ABOUT THIS RELEASE OF CENTERLINE-C++

NEW FEATURES IN CENTERLINE-C++ VERSION 2.1.1

We have added the following new features in Version 2.1.1.

Sun platform support

This release adds support for the following:

- The Solaris 2.4 operating system
- The Sun SPARCompiler C Version 3.0.1
- Sun SPARC 5 workstations running SunOS 4.1.3_UI or Solaris 2.3 or 2.4

Online documentation

The CenterLine-C++ Debugging Reference is now available online. To access the new online documentation, select Manual Browser from the Browsers menu on any primary window, click on the "?" button in the Main Window, or issue the cldoc command from a shell.

In the left panel of the Library window are one or more collections of books. Click on the name of a collection to display the names of the books in that collection in the Books panel. You can open a specific book by double-clicking on its name, or selecting its name and clicking the Open button.

You can perform a simple search from the Dynatext Library window, and you can perform more complex searches from a book window. Select Forms from the Search menu to see which other search forms are available. Use the Next, Previous, and Go Back buttons to navigate through the book.

Underlined text is hot - clicking on it scrolls the window to the section of the book referenced, or opens a new window if the reference is to another book.

You can create a history of your movement through the book by selecting New Journal from the File menu and selecting Start Record in the dialog that pops up.

Print sections by selecting Print from the File menu and highlighting the sections you want to print. Export sections to a file by selecting Export from the Edit menu, highlighting the sections you want to export, selecting Content as the Export format, providing a filename, and clicking the Export button.

Figures, tables, and tips are shown as icons. Double-click to open them.

NEW FEATURES IN CENTERLINE-C++ VERSION 2.1

Thread support on Solaris 2.3

We've added support for threaded applications on the Solaris 2.3 platform in process debugging mode (pdm), with the ability to debug threads in executables and a graphical Thread Browser to show the status of all the threads in your program. We have also added a thread-safe libC (the C++ library).

In process debugging mode, the Thread Browser gives you information about the threads and lightweight processes in your program. This information includes a list of all threads, and the state of each thread. The state information includes the function the thread is executing, the execution state (for example, running, sleeping) of the thread, and the start function for the thread.

At any given time, the Thread Browser focuses on a single thread or light-weight process (LWP), known as the "current active entity." You control execution of the current thread with the cont, next, nexti, step, and stepi commands. You can display a traceback of the thread execution stack with the where command. You can also perform these operations on another thread at the break level by making it the current active entity. To make another thread the current active thread, you use the thread command with the new thread number as an argument.

For more information about debugging threaded applications, see the thread and threads entries in the Manual Browser.

Emacs Main Window

If you're a FSF GNU Emacs 19 user, you can start up a CenterLine-C++ session from within Emacs. A compatible version of FSF GNU Emacs is available by anonymous ftp from the host ftp.centerline.com in the /pub/TOOLS/emacs directory.

First you need to add the following lines to your .emacs file:

```
(setq load-path (cons "/path-to-install/CenterLine/lib/lisp" load-path))
(load "clipc")
```

The M-x centerline-c++ command starts CenterLine-C++ as a subprocess of Emacs, with the menus from the CenterLine-C++ Main Window replacing the menus at the top of your Emacs window. Edit the path name supplied if you want to run a different version of CenterLine-C++.

All the browsers are available with the same commands and menu items you use when you invoke CenterLine-C++ from a shell. Edit your code directly in the "Source area" at the top of the Emacs Main Window. A separate Button Panel window, including your own user-defined buttons, can be launched from the Browsers menu in the Main Window.

For more information, see the emacs integration entry in the online Reference.

Error Browser layout improvements

We've streamlined both the appearance and the performance of the Error Browser. The new Error Browser uses less memory and performs faster. Note that some new buttons are only active in our full C++ software development environment, ObjectCenter.

Changing the size of your windows

We've added Motif-style resizeable panes to the Main Window and Thread Browser. For example, to change the relative size of the Source panel and Workspace, place your cursor on the pane control sash, which is a square box at the bottom of the Source panel pane, just under the Error Sentinel. Hold down the Left mouse button and drag the sash up or down to resize the panes.

X resource changes

Some X resource settings have changed since the first release of CenterLine-C++ Version 2.0. For example, the Run Window has a scroll bar, and it's easier to change fonts and geometry for most windows. For more information, refer to the Run Window and Edit Window Resources section in the X resources entry in the Reference.

The CenterLine-C++ application defaults file (in CenterLine/clc++/lib/app-defaults/CenterLine-C++) now contains specific color resources for many items in the User Interface. You can override these default settings by changing or commenting out the resources in the global application-defaults file, or in your own copy of it. The new resources are all defined in a section of the file headed "GLOBAL RESOURCE SETTINGS".

HP clcc compiler

CenterLine's C compiler is now available on the HP platform. The CenterLine-C compiler is an ANSI C optimizing compiler designed to achieve small code size. The compiler is also compliant with K&R C and is link compatible with Sun and HP compilers.

Enhanced pdm debugging

In addition to support for debugging threaded applications, we've made some other improvements to pdm. The process debugging mode in CenterLine-C++ 2.1 is based on Version 4.12 of gdb and takes advantage of its new features.

