

万水

Oracle技术丛书

王海亮 王海凤 张立民 等编著

# Oracle9i Pro\*C/C++

## 编程指南



中国水利水电出版社  
www.waterpub.com.cn

## 内 容 提 要

作者具有多年的 Oracle 开发和培训经验, 本书为开发人员提供了使用 Pro\*C/C++ 开发各种数据库应用的方法。针对各种不同类型的应用, 本书提供了非常详细、具体的开发方法, 并且为读者提供了相应的示例程序。通过学习本书, 读者一定能够快速掌握开发数据库应用的方法。

### 图书在版编目 (CIP) 数据

Oracle9i Pro\*C/C++ 编程指南 / 王海亮等编著. —北京: 中国水利水电出版社, 2003

(万水 Oracle 技术丛书)

ISBN 7-5084-1742-9

I. O… II. 王… III. ①关系数据库—数据库管理系统, Oracle9i ②C 语言—程序设计 IV. ①TP311.138 ②TP312

中国版本图书馆 CIP 数据核字 (2003) 第 088875 号

书 名	Oracle9i Pro*C/C++编程指南
作 者	王海亮 王海凤 张立民 等编著
出版、发行	中国水利水电出版社 (北京市三里河路 6 号 100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn
经 售	电话: (010) 63202266 (总机)、68331835 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	787×1000 毫米 16 开本 20.25 印张 448 千字
版 次	2004 年 1 月第一版 2004 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	36.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换

版权所有·侵权必究

# 前 言

Oracle 预编译器是 Oracle 公司提供的一整套预编译开发工具，这些预编译开发工具包括 Pro\*C/C++、Pro\*COBOL、Pro\*Fortran、Pro\*Ada，它们使得开发人员可以在应用开发语言 C/C++、COBOL、Fortran 和 Ada 中直接内嵌 SQL 语句和 PL/SQL 块，从而降低了应用开发难度。

C/C++是目前最流行的程序开发语言之一，当使用 C/C++语言开发 Oracle 数据库应用程序时，开发人员可以选择两种开发方法：一种是使用 OCI（Oracle Call Interface）函数；一种是使用 Pro\*C/C++。当使用 OCI 函数开发应用时，开发人员除了需要掌握 C/C++、SQL 和 PL/SQL 外，还必须掌握大量 OCI 函数的作用及使用方法，而当使用 Pro\*C/C++开发应用程序时，开发人员只需掌握 C/C++、SQL 和 PL/SQL。

综合以上各种原因，当使用 Oracle 预编译器开发数据库应用程序时，我们建议您采用 Pro\*C/C++。本书非常系统、具体地给大家提供了开发 Pro\*C/C++应用程序的方法和步骤，并且针对每部分内容都提供了完整的程序示例以及习题。通过本书学习，开发人员将可以快速地掌握使用 Pro\*C/C++开发应用程序的方法。

## 读者对象

本书是专门为 Oracle 应用开发人员提供的编程指南，不仅为应用开发人员提供了内嵌 SQL 语句和 PL/SQL 块的方法，而且还介绍了动态 SQL 应用、LOB 应用、对象类型应用、集合应用的开发方法。另外，应用开发人员还可以掌握 Oracle8i 的新特征（DML 返回子句、ANSI 动态 SQL 方法四）及 Oracle9i 的新特征（滚动游标）。

## 目标

在学习了本书之后，读者应该达到以下目标：

- 掌握使用 Pro\*C/C++预编译工具的方法。
- 掌握将 Pro\*C/C++集成到 Visual C++中的方法。
- 掌握内嵌 SQL 语句和 PL/SQL 块的语法及各种规则。
- 掌握开发动态 SQL 应用、LOB 应用、对象应用以及集合应用的方法。

