Getting Started with CenterLine-C++™

Read this document to learn how to start using CenterLine-C++. These topics tell you what CenterLine-C++ is and how to use the online documentation:

- What is CenterLine-C++?
- Starting CenterLine-C++
- Understanding the Main Window
- Getting help with CenterLine-C++
- Using the online examples

Read these topics for an overview of how you use CenterLine-C++:

- Loading your code into CenterLine-C++
- Setting breakpoints and actions
- Finding and fixing violations
- Examining your code

All of these topics are covered in more detail in the hardcopy documentation.

What is CenterLine-C++?

CenterLine-C++ is a complete compilation system and debugger for C++ and C. It contains:

- The C++ compilation system used in ObjectCenter, CenterLine's C++ development environment
- pdm, the symbolic debugger used in process debugging mode in ObjectCenter
- A graphical user interface

The C++ compilation system is based on cfront 3.0.2.15. It includes additional features such as *precompiled header files* to speed up compilation of programs that use common header files, and *demand-driven code generation* to speed up compilation and reduce object code size.

Use pdm to debug fully-linked executables and to access debugging information in shared libraries. You'll find that you're already familiar with many pdm commands if you've used other debuggers.

In addition to the GUI, there is an *ASCII user interface* you can use on nongraphical or low-memory workstations or for automated test runs.

Getting ready to start

Before you get started, choose which editor you want CenterLine-C++ to invoke when you select an edit symbol or enter the **edit** command. The default is **vi.** To use FSF GNU Emacs instead, enter this command in a C shell:

```
% setenv EDITOR emacs
```

or this command in a Bourne or Korn shell:

```
% EDITOR=emacs; export EDITOR
```

FSF GNU Emacs and **vi** are the only editors we support. Consult the online documentation for help connecting other editors.

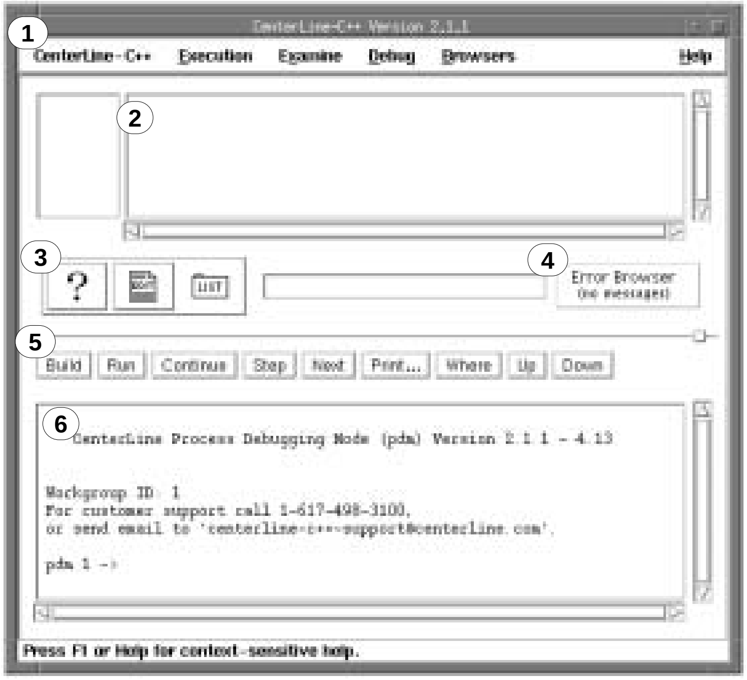
Starting CenterLine-C++

You invoke CenterLine-C++ with the **centerline-c++** command. If you don’t use any switches, CenterLine-C++ starts with the default graphical user interface style for your platform, either Motif or OPEN LOOK.

CenterLine-C++ starts by opening the Main Window.

Understanding the Main Window

To get help on any graphical object, move the cursor over the item and press the F1 or Help key.



```
  CenterLine Process Debugging Mode (pdm) Version 2.1.1 - 4.13


Workgroup ID: 1
For customer support call 1-617-498-3100,
or send email to 'centerline-c++-support@centerline.com'.

pdm 1 ->
```

Press F1 or Help for context-sensitive help.

1 *Menu Bar.* From the **CenterLine-C++** menu, you can send email to CenterLine, add custom commands and buttons, or quit. The **Browsers** menu gives you access to other CenterLine-C++ windows. Use the other menus to control your session and examine and debug your code.

2 *Source area.* When you start debugging your program, CenterLine-C++ lists your source code in the Source area with an arrow showing your current location.

3 *?, Edit, and List buttons.* Click on the **?** button to open the online documentation browser, on the **Edit** button to open your default editor, and on the **List** button to choose a new file to list in the Source area.

