

Pro*COBOL®

Getting Started

Releases 9.2 and 1.8.77 for Windows

Part No. A96113-03

December 2003

Pro*COBOL Getting Started, Releases 9.2 and 1.8.77 for Windows

Part No. A96113-03

Copyright © 1996, 2003, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Send Us Your Comments	v
Preface	vii
Intended Audience	vii
Documentation Accessibility	vii
Structure	viii
Related Documents	viii
Conventions	ix
What's New in Pro*COBOL?	xiii
Oracle9i Release 2 (9.2) New Features in Pro*COBOL	xiii
Oracle9i Release 1 (9.0.1) New Features in Pro*COBOL	xiii
Oracle9i Release 1 (9.0.1) Deprecated or Desupported Features in Pro*COBOL	xiii
1 Introducing Pro*COBOL	
What is Pro*COBOL?	1-1
Release 1.8.77	1-1
Supported Compilers	1-2
Features	1-2
Restrictions	1-2
Directory Structure	1-2
Header Files	1-3
Library File	1-3
Known Problems, Restrictions, and Workarounds	1-3
2 Building Pro*COBOL Applications	
Precompiling Pro*COBOL Applications	2-1
The Pro*COBOL Commands	2-1
Precompiler Options	2-1
Viewing the Available Options	2-2
Configuration Files	2-2
Embedding PL/SQL	2-3
Compiling and Linking Pro*COBOL Applications	2-3
MERANT Micro Focus Compiler	2-3

How to Use the IDE.....	2-3
How to Use the Animator.....	2-4
The COBOL and CBLLINK Commands.....	2-4
The COBSQL Command.....	2-4
Sample Programs.....	2-5
Building the Demonstration Tables.....	2-5
Building the Sample Programs	2-5
makeit.bat.....	2-5
Sample Files	2-6

3 Pro*COBOL for Windows

Introduction to Pro*COBOL for Windows.....	3-1
Release 1.8.77	3-1
Supported Compilers	3-2
Features.....	3-2
Restrictions.....	3-2
Directory Structure	3-2
Header Files	3-3
Library File.....	3-3
Known Problems, Restrictions, and Workarounds	3-3
Building Pro*COBOL Applications	3-4
Precompiling Pro*COBOL Applications	3-4
The Pro*COBOL Commands.....	3-4
Precompiler Options.....	3-4
Embedding PL/SQL.....	3-5
Compiling and Linking Pro*COBOL Applications.....	3-6
MERANT Micro Focus Compiler	3-6
Sample Programs	3-7
Building the Demonstration Tables.....	3-7
Building the Sample Programs	3-8
Sample Files	3-8

Index

Send Us Your Comments

Pro*COBOL Getting Started, Releases 9.2 and 1.8.77 for Windows

Part No. A96113-03

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227. Attn: Server Technologies Documentation Manager
- Postal service:

Oracle Corporation
Server Technologies Documentation Manager
500 Oracle Parkway, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This document describes the features of the Pro*COBOL precompiler that apply to the Windows NT, Windows 2000, Windows XP, and Windows Server 2003 operating systems.]

This Preface contains these topics:

- [Intended Audience](#)
- [Documentation Accessibility](#)
- [Structure](#)
- [Related Documents](#)
- [Conventions](#)

Intended Audience

*Pro*COBOL Getting Started* is intended for anyone who wants to use Pro*COBOL who perform the following tasks:

- Embed SQL statements in a COBOL program
- Build Oracle database applications with Pro*COBOL

To use this document, you need to:

- Know how to use a COBOL compiler in the Windows environment
- Be familiar with Windows commands such as deleting and copying files
- Understand the concepts of the search path, configuration files, and directory structure
- Be able to use a text editor to make changes to an ASCII text file

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

This document contains:

Chapter 1, "Introducing Pro*COBOL"

This chapter describes the Oracle programmatic interface for the COBOL language running on Windows NT and Windows 98 operating systems.

Chapter 2, "Building Pro*COBOL Applications"

This chapter contains Provides an overview of building Oracle database applications with Pro*COBOL.

Chapter 3, "Pro*COBOL for Windows"

This chapter provides information for the Pro*COBOL precompiler specific to Windows operating systems.

Related Documents

For more information, see these Oracle resources:

- *Oracle Database Installation Guide for Windows*
- *Oracle Database Release Notes for Windows*
- *Oracle Database Platform Guide for Windows*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle Net Services Administrator's Guide*
- *Oracle Real Application Clusters Quick Start*
- *Oracle Database New Features*
- *Oracle Database Reference*
- *Oracle Database Error Messages*
- *Pro*COBOL Programmer's Guide*

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle Database Sample Schemas* for information on how these schemas were created and how you can use them yourself.

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/membership/>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/documentation/>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to start SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.

Convention	Meaning	Example
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>old_release</i> .SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fsl/dbs/tbs_01.dbf /fsl/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

Convention	Meaning	Example
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - HOME_NAME > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	<pre>c:\winnt\"system32 is the same as C:\WINNT\SYSTEM32</pre>
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	<pre>C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)</pre>

Convention	Meaning	Example
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	<code>C:\> net start OracleHOME_NAME_TNSListener</code>
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none"> ■ C:\orant for Windows NT ■ C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\orann, where <i>nn</i> is the latest release number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle Database Platform Guide for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

What's New in Pro*COBOL?