## 本书组织及特点

本书共包括十六章：

- 第 1 章：Pro\*C/C++简介，介绍了 Oracle 预编译器、SQL 和 PL/SQL 的作用以及 Pro\*C/C++的优点和新特征。
- 第 2 章：使用预编译器，介绍了如何在命令行和图形界面中使用 Pro\*C/C++预编译工具，并且介绍了所有预编译选项的作用。
- 第 3 章：建立应用开发环境，介绍了将 Pro\*C/C++工具集成到 Visual C++环境中的方法，以及预编译、编译并生成应用程序的方法。
- 第 4 章：Pro\*C/C++基础，介绍了 Pro\*C/C++编程的指导方针、编程的必备知识以及编程的高级知识。
- 第 5 章：连接到 Oracle 数据库，介绍了连接到数据库、建立并发连接、使用数据库链以及控制事务的方法。
- 第 6 章：错误处理，介绍了 ANSI 错误处理方法（SQLCODE 和 SQLSTATE）、Oracle 错误处理方法（SQLCA 结构）以及使用 ORACA 结构诊断应用性能的方法。
- 第 7 章：内嵌 SQL 语句，不仅介绍了内嵌 SQL 语句的方法，而且还介绍了宿主变量和指示变量、宿主结构和指示结构的作用及使用方法。另外，还介绍了 Oracle8i 的新特征——DML 返回子句。
- 第 8 章：使用数组，介绍了宿主数组和指示数组、宿主结构数组和指示结构数组的作用及使用方法。
- 第 9 章：使用游标，介绍了使用游标和 PL/SQL 游标变量的方法，以及使用游标更新和删除数据的方法，另外还介绍了 Oracle9i 的新特征——滚动游标。
- 第 10 章：内嵌 PL/SQL 块，介绍了在 Pro\*C/C++中内嵌 PL/SQL 块的方法，以及开发和调用 PL/SQL 子程序的方法。
- 第 11 章：动态 SQL，介绍了静态 SQL 和动态 SQL 的区别，以及使用动态 SQL 方法一、动态 SQL 方法二以及动态 SQL 方法三的方法。
- 第 12 章：高级动态 SQL，不仅介绍了 Oracle 动态 SQL 方法四，而且还介绍了 Oracle8i 的新特征——ANSI 动态 SQL 方法四。
- 第 13 章：开发 LOB 应用，介绍了 LOB 类型（CLOB、BLOB、BFILE）和 LOB 语句，并介绍了开发 LOB 应用的方法。
- 第 14 章：开发对象应用，介绍了对象类型的作用以及使用联合接口和导航接口开发对象应用的方法。
- 第 15 章：开发集合应用，介绍了嵌套表和 VARRAY 的作用以及开发集合应用的具体方法。

- 第 16 章：开发 C++ 应用，介绍了 Pro\*C/C++ 所提供的 C++ 支持以及开发 C++ 应用的方法。

## 其他说明

本工作室人员都具有丰富的 Oracle 应用开发、培训和技术支持经验，曾经为电信、移动、联通、油田、银行、社保、证券期货、海关、教育等行业进行了 Oracle 技术支持和培训，并且获得用户的一致好评。如果您有 Oracle 应用开发、培训和技术支持需求，欢迎您来电来函与我们联系。

联系电话：0471-4907212

E-mail: whl88321@21cn.com

编者

2003 年 9 月

# 目 录

前言

<b>第 1 章 PRO*C/C++简介</b> .....	1
1.1 Oracle 预编译器.....	1
1.2 SQL.....	2
1.3 PL/SQL.....	3
1.4 Pro*C/C++的优点.....	3
1.5 Pro*C/C++的新特征.....	4
1.6 小结.....	5
<b>第 2 章 使用预编译器</b> .....	6
2.1 常用预编译选项.....	6
2.2 使用配置文件.....	7
2.3 预编译 Pro*C/C++源程序.....	8
2.3.1 在命令行预编译.....	8
2.3.2 使用 GUI 工具预编译.....	9
2.4 其他预编译选项.....	14
2.5 小结.....	27
2.6 习题.....	27
<b>第 3 章 建立应用开发环境</b> .....	29
3.1 集成 Pro*C/C++到 VC.....	29
3.1.1 指定执行文件路径.....	29
3.1.2 指定头文件路径.....	31
3.1.3 生成应用程序.....	33
3.2 添加 Pro*C/C++到“工具”菜单.....	41
3.3 小结.....	43
3.4 习题.....	43
<b>第 4 章 PRO*C/C++基础</b> .....	44
4.1 编程指导方针.....	44
4.2 编程必备知识.....	48
4.2.1 Oracle 数据类型.....	48
4.2.2 定义变量.....	50

