

Fortran 90 Browser

 **SunSoft**
A Sun Microsystems, Inc. Business
2550 Garcia Avenue
Mountain View, CA 94043
U.S.A.

Part No.: 802-2190-10
Revision A, March 1995

© 1995 Sun Microsystems, Inc.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

Copyright © 1993 Cray Research, Inc. All Rights Reserved. This manual or parts thereof may not be reproduced in any form unless permitted by contract or by written permission of Cray Research, Inc. CRAY is federally registered trademarks of Cray Research, Inc.

All rights reserved. This product and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX[®] and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's font suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS

Sun, the Sun logo, Sun Microsystems, Sun Microsystems Computer Corporation, Solaris, are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. OPEN LOOK is a registered trademark of Novell, Inc. PostScript and Display PostScript are trademarks of Adobe Systems, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCcompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK[®] and Sun[™] Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a product of the Massachusetts Institute of Technology.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



Contents

Preface.....	v
1. Overview	1
2. Getting Started	3
Starting f90browse	3
Opening a File	6
Looking for Objects	8
Finding a Loop.....	8
Finding External Calls.....	10
Saving Information	11
Displaying Call Trees.....	13
Call Tree Display	13
Caller/Call Tree display	15
Exiting f90browse	16

CIF Files.....	17
Generating CIFs Before Starting f90browse.....	17
Generating CIFs in the Background	18
3. Getting Help	19
Help.....	19
Context Sensitive	20
Man page.....	20
About f90browse.....	20
Glossary	21
Index.....	23

Preface

This preface is organized into the following sections.

<i>Purpose and Audience</i>	<i>page v</i>
<i>Before You Read This Book</i>	<i>page vi</i>
<i>Related Documentation</i>	<i>page vi</i>
<i>Conventions in Text</i>	<i>page viii</i>

Purpose and Audience

This guide shows how to use the Fortran 90 Browser, `f90browse` to browse Fortran 90 source code. `f90browse`, is a part of the `f90` package, which includes a set of application development tools for debugging and performance analysis. They provide a powerful, user-friendly environment for the development and enhancement of new and existing code.

The guide is intended for scientists and engineers with the following:

- Thorough knowledge of Fortran 90
- General knowledge of some operating system (experience with some OS)
- Particular knowledge of the SunOS™ commands `cd`, `pwd`, `ls`, `cat`

Before You Read This Book

If you are not familiar with Fortran 90, you may want to consult the following.

- *Fortran 90 Handbook* (Fortran 90 language definition, including intrinsics)
- *Fortran 90 Explained* (Text book introduction to Fortran 90)

Related Documentation

The related kinds of documentation included with Fortran 90 are as follows:

- Paper manuals (hard copy)
- On-line manuals (AnswerBook)
- On-line man pages
- On-line READMEs directory of information files

AnswerBook puts the paper manuals on-line and allows searches of topics. The system and manuals are included on the CD-ROM and can be installed to hard disc during installation. Installing and starting AnswerBook are described in the manual *Installing SunSoft Developer Products on Solaris*.

man Pages

Purpose

A man page is intended to answer “What does it do?” and “How do I use it?”

- *Memory Jogger*— A man page *reminds* the user of details, such as arguments and syntax. It assumes you knew and forgot. It is not a tutorial.
- *Quick Reference*—A man page helps find something *fast*. It is brief, covering major highlights. It is a *quick* reference, not a *complete* reference.

Usage

To display a man page, use the man command.

Example: Display the f90 man page.

The man command uses the MANPATH environment variable, which can effect which set of man pages are accessed.
See man(1).

```
demo$ man f90
```

Example: Display the man page for the man command.

demo\$ man man

Related man Pages

The following man pages may be of interest to Fortran 90 users.

f90(1)	Invoke the Fortran 90 compiler
f90browse(1)	Invoke the Fortran 90 Browser
asa(1)	Print files having Fortran carriage-control
dbx(1)	Debug by a command-line-driven debugger
debugger(1)	Debug by a graphical-user-interface
fsplit(1)	Print files having Fortran carriage-control
ieee_flags(3M)	Examine, set, or clear floating-point exception bits
ieee_handler(3M)	Exception handling
matherr(3M)	Error handling

READMEs

The READMEs directory has information files: bug descriptions, information discovered after the manuals were printed, feedback form, and so forth.

	Standard Installation	Nonstandard Installation to /my/dir/
Location	/opt/SUNWspro/READMEs/	/my/dir/SUNWspro/READMEs/

	Contents
File Names	
feedback	Sun programmers email template file: Send feedback comments to Sun
fortran	Fortran 90 bugs, new features, behavior changes, documentation errata

Conventions in Text

We use the following conventions in this manual to display information.

