supports Oracle Instant Client - Basic Light version.

New Command Line Option

Pro*C/C++ now introduces a new command line option “trim_password”, to prevent authentication issues caused by password strings that contain trailing blank space.

Pro*C/C++ Compatibility with vc12

Pro*C/C++ is now fully compatible with vc12 (Visual Studio 2013 compiler).

New Features in Previous Releases

This section lists new features introduced to Pro*C/C++ in previous releases.

Features in Pro*C/C++ 12.1 Production

- Support for Auto Increment Columns
- Support for 32k Columns

- Support for Prefetch By Memory
- Support for SQL Plan Management (SPM)

Features in Pro*C/C++ 11.2 Production

- Support for 8-Byte Integers

  Pro*C/C++ now supports the native C long long int datatype. Integers of up to 18 digits can be used on 32 bit and 64 bit platforms.

- Optional WITH HOLD Clause in DECLARE CURSOR Statement

  The new WITH HOLD clause in the DECLARE CURSOR statement specifies a held cursor that remains open after COMMIT.

Features in Pro*C/C++ 11.1 Production

- SQL99 syntax support: Pro*C/C++ now supports SQL99 syntax for SELECT, INSERT, DELETE, and UPDATE statements and the body of the cursor in a DECLARE CURSOR statement.
- Additional array INSERT and array SELECT syntax support: Pro*C/C++ now supports the array INSERT and array SELECT syntax of the DB2 precompiler.
- Implicit buffered insert: Pro*C/C++ now supports the implicit buffering of a single INSERT statement executed in a loop.
- Dynamic SQL statement cache: Statement caching improves the performance of dynamic SQL statements.
- Fix execution plan: To ensure better performance of a Pro*C/C++ application during deployment, the 'outline' feature of the database is used to fix the execution plan.

Bugs Fixed

The following section lists bugs fixed in Pro*C/C++. Numbers in parentheses following the description refer to bug numbers in the Oracle Bug Database.

Bugs Fixed in Pro*C/C++ Release 12.2 Production

- Pro*C/C++ no longer throws CSF-S-00000 error when common_parser=yes in the timezone.pc file (9531787)
- Pro*C/C++ no longer throws ORA-01008 error when binds are used in the Select list while common_parser=yes (14127422)
- Pro*C/C++ no longer throws ORA-932 error when precompiling with option USERID and common_parser=yes for an INSERT statement which has CASE clause and TIMESTAMP function (14335958, 19473788)
- Pro*C/C++ no longer fails to set SQLSTATE during rollback, with MODE=ANSI and without declaring SQLCODE (5891984)

- Pro*C/C++ no longer throws ORA-538976288 error when using a "SELECT INTO" statement for PIC N variable (17189633)
- Pro*C/C++ no longer throws PCB-S-00400 error while precompiling a program with a Level 88 initialised variable (20194289)
- Pro*C/C++ no longer fails to parse Solaris platform-specific keyword __thread (20901503)
- Pro*C/C++ demo makefiles from Oracle Instant Client now allows the user to specify the GCC path (14591784)
- Pro*C/C++ no longer truncates strings while fetching data from a long column into a long variable (13589112)
- Pro*C/C++ Nvarchar2 object is now correctly mapped to nvarchar2 data type (11653552)
- Pro*C/C++ now supports OCIRaw processing in the EXEC SQL OBJECT GET construct (11653512)
- Pro*C/C++ is now shipped with generic demo files instead of OSD files, and old unused files are no longer shipped (9909411)
- Pro*C/C++ now handles parsing of lengthy numeric values (9814506)
- Pro*C/C++ now accepts the TYPE token in the IS OF clause (9535800)
- Pro*C/C++ now correctly interprets a newline character between single quotes in SQL statements (9531904)
- Pro*C/C++ no longer has memory corruption issues when processing SQL statements containing N'string' (9531638)
- Pro*C/C++ now recognizes the CURSOR WITH HOLD option when parser unification is enabled (8436316)
- Pro*C/C++ now passes the correct "mode" parameter value to OCIStmtFetch2() (18509872)
- Pro*C/C++ now frees up the memory allocated to column properties (20051833)
- Pro*C/C++ no longer fails with ORA-1008 error after encountering ORA-942 error when a table is dropped (20029428)
- Pro*C/C++ no longer throws PCC-S-2201 error while parsing cursor with hold syntax in unified parser mode (20534275)
- Pro*C/C++ no longer throws PLS-S-201 error when compiling with common_parser=No (20808657)
- Len attribute of Varchar structure no longer gets corrupted during batch fetching of rows (14013076)
- Pro*C/C++ no longer throws a segmentation fault when precompiling a long filename (12565588)