4.2.3	内嵌 SQL 语句简介 .....	53
4.2.4	内嵌 PL/SQL 块简介 .....	54
4.3	编程高级知识 .....	55
4.3.1	处理字符数据 .....	55
4.3.2	等价数据类型 .....	57
4.3.3	C 预处理器 .....	58
4.3.4	Oracle 预处理器 .....	59
4.3.5	SQLLIB 公用函数 .....	60
4.3.6	嵌入 OCI 函数 .....	61
4.4	编程思路 .....	63
4.5	小结 .....	65
4.6	习题 .....	66
<b>第 5 章</b>	<b>连接到 ORACLE .....</b>	<b>67</b>
5.1	连接到数据库 .....	67
5.2	建立并发连接 .....	69
5.3	使用数据库链 .....	73
5.4	控制事务 .....	76
5.5	小结 .....	80
5.6	习题 .....	80
<b>第 6 章</b>	<b>错误处理 .....</b>	<b>82</b>
6.1	使用 SQLSTATE 和 SQLCODE .....	82
6.2	使用 SQLCA .....	89
6.3	使用 WHENEVER 语句 .....	91
6.4	使用 ORACA .....	96
6.5	小结 .....	99
6.6	习题 .....	100
<b>第 7 章</b>	<b>内嵌 SQL .....</b>	<b>101</b>
7.1	宿主变量 .....	101
7.2	指示变量 .....	105
7.3	内嵌 SQL 语句 .....	108
7.4	宿主结构 .....	112
7.5	指示结构 .....	115
7.6	小结 .....	118
7.7	习题 .....	118

<b>第 8 章</b>	<b>使用数组</b>	<b>120</b>
8.1	宿主数组	120
8.2	指示数组	127
8.3	宿主结构数组	130
8.4	指示结构数组	135
8.5	小结	138
8.6	习题	139
<b>第 9 章</b>	<b>使用游标</b>	<b>140</b>
9.1	使用非滚动游标	140
9.2	使用滚动游标	143
9.3	更新或删除游标行	147
9.4	使用游标变量	151
9.5	小结	155
9.6	习题	155
<b>第 10 章</b>	<b>内嵌 PL/SQL</b>	<b>156</b>
10.1	PL/SQL 简介	156
10.2	内嵌 PL/SQL 块	160
10.3	建立 PL/SQL 子程序	165
10.4	调用 PL/SQL 子程序	168
10.5	小结	172
10.6	习题	173
<b>第 11 章</b>	<b>动态 SQL</b>	<b>174</b>
11.1	静态 SQL 和动态 SQL	174
11.2	动态 SQL 方法一	176
11.3	动态 SQL 方法二	179
11.4	动态 SQL 方法三	182
11.5	小结	187
11.6	习题	188
<b>第 12 章</b>	<b>高级动态 SQL</b>	<b>189</b>
12.1	Oracle 动态 SQL 方法四	189
12.1.1	SQLDA 结构	189
12.1.2	使用 Oracle 动态 SQL 方法四	192
12.2	ANSI 动态 SQL 方法四	203
12.3	小结	210
12.4	习题	210

