

# *Managing the Toolset*

 *SunSoft*  
A Sun Microsystems, Inc. Business  
2550 Garcia Avenue  
Mountain View, CA 94043  
U.S.A.  
Part No.: 801-7265-10  
Revision A, August 1994

© 1994 Sun Microsystems, Inc.  
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

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<sup>®</sup> 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, Sun Microsystems, the Sun logo, Sun Microsystems Computer Corporation, the Sun Microsystems Computer Corporation logo, SunSoft, the SunSoft logo, ProWorks, ProWorks/TeamWare, ProCompiler, Sun-4, SunOS, Solaris, ONC, ONC+, NFS, OpenWindows, DeskSet, ToolTalk, SunView, XView, X11/NeWS, AnswerBook, and Magnify Help are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. UNIX is a registered trademark of Novell, Inc., in the United States and other countries; X/Open Company, Ltd., is the exclusive licensor of such trademark. OPEN LOOK<sup>®</sup> 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<sup>®</sup> and Sun<sup>®</sup> 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<sup>®</sup> 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 trademark and 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 .....	vii
<b>1. Introducing the Manager.</b> .....	<b>1-1</b>
1.1 Prerequisites For Using SPARCworks 3.0.1 .....	1-1
1.2 Prerequisites For Using ProWorks 3.0.1.....	1-2
1.3 Getting Acquainted with the Tools .....	1-2
1.4 Graphical Overview.....	1-4
1.4.1 The Manager .....	1-4
1.4.2 MakeTool .....	1-5
1.4.3 SourceBrowser.....	1-5
1.4.4 Debugger .....	1-5
1.4.5 Analyzer.....	1-6
1.4.6 FileMerge.....	1-6
1.5 Roadmap to the Tools .....	1-6
<b>2. Getting Started</b> .....	<b>2-9</b>
2.1 Starting and Quitting the Manager .....	2-9

---

2.1.1	Starting the Manager . . . . .	2-9
2.1.2	Quitting the Manager . . . . .	2-10
2.2	Manager Overview . . . . .	2-11
2.2.1	Menus, Buttons, and Pop-up Windows . . . . .	2-11
2.2.2	Palettes . . . . .	2-24
<b>3.</b>	<b>Managing the Tools . . . . .</b>	<b>3-29</b>
3.1	Working with the Manager . . . . .	3-29
3.1.1	Starting a Tool . . . . .	3-29
3.1.2	Using the Tool Startup Log Window . . . . .	3-30
3.1.3	Monitoring a Session . . . . .	3-31
3.2	Modifying the Manager . . . . .	3-31
3.2.1	Changing the Properties of the Manager . . . . .	3-31
3.2.2	Changing the Properties of a Tool . . . . .	3-33
3.2.3	Saving Manager Tool Customizations . . . . .	3-34
3.2.4	Starting a Customized Version of the Manager . . . . .	3-35
3.2.5	Duplicating and Adding Tools . . . . .	3-36
3.2.6	Deleting a Tool . . . . .	3-37
3.2.7	Restoring a Deleted Tool . . . . .	3-38
3.2.8	Starting a Non-window-based Tool . . . . .	3-39
<b>4.</b>	<b>The OPEN LOOK GUI . . . . .</b>	<b>4-41</b>
4.1	OPEN LOOK GUI Basics . . . . .	4-42
4.1.1	Mouse Buttons . . . . .	4-42
4.1.2	Mouse Actions . . . . .	4-43
4.1.3	Pinned and Unpinned Menus . . . . .	4-43

---

4.1.4	Moving Windows and Menus . . . . .	4-44
4.2	OPEN LOOK Base Window . . . . .	4-44
4.2.1	Header . . . . .	4-45
4.2.2	Control Area . . . . .	4-46
4.2.3	Pane . . . . .	4-47
4.2.4	Footer . . . . .	4-49
4.2.5	Property Windows . . . . .	4-49
4.3	Resizing and Scaling a Window . . . . .	4-49
4.3.1	Resizing a Window . . . . .	4-49
4.3.2	Scaling a Window . . . . .	4-50
4.4	Getting Help in the OPEN LOOK GUI . . . . .	4-50
<b>A.</b>	<b>Troubleshooting . . . . .</b>	<b>A-51</b>
A.1	How the Manager Finds Things . . . . .	A-51
A.2	Creating Icon Files . . . . .	A-52
A.2.1	Creating Your Own Icons . . . . .	A-53
A.3	Installation Checklist . . . . .	A-54
A.4	Troubleshooting Checklist . . . . .	A-54
A.5	Reporting Problems . . . . .	A-55
A.6	Manager Messages . . . . .	A-56
<b>B.</b>	<b>Integrating an Application into the Manager . . . . .</b>	<b>B-59</b>
B.1	Messages Description Syntax . . . . .	B-59
B.2	Session Protocol . . . . .	B-60
B.3	Manager . . . . .	B-60

---

Index .....	63
-------------	----

## *Preface*

---

This manual explains how to

- Use the Manager Graphical User Interface (GUI)
- Manage the tools
- Use the OPEN LOOK<sup>®</sup> Graphical User Interface (GUI)

The Manager is an integrated component of the toolset, which also includes:

- Analyzer
- Debugger
- FileMerge
- MakeTool
- SourceBrowser

## *Before You Begin*

This manual is written for developers who want to use the SPARCworks or ProWorks tools while programming in C, C++, FORTRAN, Pascal (for SPARCworks users only), ANSI C, and Assembler. See Chapter 1, “Introducing the Manager,” for hardware and software requirements.

This manual assumes you are familiar with

- Sun<sup>®</sup> operating system commands and concepts

- 
- The OPEN LOOK<sup>®</sup> interface and the OpenWindows<sup>™</sup> environment, particularly the use of the mouse to activate a window, select text, and click on buttons.

If you are not familiar with the OPEN LOOK interface, see Chapter 4, “The OPEN LOOK GUI,” in this manual. For more information on the OpenWindows environment, see the *OpenWindows Developer’s Guide: User’s Guide*.

## Operating Environment

The SPARCworks and ProWorks toolsets run under the Solaris<sup>™</sup> 2.x operating environments. SPARCworks also runs under Solaris<sup>™</sup> 1.x. From a usage point of view, almost all of the aspects of the SPARCworks (under Solaris 2.x and 1.x) and ProWorks toolsets are the same. This includes functionality, behavior, usage, and features. For the very few details that are different, the documentation calls out those differences.

For ProWorks, Solaris 2.x for the x86 environment implies:

- Solaris 2.1 (or later) operating environment
- SunOS<sup>™</sup> 5.1 (or later) operating system
- An Intel<sup>™</sup> computer (either a server or a workstation)
- The OpenWindows 3.x application development platform

For SPARCworks, Solaris 2.x implies:

- Solaris 2.2 (or later) operating environment
- SunOS<sup>™</sup> 5.2 (or later) operating system
- A SPARC<sup>®</sup> computer (either a server or a workstation)
- The OpenWindows 3.x application development platform

For SPARCworks, Solaris 1.x implies:

- Solaris 1.2 (or later) operating environment
- SunOS 4.1.X operating system
- A SPARC computer (either a server or a workstation)
- The OpenWindows 3.x application development platform

The SunOS 5.2 operating environment is based on the System V Release 4 (SVR4) UNIX<sup>1</sup> operating system, and the ONC<sup>™</sup> family of published networking protocols and distributed services. SunOS 4.1.X is based on the UCB BSD 4.3 operating system.

---

## *How This Book Is Organized*

This manual is organized as follows:

**Chapter 1, “Introducing the Manager,”** provides background information for running the tools and an overview of each of the tools.

**Chapter 2, “Getting Started,”** presents a graphical overview of the Manager, including formats, functions, and instructions on how to start the Manager.

**Chapter 3, “Managing the Tools,”** provides a guide for using the Manager to manage tools in a session. It includes step-by-step instructions for modifying the Manager and the other tools in the toolset.

