

# Symantec C++◆

## *Welcome to Symantec C++*

### *Part One*

- 1 Introducing Symantec C++
- 2 Introducing the IDDE



# Introducing Symantec C++ 1



**W**elcome to the Symantec C++ for Windows 95, Windows NT 3.5, Windows 3.1, and DOS. This chapter contains information on how to use this book and what you need to know about Symantec C++. For a list of the Symantec C++ version 7 features, see the *Getting Started Guide*.

## Prerequisites for Using Symantec C++

This book assumes that you know, or are learning how to program in, C or C++. You should also be familiar with the Windows and DOS operating systems. This book does not show you how to use either operating system. It assumes that you know how to choose menu commands, select options, and work with other user interface elements in Windows. For more information about Windows or DOS, see the appropriate user's manual.

You should also be familiar with fundamentals of the development cycle including editing, linking, compiling, and debugging.

## Fulfilling the prerequisites

If you are just beginning with C or C++, Symantec C++ is a great platform to learn on. However, a tutorial on C or C++ is beyond the scope of this manual. The following list of books can serve as resources on the C language:

- *Standard C* by P. J. Plauger and Jim Brodie (Microsoft Press) is a guide to writing C programs that conform to the ANSI C standard. Both authors were officers of the committee that drafted the ANSI standard.
- *Software Engineering in C* by Peter Darnell and Philip Margolis (Springer-Verlag) is an introduction to the C programming language. This book is ideal for new C programmers who have programmed in other languages.

## ◆ 1 Introducing Symantec C++

---

- *C Traps and Pitfalls* by Andrew Koenig (Addison-Wesley), written for intermediate and advanced C programmers, contains a detailed discussion of common C programming problems.
- *Numerical Recipes in C* by William Press, Brian Flannery, Saul Teukolsky, and William Vetterling (Cambridge University Press) is a detailed technical description of numerical methods with implementation examples in C.
- *Portability and the C Language* by Rex Jaeschke (Hayden Books) covers writing portable C programs and is suitable for advanced C programmers. It provides guidelines on writing programs that can be compiled with different compilers and run on multiple platforms. It points out changes introduced with ANSI C. It is useful if you port code from one platform to another or from an old (pre-ANSI) version of Zortech to Symantec C++.
- *Portable C Software* by Mark R. Horton (Prentice Hall) is also about writing portable C programs, but emphasizes porting among C compilers for UNIX and MS-DOS.
- *The C Programming Language*, 2d by Brian Kernighan and Dennis Ritchie (Prentice Hall) is an update to *The C Programming Language* that incorporates what was then the latest draft of the ANSI standard.
- *C: A Reference Manual*, 2d by Samuel Harbison and Guy Steele, Jr. (Prentice Hall) contains a detailed and authoritative discussion of the syntax, meaning, and idiomatic usage of ANSI Standard C.
- The document describing the ANSI standard is *American National Standard for Information Systems—Programming Language—C X3.159–1989*. Cost is \$50, plus \$5 for shipping and handling. To order, contact:

American National Standards Institute (ANSI)  
Sales Dept.  
1430 Broadway  
New York, NY 10018  
(212) 642-4900



The following books can help both beginning and experienced C++ programmers:

- *The Annotated C++ Reference Manual* by Margaret Ellis and Bjarne Stroustrup (Addison-Wesley, 1990).
- *The C++ Programming Language*, 2d by Bjarne Stroustrup (Addison-Wesley, 1992) includes a 10-chapter tutorial introduction to C++.
- *The C++ Primer*, 2d by Stanley Lipman (Addison-Wesley, 1992) is a solid and easy-to-read introduction to C++. It does not assume knowledge of C, but does assume knowledge of some modern block structured language.
- *Object-Oriented Programming in C++* by Ira Pohl (Benjamin/Cummings Publishing, 1993) is helpful for learning both C++ and object-oriented programming techniques.
- *The C++ Answer Book* by Tony L. Hansen (Addison-Wesley, 1990) contains useful examples, questions, and answers. Although it was written as a companion book to the first edition of *The C++ Programming Language*, it is still current and informative.
- *C++ for C Programmers* by Ira Pohl (Benjamin/Cummings Publishing, 1989) is appropriate for experienced C programmers learning C++. It introduces the features that programmers can put into immediate practice in C++.
- *The IOStreams Handbook* by Steve Teale (Addison-Wesley, 1993) is a comprehensive, detailed explanation of the standard input and output library used in C++.
- The document describing the ANSI draft proposed standard for C++ is *Working Paper for Draft Proposed International Standard for Information Systems—Programming Language—C++ X3J16*. To order the latest revision, contact:

American National Standards Institute (ANSI)  
Standards Secretariat: CBEMA  
1250 Eye Street NW

## 1 Introducing Symantec C++

---

Suite 200  
Washington, DC 20005

- *The Draft Standard C++ Library*, by P.J. Plauger (Prentice-Hall) describes the draft proposed standard C++ runtime library.

Symantec C++ can provide help in setting up the framework of your Windows applications. However, if you are writing such an application, you still need to know Windows programming basics. Some books that deal with beginning and advanced Windows programming concepts are listed below:

- *Advanced Windows* by Jeffrey Richter (Microsoft Press) covers 32-bit Windows programming concepts.
- *Programming Windows 3.1* by Charles Petzold (Microsoft Press, 1992) is the definitive book on Windows programming.
- *Advanced Windows Programming* by Martin Heller (John Wiley & Sons, Inc., 1992) builds on Petzold's *Programming Windows 3.1*.
- *Undocumented Windows* by Andrew Schulman, David Maxey, and Matt Pietrek (Addison-Wesley, 1992) is a guide and reference to the Windows API functions left undocumented or "reserved" by Microsoft.
- *Windows Programmer's Guide to DLLs and Memory Management* by Mike Klein (SAMS, A division of Prentice Hall Computer Publishing, 1992) gives in-depth coverage of designing and building DLLs and how memory is managed in the Windows environment.
- *Windows++, Writing Reusable Windows Code in C++* by Paul DiLascia (Addison-Wesley, 1992) discusses an application framework for Windows using C++ that the author developed for his own applications. This book also provides insight into using C++ to improve Windows programming productivity.
- *Writing Windows Device Drivers* by Daniel A. Norton (Addison-Wesley) is a guide and reference to writing 16-bit Windows device drivers.



Other useful titles include:

- *8087/80287/80387 for the IBM PC and Compatibles*, Third Edition, Richard Startz. (Brady Books).
- *IEEE Standard for Binary Floating-Point Arithmetic*, ANSI/IEEE Standard 754-1985. (The Institute of Electrical and Electronic Engineers, Inc.).
- *CV4 Symbolic Debug Information Specification*. (Languages Business Unit, Microsoft Corp.).
- *Microsoft Object Mapping Specification*. (Languages Business Unit, Microsoft Corp.).
- *The Microsoft Object Module Format (OMF)*. (Languages Business Unit, Microsoft Corp.).
- *ZEN of Graphics Programming*, Michael Abrash. (Coriolis Group Books).
- *ZEN of Code Optimization*, Michael Abrash. (Coriolis Group Books).
- *Pentium Processor User's Manual*, Vol. 1: Architecture and Programming Manual. (Intel Corp. Literature Sales).

## Conventions

This book uses the following typographic conventions:

- Names of menus, commands, and dialog boxes are in **boldface**.
- Filenames, code fragments, function names, variables, and information you type appear in typewriter face.
- All numbers are decimal numbers. Hexadecimal numbers are written in C notation, for example: 0x3EFA, instead of Pascal notation (\$3EFA).
- Key combinations are shown as follows: Shift+F2, Alt+F, Ctrl+F3.
- Metanames appear in *italic*.

## Finding the Information You Need

This manual is divided into the following six parts:

Part	Description
One: "Welcome to Symantec C++"	In addition to the current chapter, this part contains a chapter introducing the Integrated Development and Debugging Environment (IDDE).
Two: "Creating an Application with Symantec C++"	These chapters guide you through creating your first application in Symantec C++.
Three: "Learning Symantec C++ by Example"	These tutorial chapters take you through building DOS and Windows hypertext file reader applications.
Four: "More about Creating Programs"	These chapters provide more detail about IDDE settings, workspaces, the application framework designers, the class browsers, the text editor, and version control.
Five: "More about Testing Programs"	These chapters explain in detail the debugging features of Symantec C++.
Six: "About Managing Resources"	The chapters in this part describe in detail how to use the Symantec ResourceStudio to create and edit resources.
Seven: "Appendixes"	This part contains appendixes on expression evaluation, the relationship between IDDE settings and command-line options, and the NetBuild feature.