New features information from previous releases is also retained to help those users migrating to the current release.

The following sections describe the new features in Oracle Pro*COBOL:

- [Oracle9i Release 2 \(9.2\) New Features in Pro*COBOL](#)
- [Oracle9i Release 1 \(9.0.1\) New Features in Pro*COBOL](#)
- [Oracle9i Release 1 \(9.0.1\) Deprecated or Desupported Features in Pro*COBOL](#)

Oracle9i Release 2 (9.2) New Features in Pro*COBOL

There is no new Windows-specific feature added in Pro*COBOL for this release.

Oracle9i Release 1 (9.0.1) New Features in Pro*COBOL

The Oracle9i Release 1 (9.0.1) feature described in this section highlights the support for Windows 2000.

This section contains these topics:

- Using Oracle9i on Windows 2000

Pro*COBOL is now supported on Windows 2000. There are some differences between using Oracle9i on Windows 2000 and Windows NT 4.0.

See Also: *Oracle Database Platform Guide for Windows*

Oracle9i Release 1 (9.0.1) Deprecated or Desupported Features in Pro*COBOL

As of this release of the Oracle database server, the Pro*COBOL precompiler no longer supports the Fujitsu compiler.

Introducing Pro*COBOL

This chapter describes the Oracle programmatic interface for the COBOL language running on Windows operating systems.

This chapter contains these topics:

- [What is Pro*COBOL?](#)
- [Release 1.8.77](#)
- [Supported Compilers](#)
- [Features](#)
- [Restrictions](#)
- [Directory Structure](#)

See Also: *Pro*COBOL Programmer's Guide* for more information

What is Pro*COBOL?

To access an Oracle database, you use a high-level query language called Structured Query Language (SQL). You often use SQL through an interactive interface, such as SQL*Plus.

Pro*COBOL is a programming tool that enables you to embed SQL statements in a COBOL program. The Pro*COBOL precompiler converts the SQL statements in the COBOL program into standard Oracle run-time library calls. The generated output file can then be compiled, linked, and run in the usual manner.

Use the Pro*COBOL precompiler when rapid development and compatibility with other systems are your priorities.

Release 1.8.77

Oracle Corporation expects that any application written for Pro*COBOL release 1.8.x should precompile successfully with Pro*COBOL release 9.2.0. However, some vendor extensions may not be accepted and the application may not precompile successfully when migrating from release 1.8.x to release 9.2.0.

Note: Pro*COBOL Precompiler for Windows release 1.8.77 is now automatically installed with the Programmer installation type of the Oracle9i Client top-level component. You do not need to use the Custom installation type of the Oracle9i Client top-level component.

If you simply want to migrate a release 1.8.x application to release 9.2.0 without using any of the new features of Pro*COBOL release 9.2.0, but the application does not precompile successfully, then report the problem to Oracle Support Services.

Note: Oracle Corporation recommends that you use release 9.2.0 to develop new applications.

Supported Compilers

Pro*COBOL supports the MERANT Micro Focus NetExpress version 3.1 for 32-bit Windows NT, Windows 2000, and Windows 98.

Note: Pro*COBOL does not support Object Oriented COBOL (OOCOBOL) specifications.

Features

Pro*COBOL supports the following:

- Oracle databases release 8.1.x and higher
- Embedded PL/SQL blocks
- Bundled database calls, which can provide better performance in client/server environments
- Full ANSI compliance for embedded SQL programming
- Calls to PL/SQL stored procedures

Restrictions

Pro*COBOL does not support the following:

- User exits
- Access to the Oracle Call Interface
- Oracle object types
- Graphical user interface
- 16-bit code generation

Directory Structure

When you install Pro*COBOL, Oracle Universal Installer creates a directory called `\precomp` in your `ORACLE_BASE\ORACLE_HOME` directory.

Note: The `\precomp` directory can contain files for other products, such as Pro*C/C++.

The `\precomp` directory contains the directories listed in [Table 1-1](#).

Table 1–1 Directories

Directory Name	Contents
\admin	Configuration files
\demo\procob2	Sample programs for Pro*COBOL release 9.2.0
\demo\procob	Sample programs for Pro*COBOL release 1.8.77
\demo\sql	SQL scripts for sample programs
\doc\procob2	Readme files for Pro*COBOL 9.2.0
\doc\procob	Readme files for Pro*COBOL 1.8.77
\lib	Library files
\mesg	Message files
\public	Header files

Header Files

The `ORACLE_BASE\ORACLE_HOME\precomp\public` directory contains the Pro*COBOL header files listed in [Table 1–2](#).

