Pro*COBOL Precompiler

Getting Started

Releases 9.0.1 and 1.8.52 for Windows

June 2001

Part No. A90168-01

ORACLE

Pro*COBOL Precompiler Getting Started, Releases 9.0.1 and 1.8.52 for Windows

Part No. A90168-01

Contributors: Riaz Ahmed, Eric Belden, Sharon Castledine, Joseph Garcia, Lisa Giambruno, Ravi Kumar Gooty, Neeraj Gupta, Nancy Ikeda, Maura Joglekar, Mark Kennedy, Bernie Harris, Ana Hernandez, Mark Kennedy, Robert Knecht, Paul Lyons, Shiva Prasad, Helen Slattery, Christopher Stead, Jeff Stein, Gael Stevens, Nicole Sullivan, Ellen Tafeen, Janice Wong, Martha Woo, Janis Greenberg

Contents

2 Building Pro*COBOL Applications

Send Us Your Comments

Pro*COBOL Precompiler Getting Started, Releases 9.0.1 and 1.8.52 for Windows

Part No. A90168-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. 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?

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

- E-mail: ntdoc_us@oracle.com
- FAX - (650) 506-7365 Attn: Oracle Database for Windows Documentation
- Postal service:
  Oracle Corporation
  Oracle Database for Windows Documentation Manager
  500 Oracle Parkway, Mailstop 1op6
  Redwood Shores, CA 94065
  USA

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

If you have problems with the software, please contact your local Oracle Support Services. Contact information for Oracle Support Services is available at this Web site:

`http://www.oracle.com/support/`

Preface

This guide provides introductory information for the Pro*COBOL precompiler running on Microsoft Windows NT, Windows 95/98, and Windows 2000 operating systems.

This preface contains these topics:

Audience

*Pro*COBOL Precompiler Programmer's Guide* is intended for anyone who wants to use Pro*COBOL to 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 NT and Windows 95/98 environments
- Be familiar with Windows NT and Windows 95/98 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

Organization

Chapter 1, "Introducing Pro*COBOL"

Describes the Oracle programmatic interface for the COBOL language running on Windows NT and Windows 95/98 operating systems.

Chapter 2, "Building Pro*COBOL Applications"

Provides an overview of building Oracle database applications with Pro*COBOL release 9.0.1and 1.8.52 for Windows.

Related Documentation

For more information, see these Oracle resources:

- Oracle9i Database installation guide for Windows
- Oracle9i Database release notes for Windows
- *Oracle9i Database Administrator's Guide for Windows*
- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle9i Net Services Administrator's Guide*

- *Oracle9i Real Application Clusters Concepts*
- *Oracle9i Database New Features*
- *Oracle9i Database Reference*
- *Oracle9i Database Error Messages*
- *Pro*COBOL Precompiler Programmer's Guide*

In North America, printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

`http://www.oraclebookshop.com/`

Other customers can contact their Oracle representative to purchase printed documentation.

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://technet.oracle.com/membership/index.htm`

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://technet.oracle.com/docs/index.htm`

Conventions

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

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**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle9i Database Concepts* Ensure that the recovery catalog and target database do *not* 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 open 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.
`lowercase monospace (fixed-width font) italic`	Lowercase monospace italic font represents placeholders or variables.	You can specify the *`parallel_clause`*. Run `U`*`old_release`*`.SQL` where *`old_release`* 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 (digits [ , precision ])`
{ }	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: ■ 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 subquery;` `SELECT col1, col2, ... , coln FROM employees;`
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	
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;`
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	`CONNECT SYSTEM/system_password` `DB_NAME = database_name`

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.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

Convention	Meaning	Example

`lowercase`	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.
	
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		
		`sqlplus hr/hr`
		
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

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. For example, to start Oracle Database Configuration Assistant, you must click the Start button on the taskbar and then choose Programs > Oracle - *HOME_NAME* > Database Administration > Database Configuration Assistant.	Choose Start > Programs > Oracle - *HOME_NAME* > Database Administration > Database Configuration Assistant
`C:\>`	Represents the Windows command prompt of the current hard disk drive. Your prompt reflects the subdirectory in which you are working. Referred to as the command prompt in this guide.	`C:\oracle\oradata>`
HOME_NAME	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.	`C:\> net start Oracle`*`HOME_NAME`*`TNSListener`

Convention	Meaning	Example
ORACLE_HOME and *ORACLE_BASE*	In releases prior to 8.1, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default was: ▪ `C:\orant` for Windows NT ▪ `C:\orawin95` for Windows 95 ▪ `C:\orawin98` for Windows 98 or whatever you called your Oracle home. In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* that by default is `C:\oracle`. If you install release 9.0 on a computer with no other Oracle software installed, the default setting for the first Oracle home directory is `C:\oracle\ora90`. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. See *Oracle9i Database Getting Started for Windows* for additional information on OFA compliances and for information on installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

Documentation Accessibility

Oracle's goal is to make our products, services, and supporting documentation accessible to the disabled community with good usability. 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/`

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.

What's New in Pro*COBOL?