## Suggestions for the new users of Symantec C++

If you are new to Symantec C++, congratulations and welcome. We hope you find our product powerful and easy to use. You should read Chapter 2, "Introducing the IDDE," and work through the tutorials in Chapters 9-14 to become proficient with Symantec C++ quickly.





### **Suggestions for users new to Windows development**

If you are starting to program for Windows, Symantec C++ is a great platform. You should read Chapter 4, “Generating an Application Framework” and Chapter 7, “Adding Look and Feel with Resources.” Also you should read the tutorials (Chapters 9–14), which guide you through the development of a Windows application.

### **Suggestions for users porting to Symantec C++**

If you need to port your code from another compiler or from a previous version of Symantec C++ (Zortech C++), read the “Switching to Symantec C++” chapter in the *Compiler and Tools Guide*.

### **Suggestions for users upgrading to Symantec C++ version 7**

If you are upgrading from a previous version of Symantec C++ and want to learn about the new product features, read Chapter 4, “Generating an Application Framework,” Chapter 5, “Defining Classes and Their Hierarchies,” Chapter 7, “Adding Look and Feel with Resources,” Chapter 17, “More about AppExpress,” and Chapter 18, “More about ClassExpress.”

# ◆ 1 *Introducing Symantec C++*

---

# *Introducing the IDDE*

---

2

***T***his chapter introduces the Symantec C++ development system. The first part of the chapter describes the IDDE main window and toolboxes. The second part introduces Part Two of this manual, and outlines the steps involved in developing an application in the Symantec C++ IDDE.

## **Running Symantec C++ under Windows 3.1, Windows 95, and Windows NT**

Symantec C++ ships with three different integrated development and debugging environments (IDDEs), each tailored to a specific operating system and target. The Symantec C++ installer program asks you to define the kind of programs you plan to develop, and then installs the appropriate IDDEs on your system. All IDDEs share the same user interface, and can build applications for DOS, Windows 3.1, Windows 95, Windows NT 3.5, and DOSX. Minor differences between the three IDDEs are noted throughout this manual, as appropriate. Not every IDDE can debug every kind of executable. To debug a Windows NT program, for example, you need the full 32-bit IDDE running under Windows NT.

Screen images in this manual depict the Windows 3.1 IDDE, except where noted. The appearance of IDDE windows and dialog boxes under Windows NT and Windows 95 is slightly different.

## **Starting and exiting the IDDEs**

To start the IDDE from the Program Manager, make sure that Symantec C++ is installed properly, then double-click on the appropriate Symantec C++ IDDE icon in the Symantec C++ program group. The IDDE main window opens at the top of your screen.

## 2 Introducing the IDDE

To exit the IDDE, choose one of three commands from the IDDE **File** menu: **Exit**, **Exit & Save All**, or **Exit & Discard**. **Exit** leaves the IDDE and returns to the Program Manager. **Exit & Save All** exits and saves all the changes to the current project and options. **Exit & Discard** exits the IDDE without saving any changes to the current project or options.

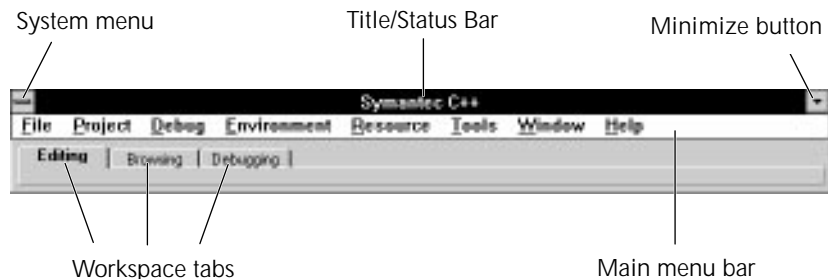
### IDDE Windows and Toolboxes

In Symantec C++, you create, edit, and debug your application in the IDDE. The IDDE provides a variety of tools for use throughout the development process.

