

Release Notes

About these Release Notes

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

It contains the following topics:

- [Documentation Accessibility](#)
- [New Features in Pro*COBOL 12.2 Production](#)
- [New Features in Previous Releases](#)
- [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.

New Features in Pro*COBOL 12.2 Production

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

Support for long Identifiers

Pro*COBOL 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*COBOL now supports Oracle Instant Client - Basic Light version.

New Command Line Option

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

Support for Micro Focus Visual COBOL 2.2 Update 2 Compiler

Pro*COBOL now supports the Micro Focus Visual COBOL 2.2 Update 2 compiler for the following platforms:

- Linux x64
- Windows 64-bit and 32-bit
- Solaris x86 (32-bit and 64-bit)
- Solaris SPARC (32-bit and 64-bit)

New Features in Previous Releases

This section lists new features introduced to Pro*COBOL in previous releases.

Features in Pro*COBOL 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*COBOL 11.2 Production

- Instant Client Mode: Pro*COBOL can run in Instant Client (IC) mode. Developing Pro*COBOL applications in an IC environment is also supported.
- Support for 8-Byte Integers: Previously, the size of integer variable declarations using COMP/ COMPU/COMP-4/ COMP-5/ BINARY storage specification was restricted to 9 digits on 32-bit platforms. Now integers up to 18 digits can be used on all platforms. Declarations greater than 9 and less than or equal to 18 are mapped to C's long long int datatype during code generation.
- Command-line Option for Held Cursor Behavior after ROLLBACK: A new command-line option, CWH_SQL99, sets the ROLLBACK behavior for held cursors. By default, CWH_SQL99, is set to YES and held cursors are closed after

ROLLBACK. Set CWH_SQL99 to NO for backward compatible behavior where held cursors remain open after ROLLBACK.

Features in Pro*COBOL 11.1 Production

- SQL99 syntax support: Pro*COBOL 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*COBOL now supports the array INSERT and array SELECT syntax of the DB2 precompiler.
- Fix execution plan: To ensure better performance of a Pro*COBOL application during deployment, the 'outline' feature of the database is used to fix the execution plan.
- Implicit buffered insert: Pro*COBOL 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.
- Scrollable cursor support: Pro*COBOL now supports a scrollable cursor to fetch data in a non sequential manner.
- Platform endian support: Pro*COBOL now supports retrieving unicode data in OS endian in the PIC N host variable.
- Flexible B Area length for Pro*COBOL: The length of B Area can now be from 8 to 253 columns.

Bugs Fixed

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

Bugs Fixed in Pro*COBOL Release 12.2 Production

- Pro*COBOL no longer throws CSF-S-00000 error when common_parser=yes in the timezone.pc file (9531787)
- Pro*COBOL no longer throws ORA-01008 error when binds are used in the Select list while common_parser=yes (14127422)
- Pro*COBOL 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*COBOL no longer throws PCB-S-00576, PLS-103 error while precompiling with embedded PL/SQL using the select /*+ index hint */ statement (953338)
- Pro*COBOL no longer fails to set SQLSTATE during rollback, with MODE=ANSI and without declaring SQLCODE (5891984)
- Pro*COBOL no longer fails to generate a proper log file (17280039)

- Pro*COBOL no longer throws PCB-S-00400 error while precompiling a program with a Level 88 initialised variable (20194289)
- Pro*COBOL no longer gets SIGSEGV for duplicate host variables when precompiling with common_parser=yes (19473788)
- Pro*COBOL no longer throws PCB-S-00576 error while precompiling a program with common_parser=yes (18800170)
- Pro*COBOL now checks for a fatal error during precompilation before deleting log and sql files (17871321)
- Pro*COBOL no longer throws ORA-538976288 error when using a "SELECT INTO" statement for PIC N variable (17189633)
- Pro*COBOL no longer creates a .sql file containing "plan_run=yes" when precompilation fails (16240153)
- Pro*COBOL no longer creates over 72 columns when a variable is modified by varying with comp5=yes (14708769)
- SQLGLS calls in Pro*COBOL now work as expected (14640230)
- Pro*COBOL no longer returns PCB-S-00400 error during precompilation when a variable identifier follows a COPY statement (14113014)
- Pro*COBOL no longer returns PCB-S-00400 error when a COPY modifier is used (13478294)
- Pro*COBOL no longer generates a UNIX Return Code of 0 when precompilation fails (10083052)
- Pro*COBOL no longer generates .cob files after errors are encountered (9303962)