- We show code listing examples in boxes.

```
WRITE( *, * ) 'Hello world'
```

- Plain typewriter font shows prompts and coding.
- In dialogs, **boldface typewriter font** shows text the user types in.

```
demo$ echo hello
hello
demo$ ■
```

- *Italics* indicate general arguments or parameters that you replace with the appropriate input. Italics also indicate emphasis.
- For Solaris 2.x, the default shell is sh and the default prompt is the dollar sign (\$). Most systems have distinct host names, and you can read some of our examples more easily if we use a symbol longer than a dollar sign. Examples generally use “demo\$” as the system prompt; where the csh shell is shown, we use “demo%” as the system prompt.
- A small clear triangle Δ shows a blank space where that is significant.

```
ΔΔ36.001
```

- We usually show Fortran 90 examples in free form, not fixed form or tab.
- We usually abbreviate “Sun Fortran 90” as “f90”.
- We usually show Fortran 90 keywords and intrinsics in uppercase, and all else in lowercase or mixed case.

Overview



The Fortran 90 Browser, `f90browse`, provides the programmer with an interactive environment in which to view and edit application source code. With `f90browse`, you can obtain information about many aspects of your code. This information is available on the routine, file, and program levels.

With `f90browse` you can do the following tasks:

- Browse Sun Fortran 90 source code.
- Create graphical displays of call trees that enable you to see quickly and clearly the structure of the code on which you are working.
- Generate information about loops, cross-references, common blocks, include files, and macro cross-references.
- Combine files into a single application in which multiple files are treated as a single entity, thus allowing you to trace variables, calls, callers, and other portions of code on a global basis.
- Locate objects when only part of the object name is known.
- Find and display compilation errors.
- Save information (displays and graphs) to files.
- Access other tools.
- Access extensive online help without ever leaving your desk or your `f90browse` session.

- Type commands within the f90browse information panel to speed execution of frequently performed tasks.
- Customize your interface through X Toolkit options and through the f90browse Options menu.

Getting Started



This section explains the following:

- Starting f90browse and the f90browse main window
- Opening a file and loading code for browsing
- Searching for objects in your code
- Saving information to a file
- Displaying various call trees
- Exiting f90browse
- Understanding CIF file use

2.1 Starting f90browse

You start f90browse by entering the f90browse command:

```
f90browse &
```

This command causes f90browse to run in the background and displays the f90browse main window on your screen. This window is shown in Figure 2-1.

Note – The f90browse displays on your screen may differ from those shown in this document due to variances among window managers.