Bugs Fixed in Pro*C/C++ Release 12.1

- Pro*C/C++ parser no longer fails when both outline and common_parser are set to 'yes' (12819524)
- Pro*C/C++ no longer generates the ANSI prototypes for public APIs when the Precompiler option 'code' is set to 'kr_c' (10250555)
- XA application no longer crashes after upgrade to 11.2 (10086495)
- Obsolete Sun SPARC compiler option 'dalign' removed from Pro*C/C++ demo make file (9590964)
- EXEC SQL COLLECTION GET from VARRARY(5) of CHAR(5) now behaves correctly (9531014)
- On 64-bit platform SQLSQLDAAlloc() no longer returns aninvalid descriptor handle (9491931)
- Pro*C/C++ no longer crashes on HP-UX after upgrading to 11g (7365514)

Bugs Fixed in Pro*C/C++ Release 11.2

- Pro*C/C++ now correctly parses the backslash ('\') character in macro definitions (#defines) (8539668)
- Pro*C/C++ now supports header files with absolute file-paths up to the maximum length of a file-path supported by the OS on which Pro*C/C++ is running (8308077)
- Pro*C/C++ no longer generates a corrupted output file name when output file path is greater that 128. Pro*C/C++ now supports output file paths up to the the length supported by the OS on which Pro*C/C++ is running (8263988)
- Pro*C/C++ no longer flags a syntax error for the use of "CREATE SCHEMA … " in EXEC SQL … statement (7644340)
- Pro*C/C++ no longer flags a syntax error for the use of "CREATE TABLE … " in EXEC SQL … statement (7644376)
- Pro*C/C++ no longer flags a syntax error for the use of "CREATE VIEW … " in EXEC SQL … statement (7644400)
- Pro*C/C++ no longer flags a syntax error for the use of "CREATE VIWE … WITH CHECK OPTION" in EXEC SQL … statement (7644435)
- In a Pro*C/C++ application, a CHARZ variable now retains blank padding after an earlier dynamic fetch of CLOB (7462575)
- Pro*C/C++ now correctly binds character strings when you bind a char buffer with a null value in a SQL statement and then use the same buffer to bind a non-null char string. Pro*C/C++ no longer fails to update the length of the buffer bound again, and no longer assumes a null string being bound – correct rows are now retrieved from the table (7395839)

- After upgrading from earlier releases to 11.1, Pro*C/C++ applications no longer crash because the bind list is freed while there is an attempt to reuse it (7308054)
- Pro*C no longer throws error PCC-2014 on some platforms when it encounters diagnostic directive #WARNING during precompilation (7252878, 6669407, 6154596)
- Pro*C/C++ no longer searches for header files in include directories when the absolute path is provided if Pro*C/C++ fails to open the file directly, and no longer reports error PCC-S-02016 - include file pathname is too long (7231977)
- On Linux x86-64, Pro*C/C++ no longer hangs when the host variable in SELECT INTO was od type pid_t (7018967)
- Semantic analysis no longer fails at precompile time when connected to TimesTen (6964328)
- Pro*C/C++ no longer appends an extra '+' sign to a SQL hint in the generated C code (6432583)
- Pro*C/C++ configuration file, pcscfg.cfg, now has a valid path for the sys_include entry. Environment variables in command-line options are now supported (5690971)
- Pro*C/C++ applications can be executed in the Instant Client (IC) environment by downloading basic.zip from the Oracle Technology Network. To develop Pro*C/C++ applications in IC mode, you need to install the required files using the Instant Client install option from the media using the Oracle Universal Installer (OUI) (5248663)

Known Bugs

The following section lists known bugs in Pro*C/C++. Numbers in parentheses following the description refer to bug numbers in the Oracle Bug Database.

Known Bugs in Pro*C/C++ Release 12.2

- ORA-00942 error is thrown while running a query that selects values from an XML table in online mode (9535704)
- PCC-S-2322 error is thrown while executing an embedded SQL statement that contains a macro (9970779)
- PCC-S-02201 error is thrown due to a c99 compliance issue with the stdbool.h file (12866048)
- Implementation of GCC extension #include_next needs to be modified in order to correctly include a file in multiple paths (21414819)
- Certain DDL statements are incorrectly parsed (20857434)
- PCC-S-02201 error is thrown while parsing the DDL statement "CREATE GLOBAL TEMPORARY TABLE" (20857400)
- PCC-S-02201 error is thrown while parsing the DDL statement "CREATE ROLE" (20857370)

- Oracle Social Network does not support special character "@" in passwords (20809065)

Support

For Pro*C/C++ support, please contact your local Oracle Support Services Center.

Pro*C/C++ Release Notes, 12*c* Release 2 (12.2)
E54618-05