The following sections describe the new features in Oracle 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

Oracle9*i* Release 1 (9.0.1) New Features in Pro*COBOL

This section contains these topics:

- **Using Oracle9*i* on Windows 2000**

  There are some differences between using Oracle9*i* on Windows 2000 and Windows NT 4.0.

  See Also: *Oracle9i Database Getting Started for Windows*

Oracle9*i* 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 Fuijitsu compiler.

1

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 1.8.52
- Supported Compilers
- Features
- Restrictions
- Directory Structure

> **See Also:** For more information on these topics, see the *Pro*COBOL Precompiler Programmer's Guide.*

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 1.8.52

Oracle Corporation expects that any application written for Pro*COBOL release 1.8.*x* should precompile successfully with Pro*COBOL release 9.0.1. However, some vendor extensions may not be accepted and the application may not precompile successfully when migrating from release 1.8.*x* to release 9.0.1.

Note: Pro*COBOL Precompiler for Windows release 1.8.52 is now automatically installed with the Programmer installation type of the Oracle9*i* Client top-level component. You do not need to use the Custom installation type of the Oracle9*i* Client top-level component.

If you simply want to migrate a release 1.8.*x* application to release 9.0.1 without using any of the new features of Pro*COBOL release 9.0.1, but the application does not precompile successfully, then use release 1.8.51. If this does not work, then report the problem to Oracle Support Services.

Note: Oracle Corporation recommends that you use release 9.0.1 to develop new applications.

Supported Compilers

Pro*COBOL supports the following compiler:

MERANT Micro Focus NetExpress version 3.1 for 32-bit Windows NT, Windows 2000, and Windows 95/98

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

Features

Pro*COBOL supports the following:

- Oracle databases release 8.1.*x* and up
- 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
- Oracle8*i* 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 following directories:

Directory Name	Contents
\admin	Configuration files
\demo\procob2	Sample programs for Pro*COBOL release 9.0.1
\demo\procob	Sample programs for Pro*COBOL release 1.8.52
\demo\sql	SQL scripts for sample programs
\doc\procob2	Readme files for Pro*COBOL 9.0.1
\doc\procob	Readme files for Pro*COBOL 1.8.52
\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.

Header File	Description
oraca.cob	Contains the Oracle Communications Area (ORACA), which helps you to diagnose runtime errors and to monitor your program's use of various Oracle resources.
oraca5.cob	ORACA5 is the COMP-5 version of ORACA.
sqlca.cob	Contains the SQL Communications Area (SQLCA), which helps you to diagnose runtime errors. The SQLCA is updated after every executable SQL statement.

Header File	Description
sqlca5.cob	SQLCA5 is the COMP-5 version of SQLCA.
sqlda.cob	Contains the SQL Descriptor Area (SQLDA), which is a data structure required for programs that use dynamic SQL Method 4.
sqlda5.cob	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 filename or directory name. For example, do not use the following formats:

- proc iname=test one.pc
- proc iname=d:\dir1\second dir\sample1.pc

2

Building Pro*COBOL Applications

This chapter provides an overview of building Oracle database applications with Pro*COBOL releases 9.0.1 and 1.8.52 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:** See the *Pro*COBOL Precompiler Programmer's Guide* for more information about Pro*COBOL commands, precompiler options, and configuration files.

The Pro*COBOL Commands

You can use one of these commands to precompile a file:

Use...	For Pro*COBOL Release...
`procob` *`filename`*	9.0.1
`procob18` *`filename`*	1.8.52

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 filenamed *`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 9.0.1, the configuration file is called `pcbcfg.cfg`. This file is located in the *ORACLE_BASE*`\`*ORACLE_HOME*`\precomp\admin` directory.
- For release 1.8.52, 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`

Check the following table to see 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. **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.

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:** For an example of a command line string, see the *Pro*COBOL Precompiler Programmer's Guide* 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 following compiler:

- 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 CBLLINK 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 `.cbl` 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 `.cbl` 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 CBLLINK Commands

`COBOL` and `CBLLINK` 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";
CBLLINK sample1
```

For static linking, the commands are:

```
COBOL sample1 /LITLINK /MAKESYN"COMP-5"="COMP";
CBLLINK 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 95/98 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 2-8 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 9.0.1, the batch files are located in *ORACLE_BASE*`\`*ORACLE_HOME*`\precomp\demo\procob2`. For release 1.8.52, 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 filenames 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 9.0.1
- *ORACLE_BASE*`\`*ORACLE_HOME*`\precomp\demo\procob` directory for release 1.8.52

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 2-4 for more information.

makeit.bat

The `makeit.bat` for release 9.0.1 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 1.8.52, 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 2–1, "Pro*COBOL Sample Programs" are located in the *ORACLE_BASE**ORACLE_HOME*\precomp\demo\procob2 (release 9.0.1) and / or *ORACLE_BASE**ORACLE_HOME*\precomp\demo\procob (release 1.8.52) 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)

Table 2–1 Pro*COBOL Sample Programs

Sample Program	Description
`lobdemo1.sql`	LOB datatypes (release 8.1.6 and on)

Index

N

O

P

R

S

U