4 *Error Browser button*. When CenterLine-C++ finds a compile or make error in your code, the Error Browser button shows a “new errors” message and changes color on color workstations. Click on the button to open the Error Browser.

5 *Button Panel*. The commands you use most often are available in the Button Panel. You can also add your own custom commands.

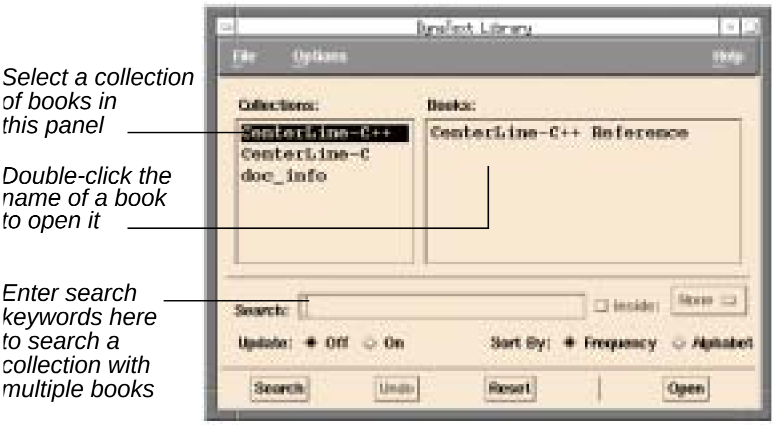
6 *Workspace.* Use the Workspace to enter pdm commands as you debug your code.

Getting help with CenterLine-C++

CenterLine-C++ comes with some hardcopy manuals, including the *CenterLine-C++ Programmer's Guide and Reference*. Chapter 7 of this manual, the reference to pdm commands, and the *CenterLine-C Programmer's Guide* are available online. The online documentation uses the *Dyna*Text™ viewer from Electronic Book Technologies, a powerful viewer whose features include full-text search and hypertext navigation.

There are two ways to open the *Dyna*Text Library window from within CenterLine-C++:

- Select Manual Browser from the Browsers menu on any primary window
- Click the "?" button in the Main Window



Select a collection of books in this panel

Double-click the name of a book to open it

Enter search keywords here to search a collection with multiple books

The Library window may take some time to initialize. As it does so, start-up messages are displayed in the Run Window.

You can also open the Library window outside of CenterLine-C++ by issuing the **cldoc** command from a shell.

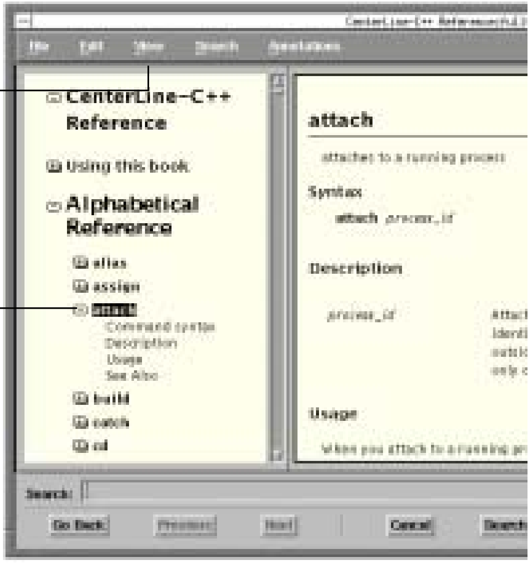
The left panel of the Library window lists collections of books, including the CenterLine-C++ and doc_info collections. The doc_info collection contains the *Dyna*Text *Reader Guide*. The right panel lists all the books in the highlighted collection in the left panel.

If there is more than one book in a collection, use the Search panel in the Library window to conduct a full-text search of all the books in the collection. If there is only one book, use the Search panel in the book window. You can search on words, patterns, and phrases, and you can conduct Boolean, context, or proximity searches.

To open a specific book, double-click on its name or select its name and click the Open button.

To open a specific *Reference* entry from within CenterLine-C++, issue the **man** command in the Workspace with the name of the command or topic, for example:

```
C++ 1 -> man attach
```



CenterLine-C++ Reference

Using this book

Alphabetical Reference

alias

assign

attach

Command syntax

Description

Usage

See Also

build

catch

cd

attach

attaches to a running process

Syntax

attach *process_id*

Description

process_id

Usage

When you attach to a running pr

Use the View menu to hide the Table of Contents or list tables or figures instead

Click on the + or - next to a section to expand or collapse the Table of Contents

Using the online examples

We provide a tutorial and a set of examples to help you get started with CenterLine-C++.