的六八



<b>第 13 章 开发 LOB 应用</b> .....	<b>211</b>
13.1 LOB 简介 .....	211
13.2 LOB 语句 .....	212
13.3 开发 LOB 应用 .....	216
13.3.1 访问 CLOB .....	216
13.3.2 访问 BLOB .....	222
13.3.3 访问 BFILE .....	228
13.4 小结 .....	233
13.5 习题 .....	233
<b>第 14 章 开发对象应用</b> .....	<b>234</b>
14.1 对象类型简介 .....	234
14.2 使用 OTT .....	236
14.3 使用联合接口 .....	238
14.4 使用导航接口 .....	244
14.5 小结 .....	252
14.6 习题 .....	253
<b>第 15 章 开发集合应用</b> .....	<b>254</b>
15.1 集合简介 .....	254
15.2 集合语句 .....	256
15.3 编写集合应用程序 .....	258
15.4 小结 .....	267
15.5 习题 .....	267
<b>第 16 章 开发 C++ 应用</b> .....	<b>269</b>
16.1 C++ 支持 .....	269
16.2 C++ 程序示例 .....	270
16.3 小结 .....	273
<b>附录 A 习题参考答案</b> .....	<b>274</b>
<b>附录 B 内嵌 SQL 命令</b> .....	<b>295</b>

# 第 1 章 Pro\*C/C++简介

Pro\*C/C++是 Oracle 公司提供的预编译工具。通过使用 Pro\*C/C++，开发人员可以在 C/C++ 语言中内嵌 SQL 语句和 PL/SQL 块，从而简化应用程序的开发。在开发数据库应用程序时，为了简化和加快应用程序的开发，开发人员可以使用各种类型的应用开发工具，例如 Developer 2000、Visual Basic、Delphi 等。而为了开发高效率的数据库应用程序，多数开发人员都会采用 C/C++ 语言。在 Oracle 数据库中，使用 C/C++ 开发应用程序有两种方法：一种是使用 OCI(Oracle Call Interface) 函数；另一种是使用 Pro\*C/C++。当使用 OCI 函数开发应用程序时，开发人员不仅要掌握 C/C++、SQL 和 PL/SQL 外，还必须掌握 Oracle 提供的大量 OCI 函数，而使用 Pro\*C/C++ 开发应用程序时，开发人员只要掌握 C/C++ 语言并能够熟练应用 SQL 语句就可以了。因为 Pro\*C/C++ 比 OCI 更加简单、更加易于应用开发，所以在开发高效率的数据库应用程序时建议您采用 Pro\*C/C++，而不是使用 OCI。在学习了本章之后，大家应该能够完成以下任务：

- 了解什么是 Oracle 预编译器。
- 了解 SQL 和 PL/SQL 的作用。
- 了解 Pro\*C/C++ 的新特征。

## 1.1 Oracle 预编译器

Oracle 预编译器是 Oracle 公司提供的一种程序设计工具，Oracle 预编译器包括 Pro\*C/C++、Pro\*COBOL、Pro\*FORTRAN、Pro\*Ada 等 Pro\* 系列工具。通过使用预编译器，Oracle 允许应用开发人员在高级语言中内嵌 SQL 语句和 PL/SQL 块，从而简化了数据库应用程序的开发。使用预编译器开发内嵌 SQL 程序的步骤如图 1-1 所示。

在各种 Oracle 预编译器中，最常用的预编译器就是 Pro\*C/C++，该预编译器使得开发人员可以在 C/C++ 语言中内嵌 SQL 语句和 PL/SQL 块。如图 1-1 所示，当开发 Pro\*C/C++ 应用程序时，第一步需要编写预编译源程序(.pc)，在该预编译源程序中包含了内嵌 SQL 和 PL/SQL；在编写了源程序之后，第二步需要将内嵌 SQL 和 PL/SQL 代码转变为对 SQLLIB 运行库函数的调用，并生成 C/C++ 源文件(.c 或.cpp)；第三步需要编译 C/C++ 源文件，生成目标代码(.obj)；第四步需要链接目标代码和 SQLLIB 运行库，生成可执行文件(.exe)，最后可以运行并调试该可执行文件，并完成应用程序的最终开发。