Unlike most Windows applications, the IDDE offers more than a single window in which to work. The IDDE is a feature-rich environment in which you work with multiple windows and toolboxes on your desktop. This section describes the general characteristics of those windows and toolboxes.

#### IDDE main window

The IDDE main window, shown in Figure 2-1, is positioned at the top of the desktop (workspace), which is your entire screen area. From the IDDE main window, you can open other windows on the desktop, load projects, set project options, and perform additional tasks. Most of the actual work, such as editing program code, is done in other windows on the desktop.



**Figure 2-1** IDDE main window

As shown in Figure 2-1, the IDDE main window has the following parts: system menu, title/status bar, main menu bar, and the minimize button. (Also shown is the Workspace toolbox docked below the main window. Docking means that the toolbox is attached to the IDDE main window, or the desktop edge.) The system menu



and the minimize button are the standard Windows user interface elements. The other elements of the main window as well as the IDDE toolboxes are described in the following sections.

## The title/status bar

The top line of the IDDE main window serves as both a title bar and a status bar. When you start the IDDE, the title bar displays the program name, Symantec C++. As actions occur in the IDDE, the title bar changes to a status bar and displays the status of the session, including the following information:

- The command description when a menu item is highlighted
- The current project that is loaded, and whether the IDDE is in debugging mode
- Various messages, the results of expressions, and other status information from the IDDE

## The menu bar

The IDDE main window menu bar is located below the title bar. Table 2-1 summarizes the functions of the IDDE menus.

**Table 2-1** The IDDE menu functions

Menu	Function
<b>File</b>	Opening and closing text files, exiting the IDDE
<b>Project</b>	Opening, closing, and editing projects, building the project, setting project options
<b>Debug</b>	Switching into debugging mode, controlling program execution while debugging, setting debugging options
<b>Environment</b>	Creating and using workspaces, customizing the IDDE
<b>Resource</b>	Creating and editing resources
<b>Tools</b>	Accessing AppExpress, ClassExpress, Global Find, and your own tools
<b>Window</b>	Opening and arranging windows and toolboxes
<b>Help</b>	Accessing online help

## 2 Introducing the IDDE

For information on how to choose menu commands, see your Windows, Windows 95, or Windows NT documentation.

### The IDDE toolboxes

The IDDE includes the following four toolboxes:

- Views
- Build
- Debug
- Workspace

Open a toolbox through the IDDE's **Window** menu. If the toolbox's name is checked, it appears in the workspace. If the name is not checked, select it to open the toolbox. The following sections describe the toolboxes in detail.

### The Views toolbox

The Views toolbox, shown in Figure 2-2, is used to open the IDDE windows. Each of the IDDE windows is represented by an icon in the toolbox.

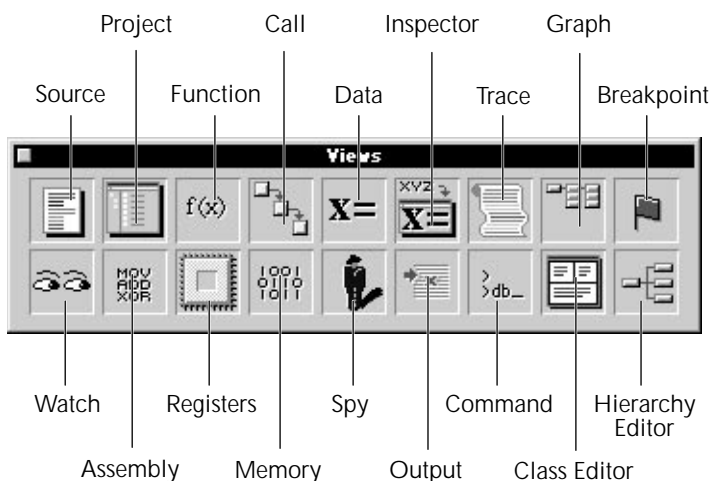


Figure 2-2 Views toolbox



To open a window from the Views toolbox, click and drag the appropriate icon from the toolbox onto the desktop. Alternatively, double-click on the icon to open the window. To replace one window on the screen with another, drag the icon of the window you want into a window on the desktop. Table 2-2 lists the IDDE windows.

