

# Pro\*C/C++ Precompiler

Getting Started

Release 8.1.6 for Windows

January 2000

Part No. A73023-01

---

Pro\*C/C++ Precompiler Getting Started, Release 8.1.6 for Windows

Part No. A73023-01

Copyright © 1994, 2000 Oracle Corporation. All rights reserved.

Contributors: Riaz Ahmed, Eric Belden, Sharon Castledine, Joseph Garcia, Lisa Giambruno, Neeraj Gupta, Bernie Harris, Ana Hernandez, Mark Kennedy, Robert Knecht, Shiva Prasad, Ali Shehade, Helen Slattery, Jeff Stein, Nicole Sullivan, Janice Wong, Martha Woo, Ravi Gooty

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; 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 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. Oracle Corporation does not warrant that this document is 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, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

**Restricted Rights Notice** Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions 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 Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Net8, Oracle8, PL/SQL, and Pro\*C/C++ are trademarks or registered trademarks of Oracle Corporation. All other company or product names mentioned are used for identification purposes only and may be trademarks of their respective owners.

---

---

# Contents

<b>Contact Us!</b> .....	vii
How to Contact Oracle Technical Publications .....	viii
How to Contact Oracle Support Services .....	ix
Resources for Oracle Partners and Developers .....	xiii
<b>Preface</b> .....	xvii
Prerequisites .....	xviii
Intended Audience .....	xviii
How This Guide Is Organized .....	xviii
Conventions .....	xix
Documentation Library .....	xxi
Related Documents .....	xxii
<b>1 Introducing Pro*C/C++</b>	
<b>What is Pro*C/C++?</b> .....	1-2
<b>Features</b> .....	1-2
<b>Restrictions</b> .....	1-2
<b>Directory Structure</b> .....	1-3
<b>2 Using Pro*C/C++</b>	
<b>Using the Graphical User Interface</b> .....	2-2
Title Bar .....	2-2
Menu Bar .....	2-3
Toolbar .....	2-3

Information Pane .....	2-4
Status Bar .....	2-5
<b>Creating and Precompiling a Pro*C/C++ Project</b> .....	2-5
Opening a Project.....	2-6
Setting the Default Extension of Output Files .....	2-6
Changing the Name of an Existing Input or Output File .....	2-6
Adding Files to the Project .....	2-7
Deleting Files from the Project.....	2-8
Setting the Precompiler Options.....	2-8
Specifying Database Connection Information.....	2-10
Precompiling a Pro*C/C++ Project .....	2-11
Checking the Results .....	2-12
Fixing Errors .....	2-12
Exiting Pro*C/C++ .....	2-13
<b>Using Pro*C/C++ at the Command Line</b> .....	2-13
<b>Header Files</b> .....	2-14
<b>Library Files</b> .....	2-15
<b>Building Multi-Threaded Applications</b> .....	2-15
<b>Precompiler Options</b> .....	2-16
Configuration File.....	2-16
CODE.....	2-16
DBMS.....	2-16
INCLUDE.....	2-16
PARSE.....	2-16
<b>Using Pro*C/C++ with the Oracle XA Library</b> .....	2-17
Compiling and Linking a Pro*C/C++ Program with XA.....	2-17
XA Dynamic Registration.....	2-18
Adding an Environmental Variable for the Current Session.....	2-18
Adding a Registry Variable for All Sessions .....	2-18
XA and TP Monitor Information .....	2-19

### 3 Sample Programs

<b>Sample Program Descriptions</b> .....	3-2
<b>Building the Demonstration Tables</b> .....	3-8
<b>Building the Sample Programs</b> .....	3-9

Setting the Path for the Sample .pre Files .....	3-10
--------------------------------------------------	------

## **A Integrating Pro\*C/C++ into Microsoft Visual C++**

<b>Integrating Pro*C/C++ within Microsoft Visual C++ Projects .....</b>	<b>A-2</b>
Specifying the Location of the Pro*C/C++ Executable.....	A-2
Specifying the Location of the Pro*C/C++ Header Files .....	A-3
Adding .pc Files to a Project .....	A-4
Adding References to .c Files to a Project .....	A-4
Adding the Pro*C/C++ Library to a Project .....	A-5
Specifying Custom Build Options.....	A-6
<b>Adding Pro*C/C++ to the Tools Menu .....</b>	<b>A-7</b>

## **Index**



---

---

# Contact Us!

## Pro\*C/C++ Precompiler Getting Started, Release 8.1.6 for Windows

Part No. A73023-01

This document describes how to contact Oracle Corporation if you have issues with the documentation or software.

---

Read the section...	If you...
<a href="#">How to Contact Oracle Technical Publications</a> on page viii	Have issues with Documentation
<a href="#">How to Contact Oracle Support Services</a> on page ix	Have issues with Software
<a href="#">Resources for Oracle Partners and Developers</a> on page xiii	Want to join an Oracle partner or application developer program

---

---

## How to Contact Oracle Technical Publications

Oracle Corporation 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 guide?
- Do you have suggestions for improvement? Please indicate the chapter, section, and page number (if available).

You can send comments regarding documentation in the following ways:

- Electronic mail - [ntdoc@us.oracle.com](mailto:ntdoc@us.oracle.com)
- FAX - (650) 506-7370 Attn: Oracle Windows Platforms Server Documentation
- Postal service:  
Oracle Corporation  
Windows Platforms Server Documentation Manager  
500 Oracle Parkway, MS 1OP8,  
Redwood Shores, CA 94065  
USA

If you would like a reply, please provide your name, address, and telephone number.



---

# How to Contact Oracle Support Services

*Please copy this form and distribute within your organization as necessary.*

Oracle Support Services can be reached at the following telephone numbers and Web sites. The hours of business are detailed in your support contract and the *Oracle Customer Support Guide* in your kit.

---

<b>Oracle Support Services In...</b>	<b>Call...</b>
United States of America	+ (650) 506-1500 for customers with support contracts. + (650) 506-5577 to obtain a support contract.
Europe	+44 1344 860 160 or the local support center in your country.
All other locations	The telephone number for your country listed at the following Web site: <a href="http://www.oracle.com/support/contact_us/sup_hot_phone.html">http://www.oracle.com/support/contact_us/sup_hot_phone.html</a>  Oracle Support Services telephone numbers are also listed in the <i>Oracle Customer Support Guide</i> in your kit.

---

Please complete the following checklist before you call. If you have this information ready, your call can be processed much quicker.

- Your CPU Support Identification Number (CSI Number) if applicable.

---

---

- The hardware name on which your application is running.

---

---

---

- 
- ❑ The operating system name and release number on which your application is running.

- To verify the operating system version on Windows NT, enter the following at the MS-DOS command prompt:

```
C:\> WINMSD
```

The *Windows NT Diagnostics* dialog box displays the operating system and Service Pack version.

---

---

---

- ❑ The release numbers of the Oracle Server and associated products involved in the current problem. For example, Oracle8i Enterprise Edition release 8.1.6.0.0 and Oracle Enterprise Manager release 2.1.0.0.0.

- To verify the release number of the Oracle Server, connect to the database using a tool such as SQL\*Plus. The release number is displayed. For example:

```
Connected to:
Oracle8i Enterprise Edition Release 8.1.6.0.0 - Production
With the Partitioning and Java options
PL/SQL Release 8.1.6.0.0 - Production
```

---

---

---

---

- ❑ The third-party software version you are using.

- To verify an application version, from the application's Help menu, select About...

---

---

---

- 
- ❑ The exact error codes and messages. Please write these down as they occur. They are critical in helping Oracle Support Services to quickly resolve your problem. Note whether there were no errors reported.

---

---

---

---

---

- ❑ A description of the issue, including:

- **What happened?** For example, the command used and its result.

---

---

---

- **When did it happen?** For example, during peak system load, or after a certain command, or after an operating system upgrade. In addition, what was happening when the problem occurred?

---

---

---

- **Where did it happen?** For example, on a particular system, or within a certain procedure or table.

---

---

---

- 
- **What is the extent of the problem?** For example, production system unavailable, or moderate impact but increasing with time, or minimal impact and stable.
  - Did the problem affect one user, several users, or all users?
  - Has anything changed? For example, if this is an operation that used to work and now fails, what is different? Can you undo any recent changes, to verify whether they are relevant to the issue?

---

---

---

- **Can the problem be reproduced?** This is a critical question for support analysts. For example, did the problem recur on the same system, under the same circumstances? Can the problem be reproduced on another system? Additionally:
  - Does installing a software component fail on all client machines, or just one?
  - Do all clients fail to connect to the server, or just one?
  - If you are able to restart the server or database, does restarting the database or rebooting the server or client machine (if applicable) make a difference?

---

---

---

- Keep copies of the Oracle alert log, any trace files, core dumps, and redo log files recorded at or near the time of the incident. Oracle Support Services may need these to further investigate your problem.

To help analyze problems:

- Archive or delete old alert logs. When the database is started without an alert log, a new one is created. In some cases, if you force the problem to recur with a new alert log, the timestamps for the recorded events may indicate which events are relevant.

- Archive or delete old trace files. To check whether the file was modified, right-click and select Properties. The *Properties* dialog box displays the modification date.
- Check the operating system error logs, especially the System log and Application log. These files are relevant to the Oracle Server. To view these files, from the Start menu, choose Programs > Administrative Tools > Event Viewer, and choose System or Application from the Log main menu.

## Resources for Oracle Partners and Developers

This section provides information on partner programs and resources for Oracle database administrators and application developers.

Information Source	Description
Oracle Corporation Home Page <a href="http://www.oracle.com">http://www.oracle.com</a>	This Web site is the starting point for general information on Oracle Corporation.
Alliance Online <a href="http://alliance.oracle.com">http://alliance.oracle.com</a>	Oracle provides leading-edge technology, education, and technical support that enables you to effectively integrate Oracle into your business. By joining the Oracle Partner Program, you demonstrate to customers that you are committed to delivering innovative Oracle-based solutions and services.  The greater your commitment to Oracle, the more we can help you grow your business. It's that simple. The value you derive is associated directly with your level of commitment.
Oracle Education <a href="http://education.oracle.com/">http://education.oracle.com/</a>	Customers come to Oracle Education with a variety of needs. You may require a complete curriculum based on your job role to enable you to implement new technology. Or you may seek an understanding of technology related to your key area of responsibility to help you meet technical challenges. You may be looking for self-paced training that can be used as an ongoing resource for reference and hands-on practice. Or, you may be interested in an overview of a new product upgrade. Whatever your training need, Oracle Education has the solution.