Table 1–2 Header Files

Header File	Description
<code>oraca.cob</code>	Contains the Oracle Communications Area (ORACA), which helps you to diagnose runtime errors and to monitor your program's use of various Oracle resources.
<code>oraca5.cob</code>	ORACA5 is the COMP-5 version of ORACA.
<code>sqlca.cob</code>	Contains the SQL Communications Area (SQLCA), which helps you to diagnose runtime errors. The SQLCA is updated after every executable SQL statement.
<code>sqlca5.cob</code>	SQLCA5 is the COMP-5 version of SQLCA.
<code>sqlda.cob</code>	Contains the SQL Descriptor Area (SQLDA), which is a data structure required for programs that use dynamic SQL Method 4.
<code>sqlda5.cob</code>	This is the COMP-5 version of SQLDA.

Library File

The `ORACLE_BASE\ORACLE_HOME\precomp\lib` directory contains the library file that you use when linking Pro*COBOL applications. The library file is called `orasql9.lib`.

Known Problems, Restrictions, and Workarounds

- Although all Windows operating systems allow spaces in filenames and directory names, the Oracle Pro*C/C++ and Oracle Pro*COBOL precompilers will not precompile files that include spaces in the file name or directory name. For example, do not use the following formats:
 - `proc iname=test one.pc`
 - `proc iname=d:\dir1\second dir\sample1.pc`
- Users running PROCOB application that are not linked using `/LITLINK` option and failing at runtime with the error,

Load error: file 'ORASQL8'

need to copy orasql9.dll to orasql8.dll in the same directory where orasql9.dll exists.

Building Pro*COBOL Applications

This chapter provides an overview of building Oracle database applications with Pro*COBOL Releases 9.2 and 1.8.77 for Windows operating systems.

This chapter contains these topics:

- [Precompiling Pro*COBOL Applications](#)
- [Compiling and Linking Pro*COBOL Applications](#)
- [Sample Programs](#)

Note: Build and execute Pro*COBOL applications in a command prompt session with the default settings for the screen buffer size and the windows size. These settings ensure successful execution of the Pro*COBOL applications.

Precompiling Pro*COBOL Applications

This section describes the basics of precompiling a Pro*COBOL application.

See Also: *Pro*COBOL Programmer's Guide* for more information about Pro*COBOL commands, precompiler options, and configuration files

The Pro*COBOL Commands

Use one of the following commands to precompile a file:

- For Pro*COBOL Release 9.2, use:
`procob filename`
- For Pro*COBOL Release 1.8.77, use:
`procobl8 filename`

By default, if no extension is provided, Pro*COBOL tries to open `filename.pco`. If the `ONAME` option is not specified, Pro*COBOL generates a file named `filename.cbl`.

Precompiler Options

Many useful options are available at precompile time. Included are options that allow you to determine how:

- Resources are used
- Errors are reported
- Input and output are formatted
- Cursors are managed

Viewing the Available Options

To see a list of available options and their default values, enter the following at the command prompt:

```
C:\> procob
```

To see the option, defaults, and the restrictions (if any) on values, enter the following at the command prompt:

```
C:\> procob /?
```

Configuration Files

Pro*COBOL reads the configuration file for options before processing options supplied at the command line.

- For release 9.2.0, the configuration file is called `pcbcfg.cfg`. This file is located in the `ORACLE_BASE\ORACLE_HOME\precomp\admin` directory.
- For release 1.8.77, the configuration file is called `pcccob.cfg`. This file is located in the `ORACLE_BASE\ORACLE_HOME\precomp\admin` directory.

The configuration file has the following two options:

- `COMP-5= yes | no`
- `include=ORACLE_BASE\ORACLE_HOME\precomp\public`

COMP-5 Option

Check the following to determine whether you should change the value of `COMP-5`:

If you are using MERANT Micro Focus COBOL, `COMP-5` can be set to `yes` (`COMP-5=yes`) or `no` (`COMP-5=no`).

- If `COMP-5=yes`:
 - All `COMP` data items (if they are potential host variables) are converted to `COMP-5`.
 - All data items generated by the precompiler will be declared as `COMP-5`.
- If `COMP-5=no`:
 - The precompiler ignores `COMP-5` host variables.
 - Precompiled files generally do not run on Intel platforms.

For a workaround during the compilation stage, use the MERANT Micro Focus COBOL compiler directive:

```
MAKESYN "COMP-5" = "COMP"
```

This statement directs the compiler to treat `COMP` items as if they are `COMP-5` items.

INCLUDE Option

The INCLUDE option enables the provided .cob files in the `ORACLE_BASE\ORACLE_HOME\precomp\public` directory to be included without an explicit INCLUDE= option at the command line.

Embedding PL/SQL

If you are using embedded PL/SQL blocks, do the following:

1. Enter the SQLCHECK option and the USERID string to connect at the precompiling command line.
2. Specify the SQLCHECK=FULL option to check the syntax or semantics of embedded SQL statements and PL/SQL blocks.

See Also: *Pro*COBOL Programmer's Guide* for an example of a command line string or review the PL/SQL MAKE file

Compiling and Linking Pro*COBOL Applications

This section describes how to compile and link Pro*COBOL applications using the MERANT Micro Focus compiler.