**Chapter 4, “The OPEN LOOK GUI,”** summarizes the OPEN LOOK GUI basics. It explains the OPEN LOOK conventions and describes the components, windows, and menus.

**Appendix A, “Troubleshooting,”** explains how to overcome problems that can arise and includes installation and troubleshooting checklists. It also tells how to report a problem and lists the Manager messages, their meanings, and what to do next.

**Appendix B, “Integrating an Application into the Manager,”** describes the ToolTalk<sup>®</sup> service messages which connect individual tools that are part of the programming environment.

---

1. UNIX is a registered trademark of UNIX System Laboratories, Inc.

---

## What Typographic Changes and Symbols Mean

The following table describes the typographic conventions and symbols used in this book.

Table P-1 Typographic Conventions

Typeface or Symbol	Meaning	Example
AaBbCc123	The names of commands, files, and directories; on-screen computer output	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. system% You have mail.
<b>AaBbCc123</b>	What you type, contrasted with on-screen computer output	system% <b>su</b> Password:
<i>AaBbCc123</i>	Command-line placeholder: replace with a real name or value	To delete a file, type <code>rm filename</code> .
<i>AaBbCc123</i>	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be root to do this.
◆	A single-step procedure	◆ <b>Click on the Apply button.</b>

Code samples are included in boxes and may display the following:

%	UNIX C shell prompt	system%
\$	UNIX Bourne and Korn shell prompt	system\$
#	Superuser prompt, all shells	system#

## How to Get Help

ProWorks and SPARCworks include the following on-line help facilities:

- **AnswerBook<sup>®</sup> system** displays all the tool manuals. You can read these manuals on line and take advantage of dynamically linked headings and cross-references.

To start the AnswerBook system, type: `answerbook`

- 
- **Magnify Help**<sup>™</sup> messages are a standard feature of the OpenWindows software environment. If you have a question, place the pointer on the window, menu, or menu button and press the Help key.
  - **Notices** are a standard feature of OPEN LOOK. Some notices inquire about whether or not you want to continue with an action. Others provide information about the end result of an action and appear only when the end result of the action is irreversible.
  - **Manual Pages** (man pages) provide information about the command-line utilities of the SunOS operating system. Each tool has at least one man page.

The manual pages for the tools include:

- analyzer(1)
- bcheck(1)
- dbx(1)
- dbxrc(4)
- debugger(1)
- er\_export(1)
- er\_mapgen(1)
- er\_mv(1)
- er\_print(1)
- er\_rm(1)
- filemerge(1)
- maketool(1)
- sbcleanup(1)
- sbinit(4)
- sbrowser(1)
- sbquery(1)
- sbtags(1)
- sparcworks(1) (for SPARCworks)
- proworks(1) (for ProWorks)

♦ **To access the man pages, type:** `man utility_name`

For example, to access the on-line documentation for `er_mapgen(1)`, type:

```
man er_mapgen
```

---

## *Related Documentation*

This manual is part of the tools document set. Other manuals in this set include:

- *Installing SunSoft Developer Products Software on Solaris*
- *SPARCworks/ProWorks Tutorial*
- *Browsing Source Code*
- *Building Programs with MakeTool*
- *Debugging a Program*
- *Merging Source Files*
- *Performance Tuning an Application*

You can find these and other related documents in the on-line AnswerBook system. To access the AnswerBook system, refer to *3.0.1 Installation AnswerBook*.

# Introducing the Manager

1 

This chapter provides background information for running the toolset and an overview of each of the tools.

This chapter includes the following sections:

<i>Prerequisites For Using SPARCworks 3.0.1</i>	<i>page 1-1</i>
<i>Prerequisites For Using ProWorks 3.0.1</i>	<i>page 1-2</i>
<i>Getting Acquainted with the Tools</i>	<i>page 1-2</i>
<i>Graphical Overview</i>	<i>page 1-4</i>
<i>Roadmap to the Tools</i>	<i>page 1-6</i>

## 1.1 Prerequisites For Using SPARCworks 3.0.1

To use SPARCworks 3.0.1, you must have:

- A SPARC™ computer running under the Solaris 2.x operating environment. For best performance, the system should have at least 16Mbytes of memory.
- ToolTalk running on your system  
To start ToolTalk, use the `ttsession` command.
- The 2.0.1 or 3.0.1 version of one of the following compilers running on your system:
  - SPARCompiler C (ANSI C)
  - SPARCompiler C++

- SPARCompiler FORTRAN
- SPARCompiler Pascal

The OpenWindows 3.x application development platform must be installed, but you may use any X11-based windows manager.

## 1.2 Prerequisites For Using ProWorks 3.0.1

To use ProWorks 3.0.1, you must have:

- A Intel™ computer running under the Solaris 2.x operating environment. For best performance, the system should have at least 16Mbytes of memory.
- ToolTalk running on your system. To start ToolTalk, use the `ttsession` command.
- The 2.0.1 or 3.0.1 version of one of the following compilers running on your system:
  - ProCompiler C (ANSI C)
  - ProCompiler C++
  - ProCompiler FORTRAN
  - ProCompiler Pascal

The OpenWindows 3.x application development platform must be installed, but you may use any X11-based windows manager.

## 1.3 Getting Acquainted with the Tools

SPARCworks is a set of programming tools for use with SPARCompiler™ products. ProWorks is a set of programming tools for use with ProCompiler™ products. The tools are

- **Manager** — manages and coordinates other tools in the toolset.
- **Analyzer** — tunes program performance, including memory allocation. (This tool available only on SPARCworks running Solaris 2.x)
- **Debugger** — dynamically analyzes programs, observes behavior while the program runs, and collects information for performance profiling.
- **FileMerge** — merges source files and coordinates source code changes with other developers.

- **MakeTool** — builds programs and browses makefiles.
- **SourceBrowser** — statically analyzes source code and program structure.

These tools speed and simplify the tasks software developers do most often—edit and merge source code, compile programs, debug programs, and tune program performance.

These tools form part of an overall development environment that includes SunOS programming utilities, the DeskSet™ productivity tools (including File Manager and Mail Tool), and the OpenWindows Developer's Guide for interactively building OPEN LOOK graphical interfaces.

Many SPARCworks and ProWorks functions are available from the terminal-mode command line, a fact important to users without access to a window environment. For information on the command-line interfaces of these tools, see the respective manual for each tool as listed in the “Preface” of this book.

This manual discusses the Manager, the OPEN LOOK window interface to the tools. There is no available command line interface to the Manager.

---

**Note** – The figures in this manual that show the Manager palette are representative of a *possible* palette. The icons displayed on your palette are dependent on the products you install. The order in which the icons are displayed is dependent on the order in which the tools are installed. For a representative icon, refer to the documentation for each product.

---

Figure 1-1 shows the Manager and the tool icons.

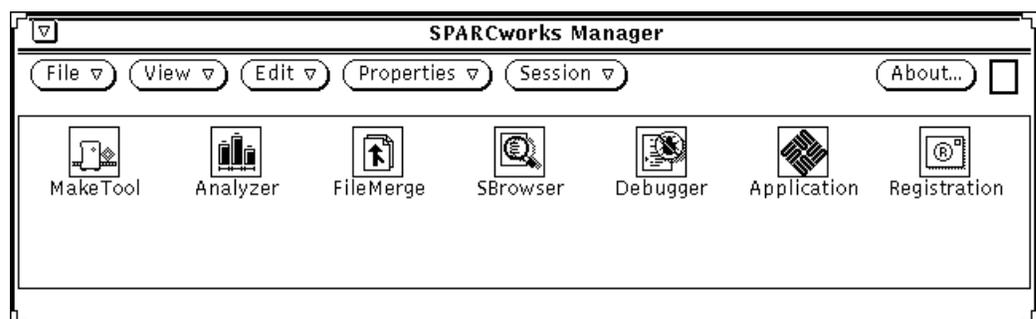


Figure 1-1 Manager Window Display

## 1.4 Graphical Overview

The tools make the development of application software easier with an integrated toolset. This section gives a brief overview of the functionality of the standard toolset. For more information on these tools, see the manuals for each tool.