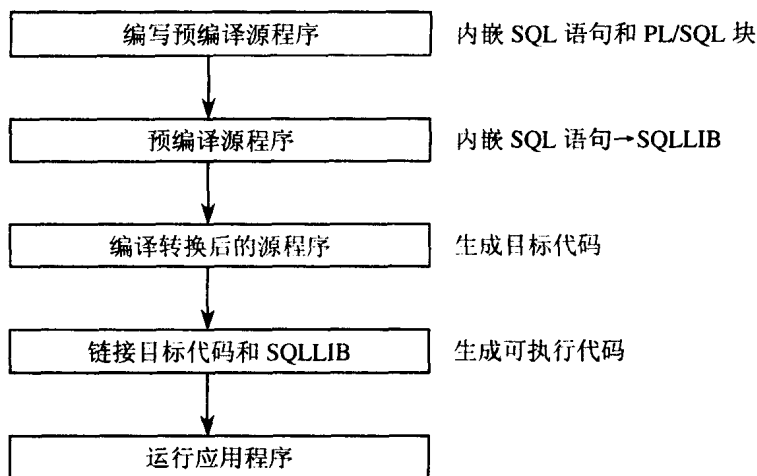


图 1-1 使用预编译器开发内嵌 SQL 程序的步骤

## 1.2 SQL

20 世纪 80 年代初, ANSI (American National Standards Institute) 数据库标准委员会开始制订有关关系语言的标准, 但是直到 1986 年, 数据库标准委员会才推出第一个 SQL 语言标准 SQL-86。随着关系数据库技术的不断发展, SQL 标准也在不断地进行扩展和修正, 并且数据库标准委员会又先后推出和制订了 SQL-89、SQL-92 和 SQL-99 标准。1979 年, Relational Software 公司 (Oracle 前身) 首先向市场推出了 SQL 执行工具。Oracle 的 SQL 完全遵从 ANSI 的 SQL 标准, 并且将最新的 SQL-99 标准集成到了 Oracle9i 数据库中。

SQL (Structure Query Language) 是关系数据库的基本操作语言, 是应用程序与数据库进行交互操作的接口。该语言将数据查询 (Data Query)、数据操纵 (Data Manipulation)、数据定义 (Data Definition) 和数据控制 (Data Control) 功能集于一体, 从而使得应用开发人员、数据库管理员、最终用户都可以通过 SQL 语言对数据库进行操作。在 Pro\*C/C++ 中可以内嵌任何类型的 SQL 语句:

- 数据查询语言: SELECT 语句。
- 数据操纵语言 (DML): INSERT、UPDATE、DELETE。
- 数据定义语言 (DDL): CREATE TABLE、ALTER TABLE、DROP TABLE 等。
- 数据控制语言 (DCL): GRANT 和 REVOKE。
- 事务控制语言: COMMIT、ROLLBACK、SAVEPOINT。

SQL\*Plus 是 Oracle 公司提供的的一个工具程序, 它可以用于运行 SQL 语句和 PL/SQL 块, 并且可以用于跟踪和调试 SQL 语句。当使用 Pro\*C/C++ 开发应用程序时, 你可以在 SQL\*Plus 中测试和调整各种 SQL 语句, 然后稍加修改就可以嵌入到 Pro\*C/C++ 源程序中。但是大家需

要注意，在 Pro\*C/C++应用程序中只能嵌入 SQL 语句和 PL/SQL 块，而不能嵌入 SQL\*Plus 命令（例如 spool 命令）。

### 1.3 PL/SQL

PL/SQL 是 Oracle 在标准 SQL 语言上的过程性扩展，它支持过程结构、变量定义以及错误处理等，在运行 Oracle 的任何平台中都可以使用 PL/SQL。通过使用 PL/SQL 块，可以在一个 PL/SQL 块中包含多条 SQL 语句和 PL/SQL 语句。在编写 Pro\*C/C++应用程序时，也可以将 PL/SQL 块内嵌到 C/C++程序中，其最大优点是可以降低网络开销、提高应用程序性能。对于其他 RDBMS（例如 SQL Server 和 Sybase）来说，当应用程序访问 RDBMS 时，每次只能发送单条 SQL 语句，如图 1-2 所示。