Bugs Fixed in Pro*COBOL Release 12.1

- Pro*COBOL no longer crashes while parsing long token/ SQL statement (13006848)
- Pro*COBOL no longer returns PCB-S-400 when precompiling SQL that includes an inline view written as embedded SQL (EXEC SQL), in spite of using common_parser=yes (12641413)
- Pro*COBOL no longer throws PCB-0400 error when using numeric as first character of a group element (10265545)
- Pro*COBOL on Windows no longer crashes when option comp1=integer is set (10040552)
- Pro*COBOL no longer returns ORA-6502 with PL/SQL bind (9905110)
- Pro*COBOL no longer crashes when a large source file contains a large number of host variables (9689604)
- Pro*COBOL now parses statements with a double dot after a COBOL COPY clause after installing patch 9218271 (9470397)
- Pro*COBOL no longer generates illegal values when a statement is declared at a remote DB and then prepared by passing a string instead of bind-var) (9402996)

- Pro*COBOL no longer fails with ORA-12899 when max_rows_insert is set (9381997)
- Pro*COBOL no longer returns PCB-S-00214 when JUSTIFIED clause is used with host variables (9266470)
- Pro*COBOL no longer returns PCB-S-400 error if SCREEN SECTION is used (9151190)
- Pro*COBOL no longer returns PCB-W-233 when host variable was used for AT clause (9147830)
- Pro*COBOL no longer returns PCB-S-400 when COPY is used in OCCURS clause of an array declaration (9128157)
- Pro*COBOL no longer returns PCB-S-400 when a COPY statement ends with double dots (9055457)
- Pro*COBOL now correctly translates SQL statements which contain a list of lists (8932394)
- Pro*COBOL no longer returns ORA-933 when cursors are declared with outline enabled (8770900)
- Pro*COBOL no longer returns PCB-S-00400 when an EXE SQL INCLUDE statement is used without an ending period in DATA SECTION (8713408)

Bugs Fixed in Pro*COBOL Release 11.2

- Pro*COBOL no longer fails with SIGSEGV when attempting to select an integer into a PIC N variable (8285576)
- Pro*COBOL no longer flags a syntax error for the use of "CREATE SCHEMA ... " in an EXEC SQL ... statement (7644365)
- Pro*COBOL no longer flags a syntax error for the use of "CREATE TABLE ... " in an EXEC SQL ... statement (7644387)
- Pro*COBOL no longer flags a syntax error for the use of "CREATE VIEW ... " in an EXEC SQL ... statement (7644423)
- Pro*COBOL no longer flags a syntax error for the use of "CREATE VIEW ... WITH CHECK OPTION" in an EXEC SQL ... statement (7644449)
- Pro*COBOL no longer goes into memory fault while pre-compiling huge files (more than 65K lines) with the EXEC SQL INCLUDE directive (7557301)
- When an EXEC SQL statement is used with an EXEC SQL INCLUDE statement in an "EXEC ORACLE IFDEF XYZ" block, Pro*COBOL no longer loses track of whether the macro "XYZ" is defined or not, and no longer includes the files mentioned in the EXEC SQL INCLUDE statement when not expected (7484318)
- When PICN_ENDIAN=OS is set, Pro*COBOL no longer tries to do conversion that results in truncation of length (7306555)
- REPLACE statements with longer pseudo-texts no longer causes Pro*COBOL precompilation to crash on certain platforms (7288085)

- Instant Client support for Pro*COBOL application execution (run time) and application development is now available on UNIX or Windows platforms (7045925)
- When DML statements are issued with a RETURNING clause in a Pro*COBOL application, the application now receives all rows affected by the DML statement (7029239)

Support

For Pro*COBOL support, contact your local Oracle Support Services Center.

Pro*COBOL® Release Notes, 12c Release 2 (12.2)

E85817-01

Copyright © 1996, 2017, 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, then 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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.