### 1.4.1 The Manager

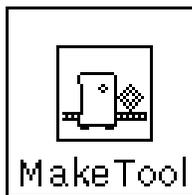


The Manager provides an easy-to-use platform for starting the tools and controlling the programming environment. Each tool can be started by double-clicking on its icon or by dragging the icon onto the screen workspace. A tool that is started from the Manager is thereafter associated with that instance of the Manager, and can be controlled as part of a Manager *session*.

In addition to managing the other standard tools, the Manager can also control custom applications written by users or third-party developers. When integrated with the tools and executed under the Manager, the standard tools and custom applications combine to provide you with the following advantages:

- **Drag-and-drop execution** — You can drag an on-screen DeskSet object (a file icon, for example) with the mouse pointer and drop it on one of the tools, which recognizes whether or not the object can be loaded or otherwise acted upon.
- **Session control** — You can group development activities into logical Manager *sessions*. Multiple sessions can proceed simultaneously, each controlled by its own Manager.
- **Workspace organization** — the Manager helps you make the best use of on-screen workspace. The tools associated with a Manager session can be closed to icons.
- **Central control** — the Manager can set environment variables and the working directory for all tools in the toolset used in a session.
- **Integration of custom applications** — You can integrate custom applications into the Manager. For information on how this is done, see *Appendix B*.

### 1.4.2 MakeTool



MakeTool is an OPEN LOOK interface to `make`, the UNIX utility for overseeing program compilation which ensures that programs are built from the newest sources. In addition, MakeTool contains a browser that helps you interpret makefiles by expanding macros and rules. For more information on MakeTool, see *Building Programs with MakeTool*.

### 1.4.3 SourceBrowser



SourceBrowser is an interactive window-based tool for analyzing source code. This tool enables you to easily find all occurrences of an identifier or string in source code. SourceBrowser maintains a list of all queries, making it easy to return to a previous query to compare results. In addition, SourceBrowser offers powerful features for customization of browsing. For example, you can restrict queries to search for symbols based on how they are used in the source.

SourceBrowser consists of the following four tools:

- **SourceBrowser** — enables you to analyze, query and edit your source code.
- **ClassBrowser** — enables you to browse C++ class definitions. You can view the base and derived classes, their data members and member functions, as well as class-defined interfaces and relationships.
- **CallGrapher** — enables you to graphically inspect the interrelationships of the functions in your program.
- **ClassGrapher** — enables you to visualize relationships among the classes of a C++ program.

SourceBrowser is integrated with Debugger and SCCS. For more information on SourceBrowser, see *Browsing Source Code*.

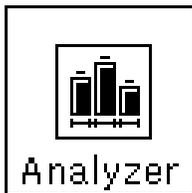
### 1.4.4 Debugger



Debugger is a sophisticated window-based tool for debugging programs and significantly reduces time in the debug-edit-compile cycle. Debugger offers a program editing facility so you need not change tools continuously.

The user-configurable graphical interface provides visual feedback and mouse control for most debugging operations. It also offers OPEN LOOK drag-and-drop support for integration with other desktop applications. For more information on Debugger, see *Debugging a Program*.

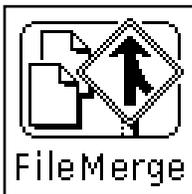
### 1.4.5 Analyzer



Analyzer is part of a set of performance analysis tools for measuring, recording, understanding, and improving the performance of an application program. Analyzer provides an easy-to-use graphical user interface for specifying and displaying data collected on a target application.

Analyzer can be used by all software developers, regardless of whether performance tuning is the developer's main responsibility. For more information on Analyzer, see *Performance Tuning an Application*.

### 1.4.6 FileMerge



FileMerge is an interactive window-based tool for comparing files and merging their differences. It displays two files side by side in read-only text comparison windows. Beneath the comparison windows is an editing subwindow that contains a merged version of the two files. The merged version contains selected lines from either or both files. You can edit this version to produce a final merged version of the two original files. For more information on FileMerge, see *Merging Source Files*.

## 1.5 Roadmap to the Tools

The Roadmap is a visual overview of the toolset. It maps out a logical path for using the tools in a software development cycle.

The toolset is used to develop, test, and analyze software systems. Each tool is individually significant with the combination of tools enhancing the productivity of the collective set. The *roadmap* shown in Figure 1-2 illustrates the use of the tools in development. It shows the interrelationship between the tools and how they might be used in a workgroup environment.

Typically, on a project, each developer works on a set of files with a standard editor. When certain milestones are met, the developer makes the files using MakeTool. After building the application, the developer browses the source

code with SourceBrowser, fixes any bugs with the Debugger, and analyzes the performance of the code with the Analyzer. The cycle can then be repeated after editing the code to meet the next milestone.

Once the individual files are completed by each developer, they are merged with the help of FileMerge and can be accessed by the entire team. Each of the other team members can access the newly merged source files and begin the edit-make-browse-debug-analyze cycle.