MERANT Micro Focus Compiler

You can build and execute a MERANT Micro Focus COBOL application in two ways:

- [How to Use the IDE](#) (using NetExpress only)
- [How to Use the Animator](#) (products other than NetExpress)
- [The COBOL and CBLINK Commands](#) (all products)
- [The COBSQL Command](#)

In each of these the COBSQL utility may be used with the following advantages:

- Pro*COBOL is run by the MERANT Micro Focus compiler and does not need to be run as a separate step.
- Animation is done using the .pco source file rather than the .cb1 file produced by Pro*COBOL.
- The MAKEYSYN directive is provided automatically and need not be specified manually.

How to Use the IDE

A program generated by Pro*COBOL can be compiled and executed from within the MERANT Micro Focus NetExpress IDE. Simply add the .cb1 file generated by Pro*COBOL to a Net Express project. To avoid potential inconsistencies when calling routines in the Oracle libraries the program should be compiled using the directive:

```
MAKEYSYN "COMP-5" = "COMP"
```

This directive can be specified in the build setting for the source file, the project settings or through a \$SET line at the start of the source file. When you select **Rebuild** or **Rebuild All** the IDE generates an executable ready to Run or Animate.

How to Use the Animator

Programs can be compiled and executed from within the MERANT Micro Focus COBOL debugger, Animator V2.

To avoid potential inconsistencies when calling routines in the Oracle libraries, select the menu option Compiler Directives, and enter:

```
MAKESYN "COMP-5" = "COMP"
```

This step is required because MERANT Micro Focus COBOL stores binary numbers in Big Endian format. Oracle libraries expect binary numbers to be stored in Little Endian format (machine format).

The COBOL and CBLINK Commands

COBOL and CBLINK can be used to build programs in two ways, depending on whether the Pro*COBOL runtime is to be statically linked or accessed through a DLL at runtime.

For dynamic linking, the commands are:

```
COBOL sample1 /MAKESYN"COMP-5"="COMP";
CBLINK sample1
```

For static linking, the commands are:

```
COBOL sample1 /LITLINK /MAKESYN"COMP-5"="COMP";
CBLINK sample1 ORACLE_BASE\ORACLE_HOME\precomp\lib\orasql9.lib
```

The previous commands produce `sample1.exe`, which can be executed like any other Windows NT, Windows 2000, or Windows 98 program.

Note: MERANT Micro Focus COBOL must be installed on the same system as Pro*COBOL to successfully execute the file.

The COBSQL Command

COBSQL can be used to simplify preprocessing and debugging. To use COBSQL, specify the following directive to the COBOL compiler:

```
PREPROCESS(COBSQL) COBSQLTYPE=ORACLE8 ENDP
```

or the short form:

```
P(COBSQL) CSQLT=ORA8 ENDP
```

COBSQLTYPE should be set to ORACLE or ORA for versions of Pro*COBOL prior to release 8.0. The directive may be set with a \$SET line at the start of the source file, on the COBOL command line, in program build settings or project settings for NetExpress, or with SQL compiler directives settings for Animator. At compile time, COBSQL runs Pro*COBOL as a background task and passes its output to the COBOL compiler together with additional information required to enable Animator to track execution using the `.pco` file rather than `.cbl` file.

When using COBSQL there is no need to deal directly with the `.cbl` file. Instead, add the `.pco` file to a NetExpress project, or open it with Animator.

Sample Programs

Oracle provides sample programs to demonstrate the use of Pro*COBOL with Oracle database features. See ["Sample Files"](#) on page 2-6 for a listing of these programs.

This section describes how to use the basic precompiling, compiling, and linking commands to build the sample programs. This section also describes the preparations required for running the Pro*COBOL sample programs.

Building the Demonstration Tables

To run the Pro*COBOL sample programs, you must have a database account with the username `scott` and the password `tiger`. If this account does not exist on your database, create one before running the sample programs.

The `scott` account must contain the `emp` and `dept` tables. If the account does not contain these tables, use the `demobld.sql` script to create them.

To run the `demobld.sql` script:

1. Start SQL*Plus.
2. Connect to the database as username `scott` with password `tiger`.
3. Run the `demobld.sql` script. For example:

```
SQL> @ORACLE_BASE\ORACLE_HOME\sqlplus\demo\demobld.sql
```

Building the Sample Programs

Pro*COBOL supplies the `makeit.bat` file which is listed in the next section, for building a MERANT Micro Focus COBOL sample file:

For release 9.2.0, the batch files are located in `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob2`. For release 1.8.77, the files are located in `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob`.

To build the sample programs:

1. Run the batch files with any sample file. Do not include the file extension. For example:


```
C:\ORACLE\ORA90\PRECOMP\DEMO\PROC0B2> makeit sample1
```
2. Ensure that all paths and file names reflect the configuration of your system, if you encounter errors when building a sample program.