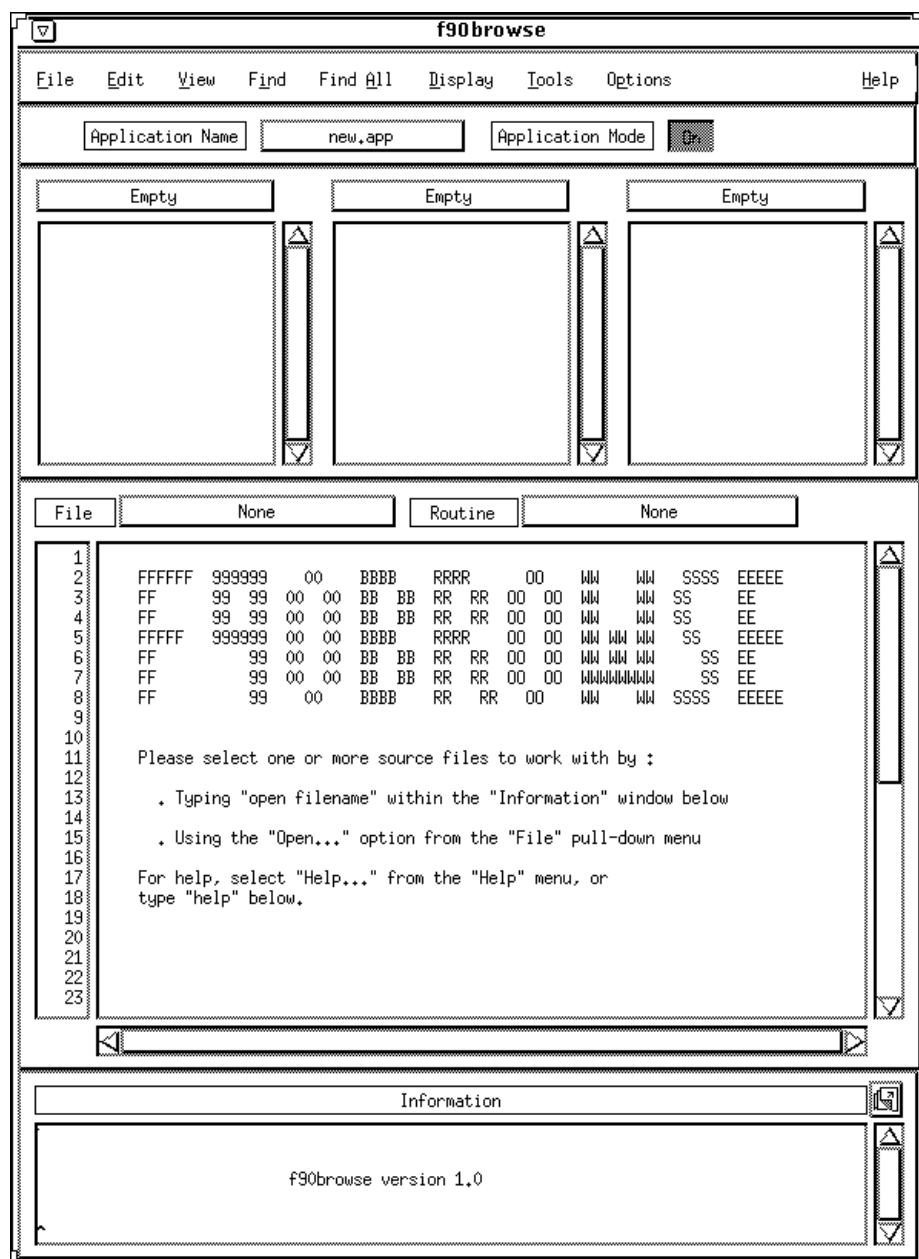


Figure 2-1 Main f90browse window

The main window is composed of the following segments:

- *Menu bar* – The menu bar is located at the top of the main window and displays buttons that open `f90browse` menus. To open a menu, position the cursor on the menu name and press the left mouse button.
- *Upper display panel* – The upper display panel is composed of three separate information lists that display names of loaded files, routines, common blocks, and so forth. All lists have vertical scroll bars on the right. If information in a list exceeds the width of the listing area, a horizontal scroll bar also is displayed.
- *Source code display* – The source code display is located in the middle of the `f90browse` window and is the largest area of the window. This display shows the source code you are currently browsing.
- *Information panel* – The information panel is located at the bottom of the main `f90browse` window and provides you with information about the status of `f90browse`. You can also type `f90browse` command equivalents for many menu options in this panel. (A list of these commands is available through the Help menu option.)

Although `f90browse` is contained within a single window, you can move some screen segments off the main window to create separate windows that you can then resize, reposition, and economy. Moving a segment off the main window is referred to as *peeling off* the window segment.

Screen segments that can be peeled off display a peel-off icon in the upper-right corner. One example of using a peel-off window is shown in subsection Section 2.4, “Saving Information.”

2.2 Opening a File

To open a file, position the cursor on the File menu button and press the left mouse button to open the menu. (This is also referred to as *clicking* on a selection.) With the menu displayed, click on the Open option. A popup window is displayed on which you enter the name of the file (or files) to be opened. Figure 2-2 shows this panel.

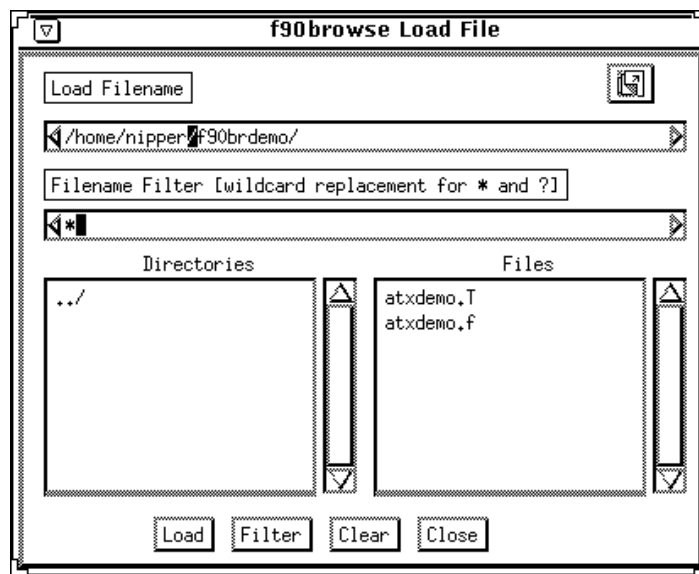


Figure 2-2 . Open display

Enter the file name(s) on the top input line and click on the Load button or press **RETURN** to open the file(s). You can use wildcards when entering file names. To enter multiple file names, separate each name with a space. If the name of the file you want to load is displayed in the Files list, simply use the left mouse button to double click on the file name to open the file. When the file is loaded, f90browse displays it in the source code display as shown in Figure 2-3.