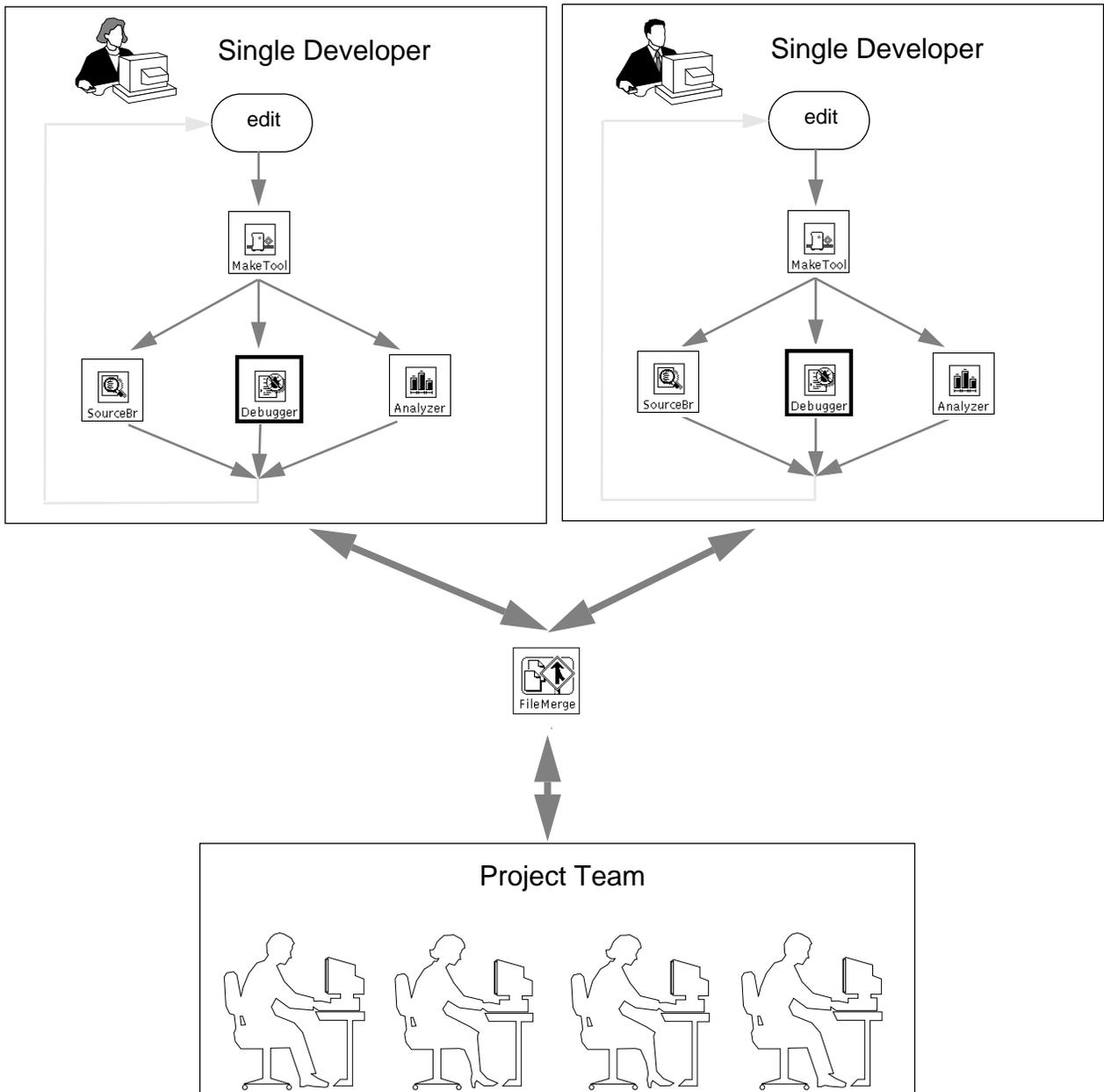


Figure 1-2 Roadmap

# Getting Started



This chapter is an introduction to the Manager that acquaints you with the Manager formats and functions.

This chapter includes the following sections:

<i>Starting and Quitting the Manager</i>	<i>page 2-9</i>
<i>Manager Overview</i>	<i>page 2-11</i>

## 2.1 Starting and Quitting the Manager

As you work with the toolset, we suggest that you start all tools for a particular project from one Manager. Doing so unifies control of opening, closing, hiding, and showing tools during a session.

### 2.1.1 Starting the Manager

To start the Manager:

◆ **For SPARCworks, type the following command at a prompt:**

`sparcworks &`

◆ **For ProWorks, type the following command at a prompt:**

`proworks &`

The Manager palette is displayed, as shown in Figure 2-1.

The Manager displays the tool icons on the palette based on one or more configuration files. The search order for configuration files is:

1. the command line
2. \$HOME/.sunpro\_defaults
3. /lib/sunpro\_defaults in the Manager installation directory

For a complete discussion of the use of the command line to specify icons, see “Saving Manager Tool Customizations” on page 3-34 and “Starting a Customized Version of the Manager” on page 3-35.

---

**Note** - The figures in this manual that show the Manager palette are representative of a *possible* palette. The icons displayed on *your* palette are dependent on the products you install. The order in which the icons are displayed is dependent on the order in which the tools are installed. Refer to the documentation for each product for a representative icon.

---

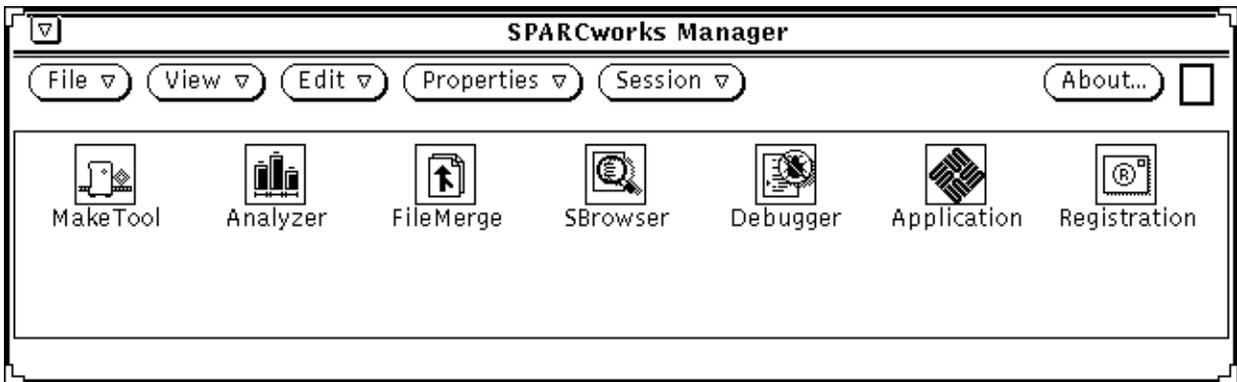


Figure 2-1 Manager Main Window

### 2.1.2 Quitting the Manager

You can quit the Manager any time through the standard OPEN LOOK window menu. If tools are still running when you quit the Manager, a prompt asks if you still want to quit. This feature is helpful when tools may be *hidden* at the time you quit the Manager.

To quit the Manager:

- ◆ **Choose Quit from the Window menu. (Press MENU on the menu button in the upper left corner of the window to display the Window menu).**

You can also terminate a session from a shell by sending the Manager a SIGUSR1 signal. Terminating the Manager in this way has the same effect as choosing Quit from the Window menu.

## 2.2 *Manager Overview*

In this section you will learn about:

- Menus, Buttons, and pop-up windows
- Using the Manager pop-up menu
- Manager formats

### 2.2.1 *Menus, Buttons, and Pop-up Windows*

The function buttons in the header of the default Manager window are:

- **File Menu** — Saves the current configuration
- **View Menu** — Displays the Compact Palette and the Tool Startup Log window
- **Edit Menu** — Deletes tools, duplicates tools, and realigns icons on the Full Palette
- **Properties Menu** — Sets the properties of a selected tool or the Manager
- **Session Menu** — Manages a tool session
- **About Button** — Displays version number, tool description, and Comments button (opens window for sending comments)
- **Drop Target** — Attaches an application to the Application icon in the Manager palette

### *File Menu*



The File menu allows you to save the current configuration of the Manager and individual tools.

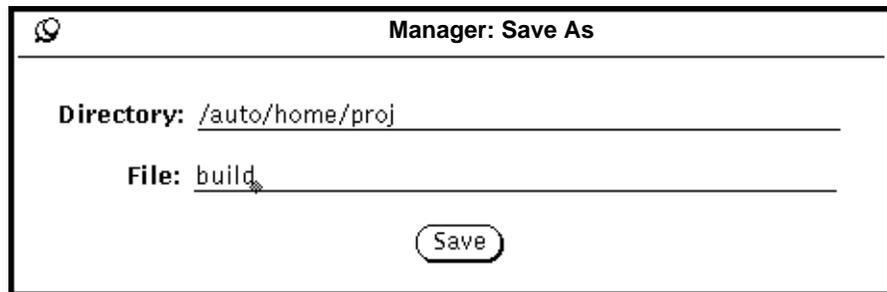
The menu displays the following options:

#### **Save**

Stores the current Manager configuration under the current file name.

#### **Save As**

Displays the Save As worksheet. The worksheet lets you specify the location in which to store the configuration file. For more on saving tool customizations, see “Saving Manager Tool Customizations” on page 3-34



## View Menu



The View menu lets you display the Compact Palette and a log window for tool startup processes. The menu displays the following options:

### Compact Palette

Changes the default Manager display to the Compact Palette format, as shown in Figure 2-2. The Compact Palette displays only the tool icons and their labels; Manager buttons and controls are no longer displayed. For instructions on switching between views, see “Palettes” on page 2-24.



Figure 2-2 Compact Palette Display

### Tool Startup Log

Displays the Tool Startup Log window, as shown in Figure 2-3. The Log window displays a list of activated tool processes. It also displays the computed values of certain environment variables. See “Using the Tool Startup Log Window” on page 3-30.