The commands to run the sample programs assume that the following are the current working directories:

- `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob2` directory for release 9.2.0
- `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob` directory for release 1.8.77

You may need to modify the sample link script to reflect the configuration of your system. See ["Compiling and Linking Pro*COBOL Applications"](#) on page 2-3 for more information.

makeit.bat

The `makeit.bat` for release 9.2.0 contains the following:

```

procob iname=%1.pco ireclen=132
cobol %1 /anim /litlink makesyn "COMP-5" = "COMP";
cbllink %1 /M%1 ORACLE_BASE\ORACLE_HOME\precomp\lib\orasql9.lib

```

For release 1.8.77, this batch file contains the following:

```

procobl8 iname=%1.pco ireclen=132
cobol %1 /anim /litlink makesyn "COMP-5" = "COMP";
cbllink %1 /M%1 ORACLE_BASE\ORACLE_HOME\precomp\lib\orasql9.lib

```

Sample Files

The Pro*COBOL sample files listed in [Table 2–1, "Pro*COBOL Sample Programs"](#) are located in the `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob2` (release 9.2.0) and/or `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob` (release 1.8.77) directories. The SQL scripts are located in the `ORACLE_BASE\ORACLE_HOME\precomp\demo\sql` directory.

Table 2–1 Pro*COBOL Sample Programs

Sample Program	Description
sample1.pco	Simple query
sample2.pco	Cursor operations
sample3.pco	Host tables
sample4.pco	Datatype equivalence
sample6.pco	Dynamic SQL Method 1
sample7.pco	Dynamic SQL Method 2
sample8.pco	Dynamic SQL Method 3
sample9.pco	Stored procedure call
calldemo.sql	Stored procedure call
sample10.pco	Dynamic SQL Method 4
sample11.pco	Cursor variable operations
sample11.sql	Cursor variable operations
sample12.pco	Dynamic SQL Method 4 using ANSI dynamic SQL
sample13.pco	Nested program
sampleco.pco	Simple query and insert
sample14.pco	Host table x (release 8.1.6 and on)
lobdemo1.pco	LOB datatypes (release 8.1.6 and on)
lobdemo1.sql	LOB datatypes (release 8.1.6 and on)

Pro*COBOL for Windows

This chapter provides information for the Pro*COBOL precompiler specific to the Windows NT, Windows 98, and Windows 2000 operating systems.

This chapter describes only the features of Oracle9i for Windows software that apply to the Windows NT, Windows 2000, Windows XP, and Windows 98 operating systems. Information on Oracle9i Personal Edition software on Windows 98 is not covered in this chapter.

This chapter contains the following sections:

- [Introduction to Pro*COBOL for Windows](#)
- [Building Pro*COBOL Applications](#)

Introduction to Pro*COBOL for Windows

This section describes the Oracle programmatic interface for the COBOL language running on Windows operating systems.

This section contains these topics:

- [Release 1.8.77](#)
- [Supported Compilers](#)
- [Features](#)
- [Restrictions](#)
- [Directory Structure](#)

Release 1.8.77

Oracle Corporation expects that any application written for Pro*COBOL release 1.8.x should precompile successfully with Pro*COBOL release 9.2.0. However, some vendor extensions may not be accepted and the application may not precompile successfully when migrating from release 1.8.x to release 9.2.0.

Note: Pro*COBOL Precompiler for Windows release 1.8.77 is now automatically installed with the Programmer installation type of the Oracle9i Client top-level component. You do not need to use the Custom installation type of the Oracle9i Client top-level component.

If you simply want to migrate a release 1.8.x application to release 9.2.0 without using any of the new features of Pro*COBOL release 9.2.0, but the application does not precompile successfully, then report the problem to Oracle Support Services.

Note: Oracle Corporation recommends that you use release 9.2.0 to develop new applications.

Supported Compilers

Pro*COBOL supports the MERANT Micro Focus NetExpress version 3.1 for 32-bit Windows NT, Windows 2000, and Windows 98.

Note: Pro*COBOL does not support Object Oriented COBOL (OOCOBOL) specifications.

Features

Pro*COBOL for Windows supports the following:

- Oracle databases release 8.1.x and higher
- Embedded PL/SQL blocks
- Bundled database calls, which can provide better performance in client/server environments
- Full ANSI compliance for embedded SQL programming
- Calls to PL/SQL stored procedures

Restrictions

Pro*COBOL for Windows does not support the following:

- User exits
- Access to the Oracle Call Interface
- Oracle object types
- Graphical user interface
- 16-bit code generation

Directory Structure

When you install Pro*COBOL, Oracle Universal Installer creates a directory called `\precomp` in your `ORACLE_BASE\ORACLE_HOME` directory.

Note: The `\precomp` directory can contain files for other products, such as Pro*C/C++.

The `\precomp` directory contains the directories listed in [Table 3-1](#).

Table 3–1 Directories

Directory Name	Contents
\admin	Configuration files
\demo\procob2	Sample programs for Pro*COBOL release 9.2.0
\demo\procob	Sample programs for Pro*COBOL release 1.8.77
\demo\sql	SQL scripts for sample programs
\doc\procob2	Readme files for Pro*COBOL 9.2.0
\doc\procob	Readme files for Pro*COBOL 1.8.77
\lib	Library files
\mesg	Message files
\public	Header files

Header Files

The `ORACLE_BASE\ORACLE_HOME\precomp\public` directory contains the Pro*COBOL header files listed in [Table 3–2](#).