**Table 2-2** IDDE windows and their functions

Window name	Function
Assembly	Debugging; shows assembly-level source
Breakpoint	Debugging; used to work with breakpoints
Call	Debugging; shows call chain
Class Editor	Used to edit class hierarchies
Command Line	Debugging; provides CodeView-like interface to the debugger
Data/Object	Debugging; shows local or global data
Function	Debugging; shows the functions in your application
Graphic Data	Debugging; shows a graphic representation of a data structure
Hierarchy Editor	Used to edit class hierarchies
Inspector	Debugging; shows local and global data
Output	Shows output of compiler, linker, and parser
Memory	Debugging; shows memory contents
Project	Displays the files in the project
Register	Debugging; shows contents of registers
Source	Used to create and edit source code
Spy	Debugging; used to view Windows messages
Thread	Debugging; shows the program's threads (Windows NT only)
Trace	Debugging; shows trace messages
Watch	Debugging; used to work with watchpoints

You can open multiple Data and Source windows. When you have more than one Data or Source window open, and then you minimize more than one of them, the IDDE adds a pop-up menu to the Data or Source icons in the Views toolbox (see Figure 2-3).

## 2 Introducing the IDDE

A small triangle in the lower-right corner of the icon indicates that a Source or Data window is minimized.



Figure 2-3 Pop-up menu in Source icon

To open a minimized Data or Source window:

1. Move the cursor over the triangle in the lower-right corner of the Data or Source window icon.
2. Click to open the pop-up menu.
3. Choose the window title you want from the pop-up menu. That window becomes active.

Alternatively, move the cursor over the window title in the pop-up menu, then click and hold the mouse button. A window icon appears to the right (see Figure 2-3). Drag the cursor over the icon, then move the cursor over either the desktop or an open window and release the mouse button.

### The Build toolbox

The Build toolbox, shown in Figure 2-4, provides quick access to project build commands.

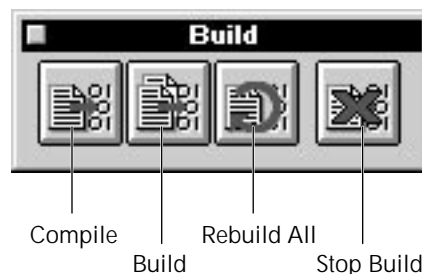


Figure 2-4 Build toolbox



To use the Build toolbox, click on the appropriate button in the toolbox. These buttons perform the following tasks:

- Compile a single source file
- Build the project
- Rebuild the entire project
- Stop the build in progress

### The Debug toolbox

Debug toolbox icons (shown in Figure 2-5) let you efficiently choose debugging commands during a debugging session. The commands available in the Debug toolbox correspond to the commands on the IDDE **Debug** menu and are described in detail in the section “Debug Toolbox Icons,” in Chapter 23, “Controlling and Configuring the Debugger.”



Figure 2-5 Debug toolbox

### The Workspace toolbox

When you first open the IDDE, the Workspace toolbox appears docked below the IDDE main window menu bar. The Workspace toolbox is used for switching between workspaces, which are customized layouts of windows that you define. The Workspace toolbox is shown in Figure 2-6.



Figure 2-6 Workspace toolbox

As you create new workspaces, their names appear on tabs in the Workspace toolbox. To switch to a different workspace, click on its tab. For more information on defining workspaces, see Chapter 3, “Starting a Project and Defining Workspaces.”

### Using toolboxes

The IDDE’s **Window** menu lists the names of all toolboxes. When a checkmark is displayed next to the toolbox name, the toolbox appears in the current workspace. To open a toolbox, select the toolbox name from the IDDE’s **Window** menu.