Information Source	Description
Oracle Technology Network <a href="http://technet.oracle.com/">http://technet.oracle.com/</a>	The Oracle Technology Network is your definitive source for Oracle technical information for developing for the Internet platform. You will be part of an online community with access to free software, Oracle Technology Network-sponsored Internet developer conferences, and discussion groups on up-to-date Oracle technology. Membership is free.
Oracle Store <a href="http://oraclestore.oracle.com/">http://oraclestore.oracle.com/</a>	This is Oracle's online shopping center. Come to this site to find special deals on Oracle software, documentation, publications, computer-based training products, and much more.
Oracle Support Services' Support Web Center <a href="http://www.oracle.com/support/">http://www.oracle.com/support/</a>	Oracle Support Services offers a range of programs so you can select the support services you need and access them in the way you prefer: by telephone, electronically, or face to face. These award-winning programs help you maintain your investment in Oracle technology and expertise.  Here are some of the resources available in the Support Web Center:
<b>OracleMetaLink</b>  <a href="http://www.oracle.com/support/elec_sup/index.html">http://www.oracle.com/support/elec_sup/index.html</a>	<b>OracleMetaLink</b> is Oracle Support Services' premier Web support service. It is available to Oracle <i>metals</i> customers (Gold, Silver, Bronze), 24 hours a day, seven days a week.
<b>OracleLifecycle</b>  <a href="http://www.oracle.com/support/sup_serv/lifecycle/index.html">http://www.oracle.com/support/sup_serv/lifecycle/index.html</a>	<b>OracleLifecycle</b> is designed to deliver customized, industry-focused, full life-cycle support solutions that enable industry leaders to use Oracle technology to make smart business decisions, achieve operational excellence, and succeed in their markets.
<b>ExpertONLINE</b>  <a href="http://www.oracle.com/support/sup_serv/online/index.html">http://www.oracle.com/support/sup_serv/online/index.html</a>	Oracle Support Services has launched a new line of services called <b>ExpertONLINE</b> . These services provide online database administration for companies looking to supplement their existing DBA staff or fill a DBA role. Services range from <b>ExpertDETECT</b> , a monitoring, diagnostic, and recommendation service, to <b>ExpertDBA</b> , a full online database administration service.
<b>Virtual Support Analyst (VSA)</b>  <a href="http://www.oracle.com/support/sup_serv/vsa_start.html">http://www.oracle.com/support/sup_serv/vsa_start.html</a>	<b>VSA</b> is Oracle's Internet e-mail service; it is available to U.S. customers with an Oracle <i>metals</i> support agreement. With <b>VSA</b> , you can initiate a request for assistance through e-mail, bypassing the queues you may encounter when using telephone support. <b>VSA</b> also enables you to access Oracle's bug database.

---

Information Source	Description
<p>Customer Service</p> <p><a href="http://www.oracle.com/support/cus_serv/index.html">http://www.oracle.com/support/cus_serv/index.html</a></p>	<p>This site provides resources to make your interactions with Oracle as easy as possible. Among the things you can do are:</p> <ul style="list-style-type: none"> <li>■ Learn what is a CPU Support Identification (CSI) number</li> <li>■ Update your technical contact information</li> <li>■ Find out whom to contact for invoice and collection issues</li> <li>■ Request product update shipments</li> <li>■ Access a glossary of Oracle Support Services terms</li> </ul>
<p>U.S. Customer Visit Program</p> <p><a href="http://www.oracle.com/support/cus_serv/cus_visit.html">http://www.oracle.com/support/cus_serv/cus_visit.html</a></p>	<p>This U.S.-based program has been established to help our customers understand and obtain maximum benefit from the support services they have purchased.</p> <p>The visit typically offers a customized orientation presentation, a comprehensive overview and demonstration of Oracle's electronic services, and helpful tips on working more effectively with Oracle Support Services.</p>
<p>Support Web Center Library</p> <p><a href="http://www.oracle.com/support/library/index.html">http://www.oracle.com/support/library/index.html</a></p>	<p>This site contains articles, guides, and other documentation to help you leverage the wealth of knowledge and reference material that Oracle Support Services produces.</p>





---

---

# Preface

This guide provides introductory information for the Pro\*C/C++ precompiler running on Microsoft Windows operating systems. Specific topics discussed in this preface are:

- [Prerequisites](#)
- [Intended Audience](#)
- [How This Guide Is Organized](#)
- [Conventions](#)
- [Documentation Library](#)
- [Related Documents](#)

## Prerequisites

This guide assumes that you are familiar with:

- Commands for deleting and copying files and the concepts of the search path, subdirectories, and path names.
- Microsoft Windows NT or Windows 95/98 operating system.
- Microsoft Visual C++ version 5.0 or higher.

## Intended Audience

This guide is necessary for anyone who wants to use the Pro\*C/C++ precompiler on Microsoft Windows.

## How This Guide Is Organized

This guide is organized as follows:

### **Chapter 1, "Introducing Pro\*C/C++"**

Describes Pro\*C/C++, the Oracle programmatic interface for the C and C++ languages running on Microsoft Windows NT and Windows 95/98 operating systems.

### **Chapter 2, "Using Pro\*C/C++"**

Explains how to create and precompile a project. Also describes the Pro\*C/C++ graphical user interface, from which you execute commands with Windows menus and icons or with keyboard equivalents, and using Pro\*C/C++ at the command line.

### **Chapter 3, "Sample Programs"**

Describes how to build Oracle database applications with Pro\*C/C++ using the sample programs that are included with this release, and provides an overview of how to build multi-threaded applications.

### **Chapter A, "Integrating Pro\*C/C++ into Microsoft Visual C++"**

Describes how to integrate Pro\*C/C++ into the Microsoft Visual C++ integrated development environment.

# Conventions

The following conventions are used in this guide.

Convention	Example	Meaning
All uppercase plain	SQL> ALTER DATABASE	Indicates command names, SQL reserved words, and keywords.
<i>Italic</i>	Italic is used to indicate a variable: <i>filename</i>	Indicates a value that you must provide. For example, if a command asks you to type <i>filename</i> , you enter the actual name of the file.  Italic is also used for emphasis in the text and to indicate the titles of other guides.
square brackets [ ]	x:\[pathname]\oracle\home_name	Encloses optional items. For example, when you create an OFA-compliant Oracle home directory, you can place an optional pathname before the \oracle pathname.  Square brackets also indicate a function key, for example [Enter].
C:\>	C:\ORACLE>	Represents the Windows platforms command prompt of the current hard disk drive. Your prompt may differ and may, at times, reflect the subdirectory in which you are working. Referred to as the <i>MS-DOS command prompt</i> in this guide.
Backslash (\) before a directory name	\bin	Indicates that the directory is a subdirectory of the root directory.

Convention	Example	Meaning
<i>oracle_home</i> and <i>oracle_base</i>	Go to the <i>oracle_base\oracle_home\bin</i> directory.	<p>In this Optimal Flexible Architecture (OFA)-compliant release, all subdirectories are no longer under a top level <i>oracle_home</i> directory. There is now a new top-level directory called <i>oracle_base</i> that by default is <i>c:\oracle</i>. The Oracle home directories are located directly under <i>oracle_base</i>.</p> <p>If you install Oracle8i release 8.1.6 on a computer where there is no other Oracle software on the computer, the default settings for the first Oracle home directory is <i>c:\oracle\ora81</i>. If you run Oracle Universal Installer again and install release 8.2.x, the second Oracle home directory is called <i>\ora82</i>.</p> <p>All directory path examples in this guide follow OFA conventions. For more information on OFA, see <i>Oracle8i Administrator's Guide for Windows NT</i>.</p>
<i>HOME_NAME</i>	OracleHOME_NAMETNSListener	Represents the Oracle home name. The home name can be up to sixteen alphanumeric characters. The only special character allowed in the home name is the underscore.
<i>HOMEID</i>	HOME0, HOME1, HOME2	Represents a unique registry subkey for each Oracle home directory in which you install products. A new <i>HOMEID</i> is created and incremented each time you install products to a different Oracle home directory on one machine. Each <i>HOMEID</i> contains its own configuration parameter settings for installed Oracle products.

Convention	Example	Meaning
Symbols	period . comma , hyphen - semicolon ; colon : equal sign = backslash \ single quote ' double quote " parentheses ()	Symbols other than brackets and vertical bars must be entered in commands exactly as shown.

## Documentation Library

This guide is part of a larger library of Oracle documentation. The Oracle documentation library consists of two types of documentation:

Documentation Type	Describes...
Operating System-specific	Installation, configuration, and use of Oracle products in a Windows NT or Windows 95/98 environment. Operating system-specific documents are occasionally referred to in the generic documentation set. These documents are easy to identify because they always mention their specific operating system in their title.
Generic	<p>Oracle database, Oracle networking, and Application Programming Interfaces information that is uniform across all operating system platforms. The majority of documents in your documentation set belong to this category. While reading through the generic documentation set, you are occasionally asked to refer to your platform (or operating system) documentation for procedures specific to the Windows NT or Windows 95/98 operating systems.</p> <p>To easily identify where these generic documentation references are described in your operating system documentation, see the index of this guide for the following entry:</p> <p>generic documentation references</p> <p>All generic documentation references described in this guide appear under this index entry.</p>

## Related Documents

For more information, see the following manuals.

- *Oracle8i Installation Guide for Windows NT*
- *Oracle8i Release Notes for Windows NT*
- *Oracle8i Administrator's Guide for Windows NT*
- *Oracle Enterprise Manager Administrator's Guide*
- *Net8 Administrator's Guide*
- *Oracle8i Parallel Server Concepts*
- *Oracle Parallel Server Administrator's Guide for Windows NT*
- *Getting to Know Oracle8i*
- *Oracle8i Concepts*
- *Oracle8i Reference*
- *Oracle8i Error Messages*
- *Pro\*C/C++ Precompiler Programmer's Guide*

---

---

# Introducing Pro\*C/C++

This chapter describes Pro\*C/C++, the Oracle programmatic interface for the C and C++ languages running on Window operating systems. Pro\*C/C++ enables you to build Oracle database applications in a Win32 environment. Specific topics discussed are:

- [What is Pro\\*C/C++?](#)
- [Features](#)
- [Restrictions](#)
- [Directory Structure](#)

**Note:** See the *Pro\*C/C++ Precompiler Programmer's Guide* for additional information.

## What is Pro\*C/C++?

Pro\*C/C++ precompiler allows you to create applications that access your Oracle database whenever rapid development and compatibility with other systems are your priorities.

The Pro\*C/C++ programming tool enables you to embed Structured Query Language (SQL) statements in a C or C++ program. The Pro\*C/C++ precompiler translates these statements into standard Oracle runtime library calls, then generates a modified source program that you can compile, link, and run in the usual way.

## Features

Pro\*C/C++ supports the following features:

- Remote access with Net8 or local access to Oracle databases
- Embedded PL/SQL blocks
- Bundled database calls, which can provide better performance in client/server environments
- Full ANSI compliance for embedded SQL programming
- PL/SQL version 8.0 and host language arrays in PL/SQL procedures
- Multi-threaded applications
- Full ANSI C compliance
- Microsoft Visual C++ support, version 6.0 for 32-bit applications

---

---

**Note:** Borland C++ is no longer supported.

---

---

## Restrictions

Pro\*C/C++ release 8.1.6 does not support 16-bit code generation.