Table 3–2 Header Files

Header File	Description
<code>oraca.cob</code>	Contains the Oracle Communications Area (ORACA), which helps you to diagnose runtime errors and to monitor your program's use of various Oracle resources.
<code>oraca5.cob</code>	ORACA5 is the COMP-5 version of ORACA.
<code>sqlca.cob</code>	Contains the SQL Communications Area (SQLCA), which helps you to diagnose runtime errors. The SQLCA is updated after every executable SQL statement.
<code>sqlca5.cob</code>	SQLCA5 is the COMP-5 version of SQLCA.
<code>sqllda.cob</code>	Contains the SQL Descriptor Area (SQLDA), which is a data structure required for programs that use dynamic SQL Method 4.
<code>sqllda5.cob</code>	This is the COMP-5 version of SQLDA.

Library File

The `ORACLE_BASE\ORACLE_HOME\precomp\lib` directory contains the library file that you use when linking Pro*COBOL applications. The library file is called `orasql9.lib`.

Known Problems, Restrictions, and Workarounds

- Although all Windows operating systems allow spaces in filenames and directory names, the Oracle Pro*C/C++ and Oracle Pro*COBOL precompilers will not precompile files that include spaces in the file name or directory name. For example, do not use the following formats:
 - `proc iname=test one.pc`
 - `proc iname=d:\dir1\second dir\sample1.pc`
- Users running PROCOB application that are not linked using /LITLINK option and failing at runtime with the error,

```
Load error: file 'ORASQL8'
```

need to copy `orasql9.dll` to `orasql8.dll` in the same directory where `orasql9.dll` exists.

Building Pro*COBOL Applications

This section provides an overview of building Oracle database applications with Pro*COBOL Releases 9.2 and 1.8.77 for Windows operating systems.

This section contains these topics:

- [Precompiling Pro*COBOL Applications](#)
- [Compiling and Linking Pro*COBOL Applications](#)
- [Sample Programs](#)

Note: Build and execute Pro*COBOL applications in a command prompt session with the default settings for the screen buffer size and the windows size. These settings ensure successful execution of the Pro*COBOL applications.

Precompiling Pro*COBOL Applications

This section describes the basics of precompiling a Pro*COBOL application.

The Pro*COBOL Commands

Use one of the following commands to precompile a file:

- For Pro*COBOL Release 9.2, use:
`procob filename`
- For Pro*COBOL Release 1.8.77, use:
`procob18 filename`

By default, if no extension is provided, Pro*COBOL tries to open `filename.pco`. If the `ONAME` option is not specified, Pro*COBOL generates a file named `filename.cbl`.

Precompiler Options

Many useful options are available at precompile time. Included are options that allow you to determine how:

- Resources are used
- Errors are reported
- Input and output are formatted
- Cursors are managed

Viewing the Available Options

To see a list of available options and their default values, enter the following at the command prompt:

```
C:\> procob
```

To see the option, defaults, and the restrictions (if any) on values, enter the following at the command prompt:

```
C:\> procob /?
```

Configuration Files

Pro*COBOL reads the configuration file for options before processing options supplied at the command line.

- For release 9.2.0, the configuration file is called `pcbcbfg.cfg`. This file is located in the `ORACLE_BASE\ORACLE_HOME\precomp\admin` directory.
- For release 1.8.77, the configuration file is called `pcccob.cfg`. This file is located in the `ORACLE_BASE\ORACLE_HOME\precomp\admin` directory.

The configuration file has the following two options:

- `COMP-5= yes | no`
- `include=ORACLE_BASE\ORACLE_HOME\precomp\public`

COMP-5 Option

Check the following to determine whether you should change the value of `COMP-5`:

If you are using MERANT Micro Focus COBOL, `COMP-5` can be set to `yes` (`COMP-5=yes`) or `no` (`COMP-5=no`).

- If `COMP-5=yes`:
 - All `COMP` data items (if they are potential host variables) are converted to `COMP-5`.
 - All data items generated by the precompiler will be declared as `COMP-5`.
- If `COMP-5=no`:
 - The precompiler ignores `COMP-5` host variables.
 - Precompiled files generally do not run on Intel platforms.

For a workaround during the compilation stage, use the MERANT Micro Focus COBOL compiler directive:

```
MAKESYN "COMP-5" = "COMP"
```

This statement directs the compiler to treat `COMP` items as if they are `COMP-5` items.

INCLUDE Option

The `INCLUDE` option enables the provided `.cob` files in the `ORACLE_BASE\ORACLE_HOME\precomp\public` directory to be included without an explicit `INCLUDE=` option at the command line.

Embedding PL/SQL

If you are using embedded PL/SQL blocks, do the following:

1. Enter the `SQLCHECK` option and the `USERID` string to connect at the precompiling command line.
2. Specify the `SQLCHECK=FULL` option to check the syntax or semantics of embedded SQL statements and PL/SQL blocks.

See Also: *Pro*COBOL Programmer's Guide* for an example of a command line string or review the PL/SQL MAKE file

Compiling and Linking Pro*COBOL Applications

This section describes how to compile and link Pro*COBOL applications using the MERANT Micro Focus compiler.