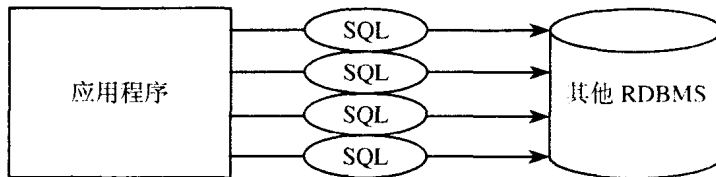


图 1-2 应用程序访问其他 RDBMS

如图 1-2 所示，执行四条 SQL 语句需要在网络上发送四次语句。而对于 Oracle 数据库来说，通过使用 PL/SQL 块，可以将多条 SQL 语句组织到同一个 PL/SQL 块中，从而大大降低了网络开销，进而提高了应用性能，如图 1-3 所示。

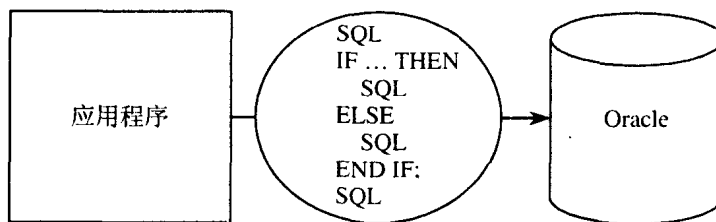


图 1-3 应用程序访问 Oracle

如图 1-3 所示，通过使用内嵌 PL/SQL 块，只需要发送一次 PL/SQL 块语句就可以完成所有 SQL 语句的发送和数据处理工作。

### 1.4 Pro\*C/C++的优点

Pro\*C/C++提供了以下一些特征和优点：

- 允许使用 C/C++ 语言编写 Oracle 应用程序。
- 允许在 C/C++ 语言中内嵌 SQL 语句，并遵从 ANSI/ISO 标准。
- 支持动态 SQL 特征，从而使得应用程序可以在运行时接收和建立 SQL 语句。
- 允许编写共享服务器进程应用。
- 可以自动在 Oracle 数据类型和 C 数据类型之间进行转换。
- 允许内嵌 PL/SQL 块，从而提高应用性能。
- 提供了大量的预编译选项，并且在预编译源程序时允许改变选项值。
- 提供了数据类型等价，使得开发人员可以控制数据输入和数据输出。
- 允许单独预编译多个程序模块，然后链接为一个可执行程序。
- 当预编译源程序时，可以检查内嵌 SQL 语句和 PL/SQL 块的语法和语义。
- 通过使用 Oracle Net，允许并发访问多个 Oracle 数据库。
- 允许使用宿主数组作为输入和输出变量，从而降低网络和处理开销，提高应用性能。
- 允许根据条件预编译应用代码，从而使得应用程序可以兼容于不同平台。
- 通过提供 SQLCA 结构、WHENEVER 和 DO 语句，使得应用程序可以处理 SQL 错误和警告。
- 支持对象类型，从而使得开发人员可以开发对象类型应用。
- 支持集合类型（变长数组和嵌套表），从而使得开发人员可以开发集合类型应用。
- 支持 LOB 类型，从而使得开发人员可以开发 LOB 应用。

## 1.5 Pro\*C/C++的新特征

### 1. Oracle8i 的新特征

Oracle8i 具有以下新特征：

- 提供了内嵌的 LOB 接口。通过使用内嵌的 LOB 接口，应用开发人员可以在 Pro\*C/C++ 应用程序中直接实现 DBMS\_LOB 包的相同功能。
- 提供了对 ANSI 动态 SQL 接口的支持。Oracle 动态 SQL 方法四不支持对象类型和集合类型，而 ANSI 动态 SQL 方法四不但可以支持所有的 Oracle 类型，而且简化了动态 SQL 应用的开发。
- 支持 DML 返回子句。通过在内嵌 INSERT、UPDATE 和 DELETE 上使用返回子句，可以取得 DML 操作后的数据。
- 扩展了对运行上下文（Runtime Context）的支持，允许开发人员指定特定的运行上下文。
- 支持外部存储过程，允许使用 Pro\*C/C++ 开发外部存储过程。
- 支持预提取行特征，从而降低了网络开销，提高了应用性能。
- 允许使用 CALL 语句调用 Java 存储过程。