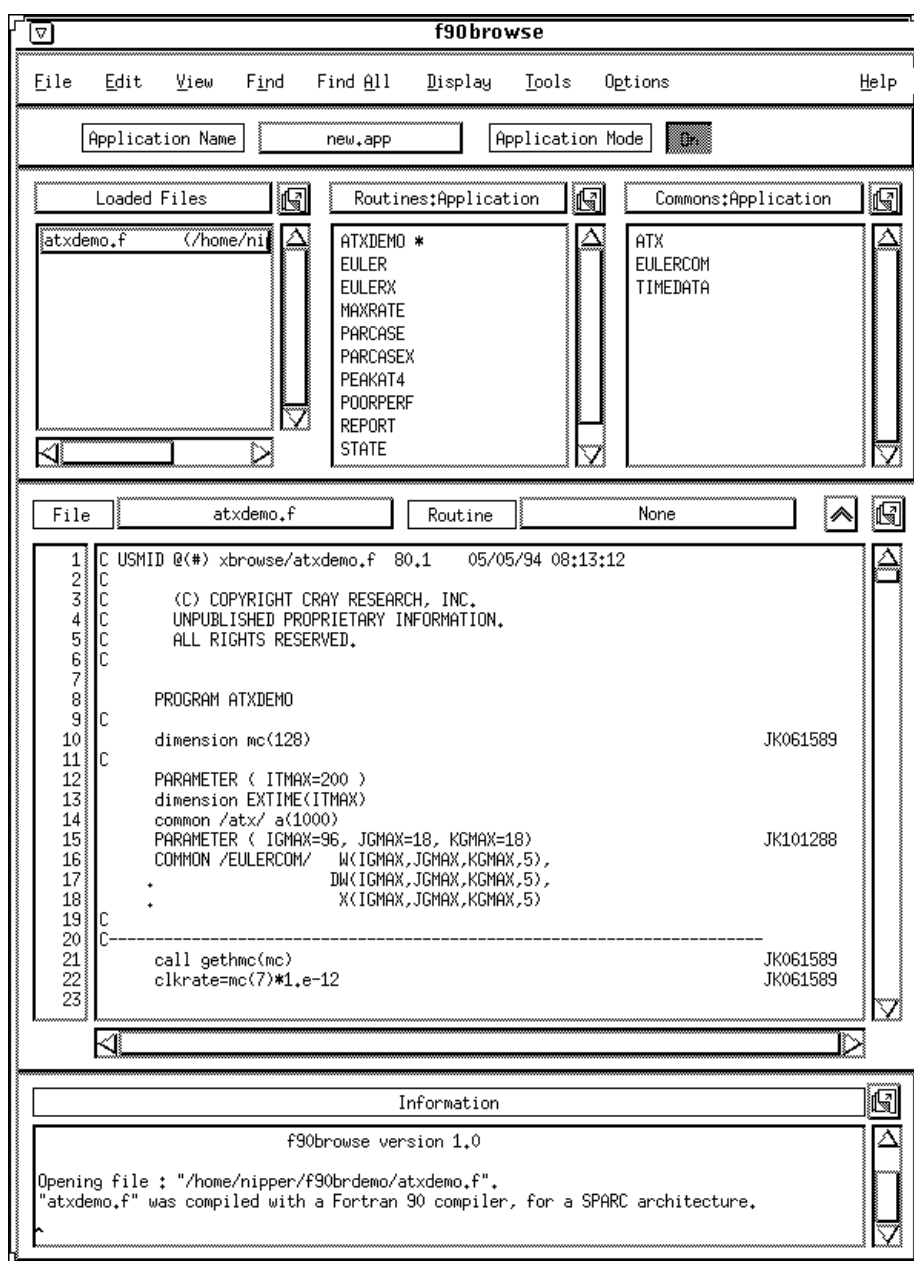


Figure 2-3 atxdemo.f file opened

2.3 *Looking for Objects*

A common activity while browsing code is to look for various types of objects. The following list names just a few of the objects you can locate using `f90browse`:

- Calls and callers
- Common blocks
- Loops
- Routines
- Variables

You look for routine-based information by using Find menu options. Find All menu options let you look for file- or application-based information. Most options work in a similar manner, so trying a couple of searches will give you an idea of how to find objects in your code. The next two subsections illustrate how to find loops and external calls in our sample Fortran program.

2.3.1 *Finding a Loop*

Because loops are displayed by default when a routine name in a Fortran file is selected, finding loops is very easy. To display loops, click on the routine name for which you want information displayed. Loop information is displayed in the center information list of the upper display panel and the area of your code where the loop occurs is highlighted in the source code display.