ANSI C libC

We have also provided an ANSI C libC (the C++ library) for CenterLine-C++. The K&R C libC is the default. To use the ANSI C libC, use the the +a1 switch. The libraries are installed in CenterLine/<arch>/lib/a0/libC.a and CenterLine/<arch>/lib/a1/libC.a. See the UNIX manual page for CC for information about the +a0 and +a1 switches.

Licensing and installation enhancements

To make the installation process easier, we've made some changes to the RUN_ME script and made the licensing error messages more

informative. We've also updated our installation manuals, Installing CenterLine Evaluations and Installing and Managing CenterLine Products.

CHANGES BETWEEN VERSION 2.0.2 AND VERSION 2.1

This section describes improvements made in CenterLine-C++ in the point releases since Version 2.0.2.

New pdm option and switch

The option full_symbols has these characteristics:

```
Type: Boolean
Default Value: FALSE
Commands Affected: debug
```

When full_symbols is false, CenterLine-C++ reads only part of the debugging information to shorten initialization time. Additional debugging information will be read as needed, such as when you issue the whatis or list command. When full_symbols is true, CenterLine-C++ reads all the debugging information at initialization.

You can set the value of full_symbols by adding this line to your .pdminit file:

```
setopt full_symbols
```

You can also set it by issuing this command in the Workspace before issuing the debug command:

```
-> setopt full_symbols
```

Use the unsetopt command to reset the full_symbols option to false.

For more information about setting, unsetting, and displaying options in the Workspace, see the setopt, unsetopt, and printopt commands.

The new switch -full_symbols corresponds to the option full_symbols. For example, with the following command, you can invoke CenterLine-C++ in pdm mode with full_symbols set:

```
% centerline-c++ -full_symbols
```

Longer line limit in the Source area

The longest line the Source area could display used to be 500 characters. Now, the Source area can display up to 10,000 characters per line.

CENTERLINE-C++ DOCUMENTATION

This section describes the documentation for CenterLine-C++. CenterLine-C++ Version 2.1.1 comes with online documentation and hardcopy documentation. In addition, we provide

USL C++ Language System documentation.

Online documentation

Access an online version of the Reference to pdm, the process debugger used with CenterLine-C++, by selecting Manual Browser from the Browsers menu or by typing cldoc in a shell. Appendixes to the online Reference contain information about the current release, platform-specific information, and selected readings and Readme files about the C++ Language System.

In addition to the online documentation described above, the following information is available:

- Access context-sensitive help in the GUI version of CenterLine-C++ by moving the cursor over the item you want information about and pressing F1 or the Help key if your keyboard has one. A Help window appears describing that item. If you have bound the F1 key to a window manager operation, you are unable to access context-sensitive help with the F1 key.

- Access information on a variety of topics from the Help menu, which appears on every primary window.

- Access information about a command by typing help in the Workspace followed by the name of the command.

- Access any entry in the online Reference by typing man and the name of the entry in the Workspace.

The online documentation available outside the CenterLine-C++ environment is in this directory:

<path>/CenterLine/clc++/<arch>/docs

The word <path> represents the path to the CenterLine directory, and <arch> is a platform-specific directory, for example sparc-sunos4, sparc-solaris2, or pa-hpux8. The online directory contains a file called README, which describes the files in this directory.

Among the files are

- bugs.open, which describes the known bugs, limitations, and workarounds for CenterLine-C++

- bugs.fixed, which describes bugs fixed since the most recent version of CenterLine-C++

Hardcopy documentation

This is the hardcopy documentation that comes with CenterLine-C++:

CenterLine-C++ Read Me First Release Bulletin
The latest hardcopy information, containing any updates necessary to other documentation.

Installing and Managing CenterLine Products
How to install CenterLine-C++ and administer it, including how to troubleshoot licensing problems.

CenterLine-C++ Programmer's Guide and Reference
Information about the CenterLine-C++ compilation system and pdm, our process debugger.

CenterLine-C Programmer's Guide
Information about the CenterLine-C compiler.

USL documentation

CenterLine-C++ also comes with the following USL C++ Language System documentation:

AT&T C++ Language System Product Reference Manual
We ship this complete manual with CenterLine-C++. This manual provides a complete definition of the C++ language supported by Release 3.0 of the C++ Language System.

AT&T C++ Language System Library Manual
We ship this complete manual with CenterLine-C++. This manual describes class libraries shipped with Release 3.0: the iostream library, complexlibrary, and task library. (We do not, however, support or supply the task library libtask.a itself.)

PRODUCT LIMITATIONS

The following is a list of known current limitations with our product.

GUI behavior
CenterLine-C++ currently does not allow you to change the placement of the scrollbar. In addition, you cannot set the scrollbar placement with a window manager X resource (such as Motif's XmNscrollLeftSide or XmNscrollRightSide or OPEN LOOK's OpenWindows.ScrollbarPlacement). This is because CenterLine-C++ is developed using the Object Interface (OI) toolkit, not the Motif or OPEN LOOK (XView) toolkits. The OI toolkit currently does not offer a resource to change scrollbar placement.

Data Browser
Due to window size limitations in X, the Data Browser has a limit to the number of items it can contain. The limit is determined at run time. The font and model (Motif or OPEN LOOK) you use affect the limit. Using default fonts, you can create about 900 fields per data item under Motif and about 1500 under OPEN LOOK.

Source area
There is a limitation to the number of lines that you can list in the Source area. You can list files up to about 30,000 lines.

Limited thread support
CenterLine-C++ currently supports threads on the Solaris 2.3

platform only.