Pro*C/C++

Release Notes

12*c* Release 1 (12.1)

E18409-04

April 2013

About these Release Notes

This document contains important information about Pro*C/C++ 12*c* Release 1 (12.1).

It contains the following topics:

- Documentation Accessibility
- Compatibility and Migration Issues
- New Features in Pro*C/C++
- 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++

This section briefly describes new features introduced in Pro*C/C++ 12*c* Release 1 (12.1).

Support for Auto Increment Columns

Pro*C/C++ now supports auto increment columns. This simplifies the development of ticketing, HR, and tracking applications, and assists in migrating databases to Oracle Database.

Support for 32k Columns

Pro*C/C++ now supports 32k column data with VARCHAR2, NVARCHAR2 and RAW data. This feature also assists with the migration of databases to Oracle Database.

Support for Prefetch By Memory

New inline and command-line option allows you to specify the memory used for pre-fetching rows. The new MEMFORPREFETCH option will provide assistance in the migration of databases to Oracle Database. The existing PREFETCH option remains unchanged.

Support for SQL Plan Management (SPM)

New command-line options generate plan baseline SQL statements, and control the name and format of generated SQL and LOG files. These new options enhance existing precompiler SQL Plan Management. They provide a means of performance management and assist in the upgrade of compiler applications.

New Features in Previous Releases

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

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.

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 11.2

- Literals can be assigned to binary_double datatypes by using suffix "d";. But the statement is not recognized by Pro*C/C++ causing error PCC-S-02201 during precompilation (8551402)
- Hints in EXPLAIN PLAN disappear after precompile (1466269)
- Backslashes in split statements are not escaped properly (1323304)
- SQLCHECK=FULL doesn't detect invalid columns in an UPDATE WHERE CURRENT OF statement (658837)

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 12.1

- 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)

Support

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

Pro*C/C++ Release Notes, 12*c* Release 1 (12.1)
E18409-04

Copyright © 1996, 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.