Figure 2-4 shows loop 1010 of the MAXRATE routine selected in the information list with the area where the loop occurs highlighted in the source code display. The information list shows the following information about the selected loop:

<u>Value</u>	<u>Description</u>
1010	Loop label. If a loop has no label, the word Loop is displayed.
75 - 80	Starting and ending line numbers.
1	Nesting level. A nesting level of 1 indicates an outer loop, 2 indicates a single nest, and so forth.

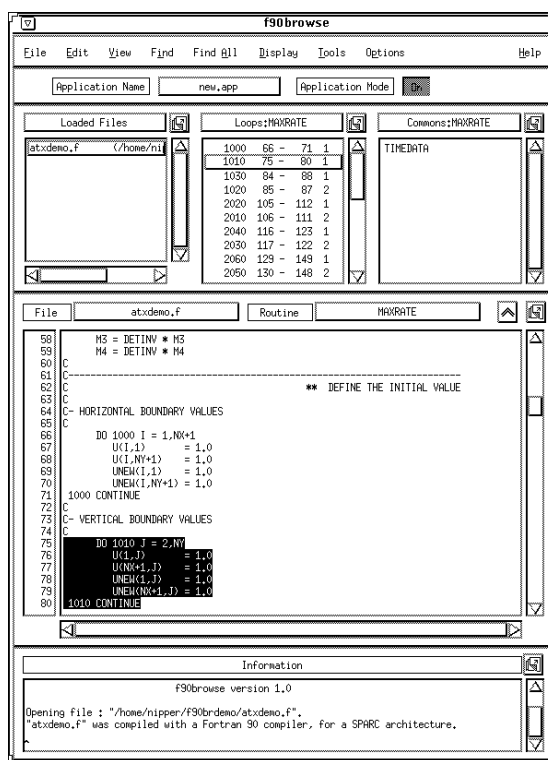


Figure 2-4 Loop option output

2.3.2 *Finding External Calls*

The **Externals** option of the **Find All** menu lists external calls in a file or application. (An external call is one for which **f90browse** does not have the source code loaded.) Because external calls are not displayed by default, follow this procedure:

1. In the upper display panel, click on a file name to select the file.
2. Click on the **Find All** button to open the menu.
3. Click on the **Externals** option.

The **Externals** option displays the names of external calls in the upper display panel. To view every occurrence of an individual call, select an external call by clicking on the call name; each occurrence of the call is highlighted in the source code display.

Figure 2-5 shows output from the **Externals** option for the `atxdemo.f` file.

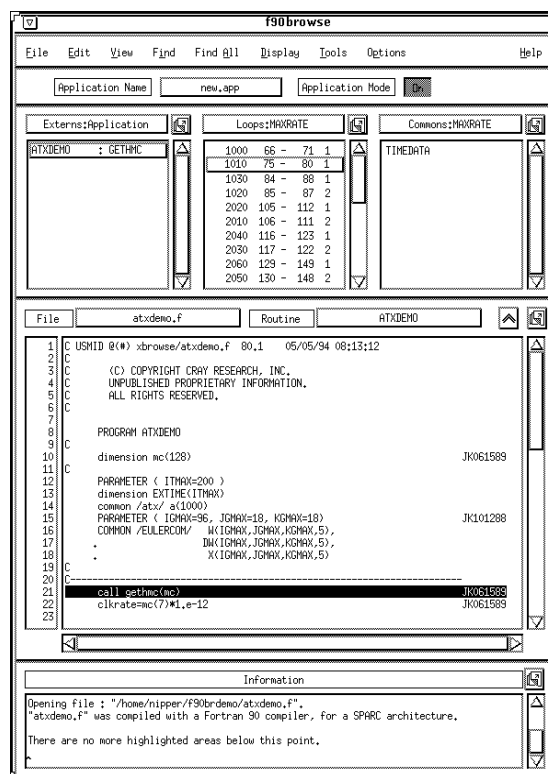


Figure 2-5 Externals option output

2.4 Saving Information

f90browse makes it easy to save output from menu options by using the peel-off feature, which was introduced on page 2-5. Information displayed in any segment of the f90browse window that you can peel off can be saved to a file. For example, to save the MAXRATE loop information shown in Figure 2-4, refer to the following procedure.

1. Position the cursor on the peel-off icon located to the right of the loops:MAXRATE title bar (shown in Figure 2-4 on page 2-9) and press the left mouse button. When you select **Save List**, a popup window opens that allows you to choose among several options.