## 2. Oracle9i 的新特征

Oracle9i 具有以下新特征:

- 支持 UNICODE。
- 支持 Oracle9i 所提供的新的日期时间类型。
- 支持 Oracle9i 所提供的对象类型继承。
- 支持 Oracle9i 所提供的多级集合类型嵌套。
- 支持在 Pro\*C/C++应用程序中使用滚动游标。
- 支持在 Pro\*C/C++应用程序中使用连接池。

## 1.6 小结

本章简单介绍了 Pro\*C/C++的作用以及 SQL 和 PL/SQL 的作用。在学习了本章之后,大家应该了解以下一些内容:

- 了解使用 Pro\*C/C++和 OCI 开发数据库应用程序的区别。
- 了解使用 Pro\*C/C++开发数据库应用程序的大致步骤。
- 了解 Pro\*C/C++的优点以及各种新特征。

## 第 2 章 使用预编译器

当开发 Pro\*C/C++ 应用程序时，因为其源程序代码包含了内嵌的 SQL 语句和 PL/SQL 块，所以不能直接使用 C/C++ 工具进行编译和链接，而必须首先将内嵌的 SQL 语句和 PL/SQL 块转换为对 SQLLIB（Oracle 运行库）的调用，该项任务是通过 Pro\*C/C++ 预编译器来完成的，本章将详细介绍预编译选项以及预编译器的使用方法。在学习完本章之后，大家应该能够完成以下一些任务：

- 掌握常用的预编译选项。
- 学会使用配置文件。
- 学会在命令行使用 Pro\*C/C++ 预编译器。
- 学会使用 Pro\*C/C++ 预编译器图形工具。
- 了解其他的预编译选项。

### 2.1 常用预编译选项

当将 Pro\*C/C++ 源程序转换为 C/C++ 代码时，必须进行预编译。而在使用预编译工具时，必须提供相应的预编译选项。常用的预编译选项包括 INAME、ONAME、INCLUDE、PARSE、SQLCHECK 和 USERID，大家必须清楚这些预编译选项的作用。

#### 1. INAME

作用：该选项用于指定输入文件名称（Pro\*C/C++ 源程序名），并且该选项是必须的。当指定 Pro\*C/C++ 源程序时，文件扩展名 .pc 可以忽略。如果该选项是命令行的第一个选项，那么可以忽略 “INAME=”。

语法：

```
INAME=path_and_filename
```

#### 2. ONAME

作用：该选项用于指定输出文件名称，输出文件是预编译后生成的 C/C++ 代码文件。如果在预编译时不指定该选项，那么输出文件的名称会自动使用 INAME 所指定的输入文件名，而文件名后缀则由选项 CPP\_SUFFIX 确定。

语法：

```
ONAME=path_and_filename
```

#### 3. INCLUDE

作用：该选项用于指定 #include 或 EXEC SQL INCLUDE 语句的头文件所在路径。

语法:

```
INCLUDE=路径名或 INCLUDE=(路径 1,路径 2,...)
```

说明: 当执行预编译命令时, 预编译器会按照以下顺序搜索头文件:

- (1) 当前目录。
- (2) `SYS_INCLUDE` 选项所指定的系统目录。
- (3) `INCLUDE` 选项所指定的目录。
- (4) 标准头文件的内置目录。

#### 4. PARSE

作用: 该选项用于指定预编译器解析 Pro\*C/C++源文件的方法, 默认值为 `FULL`。

语法:

```
PARSE={FULL | PARTIAL | NONE}
```

注意事项:

- (1) 如果要生成 C++代码, 则必须将 `PARSE` 设为 `NONE` 或 `PARTIAL`。
- (2) 如果 `PARSE` 被设置为 `NONE` 或 `PARTIAL`, 那么必须在定义部分内定义宿主变量。