The tutorial uses a simple program named Bounce. The Bounce program creates a new window and bounces a rectangle within the window. In the tutorial you will find and fix an error in the program, and then enhance it so that it bounces a linked list of circles.

We provide the tutorial code and a makefile to build it online. Enter the following in a shell to create the directory containing the examples used in the tutorial and to change to the examples directory:

```
$ c++examples
$ cd c++examples_dir
```

The **c++examples_dir** also contains other examples used in the manual.

To start CenterLine-C++ and make the Bounce program, enter the **centerline-c++** startup command in a shell, and then enter this command in the Workspace:

```
pdm 1 -> make tutorial
```

Go to Chapter 5 of the *CenterLine-C++ Programmer's Guide and Reference* to work through the tutorial.

Getting your code into CenterLine-C++

To use the pdm debugger, build your application outside CenterLine-C++ and then debug the fully-linked executable. pdm is similar to debuggers such as **dbx** and **gdb** and uses most of the same commands, but it provides the Data Browser and Error Browser to help you visualize and edit your code, and it can access debugging information generated by CenterLine's compilation system.

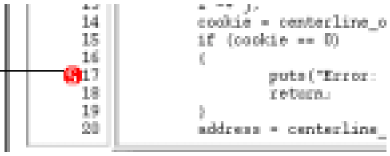
Load the executable with the **debug** command. You can also debug an executable with a corefile, or attach to a running process.

Issue the **run** command to initialize variables and run your program.

Setting breakpoints and actions

Set breakpoints and conditional breakpoints with the **stop** and **when** commands.

The GUI also provides several ways to set breakpoints. For example, the simplest way to set a breakpoint is to click the left mouse button on a line in the Source area.



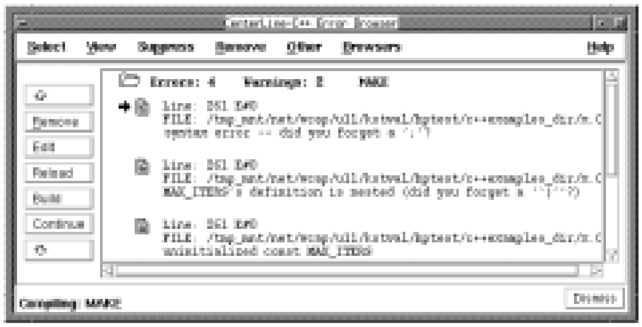
Clicking on the number 17 sets a breakpoint on line 17

```
14      cookie = centerline_o
15      if (cookie == 0)
16      {
17              puts("Error:
18              return;
19      }
20      address = centerline_
```

Finding and fixing violations

If CenterLine-C++ finds static errors when you compile your code, the Error Browser lists the errors. You can:

- List the code containing an error in the Source area by clicking on a message
- Invoke your editor with the cursor on the line containing the error by clicking on the edit icon in the Error Browser



CenterLine-C++ Error Browser
Select View Suggress Remove Other Browsers Help
Remove
Edit
Reload
Build
Continue
Dismiss

```
Errors: 4    Warnings: 2    MAKE

Line: 261 E#0
FILE: /tmp_mnt/net/wcsp/u11/kstval/bgtest/c++examples_dir/m.C
syntax error -- did you forget a ';'?

Line: 261 E#0
FILE: /tmp_mnt/net/wcsp/u11/kstval/bgtest/c++examples_dir/m.C
MAX_ITEMS's definition is nested (did you forget a ``}''?)

Line: 261 E#0
FILE: /tmp_mnt/net/wcsp/u11/kstval/bgtest/c++examples_dir/m.C
uninitialized const MAX_ITEMS
```

Compiling: MAKE

Examining your code

When execution stops (because you entered Control-C, or CenterLine-C++ encountered a breakpoint, a run-time error, or a signal), you are at a break level, and you can **step** through your code entering functions, execute the **next** line without entering functions, or continue execution with the **cont** command.

From a break level, you can use the Data Browser to display data structures and expressions. There are also CenterLine-C++ commands that let you step through machine instructions and examine the execution stack.

For further help

Some CenterLine-C++ documentation is distributed online. See “Getting help with CenterLine-C++” on page 4 for how to open it.

See *Installing CenterLine Evaluations* or *Installing and Managing CenterLine Products* for information about installing CenterLine-C++.

See the *AT&T C++ Product Reference Manual* for information about cfront and the *AT&T C++ Library Manual* for information about the complex and standard C++ libraries.

Use the UNIX **man** command to view manual pages for the following shell commands: **CC**, **clcc**, **cldoc**, **clezstart**, **c++examples**, and **centerline-c++**.