## Directory Structure

Installing Oracle software creates a directory structure on your hard drive for the Oracle products. A main Oracle directory contains the Oracle subdirectories and files that are necessary to run Pro\*C/C++.

When you install Pro\*C/C++, Oracle Universal Installer creates a directory called `\precomp` in the `oracle_base\oracle_home` directory. This subdirectory contains the Pro\*C/C++ executable files, library files, and sample programs listed in [Table 1-1, "precomp Directory Structure"](#).

**Table 1-1** *precomp Directory Structure*

Directory Name	Contents
<code>\admin</code>	Configuration files
<code>\demo\proc</code>	Sample programs for Pro*C/C++
<code>\demo\sql</code>	SQL scripts for sample programs
<code>\doc\proc</code>	Readme files for Pro*C/C++
<code>\help\proc</code>	Help files for Pro*C/C++
<code>\lib\msvc</code>	Library files for Pro*C/C++
<code>\msg</code>	Message files
<code>\misc\proc</code>	Miscellaneous files for Pro*C/C++
<code>\public</code>	Header files

---



---

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

---



---



---

---

## Using Pro\*C/C++

This chapter explains how to create and precompile a project. It also describes the Pro\*C/C++ graphical user interface, from which you execute commands with Windows menus and icons or with keyboard equivalents, and using Pro\*C/C++ at the command line. Specific topics discussed are:

- [Using the Graphical User Interface](#)
- [Creating and Precompiling a Pro\\*C/C++ Project](#)
- [Using Pro\\*C/C++ at the Command Line](#)
- [Header Files](#)
- [Library Files](#)
- [Building Multi-Threaded Applications](#)
- [Precompiler Options](#)
- [Using Pro\\*C/C++ with the Oracle XA Library](#)

**Note:** See the *Pro\*C/C++ Precompiler Programmer's Guide* for additional information about Pro\*C/C++.

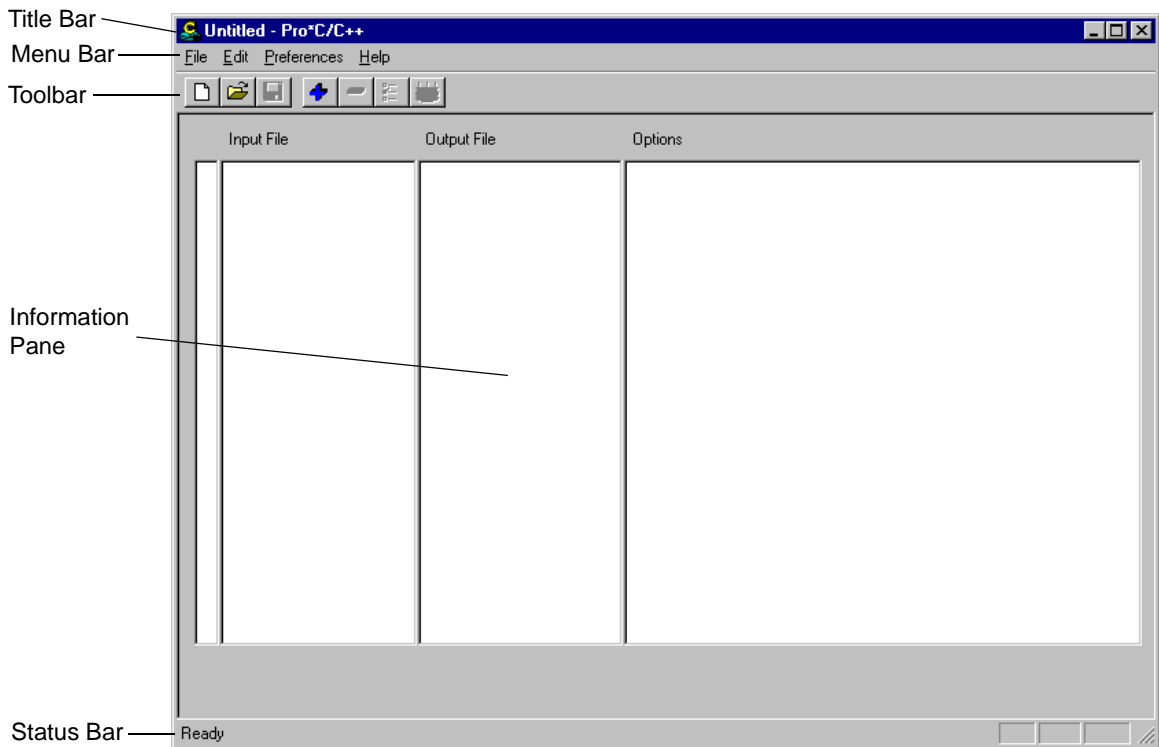
## Using the Graphical User Interface

Before you follow the instructions for creating and precompiling a Pro\*C/C++ project, you should familiarize yourself with the basic commands, dialog boxes, menus, and buttons of the Pro\*C/C++ graphical user interface.

### Starting Pro\*C/C++ Graphical Interface

To start the graphical user interface, choose Start > Programs > Oracle - *HOME\_NAME* > Application Development > Pro C\_C++.

The Pro\*C/C++ precompile environment contains five elements noted in the following illustration:



### Title Bar

The title bar displays the name of the Pro\*C/C++ project. If you have not assigned a name to the current project, the word "Untitled" appears instead.

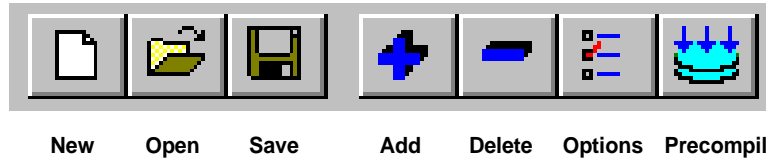
## Menu Bar

The menu bar contains the following menus:

Menu	Description
File	Contains commands to create a new Pro*C/C++ project, open an existing Pro*C/C++ project, save the active Pro*C/C++ project under the same name or under a different name, specify a connect string to an Oracle database, precompile a Pro*C/C++ project, and exit the application.
Edit	Contains commands to add files to a Pro*C/C++ project, delete files from a Pro*C/C++ project, and display or change precompiler options.
Preferences	Contains commands to set the default file extension of output files.
Help	Contains the About Pro*C/C++ command, which displays the version number of the application and copyright information.

## Toolbar

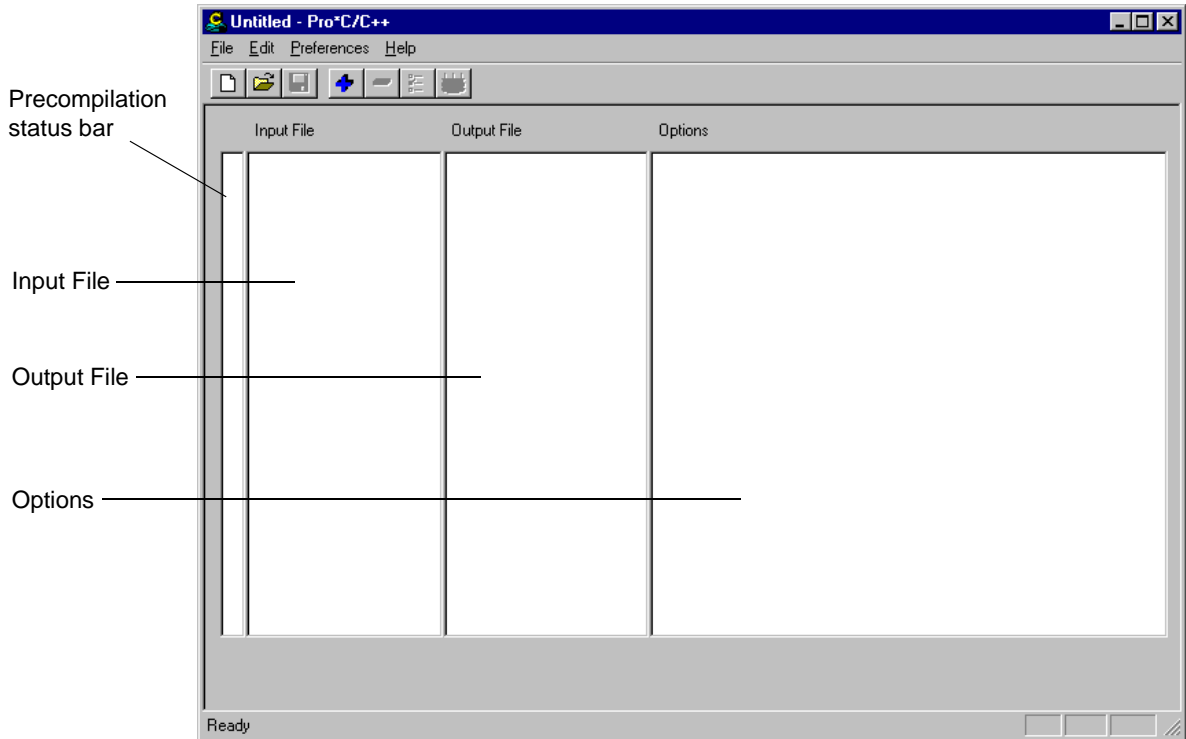
The toolbar enables you to execute commands by clicking a button:



Button	Description
New	Create a new Pro*C/C++ project
Open	Open an existing Pro*C/C++ project
Save	Save the active Pro*C/C++ project under the same name
Add	Add files to a Pro*C/C++ project
Delete	Delete files from a Pro*C/C++ project
Options	Display or change precompiler options
Precompile	Precompile a Pro*C/C++ project

## Information Pane

The information pane consists of four elements noted in the following illustration:



Element	Description
Precompilation Status Bar	Indicates whether the precompilation for a file was successful or unsuccessful.
Input File	Shows the files of a Pro*C/C++ project to be precompiled.
Output File	Shows the output files of a Pro*C/C++ project after precompilation.
Options	Displays precompile options that are different from the default options.

Look for one of the three status icons in the precompilation status bar after the precompile process is complete.

- A green check indicates that the file precompiled successfully.
- A yellow check indicates that the file precompiled successfully, but there are one or more warnings.
- A red X indicates that the file did not precompile successfully.

Double clicking a status icon opens the *Precompilation Status* dialog box. This dialog box provides detailed information on the reason for a warning or failure.

## Status Bar

The status bar at the bottom of the window displays information about the progress of a precompilation. The status bar also identifies the purpose of a toolbar button or menu command when you place the mouse pointer over the toolbar button or menu command.