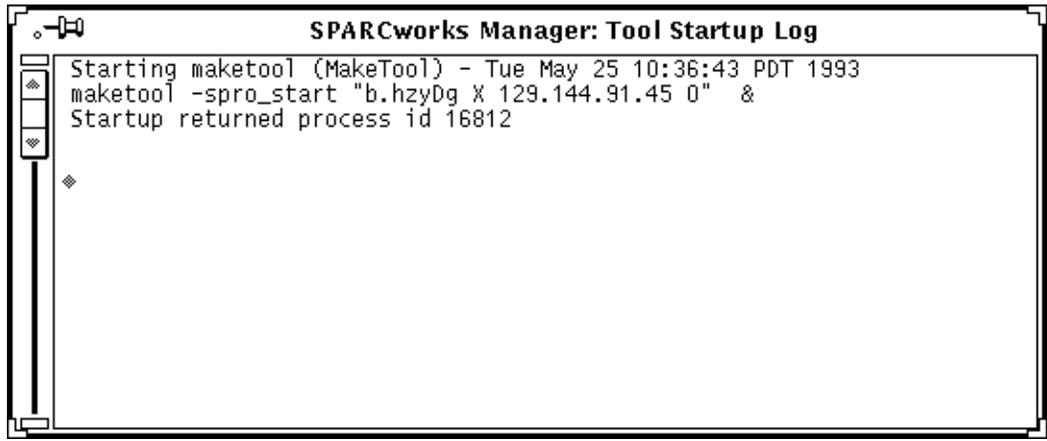


Figure 2-3 Tool Startup Log Window

### Edit Menu



The Edit menu lets you duplicate and delete the tools on the palette, as well as automatically align the tool icons. The menu displays the following options:

#### Delete Tool

Deletes a selected tool from the palette. Deleted tools are listed at the bottom of the Edit button menu and can be restored by selecting them from the menu.

#### Duplicate Tool

Makes a copy of a selected tool. The duplicated tool has all the properties and functionality of the original with the exception of a unique label. The properties of a duplicated tool can be modified through the use of the Tool Properties sheet. The customized properties can be saved under a new label for future use.

### ***Snap Icons to Grid***

Automatically realigns icons to an invisible grid at the top of the window. The option is particularly helpful when a significant number of tool icons are displayed or after you rearrange the icons on the Manager palette.

### ***Properties Menu***



The Properties menu lets you customize the properties of a selected tool or the Manager itself. The menu displays the following options:

### ***Selected Tool***

Displays a property sheet for the selected tool so you can customize the tool, its label, and its Manager icon. Figure 2-4 on page 15 shows an example of the Tool Icon and Startup Properties window. The Tool Properties window allows you to customize a tool configuration file and create a unique version of a tool.

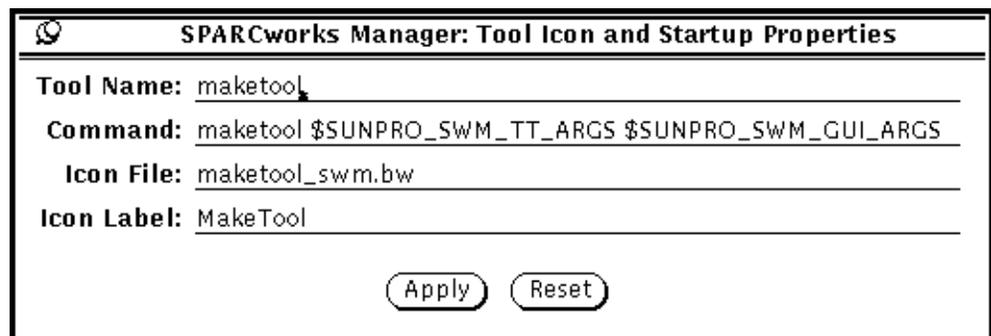


Figure 2-4 Tool Icon and Startup Properties Window

The properties are indexed by the name of the tool. Beneath the Tool Name entry is the command line executed to start the tool. You can edit the command line and specify different commands for a tool, thus creating a customized version of a tool that can be saved under a unique label name.

### ***Internal Variables***

The Manager uses the following internal variables in its tool properties. Although the naming convention is similar to that of UNIX environment variables, these are not environment variables. You should not try to change their values.

`$SUNPRO_SWM_TT_ARGS`

Special command line argument that all the tools in the toolset use so that the Manager session commands are recognized. This variable is set by the Manager. Applications that support the session management protocol must specify this variable as the first command-line argument.

`$SUNPRO_SWM_GUI_ARGS`

Command line arguments that specify the initial position of windows when drag and drop is used. If double click is used to open a window, then this value is set to zero. This variable is set each time a tool is started.

`$SUNPRO_SWM_APP_DIR`

Variable that is set when a pathname or a FileManager icon is dropped on the Manager Drop Target. The current working directory is changed to the directory component of the path name and the `$SUNPRO_SWM_APP_DIR` environment variable is updated. Each time a tool is started from the Manager its current working directory variable is set to `$SUNPRO_SWM_APP_DIR`.

`$SUNPRO_SWM_APP_FILE`

Variable that is set when a pathname or FileManager icon is dropped on the Manager Drop Target. The `$SUNPRO_SWM_APP_FILE` is set to the file component of the path. The tool icon labeled Application uses `$SUNPRO_SWM_APP_FILE` as its command name.

`$SUNPRO_SWM_DIRECTORY`

Directory path that is used to locate tools relative to the installed location of the Manager. This path can be used to help ensure that the version of the tools match that of the Manager you're using.

---

**Note** – For information on environment variables, see the operating system documentation for the `sh` and `setenv` commands.

---

### **All Tools**

Displays a properties window for the current Manager. From the properties window, you can:

- Change the working directory for starting the tools
- Edit environment variable values

The following is a sample of the Manager Properties window:

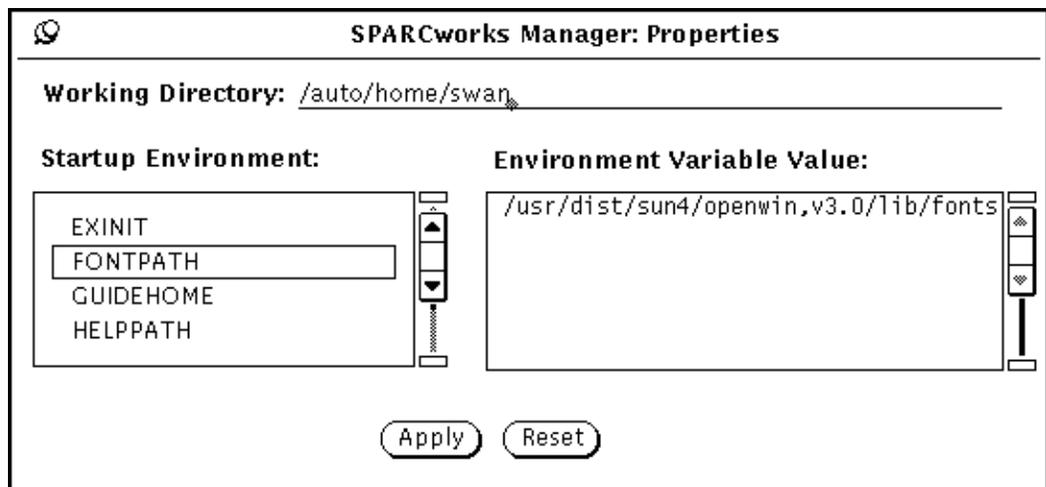


Figure 2-5 Manager Properties Window

### Session Menu



The Session menu lets you manage the display of tools that are running from its Manager. The menu displays the following options:

#### **Open**

Simultaneously opens all tools that have been started from the current Manager and closed to icons. If none of the running tools has been closed to icons, then Open has no effect.

#### **Close**

Simultaneously closes all tools that have been started from the current Manager. The closed tools are represented as icons.

The Open and Close features do not have any impact on tools that are hidden.

#### **Hide**

Simultaneously unmaps, or *hides*, all tools started from the current Manager. All tools of the current Manager session are hidden from display.

#### **Show**

Simultaneously maps, or *shows*, all hidden tools started from the current Manager. *Showing* tools reverses the effect of *hiding* tools. The next two examples illustrate the usefulness of the Hide and Show options. In Figure 2-6, two Manager sessions are showing, before the user selects the Hide option; in Figure 2-7, the selected session is hidden.