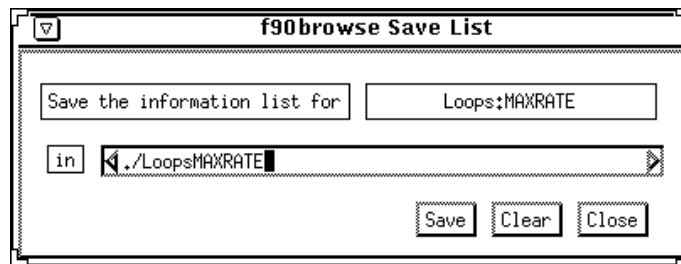


Figure 2-6 Save List display

2. To save information to a file using the default file name, click on the **Save** button. To change the file name, click on the **Clear** button to erase the default file name and then enter a new file name.
3. After you enter a new file name or use the default name, click on the **Save** button to save the information.

Note – Unless you specify a full path name for the file, files created from this type of save operation are saved in the directory from which you invoked f90browse.

4. Click on the **Close** button to close the display.

2.5 *Displaying Call Trees*

Call trees make it easy to see the structure of your code through a clear graphic format. The Display menu provides you with the following two call tree options:

- Call Tree
- Caller/Call Tree

f90browse gives you visual clues to help you identify various parts of your code within the call trees. On monochrome screens, external routines are marked with an E following the routine name and intrinsic routines are marked with an I. On color screens, external routines are shown in green and intrinsic routines are shown in yellow. (An external call is one for which f90browse does not have the source code loaded.)

2.5.1 *Call Tree Display*

The Call Tree option produces a static call tree of loaded source code. The name of the application or file used by this option is shown at the top of the display and called routines and subprograms are displayed in the middle.

The small, empty boxes following some nodes (routine names) on the tree indicate there are subtrees that stem from these nodes. To open the tree one level, position the cursor on the box and press the left mouse button. To close the tree one level, position the cursor on the node name and press the right mouse button. When you click on a node, that node becomes the current node and is displayed in the main f90browse window. To close the display, click on the Close button.

Figure 2-7 shows the application call tree for ATXDEMO with the STATE routine opened up one level.

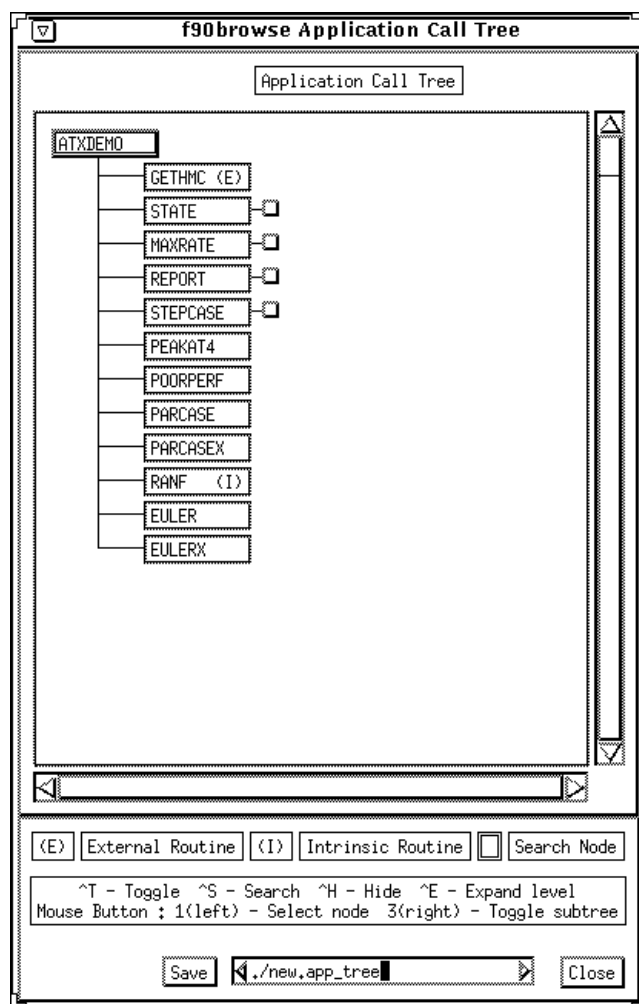


Figure 2-7 Sample Application Call Tree display

2.5.2 *Caller/Call Tree display*

The **Caller/Call Tree** option displays a static call tree of routines that call a specified subprogram and, in turn, displays any subprograms called by the specified subprogram. Selecting a node on the tree highlights the subprogram. The name of the source file (or the application when in application mode) used by this option is shown at the top of the window. Calling sequences are displayed in the middle of the window. To close the display, click on the **Close** button.

Figure 2-8 shows sample output from the **Caller/Call Tree** option. In this example, the **STATE** routine is called by **ATXDEMO** and calls both the **CLOCK** and **DATE** routines.