## Creating and Precompiling a Pro\*C/C++ Project

This section describes the steps involved in creating and precompiling a Pro\*C/C++ project. After starting the Pro\*C/C++ application, perform the following steps:

- [Opening a Project](#)
- [Setting the Default Extension of Output Files](#)
- [Changing the Name of an Existing Input or Output File](#)
- [Adding Files to the Project](#)
- [Deleting Files from the Project](#)
- [Setting the Precompiler Options](#)
- [Specifying Database Connection Information](#)
- [Precompiling a Pro\\*C/C++ Project](#)
- [Checking the Results](#)
- [Fixing Errors](#)
- [Exiting Pro\\*C/C++](#)

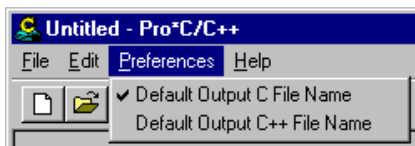
## Opening a Project

Pro\*C/C++ opens only one project at a time. A project consists of one or more precompilable files. Project files have an extension of `.pre`.

- To create a new project, choose File > New Project.
- To open an existing project, choose File > Open Project.

## Setting the Default Extension of Output Files

Use the Preferences menu to determine the default extension of the output files.



This setting only affects input files that you add later. An existing output file name will not change. However, you can change an existing output file name by double-clicking the output file and entering a new name.

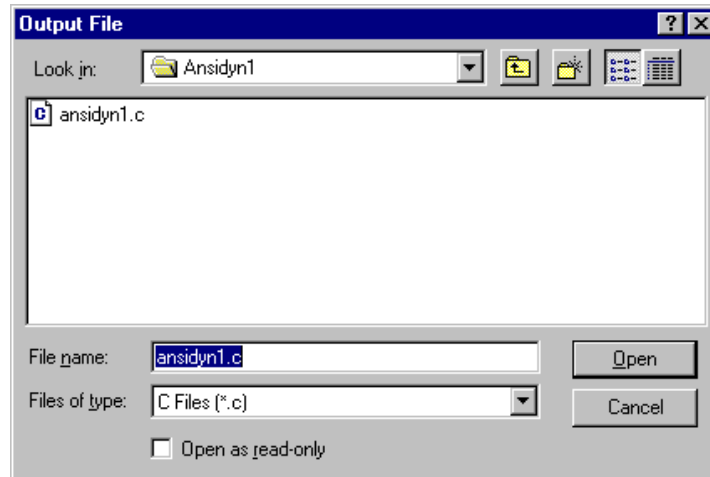
- If you select Default Output C File Name, the default extension of the output files is `.c`.
- If you select Default Output C++ File Name, the default extension of the output files is `.cpp`.
- If you deselect both Default Output C File Name and Default Output C++ File Name, the *Output File* dialog box appears when you add an output file.
  - Enter an output file name for the file selected. After you select or enter a file name, it appears in the Output File area of the information pane.

## Changing the Name of an Existing Input or Output File

**To change the name of an existing input or output file:**

1. Double-click the file name in the Input File or Output File area of the information pane. The *Input File* or *Output File* dialog box appears.



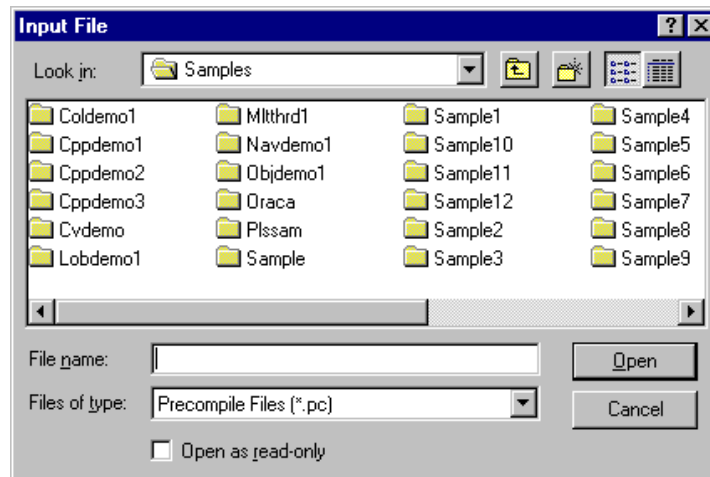


2. Replace the old file name with the new file name.
3. Click Open.

## Adding Files to the Project

**To add files to the project:**

1. Choose Edit > Add. The *Input File* dialog box appears.



2. Select one or more .pc files. Use the Ctrl key and the mouse to select files that are not adjacent.
3. Click Open. The selected files appear in the information pane.

## Deleting Files from the Project

If you need to, you can easily delete one or more files from the project.

### To delete files from the project:

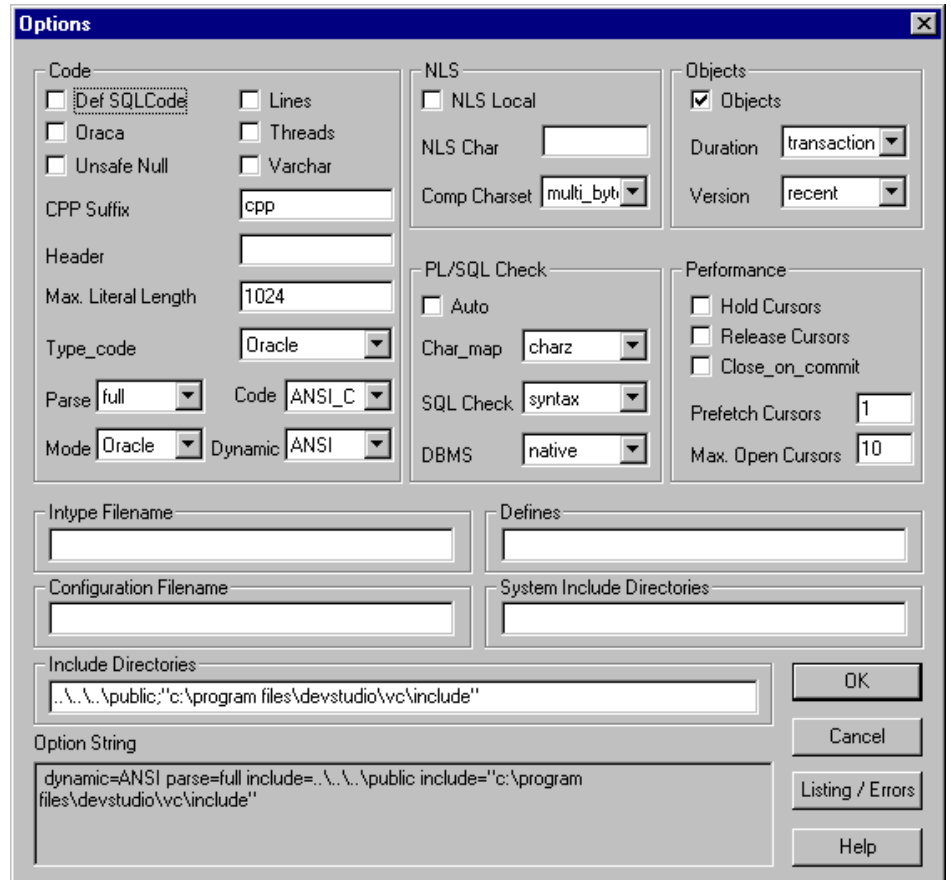
1. Highlight the file(s) in the information pane.
2. Choose Edit > Delete.
3. Click Yes.

## Setting the Precompiler Options

The Precompiler options enable you to control how resources are used, how errors are reported, how input and output are formatted, and how cursors are managed.

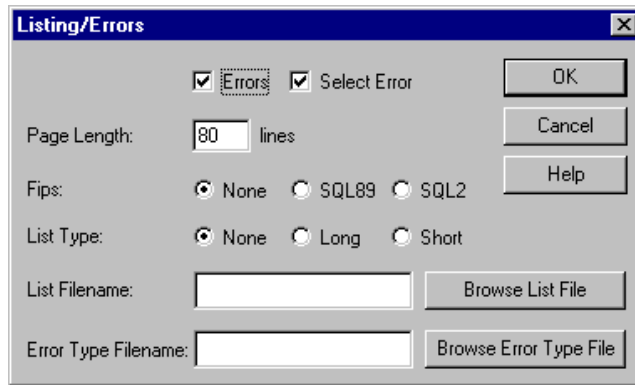
### To set the precompile options:

1. Select one or more files in the Input File list.
2. Choose Edit > Options. The *Options* dialog box appears.



Default options are in effect for all newly added files. When you change an option's default setting, a description of the change appears in the Option String edit field at the bottom of the *Options* dialog box and in the Options area of the information pane. For additional information on options, see ["Precompiler Options"](#) on page 2-16.

3. To change the format of the output list file that the precompiler writes to disk, click the Listing / Errors button. The *Listing/Errors* dialog box appears.



The settings include the type of error information generated and the name of the list file.

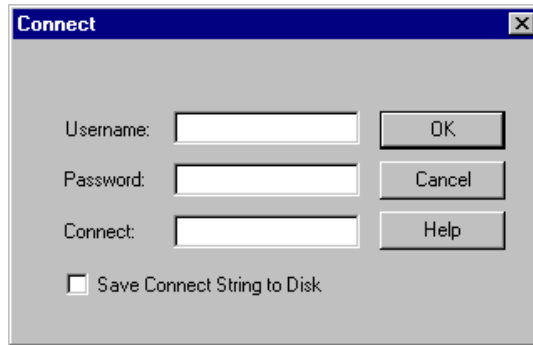
4. After you set the options in the *Options* dialog box, click OK.

## Specifying Database Connection Information

If you selected semantics or full for the SQL Check option in the *Options* dialog box, you may need to specify database connection information to the Oracle database. You do not need to connect to the Oracle database if every table referenced in a data manipulation statement or PL/SQL block is defined in a DECLARE TABLE statement.

**To specify database connection information:**

1. Choose File > Connect. The *Connect* dialog box appears.



2. Use this dialog box to specify database connection information prior to precompiling. No database connection is performed at this time. Only one set of database connection information can be specified for all files requiring semantic or full SQLCHECKing.
3. The *Connect* dialog box appears automatically at precompile time if you have not previously responded. Enter the user name, the password, and the network service name (database alias). The network service name is not required for a local database.
4. If you want to save the connection information between Pro\*C/C++ sessions, select the Save Connect String to Disk check box. If you do not select the check box, you must enter this information each time you precompile.
5. Click OK.

## Precompiling a Pro\*C/C++ Project

You can precompile any number of files in the Input File list.

**To precompile:**

1. Select one or more files in the Input File list. You can use the Ctrl button to highlight files that are not adjacent to each other (for example, the first and third files in a list).
2. Choose File > Precompile.

When precompiling is completed, the message in the dialog box indicates “Precompiling Finished!”, and the Cancel button changes to OK.

3. Click OK.

---

---

**Note:** Although Cancel does not interrupt the precompile for a file already in process, it does halt the precompile chain for remaining files.