---

**Note** – User-defined application tools may not support the ToolTalk protocol and therefore operate differently from tools under the Manager. The main difference is that the Session options are not supported.

---

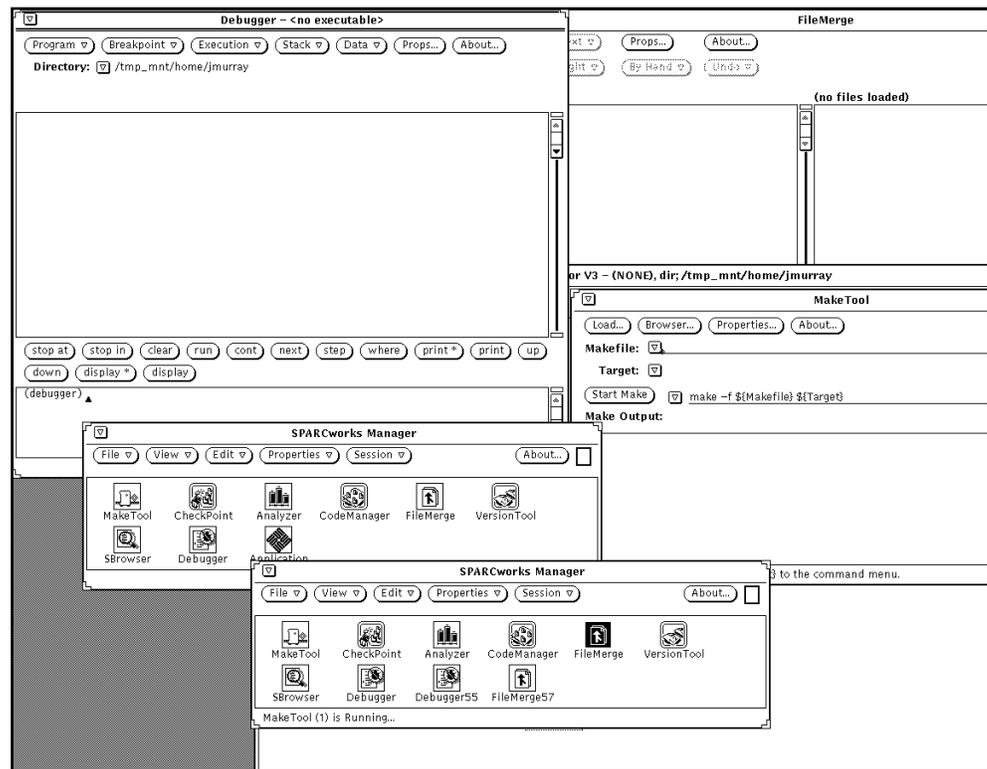


Figure 2-6 Multiple Managers and Multiple Open Tools

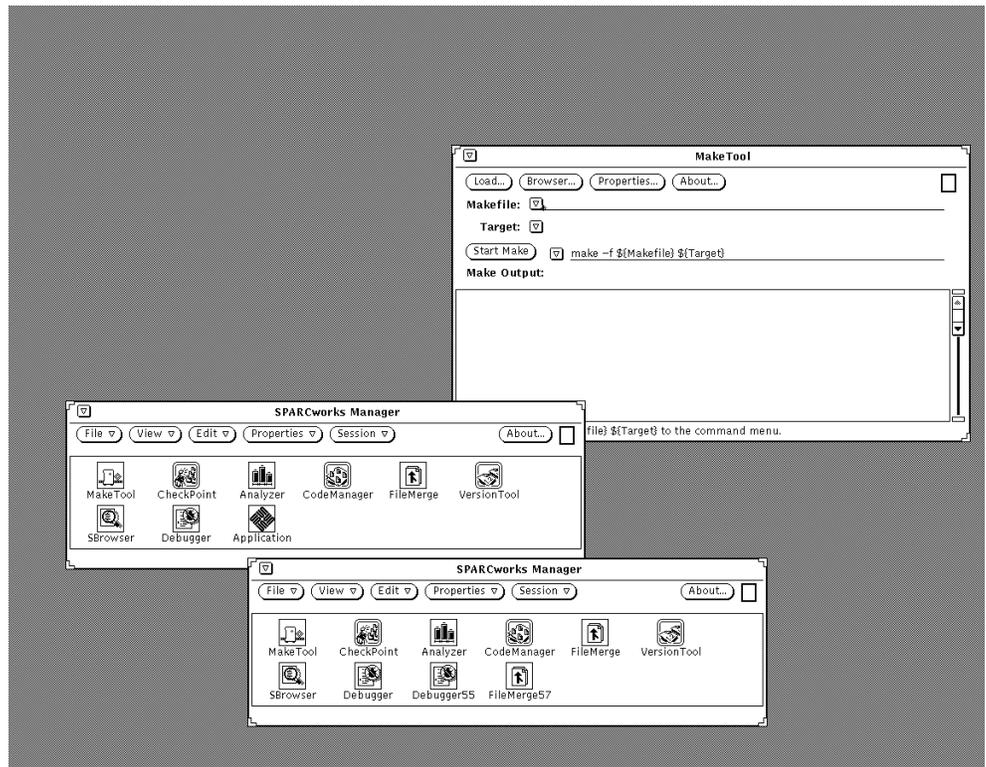
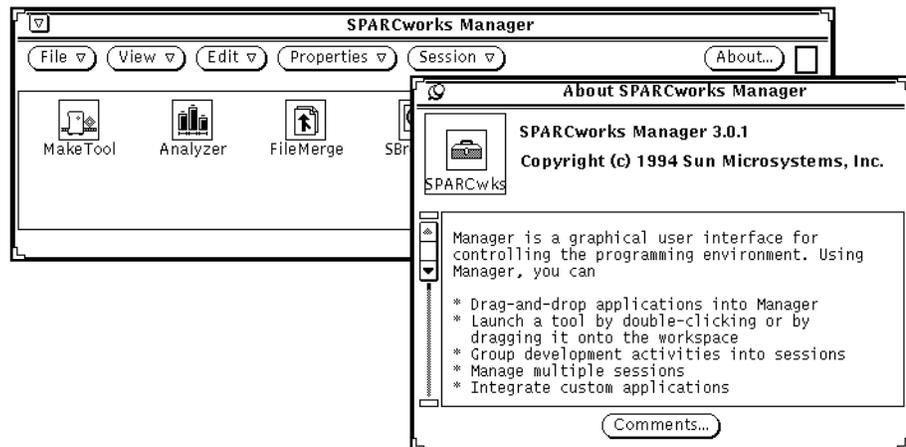


Figure 2-7 Tools of One Manager Hidden

## About Button

The About button displays a pop-up window that gives the version number, copyright, and a brief description of the Manager. Clicking on the Comments button displays another window from which you can send comments about the tool to SunSoft.



**Note** – The About box for ProWorks is identical to the SPARCworks version shown the previous figure except for the product name.

## Drop Target

The Drop Target provides drag-and-drop capabilities for attaching an application to the Application icon in the Manager palette. This sets the current directory of the Manager to the directory of the application, providing a focus for future invocations of tools. It also sets the environment variable `$SUNPRO_SWM_APP_FILE`, which can be referenced in the command line field of other tools. For example, setting the Command field of the Tool Icon and Startup Properties window for the Debugger to

```
debugger $SUNPRO_SWM_TT_ARGS $SUNPRO_SWM_GUI_ARGS \  
$SUNPRO_SWM_APP_FILE
```

makes the next invocation of the Debugger start with the current application as the program to debug.

The Drop Target is located in the upper right corner of the Manager control area as shown in Figure 2-8 on page 22.