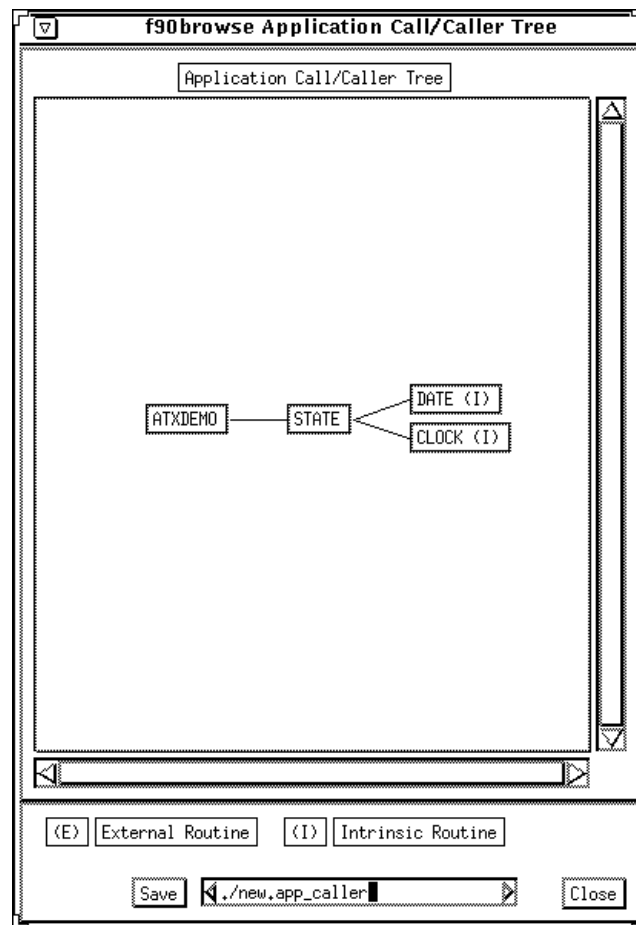


Figure 2-8 Sample Caller/Call Tree display

2.6 Exiting f90browse

To exit the f90browse tool, select Quit f90browse from the File menu. f90browse displays a popup window on which you can confirm or cancel your decision to exit.

2.7 CIF Files

Compiler information files (CIFs) are generated by the Fortran compiler and contain information about source code. Depending upon the options you select on the compiler command, a CIF may contain information about the characteristics of the machine for which the code was compiled; information about variables, constants, name lists, labels, loops, and procedures; cross-reference information; and compiler messages. `f90browse` obtains information about source code from CIFs generated by the compilers. These files are named `filename.T`.

2.7.1 Generating CIFs Before Starting `f90browse`

If a CIF exists for a particular source file, `f90browse` will use that file. If no CIF exists, or if it is found to be out-of-date (that is, older than the source file), `f90browse` creates a new CIF by invoking the proper compiler for the source file. You can decrease `f90browse` start-up time by using compiler options to generate CIFs before you start `f90browse`.

Individual compiler options are as follows:

Compiler	Option
Fortran 90	-db <i>filename.f</i>

If you compile your code with the compiler option on, `f90browse` reads from the already created CIF rather than creating a new CIF when you invoke the `f90browse` tool.

`f90browse` creates a binary format of the CIF file and adds a file extension of `.TT` to the file name. By default, these files are created in the same directory as the source code.

You can, however, specify that all CIFs created by `f90browse` be placed in the directory pointed to by the `CIFDIR` environment variable as shown in the following examples:

```
mkdir /tmp/cifs          (csh users)
setenv CIFDIR /tmp/cifs
```

```
mkdir /tmp/cifs          (ksh and sh users)
CIFDIR=/tmp/cifs
export CIFDIR
```

In these examples, all `f90browse`-created CIFs will be placed in and read from the `/tmp/cifs` directory.

2.7.2 Generating CIFs in the Background

By default, `f90browse` creates a CIF as a foreground process, thereby stopping all other activity within the tool until the CIF has been created. If you want CIFs to be generated in the background, set the `NPROC` environment variable to a value greater than zero.

The online help system available within `f90browse` puts information at your fingertips with the click of a button and ensures that help is always available when you need it. The Help menu gives you easy access to information about every aspect of the tool. By selecting among the following menu options, you can view documentation online without ever leaving your `f90browse` session:

- Help
- Context Sensitive
- Man page
- About `f90browse`

3.1 Help

The Help option displays in the main window. It displays an online file with complete information about `f90browse`. You can think of this option as a replacement for the usual paper documentation. It contains information about the following topics:

- Information that introduces the various portions of the `f90browse` window and helps you understand how to navigate through `f90browse`
- Full descriptions of all available menu options

- Lists of X Window System Toolkit options and environment variables available with `f90browse`
- A list of command equivalents for menu options

3.2 *Context Sensitive*

Context-sensitive help provides information relevant to the area of the `f90browse` window in which your cursor is positioned. When you select context-sensitive help, `f90browse` displays a popup window that gives a brief description of the area of the window on which the cursor is positioned. You can think of context-sensitive help as an online quick-reference guide.