MERANT Micro Focus Compiler

You can build and execute a MERANT Micro Focus COBOL application in two ways:

- [How to Use the IDE](#) (using NetExpress only)
- [How to Use the Animator](#) (products other than NetExpress)
- [The COBOL and CBLINK Commands](#) (all products)
- [the COBSQL Command](#)

In each of these the COBSQL utility may be used with the following advantages:

- Pro*COBOL is run by the MERANT Micro Focus compiler and does not need to be run as a separate step.
- Animation is done using the .pco source file rather than the .cb1 file produced by Pro*COBOL.
- The MAKEYSYN directive is provided automatically and need not be specified manually.

How to Use the IDE

A program generated by Pro*COBOL can be compiled and executed from within the MERANT Micro Focus NetExpress IDE. Simply add the .cb1 file generated by Pro*COBOL to a Net Express project. To avoid potential inconsistencies when calling routines in the Oracle libraries the program should be compiled using the directive:

```
MAKESYN "COMP-5" = "COMP"
```

This directive can be specified in the build setting for the source file, the project settings or through a \$SET line at the start of the source file. When you select **Rebuild** or **Rebuild All** the IDE generates an executable ready to Run or Animate.

How to Use the Animator

Programs can be compiled and executed from within the MERANT Micro Focus COBOL debugger, Animator V2.

To avoid potential inconsistencies when calling routines in the Oracle libraries, select the menu option Compiler Directives, and enter:

```
MAKESYN "COMP-5" = "COMP"
```

This step is required because MERANT Micro Focus COBOL stores binary numbers in Big Endian format. Oracle libraries expect binary numbers to be stored in Little Endian format (machine format).

The COBOL and CBLINK Commands

COBOL and CBLINK can be used to build programs in two ways, depending on whether the Pro*COBOL runtime is to be statically linked or accessed through a DLL at runtime.

For dynamic linking, the commands are:

```
COBOL sample1 /MAKESYN"COMP-5"="COMP";
CBLINK sample1
```

For static linking, the commands are:

```
COBOL sample1 /LITLINK /MAKESYN"COMP-5"="COMP";
CBLINK sample1 ORACLE_BASE\ORACLE_HOME\precomp\lib\orasql9.lib
```

The previous commands produce `sample1.exe`, which can be executed like any other Windows NT, Windows 2000, or Windows 98 program.

Note: MERANT Micro Focus COBOL must be installed on the same system as Pro*COBOL to successfully execute the file.

the COBSQL Command

COBSQL can be used to simplify preprocessing and debugging. To use COBSQL, specify the following directive to the COBOL compiler:

```
PREPROCESS(COBSQL) COBSQLTYPE=ORACLE8 ENDP
```

or the short form:

```
P(COBSQL) CSQLT=ORA8 ENDP
```

COBSQLTYPE should be set to ORACLE or ORA for versions of Pro*COBOL prior to release 8.0. The directive may be set with a \$SET line at the start of the source file, on the COBOL command line, in program build settings or project settings for NetExpress, or with SQL compiler directives settings for Animator. At compile time, COBSQL runs Pro*COBOL as a background task and passes its output to the COBOL compiler together with additional information required to enable Animator to track execution using the .pco file rather than .cbl file.

When using COBSQL there is no need to deal directly with the .cbl file. Instead, add the .pco file to a NetExpress project, or open it with Animator.

Sample Programs

Oracle provides sample programs to demonstrate the use of Pro*COBOL with Oracle database features. See ["Sample Files"](#) on page 3-8 for a listing of these programs.

This section describes how to use the basic precompiling, compiling, and linking commands to build the sample programs. This section also describes the preparations required for running the Pro*COBOL sample programs.

Building the Demonstration Tables

To run the Pro*COBOL sample programs, you must have a database account with the username `scott` and the password `tiger`. If this account does not exist on your database, create one before running the sample programs.

The `scott` account must contain the `emp` and `dept` tables. If the account does not contain these tables, use the `demobld.sql` script to create them.

To run the `demobld.sql` script:

1. Start SQL*Plus.
2. Connect to the database as username `scott` with password `tiger`.

3. Run the `demobld.sql` script. For example:

```
SQL> @ORACLE_BASE\ORACLE_HOME\sqlplus\demo\demobld.sql
```

Building the Sample Programs

Pro*COBOL supplies the `makeit.bat` file which is listed in the next section, for building a MERANT Micro Focus COBOL sample file:

For release 9.2.0, the batch files are located in `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob2`. For release 1.8.77, the files are located in `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob`.

To build the sample programs:

1. Run the batch files with any sample file. Do not include the file extension. For example:

```
C:\ORACLE\ORA90\PRECOMP\DEMO\PROCOB2> makeit sample1
```

2. Ensure that all paths and file names reflect the configuration of your system, if you encounter errors when building a sample program.

The commands to run the sample programs assume that the following are the current working directories:

- `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob2` directory for release 9.2.0
- `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob` directory for release 1.8.77

You may need to modify the sample link script to reflect the configuration of your system. See "[Compiling and Linking Pro*COBOL Applications](#)" on page 3-6 for more information.