#### 5. SQLCHECK

作用: 该选项用于指定 SQL 语法和语义的检查方式, 默认值为 `SYNTAX`。该选项既可以内嵌到程序中, 也可以在命令行键入。

语法:

```
SQLCHECK={SEMANTICS | FULL | SYNTAX }
```

注意事项:

(1) 如果 Pro\*C/C++源程序内嵌了 PL/SQL 块, 那么在预编译时必须将该选项设置为 `SEMANTICS` 或 `FULL`。

(2) 当将 `SQLCHECK` 设置为 `SEMANTICS` 或 `FULL` 时, 在预编译源程序时需要进行语义检查。因为语义检查需要访问数据字典, 所以必须设置 `USERID`。

#### 6. USERID

作用: 该选项用于指定用户名、口令和网络服务名。

语法:

```
USERID=用户名/口令@网络服务名
```

注意事项: 当将 `SQLCHECK` 设置为 `SEMANTICS` 或 `FULL` 时, 必须设置 `USERID`。

## 2.2 使用配置文件

配置文件是一个可以编辑的文本文件, 用于存放 Pro\*C/C++的预编译选项。当使用配置文件时, 在其中的每一行上只能设置一个选项。需要注意, 如果在配置文件的某一行上包含有多个选项, 那么预编译命令只会使用第一个选项。配置文件又包括系统配置文件和用户配置文件两种。



## 1. 系统配置文件

系统配置文件是指由 Oracle 提供的默认配置文件。默认情况下，当运行预编译工具时，预编译工具会自动使用系统配置文件中的相应选项。当安装 Pro\*C/C++ 时，Oracle 会自动建立名称为 `pcscfg.cfg` 的配置文件，在 Windows 95/98/2000/XP 平台中该文件位于 `%ORACLE_HOME%\precomp\admin` 子目录下，其中 `ORACLE_HOME` 为 Oracle 软件安装路径。

当使用 Pro\*C/C++ 预编译工具执行预编译操作时，需要指定各种预编译选项。在预编译选项中，某些选项（例如 `INCLUDE`、`PASRE`）必须设置，并且其值相对固定。为了避免每次预编译都设置这些选项，可以在系统配置文件中设置这些常用的预编译选项。示例如下：

```
parse=full
include=g:\oracle\ora92\precomp\public
include=g:\oracle\VC98\include
include=g:\oracle\ora92\oci\include
```

如上例所示，`PARSE=FULL` 表示要解析源文件的所有语法，包括 C 语法和内嵌 SQL 的语法，而 `include` 则指定了头文件所在路径，包括 VC 头文件所在路径、Pro\*C/C++ 头文件所在路径以及 OCI（Oracle Call Interface）头文件所在路径。

## 2. 用户配置文件

用户配置文件是指由开发人员所定义的私有配置文件。对于系统配置文件来说，当执行预编译操作时，Oracle 会自动调用该配置文件；如果要使用用户配置文件，那么在预编译时必须设置 `CONFIG` 选项。另外，需要注意，应该将公用的预编译选项放在系统配置文件 `pcscfg.cfg` 中，而用户配置文件应该存放私有的预编译选项。

## 2.3 预编译 Pro\*C/C++ 源程序

当编写了 Pro\*C/C++ 源程序之后，因为源程序包含了内嵌的 SQL 和 PL/SQL 代码，所以不能直接进行编译和链接，而必须首先进行预编译。预编译 Pro\*C/C++ 源程序有两种方法：一种是使用命令行预编译工具，这种方法适合于任何 OS 平台；另一种则是使用图形界面工具（GUI），这种方法只适用于 Windows 平台。本节将详细介绍这两种工具的使用方法。

### 2.3.1 在命令行预编译

当在命令行执行预编译操作时，可以使用 Oracle 提供的预编译工具程序 `proc` 来完成，语法如下：

```
proc [OPTION_NAME=value] [OPTION_NAME=value] ...
```

其中，`option_name` 用于指定要设置的预编译选项，而 `value` 则用于指定预编译选项的值。需要注意，为了简化命令行的选项设置，应该在系统配置文件中设置常用的、固定的预编译选项。下面是使用 `proc` 工具的一些示例。