Because output is displayed in a separate window, you can resize, move, and iconify the display. In addition to clicking on the menu option, you can display Context Sensitive help by pressing either the `F1` key or the key designated as the `HELP` key on your workstation.

3.3 *Man page*

When you select this option, `f90browse` pops up a window on which to enter the name of the man page you want to view. The man page is displayed in the main `f90browse` window. In addition, by using the peel-off feature, you can save man page output to a file and print the contents.

3.4 *About f90browse*

When you select this option, `f90browse` displays version and copyright information about the tool.

Glossary

application mode	The mode in which two or more files are combined so that many f90browse options apply to an entire program rather than to an individual routine. This is the default mode of f90browse.
button	An icon that allows you to open menus, execute commands, and display popup windows. To activate, position the cursor on the button and press the left mouse button. (This is also called clicking on a button.) Buttons may be boxed as are the Load, Filter, Clear, and Close buttons of the Open display shown in subsection Section 2.2, “Opening a File,” on page 6, or they may be unboxed as are the menu buttons at the top of the main f90browse window.
call tree	A graphical display of selected source code in which the name of the source file is shown at the top of the f90browse display and called routines and subprograms are displayed below.
click	To press a mouse button once and release it without moving the pointer.
compiler information file	A file produced by the Fortran 90 compiler that describes a source file. Depending upon compiler options, this file may contain information about procedures, variables, constants, labels, compiler messages, cross-reference information, and so forth.
external call	A call to a subprogram for which f90browse does not have the source code loaded.
information panel	Screen area located at the bottom of the main f90browse window that displays acknowledgments, messages, and errors. You also can type f90browse commands in this panel.
menu buttons	Buttons located on the menu bar at the top of the main f90browse display. Clicking on a menu name (button) opens the associated menu.

node	A point defined by a unique name.
peel-off capability	Allows you to convert a screen segment into a separate X Window System window that can be moved, resized, iconified, or otherwise manipulated in the same manner as any other X Window System window.
popup window	A window that is displayed to perform a specific function and that is then dismissed.
source code display	Screen area located in the middle of the main f90browse window that displays the source code you currently are browsing.
source file	The actual code being examined by f90browse.
upper display panel	Screen area located near the top of the main f90browse window that displays three information lists that contain output from menu options.

Index

Context-sensitive help 20
Copyright information
 displaying 20
Help
 context sensitive 20
 online 19
 types available 19
 viewing through xbrowse 20
Online help 19
Version information
 displaying 20

A

addenda for manuals, read me file, vii
audience, v

C

Call trees
 types, 13
Commands
 entering in the information panel, 5
 f90browse, 3
comments
 to Sun, vii

Compiler information file
 definition, 17
 generating in the background, 18
 using to reduce start-up time, 17
Compiler options, 17
Context-sensitive help, 20
conventions in text, viii
Copyright information
 displaying, 20

D

Definition
 clicking, 6
 compiler information file, 17
 external call, 10
 information panel, 5
 menu bar, 5
 peel-off icon, 5
 peel-off window, 5
 source code display, 5
 upper display panel, 5

E

email
 send feedback comments to Sun, vii
errata and addenda for manuals, read me
 file, vii

External calls
 definition, 10
 finding, 10

F

f90browse
 exiting, 16
 main window, 4
 starting, 3
f90browse command, 3
feedback file for email to Sun, vii
Files
 .T, 17
 .TT, 17
 information files, vii
 opening, 6
 saving output to, 11
font conventions, viii

H

Help
 context sensitive, 20
 online, 19
 types available, 19

I

Information
 about external calls, 10
 about loops, 8
 files, vii
 saving, 11
Information panel
 definition, 5
 typing commands in, 5

L

Loops, finding, 8

M

Man pages
 viewing through xbrowse, 20

man pages, vi
Menu bar, definition, 5

O

on-line documentation, vi
Online help, 19
Output, saving, 11

P

Peel off
 icon, example, 5
 window, definition, 5
 windows, usage, 5
 windows, using to save
 information, 11
prerequisites, v
prompt conventions, viii
purpose of manual, v

R

READMEs, vii

S

Saving output to a file, 11
Source code display, definition, 5
standard
 conformance to standards, vi
Sun, sending feedback to, vii

T

triangle as blank space, viii

U

Upper display panel, definition, 5

V

Version information
 displaying, 20