---

---

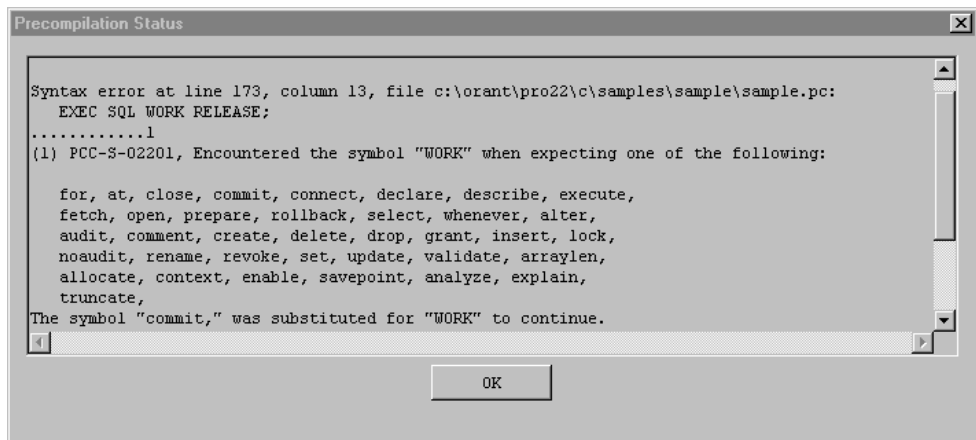
## Checking the Results

Precompiling can result in success, success with warning(s), or failure. When precompiling is finished, check the precompilation status bar.

- A green check indicates that the file compiled successfully.
- A yellow check indicates that the file compiled successfully, but there are one or more warnings.
- A red X indicates that the file did not compile successfully.

## Fixing Errors

If you see a yellow check or a red X, double-click the icon in status bar. The *Precompilation Status* dialog box appears. This dialog box lists warning messages or reasons why the precompilation failed. For example:



Switch to your development environment to fix the problem(s). After you correct the errors, precompile again.

---

---

**Note:** If you receive a PCC-S-02014 error (syntax error at line *num*, column *colnam*, file *name*), do the following:

- Copy the batch files `mod_incl.bat` and `add_newl.bat` from the `oracle_base\oracle_home\precomp\misc\proc` directory to the directory that contains the problematic INCLUDE file.
  - Run `mod_incl.bat`.
- 
- 

## Exiting Pro\*C/C++

To exit Pro\*C/C++, choose File > Exit. If your project changed in any way, you are prompted to save it.

---

---

**Suggestion:** If you want to keep an original file, as well as a version of the file with your changes, choose the Save As command. The Save command overwrites the previous version.

---

---

## Using Pro\*C/C++ at the Command Line

To precompile a file at the command line, enter the following command:

```
C:\> proc iname=filename.pc
```

where `filename.pc` is the name of the file. If the file is not in your current working directory, include the file's full path after the INAME argument.

Pro\*C/C++ generates `filename.c`, which can be compiled by your C compiler.

## Header Files

The `oracle_base\oracle_home\precomp\public` directory contains the Pro\*C/C++ header files.

**See Also:** See the *Pro\*C/C++ Precompiler Programmer's Guide* for more information about `oraca.h`, `sqlca.h`, and `sqllda.h`.

---

Header File	Description
<code>oraca.h</code>	Contains the Oracle Communications Area (ORACA), which helps you to diagnose runtime errors and to monitor your program's use of various Oracle8 resources.
<code>sql2oci.h</code>	Contains SQLLIB functions that enable the Oracle Call Interface (OCI) environment handle and OCI service context to be obtained in a Pro*C/C++ application.
<code>sqlapr.h</code>	Contains ANSI prototypes for externalized functions that can be used in conjunction with OCI.
<code>sqlca.h</code>	Contains the SQL Communications Area (SQLCA), which helps you to diagnose runtime errors. The SQLCA is updated after every executable SQL statement.
<code>sqlcpr.h</code>	Contains platform-specific ANSI prototypes for SQLLIB functions that are generated by Pro*C/C++. By default, Pro*C/C++ does not support full-function prototyping of SQL programming calls. If you need this feature, include <code>sqlcpr.h</code> before any EXEC SQL statements in your application source file.
<code>sqllda.h</code>	Contains the SQL Descriptor Area (SQLDA), which is a data structure required for programs that use dynamic SQL Method 4.
<code>sqlkpr.h</code>	Contains K&R prototypes for externalized functions that can be used in conjunction with OCI.
<code>sqlproto.h</code>	The <code>sqlproto.h</code> header file was obsoleted in Pro*C/C++ release 8.0.3. Use <code>sqlcpr.h</code> instead of <code>sqlproto.h</code> . However, applications that were built using <code>sqlproto.h</code> can be created without modification: a dummy <code>sqlproto.h</code> file that includes <code>sqlcpr.h</code> has been provided in the <code>oracle_base\oracle_home\precomp\public</code> directory.

---



## Library Files

The `oracle_base\oracle_home\precomp\lib\msvc` directory contains the library file that you use when linking Pro\*C/C++ applications. The library file is called `orasql8.lib`.

Pro\*C/C++ application program interface (API) calls are implemented in DLL files provided with your Pro\*C/C++ software. To use the DLLs, you must link your application with the import libraries (.lib files) that correspond to the Pro\*C/C++ DLLs. Also, you must ensure that the DLL files are installed on the computer that is running your Pro\*C/C++ application.

Microsoft provides you with three libraries: `libc.lib`, `libcmtd.lib`, and `msvcrt.lib`. The Oracle DLLs use the `msvcrt.lib` runtime library. You must link with `msvcrt.lib` rather than the other two Microsoft libraries.

## Building Multi-Threaded Applications

Build multi-threaded applications if you are planning to perform concurrent database operations.

Windows NT and Windows 95/98 schedule and allocate threads belonging to processes. A thread is a path of a program's execution. It consists of a kernel stack, the state of the CPU registers, a thread environment block, and a users stack. Each thread shares the resources of a process. Multi-threaded applications use the resources of a process to coordinate the activities of individual threads.

When building a multi-threaded application, make sure that your C/C++ code is reentrant. This means that access to static or global data must be restricted to one thread at a time. If you mix multi-threaded and non-reentrant functions, one thread can modify information that is required by another thread.

The Pro\*C/C++ precompiler automatically creates variables on the local stack of the thread. This ensures that each thread using the Pro\*C/C++ function has access to a unique set of variables and is reentrant.

**See Also:** See the *Pro\*C/C++ Precompiler Programmer's Guide* for additional information on how to write multi-threaded applications with Pro\*C/C++.

## Precompiler Options

This section highlights issues related to Pro\*C/C++ for Windows platforms.

**See Also:** See the "Precompiler Options" chapter of the *Pro\*C/C++ Precompiler Programmer's Guide* for more information on the precompiler options.

### Configuration File

A configuration file is a text file that contains precompiler options.

For this release, the system configuration file is called `pcscfg.cfg`. This file is located in the `oracle_base\oracle_home\precomp\admin` directory.

### CODE

The CODE option has a default setting of ANSI\_C. Pro\*C/C++ for other operating systems may have a default setting of KR\_C.

### DBMS

DBMS=V6\_CHAR is not supported when using CHAR\_MAP=VARCHAR2. Instead, use DBMS=V7.

### INCLUDE

For the Pro\*C/C++ graphical user interface, use the Include Directories field of the *Options* dialog box to enter INCLUDE path directories. If you want to enter more than one path, separate each path with a semicolon, but do not insert a space after the semicolon. This causes a separate "INCLUDE=" string to appear in front of each directory.

For sample programs that precompile with PARSE=PARTIAL or PARSE=FULL, an include path of `c:\program files\devstudio\vc\include` has been added. If Microsoft Visual C++ has been installed in a different location, modify the Include Directories field accordingly for the sample programs to precompile correctly.

### PARSE

The PARSE option has a default setting of NONE. Pro\*C/C++ for other operating systems may have a default setting of FULL.

## Using Pro\*C/C++ with the Oracle XA Library

The XA Application Program Interface (API) is typically used to enable an Oracle database to interact with a transaction processing (TP) monitor, such as:

- BEA Tuxedo
- IBM Transarc Encina
- IBM CICS

You can also use TP monitor statements in your client programs. The use of the XA API is also supported from both Pro\*C/C++ and OCI.

The Oracle XA Library is automatically installed as part of Oracle8i Enterprise Edition. The following components are created in your Oracle home directory:

Component	Location
<code>oraxa8.lib</code>	<code>oracle_base\oracle_home\rdbms\xa</code>
<code>xa.h</code>	<code>oracle_base\oracle_home\rdbms\xa</code>

## Compiling and Linking a Pro\*C/C++ Program with XA

**To compile and link a Pro\*C/C++ program with XA:**

1. Precompile `filename.pc` using Pro\*C/C++ to generate `filename.c`.
2. Compile `filename.c`, making sure to include `oracle_base\oracle_home\rdbms\xa` in your path.
3. Link `filename.obj` with the following libraries:

Library	Location
<code>oraxa8.lib</code>	<code>oracle_base\oracle_home\rdbms\xa</code>
<code>oci.lib</code>	<code>oracle_base\oracle_home\oci\lib\msvc</code>
<code>orasql8.lib</code>	<code>oracle_base\oracle_home\precomp\lib\msvc</code>

4. Run `filename.exe`.

## XA Dynamic Registration

Oracle supports the use of XA dynamic registration. XA dynamic registration improves the performance of applications that interface with XA-compliant TP monitors.

For TP monitors to use XA dynamic registration with an Oracle database on Windows NT, you must add either an environmental variable or a registry variable to the Windows NT computer on which your TP monitor is running. See either of the following sections for instructions:

- [Adding an Environmental Variable for the Current Session](#)
- [Adding a Registry Variable for All Sessions](#)

### Adding an Environmental Variable for the Current Session

Adding an environmental variable at the MS-DOS command prompt affects only the current MS-DOS session.

**To add an environmental variable for the current session:**

1. Go to the computer where your TP monitor is installed.
2. Enter the following at the MS-DOS command prompt:

```
C:\> set ORA_XA_REG_DLL = vendor.dll
```

where *vendor.dll* is the TP monitor DLL provided by your vendor.

### Adding a Registry Variable for All Sessions

Adding a registry variable affects all sessions on your Windows NT computer. This is useful for computers where only one TP monitor is running.

**To add a registry variable for all sessions:**

1. Go to the computer where your TP monitor is installed.
2. Enter the following at the MS-DOS command prompt:

```
C:\> regedit
```

The *Registry Editor* window appears.

3. Go to HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE.
4. Choose the Add Value option in the Edit menu.

The *Add Value* dialog box appears.

5. Enter `ORA_XA_REG_DLL` in the Value Name field.
6. Select `REG_EXPAND_SZ` from the Data Type drop-down list box.
7. Click OK.

The *String Editor* dialog box appears.