The `makeit.bat` for release 9.2.0 contains the following:

```
procob iname=%1.pco ireclen=132
cobol %1 /anim /litlink makesyn "COMP-5" = "COMP";
cbllink %1 /M%1 ORACLE_BASE\ORACLE_HOME\precomp\lib\orasql9.lib
```

For release 1.8.77, this batch file contains the following:

```
procob18 iname=%1.pco ireclen=132
cobol %1 /anim /litlink makesyn "COMP-5" = "COMP";
cbllink %1 /M%1 ORACLE_BASE\ORACLE_HOME\precomp\lib\orasql9.lib
```

Sample Files

The Pro*COBOL sample files listed in [Table 3-3](#) are located in the `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob2` (release 9.2.0) and/or `ORACLE_BASE\ORACLE_HOME\precomp\demo\procob` (release 1.8.77) directories. The SQL scripts are located in the `ORACLE_BASE\ORACLE_HOME\precomp\demo\sql` directory.

Table 3-3 Pro*COBOL Sample Programs

Sample Program	Description
<code>sample1.pco</code>	Simple query
<code>sample2.pco</code>	Cursor operations
<code>sample3.pco</code>	Host tables

Table 3–3 (Cont.) Pro*COBOL Sample Programs

Sample Program	Description
sample4.pco	Datatype equivalence
sample6.pco	Dynamic SQL Method 1
sample7.pco	Dynamic SQL Method 2
sample8.pco	Dynamic SQL Method 3
sample9.pco	Stored procedure call
calldemo.sql	Stored procedure call
sample10.pco	Dynamic SQL Method 4
sample11.pco	Cursor variable operations
sample11.sql	Cursor variable operations
sample12.pco	Dynamic SQL Method 4 using ANSI dynamic SQL
sample13.pco	Nested program
sampleco.pco	Simple query and insert
sample14.pco	Host table x (release 8.1.6 and on)
lobdemo1.pco	LOB datatypes (release 8.1.6 and on)
lobdemo1.sql	LOB datatypes (release 8.1.6 and on)

Index

A

Animator, 2-4
ANSI compliance, 1-2, 3-2
ANSI dynamic SQL, 2-6, 3-9

B

batch files, for sample programs, 2-5, 3-8

C

CBLINK command, 2-4
COBSQL, 2-4, 3-7
comp5, 2-2, 3-5
compiling, 2-3, 3-6
configuration files, 2-2
 location, 2-2

D

default file extension, 2-1, 3-4
demobld.sql script, 2-5, 3-7
demonstration tables, building, 2-5, 3-7
DEPT table, 2-5, 3-7
directory structures, 1-2, 3-2
dynamic SQL (ANSI), 2-6, 3-9

E

embedded PL/SQL, 2-3, 3-5
EMP table, 2-5, 3-7

F

features
 new, 0-xiii
file extension, default, 2-1, 3-4

G

generic documentation references
 comp5, 2-2, 3-5
 compiling and linking, 2-3, 3-6
 sample programs, 2-5, 3-7
 user exits, not supported, 1-2, 3-2

H

header files, 1-3, 3-3

I

IDE, 2-3, 3-6
INCLUDE option, 2-3, 3-5
installation
 Pro*COBOL release 1.8.x, 1-1, 3-1

L

library file, 1-3, 3-3
linking, 2-3, 3-6

M

makeit.bat, 2-5, 3-8
MERANT Micro Focus COBOL
 Animator, 2-4
 building sample programs, 2-5
 COBSQL, 2-4, 3-7
 comp5, 2-2, 3-5
 compiling and linking, 2-3, 3-6
 IDE, 2-3, 3-6
 NetExpress, 2-3, 3-6
 version supported, 1-2, 3-2

N

NetExpress, 2-3, 3-6

O

ONAME option, 2-1, 3-4
options, 2-1, 3-4
ORACA, 1-3, 3-3
orasql9.lib library file, 1-3, 3-3

P

pbcfg.cfg, 2-2
pcccob.cfg, 2-2
PL/SQL, 2-3, 3-5
precompiling, 2-1, 3-4
 commands, 2-1, 3-4

Pro*COBOL
 building applications, 2-1, 3-4
 compilers supported, 1-2, 3-2
 compiling and linking, 2-3, 3-6
 configuration files, 2-2
 directory structures, 1-2, 3-2
 features, 1-2, 3-2
 options, 2-1, 3-4
 overview, 1-1
 release 1.8.x
 described, 1-1, 3-1
 installation, 1-1, 3-1
 restrictions, 1-2, 3-2

R

release 1.8.x
 described, 1-1, 3-1
 installation, 1-1, 3-1
restrictions
 Pro*COBOL, 1-2, 3-2

S

sample programs
 batch files, 2-5, 3-8
 building, 2-5, 3-7
 file names, 2-6, 3-8
 location of, 1-3, 3-3
SCOTT/TIGER account, 2-5, 3-7
SQL (Structured Query Language), 1-1
SQL*Plus, 1-1, 2-5, 3-7
SQLCA, 1-3, 3-3
SQLCHECK option, 2-3, 3-5
SQLDA, 1-3, 3-3
Structured Query Language (SQL), 1-1

U

user exits, not supported, 1-2, 3-2