To attach an application to the Application icon, do one of the following:

- ◆ **Drag and drop a File Manager icon onto the Drop Target.**
- ◆ **Select the path name of the application in a Command Tool window, then drag and drop it onto the Drop Target.**

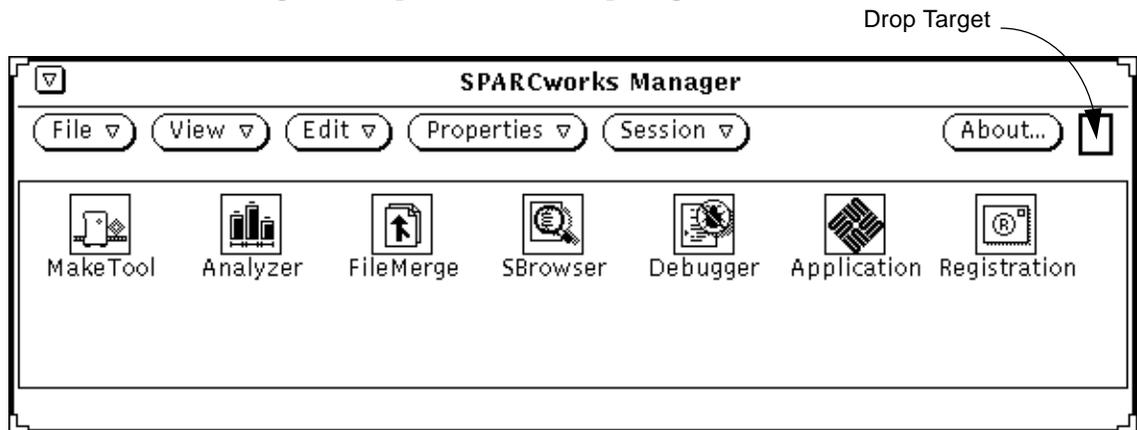


Figure 2-8 Drop Target

Once you drop a File Manager icon or path name onto the Drop Target, the current working directory changes to the directory component of the path name. The path name of the application is displayed in the message area at the bottom of the Manager window, and for the remainder of that session only is represented by the generic Application icon.

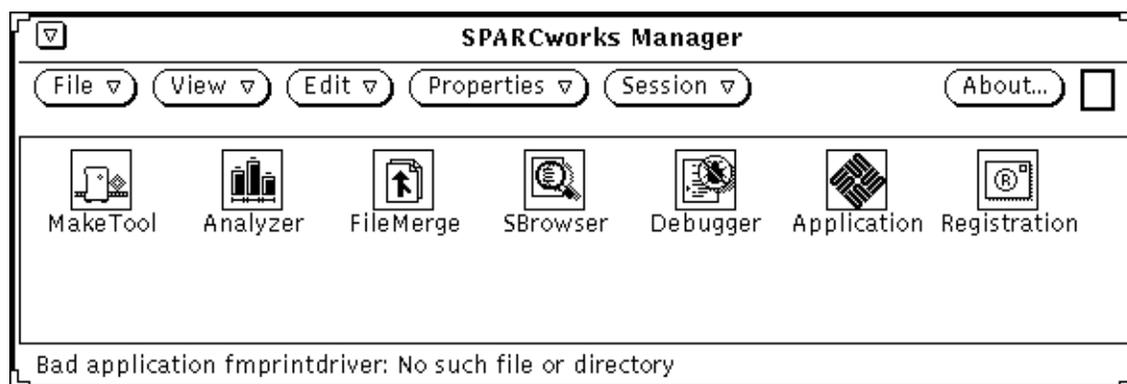


Figure 2-9 Drop Target Error Message

To start the application you dropped on the Drop Target do one of the following:

- ◆ **Double-click on the Application icon.**
- ◆ **Drag the Application icon onto the workspace.**

---

**Note** – Dropping a path name onto the Drop Target that is not the path of an application results in an error message, as shown in Figure 2-9.

---

### *Pop-up Menu*

The Manager pop-up menu can be accessed at all times. The pop-up menu offers the same options as the Session menu and the View menu combined. For more information on the functionality of the Session and View menus, see “Session Menu” on page 2-18 and “View Menu” on page 2-13.

To display the Manager pop-up menu:

- ◆ **Move the pointer into the window’s background area and press MENU.**  
The pop-up menu is displayed as shown in Figure 2-10 on page 24.

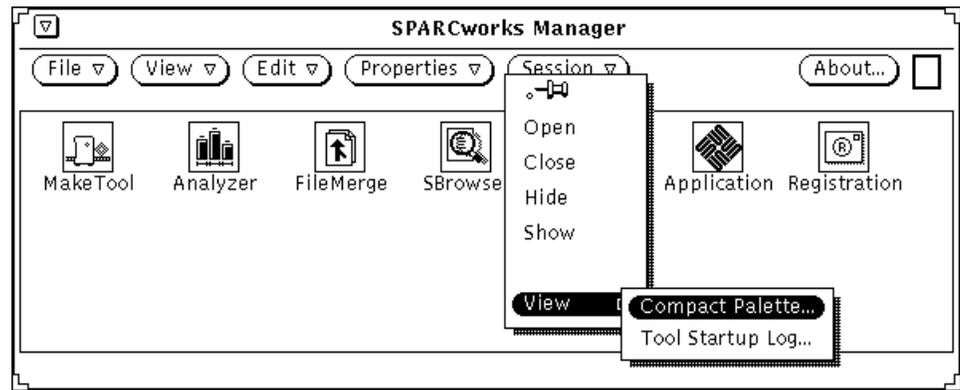


Figure 2-10 Manager Pop-up Menu

### 2.2.2 Palettes

The Manager offers the following display formats:

- Full Palette
- Compact Palette

The Full Palette displays OPEN LOOK standard control buttons (as in the figure above). The control buttons let you change the palette display, duplicate and delete tools, change the properties of the Manager and individual tools, and monitor a session.

The Compact Palette provides condensed organization for displaying the tool icons and labels. It allows easy access to the tools with a minimum workspace requirement.

## Viewing the Compact Palette

To display the Compact Palette:



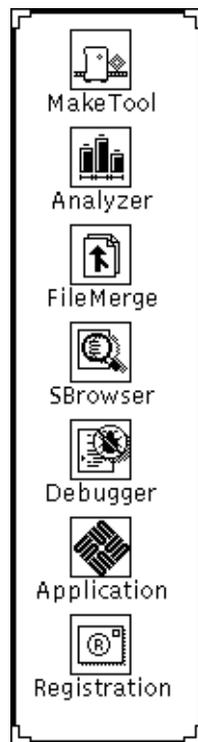
♦ **Choose Compact Palette from the View menu.**

The Compact Palette is displayed, as shown in Figure 2-11.



Figure 2-11 Compact Palette Display

### *Resizing the Compact Palette*



You can resize the Compact Palette display by dragging a resize corner to create the desired view.

Figure 2-12 Vertical Compact Palette

### *Redisplaying the Full Palette*

To redisplay the Manager window with controls and buttons:

- ◆ **Choose View from the pop-up menu and drag right to select Full Palette from the pull-right menu. (See Figure 2-13 on page 2-27).**  
 If the Compact Palette was resized to a vertical display, select Snap Icons to Grid from the Edit menu. This realigns the icons to a horizontal format.

You can also double-click on the background of the palette to switch between Compact and Full Palette display.

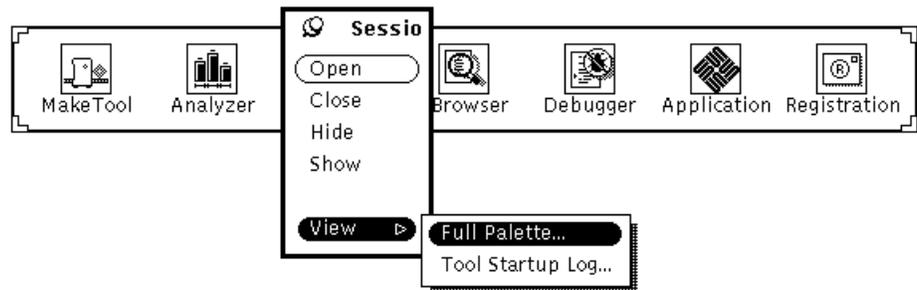


Figure 2-13 Redisplaying the Full Palette

### *Aligning Tool Icons*

The Manager provides an option for quickly realigning tool icons. Snap Icons To Grid is particularly helpful when you have a significant number of tool icons displayed in a palette. For instance, you can rearrange the icons so the tools you use most often are on the top line, then use Snap Icons to Grid to realign the icons in a linear format according to their new positions.