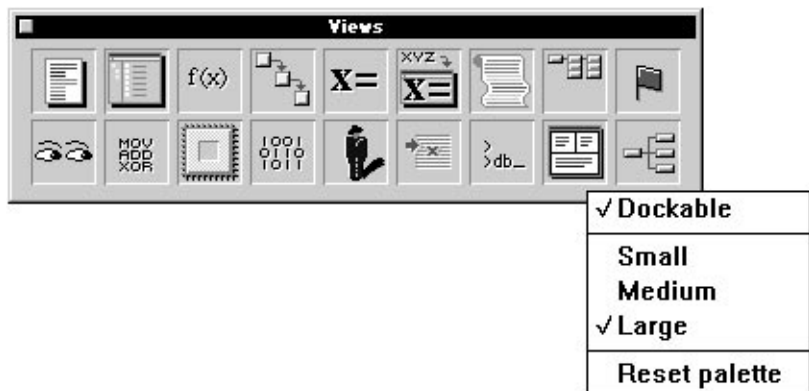
## 2 Introducing the IDDE

You can position a toolbox on the desktop by clicking on its title bar and dragging the toolbox to the desired position on the screen. To dock a toolbox, position it on the IDDE main window or on the edge of the screen. To undock the toolbox, click on the toolbox and drag it away from the IDDE main window or the desktop edge.

You can change the shape of a toolbox by clicking the toolbox edge and dragging the outline of the toolbox to the desired shape.

If you want to remove an icon from a toolbox, right-click on the icon you want to remove, drag it from the toolbox, and drop it.

Toolboxes can be configured with commands in their pop-up menus. To access a toolbox's pop-up menu, right-click on an empty part of the toolbox (as shown in Figure 2-7). When the first menu item, **Dockable**, is checked, you can dock the toolbox. The next three items (**Small**, **Medium**, and **Large**) let you change the size of the toolbox and its icons. The last item on the menu, **Reset Palette**, lets you restore any icons or buttons you have removed from the toolbox.



**Figure 2-7** The toolbox pop-up menu

To identify a toolbox icon, hold the mouse cursor over the icon for a few seconds. A small yellow tag appears, showing the name of the icon.

When you want to close a toolbox, click on the Close box. Each toolbox has a Close box in the upper-left corner, as shown in Figure 2-8.

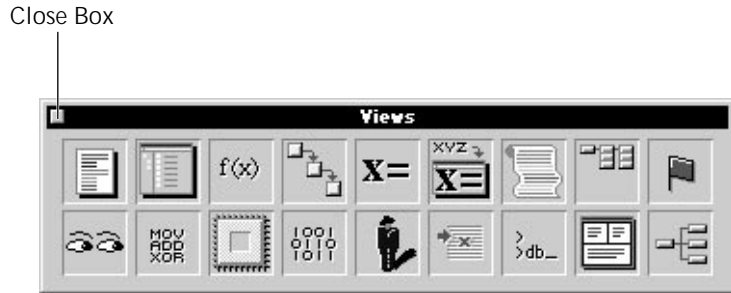


Figure 2-8 Close box

## The IDDE windows

This section describes the general properties of the IDDE windows and how to work with them.

### Opening and closing windows in the IDDE

Symantec C++ supports both standard Windows methods and a few unique methods for displaying and managing windows. In the IDDE, you can open a window in any of the following ways:

- From the Views toolbox.
- Through the **Goto View** submenu of the IDDE's **Window** menu. The **Goto View** submenu lists all of the windows you can open, along with the shortcut keys that open those windows.
- By choosing **Window List** from the IDDE's **Window** menu or from the system menu of any IDDE window. A dialog box listing all IDDE windows opens. Choose a window name from the list and click OK.

When you open a Source window to edit a specific source file, the filename is added to the **Window List** dialog box. You can then activate this particular window from the **Window List** dialog box.

Each window has a Close box similar to that of a toolbox (see Figure 2-8). Click on the box to close the window.

### **Using the drag-and-drop feature**

The Symantec C++ IDDE provides a drag-and-drop feature that lets you execute commands by dragging from one window onto another or onto the desktop. For example, you can drag a module's name from the project window onto the desktop in order to edit that module in a Source window. When something cannot be dropped on a particular window or the desktop, the cursor changes from that icon to a "No" sign (a circle with a diagonal line through it). Most of the drag-and-drop functionality is available in debugging mode and is described in Chapter 24, "Commands Available in Debugging Mode."