8. Enter `vendor.dll` in the String field, where `vendor.dll` is the TP monitor DLL provided by your vendor.
9. Click OK.

The Registry Editor adds the parameter.

10. Choose Exit from the Registry menu.

The registry exits.

## XA and TP Monitor Information

Refer to the following for more information about XA and TP monitors:

- *Distributed TP: The XA Specification* published by X/Open (now part of OpenGroup). See the web site at:  
[http://www.opengroup.org/public/pubs/catalog/full\\_1st.htm](http://www.opengroup.org/public/pubs/catalog/full_1st.htm)
- *Transaction Processing XPG4 X/Open CAE Specification XO/CAE/91/300 or C193 2/92*
- X/Open Company, Ltd., 1010 El Camino Real, Suite 380, Menlo Park, CA 94025, U.S.A.
- Your specific TP monitor documentation

**See Also:** For more information about the Oracle XA Library and using XA dynamic registration, see *Oracle8i Application Developer's Guide - Fundamentals*.



---

---

## Sample Programs

This chapter describes how to build Oracle database applications with Pro\*C/C++ using the sample programs that are included with this release. Specific topics discussed are:

- [Sample Program Descriptions](#)
- [Building the Demonstration Tables](#)
- [Building the Sample Programs](#)

## Sample Program Descriptions

When you install Pro\*C/C++, Oracle Installer copies a set of Pro\*C/C++ sample programs to the `oracle_base\oracle_home\precomp\demo\proc` directory. These sample programs are listed in [Table 3–1, "Sample Programs"](#). The SQL scripts are located in the `oracle_base\oracle_home\precomp\demo\sql` directory.

Oracle recommends that you build and run these sample programs to verify that Pro\*C/C++ has been installed successfully and operates correctly. You can delete the programs after you use them.

When built, the sample programs that Oracle provides in Pro\*C/C++ produce `.exe` executables. For some sample programs, you must run the SQL scripts in the sample directory before you precompile and run the sample program. The SQL scripts set up the correct tables and data so that the sample programs run correctly.

**Table 3–1 Sample Programs**

Sample Program	Source Files	Pro*C/C++ GUI Project File	MSVC Compiler Project File	Notes
ANSIDYN1	<code>ansidyn1.pc</code>	<code>ansidyn1.pre</code>	<code>ansidyn1.dsp</code>	
ANSIDYN2	<code>ansidyn2.pc</code>	<code>ansidyn2.pre</code>	<code>ansidyn2.dsp</code>	
COLDEMO1	<code>coldemo1.h</code> <code>coldemo1.pc</code> <code>coldemo1.sql</code> <code>coldemo1.typ</code>	<code>coldemo1.pre</code>	<code>coldemo1.dsp</code>	Run <code>coldemo1.sql</code> and the Object Type Translator before building <code>coldemo1</code> .
CPPDEMO1	<code>cppdemo1.pc</code>	<code>cppdemo1.pre</code>	<code>cppdemo1.dsp</code>	
CPPDEMO2	<code>cppdemo2.pc</code> <code>empclass.pc</code> <code>cppdemo2.sql</code> <code>empclass.h</code>	<code>cppdemo2.pre</code>	<code>cppdemo2.dsp</code>	Run <code>cppdemo2.sql</code> before building <code>cppdemo2</code> .
CPPDEMO3	<code>cppdemo3.pc</code>	<code>cppdemo3.pre</code>	<code>cppdemo3.dsp</code>	
CVDEMO	<code>cv_demo.pc</code> <code>cv_demo.sql</code>	<code>cv_demo.pre</code>	<code>cv_demo.dsp</code>	Run <code>cv_demo.sql</code> before building <code>cv_demo</code> .
EMPCLASS	<code>cppdemo2.pc</code> <code>empclass.pc</code> <code>cppdemo2.sql</code> <code>empclass.h</code>	<code>empclass.pre</code>	<code>empclass.dsp</code>	Run <code>cppdemo2.sql</code> before building <code>empclass</code> .



<b>Sample Program</b>	<b>Source Files</b>	<b>Pro*C/C++ GUI Project File</b>	<b>MSVC Compiler Project File</b>	<b>Notes</b>
LOBDEMO1	lobdemo1.h lobdemo1.pc lobdemo1.sql	lobdemo1.pre	lobdemo1.dsp	Run lobdemo1.sql before building lobdemo1.
MLTTHRD1	mltthrd1.pc mltthrd1.sql	mltthrd1.pre	mltthrd1.dsp	Run mltthrd1.sql before building mltthrd1.
NAVDEMO1	navdemo1.h navdemo1.pc navdemo1.sql navdemo1.typ	navdemo1.pre	navdemo1.dsp	Run navdemo1.sql and the Object Type Translator before building navdemo1.
OBJDEMO1	objdemo1.h objdemo1.pc objdemo1.sql objdemo1.typ	objdemo1.pre	objdemo1.dsp	Run objdemo1.sql and the Object Type Translator before building objdemo1.
ORACA	oraca.pc oracatst.sql	oraca.pre	oraca.dsp	Run oracatst.sql before building oraca.
PLSSAM	plssam.pc	plssam.pre	plssam.dsp	
SAMPLE	sample.pc	sample.pre	sample.dsp	
SAMPLE1	sample1.pc	sample1.pre	sample1.dsp	
SAMPLE2	sample2.pc	sample2.pre	sample2.dsp	
SAMPLE3	sample3.pc	sample3.pre	sample3.dsp	
SAMPLE4	sample4.pc	sample4.pre	sample4.dsp	
SAMPLE5	sample5.pc exampbld.sql exemplod.sql	sample5.pre	sample5.dsp	Run exampbld.sql, then run exemplod.sql, before building sample5.
SAMPLE6	sample6.pc	sample6.pre	sample6.dsp	
SAMPLE7	sample7.pc	sample7.pre	sample7.dsp	
SAMPLE8	sample8.pc	sample8.pre	sample8.dsp	
SAMPLE9	sample9.pc calldemo.sql	sample9.pre	sample9.dsp	Run calldemo.sql before building sample9.
SAMPLE10	sample10.pc	sample10.pre	sample10.dsp	
SAMPLE11	sample11.pc sample11.sql	sample11.pre	sample11.dsp	Run sample11.sql before building sample11.
SAMPLE12	sample12.pc	sample12.pre	sample12.dsp	

Sample Program	Source Files	Pro*C/C++ GUI Project File	MSVC Compiler Project File	Notes
SQLVCP	sqlvcp.pc	sqlvcp.pre	sqlvcp.dsp	
WINSAM	resource.h winsam.h winsam.ico winsam.pc winsam.rc	winsam.pre	winsam.dsp	

The following subsections describe the functionality of the sample programs.

**ANSIDYN1**

Demonstrates using ANSI dynamic SQL to process SQL statements that are not known until runtime. This program is intended to demonstrate the simplest (though not the most efficient) approach to using ANSI dynamic SQL.

**ANSIDYN2**

Demonstrates using ANSI dynamic SQL to process SQL statements that are not known until runtime. This program uses the Oracle extensions for batch processing and reference semantics.

**COLDEMO1**

Fetches census information for California counties. This program demonstrates various ways to navigate through collection-typed database columns.

**CPPDEMO1**

Prompts the user for an employee number, then queries the EMP table for the employee's name, salary, and commission. This program uses indicator variables (in an indicator struct) to determine whether the commission is NULL.

**CPPDEMO2**

Retrieves the names of all employees in a given department from the EMP table (dynamic SQL Method 3).

**CPPDEMO3**

Finds all salespeople and prints their names and total earnings (including commissions). This program is an example of C++ inheritance.

**CVDEMO**

Declares and opens a ref cursor.

**EMPCLASS**

The EMPCLASS and CPPDEMO2 files were written to provide an example of how to write Pro\*C/C++ programs within a C++ framework. EMPCLASS encapsulates a specific query on the emp table and is implemented using a cursor variable. EMPCLASS instantiates an instance of that query and provides cursor variable functionality (that is: open, fetch, close) through C++ member functions that belong to the emp class. The empclass.pc file is NOT a standalone demo program. It was written to be used by the cppdemo2 demo program. To use the emp class, you have to write a driver (cppdemo2.pc) which declares an instance of the emp class and issues calls to the member functions of that class.

**LOBDEMO1**

Fetches and adds crime records to the database based on the person's Social Security number. This program demonstrates the mechanisms for accessing and storing large objects (LOBs) to tables and manipulating LOBs through the stored procedures available through the DBMS\_LOB package.

**MLTTHR1**

Shows how to use threading in conjunction with precompilers. The program creates as many sessions as there are threads. See "[Building Multi-Threaded Applications](#)" on page 2-15.

**NAVDEMO1**

Demonstrates navigational access to objects in the object cache.

**OBJDEMO1**

Demonstrates the use of objects. This program manipulates the object types *person* and *address*.

**ORACA**

Demonstrates how to use ORACA to determine various performance parameters at runtime.

**PLSSAM**

Demonstrates the use of embedded PL/SQL blocks. This program prompts you for an employee name that already resides in a database. It then executes a PL/SQL block, which returns the results of four SELECT statements.

**SAMPLE**

Adds new employee records to the personnel database and checks database integrity. The employee numbers in the database are automatically selected using the current maximum employee number +10.

**SAMPLE1**

Logs on to an Oracle database, prompts the user for an employee number, queries the database for the employee's name, salary, and commission, and displays the result. The program continues until the user enters 0 as the employee number.

**SAMPLE2**

Logs on to an Oracle database, declares and opens a cursor, fetches the names, salaries, and commissions of all salespeople, displays the results, and closes the cursor.

**SAMPLE3**

Logs on to an Oracle database, declares and opens a cursor, fetches in batches using arrays, and prints the results using the *print\_rows()* function.

**SAMPLE4**

Demonstrates the use of type equivalencing using the LONG VARRAW external datatype.

**SAMPLE5**

Prompts the user for an account number and a debit amount. The program verifies that the account number is valid and that there are sufficient funds to cover the withdrawal before it debits the account. This program shows the use of embedded SQL.

**SAMPLE6**

Creates a table, inserts a row, commits the insert, and drops the table (dynamic SQL Method 1).

**SAMPLE7**

Inserts two rows into the EMP table and deletes them (dynamic SQL Method 2).

**SAMPLE8**

Retrieves the names of all employees in a given department from the EMP table (dynamic SQL Method 3).

**SAMPLE9**

Connects to an Oracle database using the SCOTT/TIGER account. The program declares several host arrays and calls a PL/SQL stored procedure (GET\_EMPLOYEES in the CALLEDemo package). The PL/SQL procedure returns up to ASIZE values. The program keeps calling GET\_EMPLOYEES, getting ASIZE arrays each time, and printing the values, until all rows have been retrieved.

**SAMPLE10**

Connects to an Oracle database using your user name and password and prompts for a SQL statement. You can enter any legal SQL statement, but you must use regular SQL syntax, not embedded SQL. Your statement is processed. If the statement is a query, the rows fetched are displayed (dynamic SQL Method 4).

**SAMPLE11**

Fetches from the EMP table, using a cursor variable. The cursor is opened in the stored PL/SQL procedure *open\_cur*, in the EMP\_DEMO\_PKG package.

**SAMPLE12**

Demonstrates how to do array fetches using dynamic SQL Method 4.

**SQLVCP**

Demonstrates how you can use the *sqlvcp()* function to determine the actual size of a VARCHAR struct. The size is then used as an offset to increment a pointer that steps through an array of VARCHARs.

This program also demonstrates how to use the *SQLStmtGetText()* function to retrieve the text of the last SQL statement that was executed.

**WINSAM**

Adds new employee records to the personnel database and checks database integrity. You can enter as many employee names as you want and perform the SQL commands by selecting the appropriate buttons in the *Employee Record* dialog box. This is a GUI version of the sample program.

## Building the Demonstration Tables

To run the sample programs, you must have a database account with the user name SCOTT and the password TIGER. Also, you must have a database with the sample tables EMP and DEPT. This account is included in the starter database for your Oracle8i server. If the account does not exist on your database, create the account before running the sample programs.

See *Oracle8i Administrator's Guide for Windows NT* for more information. If your database does not contain EMP and DEPT tables, you can use the `demobld.sql` script to create them.

### To build the sample tables:

1. Start SQL\*Plus
2. Connect as user name SCOTT with the password TIGER.
3. Run the `demobld.sql` script:

```
SQL> @ORACLE_BASE\ORACLE_HOME\SQLPLUS\DEMO\DEMOBLD.SQL;
```

## Building the Sample Programs

Microsoft Visual C++ 6.0 project files have an extension of `.dsp`. The `.dsp` files in the `oracle_base\oracle_home\precomp\demo\proc` directory guide and control the steps necessary to precompile, compile, and link the sample programs.

Pro\*C/C++, SQL\*Plus, and the Object Type Translator have been integrated into the Microsoft Visual C++ sample project files. You do not have to run Pro\*C/C++, SQL\*Plus, and the Object Type Translator separately before compilation. See [Appendix A, "Integrating Pro\\*C/C++ into Microsoft Visual C++"](#) for more information.

**See Also:** For more information on OTT, see the *Pro\*C/C++ Precompiler Programmer's Guide*.

### To build a sample program:

1. Open a Visual C++ project file, such as `sample1.dsp`.
2. Check the paths in the project file to ensure that they correspond to the configuration of your system. If they do not, change the paths accordingly. Your system may produce error messages if the paths to all components are not correct. See ["Setting the Precompiler Options"](#) on page 2-8 and ["Setting the Path for the Sample .pre Files"](#) on page 3-10.

---

---

**Note:** All of the sample programs were created with `c:\oracle\ora81` as the default drive.

---

---

3. Choose **Build > Rebuild All**. Visual C++ creates the executable.

## Setting the Path for the Sample .pre Files

By default the sample .pre files search for their corresponding .pc files in the `c:\oracle\ora81` directory where `c:\` is the drive that you are using, and `oracle\ora81` represents the location of the Oracle home. If the Oracle base and Oracle home directories are different on your computer, you must change the directory path to the correct path.

To change the directory path for a sample .pre file:

1. In Pro\*C/C++, open the .pre file.
2. Double-click the file name in the Input File area to display the Input File dialog box.
3. Change the directory path to the correct path.
4. Click Open.



# A

---

---

## Integrating Pro\*C/C++ into Microsoft Visual C++

This appendix describes how to integrate Pro\*C/C++ into the Microsoft Visual C++ integrated development environment. Specific topics discussed are:

- [Integrating Pro\\*C/C++ within Microsoft Visual C++ Projects](#)
- [Adding Pro\\*C/C++ to the Tools Menu](#)

## Integrating Pro\*C/C++ within Microsoft Visual C++ Projects

This section describes how to fully integrate Pro\*C/C++ within Microsoft Visual C++ projects.

All the precompiler errors and warnings are displayed in the output box where Microsoft Visual C++ displays compiler and linker messages. You do not have to precompile a file separately from the Microsoft Visual C++ build environment. More importantly, Microsoft Visual C++ maintains the dependencies between .c and .pc files. Microsoft Visual C++ maintains the dependency and precompile files, if needed.

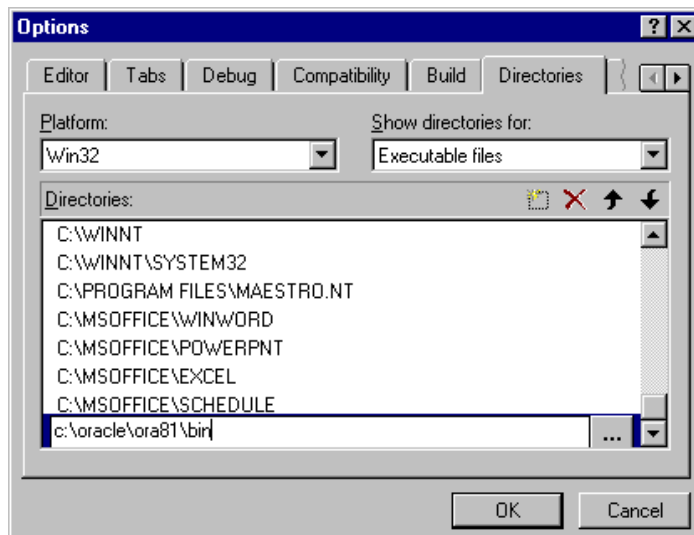
All of the procedures in this section are performed within Microsoft Visual C++.

### Specifying the Location of the Pro\*C/C++ Executable

For Microsoft Visual C++ to run Pro\*C/C++, it needs to know the location of the Pro\*C/C++ executable. If Microsoft Visual C++ was installed before any Oracle 8.1 products were installed, then you must add the directory path.

**To specify the location of the Pro\*C/C++ executable:**

1. Choose Options from the Tools menu. The *Options* dialog box appears.



2. Click the Directories tab.

3. Select Executable files from the Show directories for list box.
4. Scroll to the bottom of the Directories box and click the dotted rectangle.
5. Enter the `oracle_base\oracle_home\bin` directory. For example:  
`c:\oracle\ora81\bin`
6. Click OK.

## Specifying the Location of the Pro\*C/C++ Header Files

### To specify the location of the Pro\*C/C++ header files:

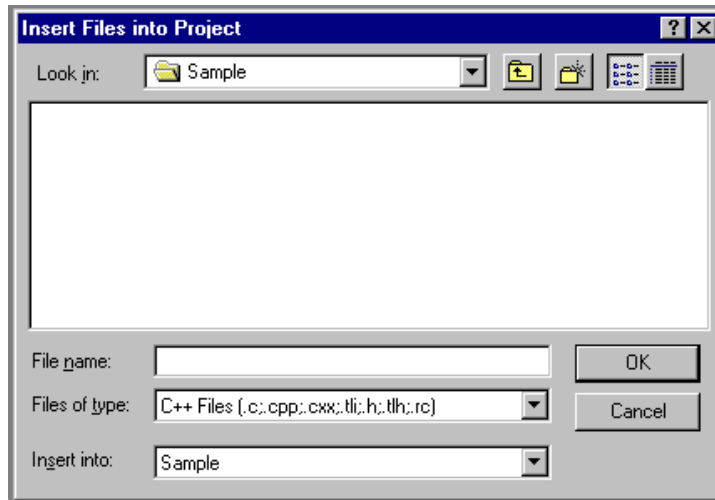
1. Choose Options from the Tools menu. The *Options* dialog box appears.
2. Click the Directories tab.
3. Select Include Files from the Show directories for list box.
4. Scroll to the bottom of the Directories box and click the dotted rectangle.
5. Enter the `oracle_base\oracle_home\precomp\public` directory. For example:  
`c:\oracle\ora81\precomp\public`
6. Click OK.

## Adding .pc Files to a Project

After you create a project, you need to add the .pc file(s).

### To add a .pc file to a project:

1. Choose Add To Project from the Project menu, and then choose Files. The *Insert Files into Project* dialog box appears.



2. Select All Files from the Files of type list box.
3. Select the .pc file.
4. Click OK.

## Adding References to .c Files to a Project

For each .PC file, you need to add a reference to the .C file that will result from precompiling.

### To add a reference to a .c file to a project:

1. Choose Add To Project from the Project menu, and then choose Files. The *Insert Files into Project* dialog box appears.
2. Type the name of the .c file in the File Name box.

3. Click OK. Because the `.c` file has not been created yet, Microsoft Visual C++ displays the following message: “The specified file does not exist. Do you want to add a reference to the project anyway?”
4. Click Yes.

## Adding the Pro\*C/C++ Library to a Project

Pro\*C/C++ applications must link with the library file `orasql8.lib`.

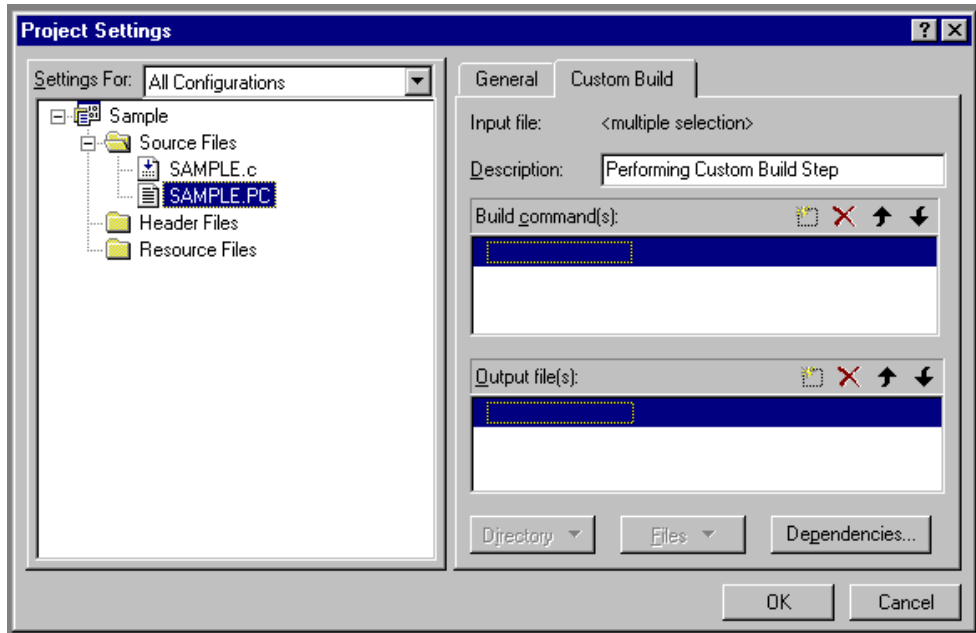
### To add the Pro\*C/C++ library to a project:

1. Choose Add To Project from the Project menu, and then choose Files.  
The *Insert Files into Project* dialog box appears.
2. Select All Files from the Files of type list box.
3. Select `orasql8.lib` from the `oracle_base\oracle_home\precomp\lib\msvc` directory.
4. Click OK.