### **Using the IDDE**

The following sections serve as an overview of Part Two of this manual and introduce you to the process involved with creating an application in the Symantec C++ environment.

### **Creating a project**

Projects are integral to the Symantec C++ development system. You cannot build an application without creating a project. A project is a collection of source files, headers, resource files, and other components that you need to build an application.

To help you create a new project, a new tool called ProjectExpress is included. This tool lets you select the project target (the result of your development efforts), the directory for your project files, and other options. While in ProjectExpress, you can also use AppExpress to generate the framework for your application.

AppExpress is a tool that automatically generates an application framework. AppExpress lets you select from a variety of application types, then creates a working skeleton of source code and resource definitions. After the framework is generated, you can concentrate on customizing the framework and implementing the features of your application.

## Using workspaces

Because the IDDE provides a variety of windows used for specialized tasks, you probably do not need all of them open simultaneously. The IDDE workspaces provide a convenient way to switch from one screen layout to another. Workspaces are task-oriented, as opposed to project-oriented. You create workspaces for different tasks, such as editing, browsing, or debugging, which you perform in different projects. The **Workspace** submenu of the IDDE's **Environment** menu provides access to workspace commands.

## Creating and editing your application

Once you have created a project, you can start working on the program's code. This section introduces the IDDE tools.

### Creating and editing resources

If you are writing a Windows application, you probably need to create resources such as menus and dialog boxes. Commands on the IDDE's **Resource** menu access the Symantec C++ ResourceStudio, a powerful tool for creating and editing resources.

### Binding resources to classes

ClassExpress is a tool for binding resources, such as dialog box controls, to the classes in your program. You need to do this in order to be able to respond to user actions. You can launch ClassExpress by choosing **ClassExpress** from the IDDE's **Tools** menu, or from within ResourceStudio.

### Editing class hierarchies

The IDDE provides two tools for editing class hierarchies: Class Editor and Hierarchy Editor. You can open a Class or Hierarchy Editor window from the Views toolbox, or from the **Goto View** submenu of the IDDE's **Window** menu. Both editors have the same functionality, but different interfaces. The Class Editor emphasizes member editing, while the Hierarchy Editor emphasizes inheritance relationships.

### **Editing code**

While you can greatly reduce the amount of work needed to write a complex application using the tools Symantec C++ provides, eventually you need to edit the raw source code. The IDDE includes a powerful, scriptable text editor with features such as customizable key bindings, automatic token coloring, delimiter matching, and other features. Edit your code in the Source window, which you can open from the Views toolbox or the **Goto View** submenu of the IDDE's **Window** menu.

### **Debugging your application**

After you write most or all of your application's source code, you need to test and debug it. You do this by switching the IDDE into debugging mode and using the commands on the IDDE's **Debug** menu, the Debug toolbox, and various debugging windows you open from the Views toolbox.

### **Using help**

The following sections describe several ways to get online help in the IDDE.

#### **The title/status bar**

When you click and hold on a command in a menu, the IDDE main window title/status bar shows you a brief description of the menu command.

#### **The IDDE Help menu**

The IDDE provides several online help systems through the IDDE **Help** menu. The first items on the IDDE **Help** menu access the IDDE Help system, shown in Figure 2-9. To navigate the IDDE Help system, click on the icon representing the type of help you need.

Alternatively, choose **Search** from the IDDE **Help** menu to find a particular topic in the IDDE Help system.



Figure 2-9 The IDDE Help system

Other commands on the Help menu access Windows API and Microsoft Foundation Class reference material.

Finally, the **About Symantec C++** command on the IDDE **Help** menu displays a dialog box with version information on the Symantec C++ Development System.

### Other ways to launch online help

To view help information for a Symantec C++ run-time library function, you can highlight the name of the function in a Text Editor window and type CTRL+ALT+F1.

You can view help topics you use frequently without navigating IDDE Help, by associating specific .HLP files with key combinations. (For example, to view help for ResourceStudio, you could map RSTUDIO.HLP with the key combination CTRL+ALT+F4.)

To associate a key combination with a help file, choose **Edit/Browsing Settings** from the IDDE's **Environment** menu. Click Text, then click Help files.

## ◆ 2 *Introducing the IDDE*

---