## Specifying Custom Build Options

To specify custom build options:

1. In FileView, right-click a .pc file and choose Settings. The *Project Settings* dialog box appears with the Custom Build tab displayed.



2. In the Build command(s) box, enter the following on one line:

```
$(ProjDir)\..\..\..\..\bin\proc $(ProjDir)\$(InputName).pc
include=$(ProjDir)\..\..\..\public include="$(MSDEVDIR)\..\vc\include"
```

\$(ProjDir) and \$MSDEVDIR are macros for custom build commands in Microsoft Visual C++. See the Microsoft Visual C++ documentation for more information.

3. In the Output file(s) box, enter one of the following:

If You Are Generating...	Enter...
.C files	\$(ProjDir)\\$(InputName).c

If You Are Generating...	Enter...
.CPP files	<code>\$(ProjDir)\\$(InputName).cpp</code>

When the project is built, Microsoft Visual C++ will check the date of the output files to determine whether they need to be rebuilt for any new modifications made to the source code. See the Microsoft Visual C++ documentation for more information.

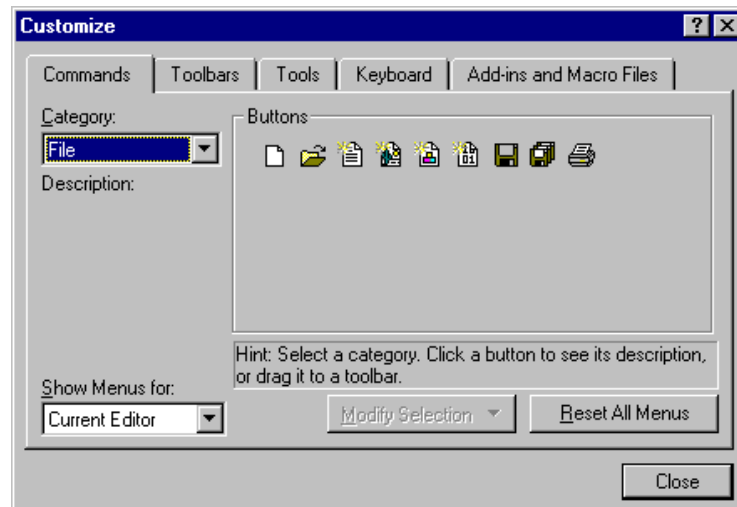
4. Click OK.

## Adding Pro\*C/C++ to the Tools Menu

You can include Pro\*C/C++ as a choice in the Tools menu of Microsoft Visual C++.

### To add Pro\*C/C++ to the Tools menu:

1. From within Microsoft Visual C++, choose Customize from the Tools menu. The *Customize* dialog box appears.



2. Click the Tools tab.
3. Scroll to the bottom of the Menu contents box and click the dotted rectangle.
4. Enter the following text:

Pro\*C/C++

- In the Command box, type the path and file name of the graphical Pro\*C/C++ executable, or use the Browse button to the right of the box to select the file name. For example:

```
c:\oracle\ora81\bin\procui.exe
```

- In the Arguments box, enter the following text.

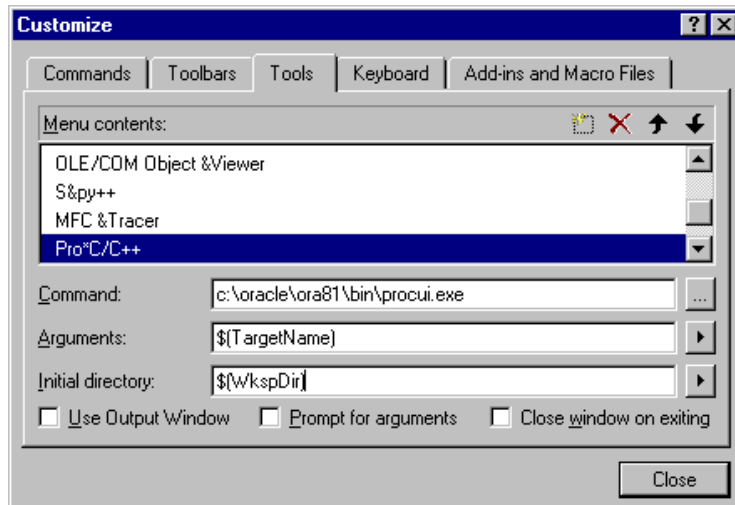
```
$(TargetName)
```

When you choose Pro\*C/C++ 8.1 from the Tools menu, Microsoft Visual C++ uses the \$(TargetName) argument to pass the name of the current development project to Pro\*C/C++. Pro\*C/C++ then opens a precompile project with the same name as the opened project, but with a .PRE extension in the project directory.

- In the Initial directory box, enter the following text:

```
$(WkspDir)
```

The *Customize* dialog box should now look like the following graphic (although the Oracle home directory may be different on your computer).



- Click Close. Microsoft Visual C++ adds Pro\*C/C++ to the Tools menu.



---

---

# Index

## Numerics

---

16-bit code, not supported, 1-2

## A

---

add\_newl.bat, 2-13

adding files, 2-7

ANSI compliance, 1-2

ANSI dynamic SQL, 3-4

## C

---

CODE option, 2-16

command line, precompiling from, 2-13

configuration files, 2-16

location, 2-16

Connect dialog box, 2-11

connect string, 2-10

## D

---

database connect string, 2-10

DBMS option, 2-16

Default Output

C File Name command, 2-6

C++ File Name command, 2-6

deleting files, 2-8

directory structure, 1-3

documentation

generic, xxi

.dsp files, 3-9

Dynamic Link Libraries (DLLs), 2-15

dynamic SQL

method 1, 3-6

method 2, 3-6

method 3, 3-4, 3-6

method 4, 3-7

## E

---

Edit menu, 2-3

embedded SQL, 3-5

## F

---

File menu, 2-3

## G

---

generic documentation references

default values for options, 2-16

demo directory, 1-3

header files, location of, 2-14

linking, 2-15

Oracle XA, 2-17

graphical user interface, 2-2 to 2-5

## H

---

header files

location of, 2-14

oraca.h, 2-14

sql2oci.h, 2-14

sqlapr.h, 2-14

sqlca.h, 2-14

sqlcpr.h, 2-14

sqlda.h, 2-14

- sqlkpr.h, 2-14
- sqlproto.h, 2-14
- Help menu, 2-3

---

## I

- INCLUDE option, 2-16
- Input File dialog box, 2-7

---

## L

- large objects, 3-5
- linking, 2-15
- Listing/Errors dialog box, 2-10
- LOBs, 3-5

---

## M

- menu bar, 2-3
- Microsoft Visual C++
  - integrating Pro\*C/C++ into, A-1 to A-8
- mod\_incl.bat, 2-13
- msvcrt.lib runtime library, 2-15
- multi-threaded applications, 2-15, 3-5

---

## N

- Net8, 1-2
- New toolbar button, 2-6

---

## O

- Object Type Translator (OTT), 3-9
- objects
  - demonstration program, 3-5
- Open toolbar button, 2-6
- options dialog box, 2-8
- oraca.h header file, 2-14
- Oracle base
  - described, xx
- Oracle home
  - described, xx
- Oracle XA, 2-17
- Oracle XA Library
  - additional documentation, 2-19
- orasql8.lib library file, 2-15

- OTT (Object Type Translator), 3-9
- output file names, 2-6

---

## P

- PARSE option, 2-16
- paths
  - checking, 3-9
  - checking the .pre files, 3-10
- PCC-S-02014 error, 2-13
- pccsf.cfg configuration file, 2-16
- .pre files, 2-6
  - checking the paths, 3-10
- precompiling, steps involved, 2-5 to 2-13
- Preferences menu, 2-3, 2-6
- Pro\*C/C++
  - command line interface, 2-13
  - configuration files, 2-16
  - features, 1-2
  - graphical user interface, 2-2 to 2-5
  - integrating into Microsoft Visual C++, A-1 to A-8
  - introduction, 1-1
  - library file, A-5
  - linking, 2-15
  - overview, 1-2
  - starting, 2-2
- project files, 2-6, 3-9

---

## R

- reentrant functions, 2-15

---

## S

- sample programs
  - ANSIDYN1, 3-2, 3-4
  - ANSIDYN2, 3-2, 3-4
  - building, 3-9
  - COLDEMO1, 3-2, 3-4
  - CPPDEMO1, 3-2, 3-4
  - CPPDEMO2, 3-2, 3-4
  - CPPDEMO3, 3-2, 3-4
  - CV\_DEMO, 3-2, 3-4
  - default drive, 3-9

- described, 3-4 to 3-7
- EMPCLASS, 3-2, 3-5
- INCLUDE path, 2-16
- LOBDEMO1, 3-3, 3-5
- location of, 1-3, 3-2
- MLTTHRD1, 3-3, 3-5
- NAVDEMO1, 3-3, 3-5
- OBJDEMO1, 3-3, 3-5
- ORACA, 3-3, 3-5
- PLSSAM, 3-3, 3-5
- SAMPLE, 3-3, 3-5
- SAMPLE1, 3-3, 3-6
- SAMPLE10, 3-3, 3-7
- SAMPLE11, 3-3, 3-7
- SAMPLE12, 3-3, 3-7
- SAMPLE2, 3-3, 3-6
- SAMPLE3, 3-3, 3-6
- SAMPLE4, 3-3, 3-6
- SAMPLE5, 3-3, 3-6
- SAMPLE6, 3-3, 3-6
- SAMPLE7, 3-3, 3-6
- SAMPLE8, 3-3, 3-6
- SAMPLE9, 3-3, 3-6
- setting the path, 3-9
- setting the path for the .pre files, 3-10
- SQLVCP, 3-4, 3-7
- WINSAM, 3-4, 3-7
- sample tables
  - building, 3-8
- Save As command, 2-13
- SQL (Structured Query Language), 1-2
- sql2oci.h header file, 2-14
- sqlapr.h header file, 2-14
- sqlca.h header file, 2-14
- sqlcpr.h header file, 2-14
- sqlda.h header file, 2-14
- sqlkpr.h header file, 2-14
- sqllib80.lib, A-5
- sqlproto.h header file, 2-14
- SQLStmtGetText() function, 3-7
- sqlvcp() function, 3-7
- starting Pro\*C/C++, 2-2
- status bar, 2-5
- Structured Query Language (SQL), 1-2

## T

---

- thread, defined, 2-15
- title bar, 2-2
- toolbar buttons
  - New, 2-6
  - Open, 2-6
- transaction processing monitor
  - additional documentation, 2-19