Snap Icons to Grid is recommended if you downsize to the Compact Palette, reshape the palette, and then return to the Full Palette format. Snap Icons to Grid realigns the icons to a horizontal format.

The Snap Icons to Grid option is located under the Edit button. To realign moved icons:



◆ **Choose Snap Icons to Grid from the Edit menu.**

The icons are automatically realigned in a horizontal format along the top of the window.

Figure 2-14 and Figure 2-15 show the Manager before and after Snap Icons to Grid was used.

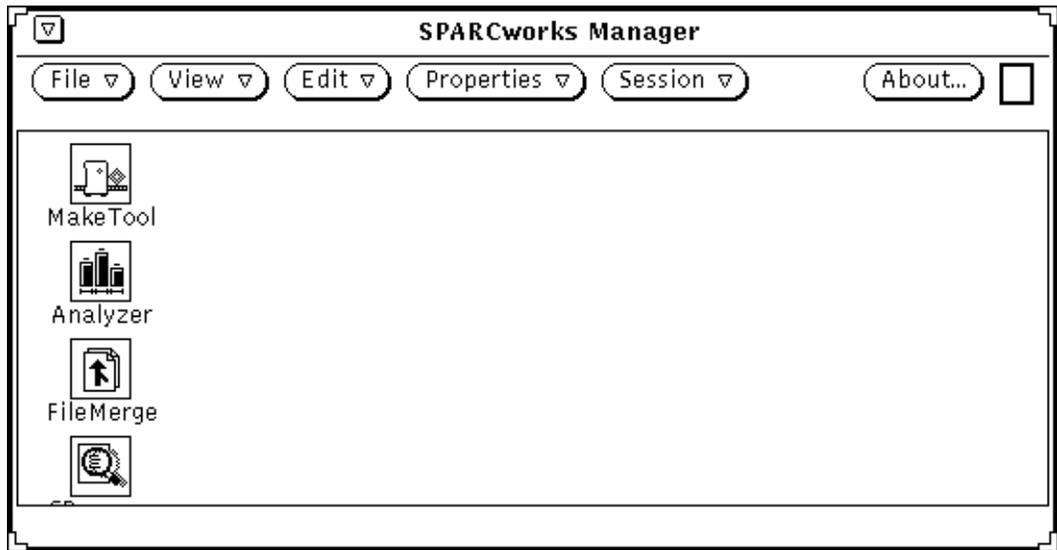


Figure 2-14 Before Using Snap Icons to Grid

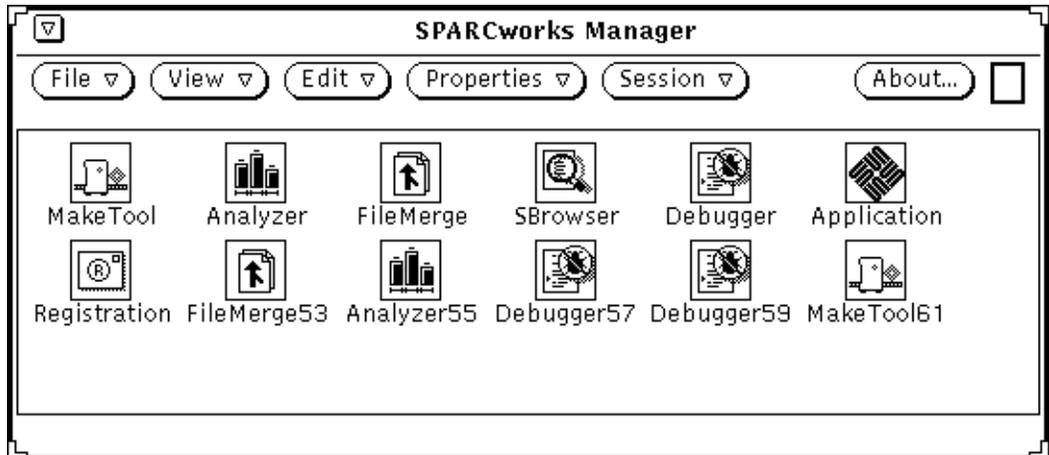


Figure 2-15 After Using Snap Icons to Grid

## Managing the Tools

3 

This chapter is a guide for using the Manager to manage tools in a session. It includes step-by-step instructions for modifying the Manager and the tools.

This chapter includes the following sections:

<i>Working with the Manager</i>	<i>page 3-29</i>
<i>Modifying the Manager</i>	<i>page 3-31</i>

### 3.1 Working with the Manager

This section discusses the basics of working with the Manager. You will learn how to

- Start a tool
- Use the Tool Startup Log window
- Monitor a session

#### 3.1.1 Starting a Tool

Using the Manager, programmers can start all tools for a particular project from one Manager. Doing so unifies control of opening, closing, hiding, and showing tools during a programming session. More than one Manager can be active at one time. To start a tool, do one of the following:

- ♦ **Drag the tool icon onto the workspace.**

- ◆ Double-click on the tool icon.

### 3.1.2 Using the Tool Startup Log Window

The Tool Startup Log window displays a log of startup processes for tools. It also displays the command used to start the tool, including the `$SUNPRO_SWM_TT_ARGS` and `$SUNPRO_SWM_GUI_ARGS` environment variables if they are specified in the command line. The values for these variables are computed each time the tool is started. The log entries for these environment variables can be important when debugging a Manager command if the tool did not start as you expected.

To bring up the Log Window:

- ◆ **Choose Tool Startup Log from the View menu.**  
Once the Log window is displayed, startup messages appear for tools that are launched. In Figure 3-1, the startup messages for MakeTool are displayed.

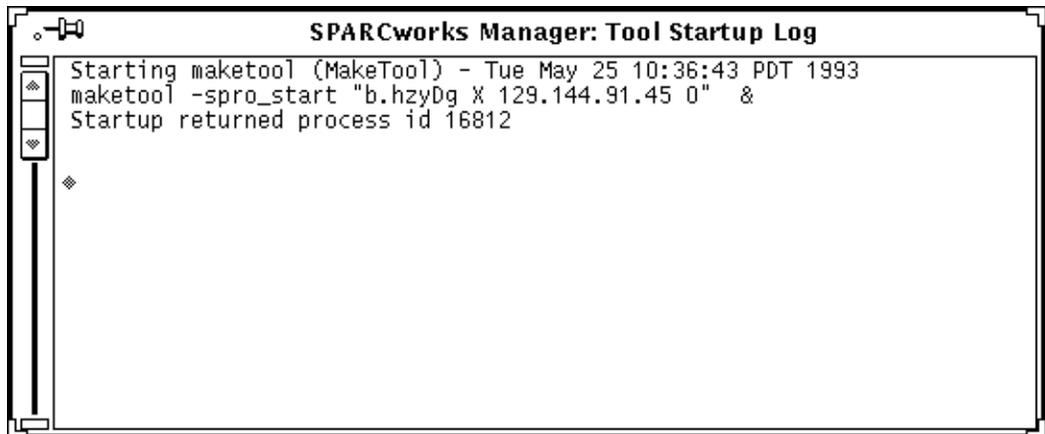


Figure 3-1 Tool Startup Log Window

### 3.1.3 Monitoring a Session



You can monitor a session using the Session options, which offer the ability to open, close, hide, or show all the tools associated with the Manager. This functionality is particularly helpful when working with multiple Managers.

## 3.2 Modifying the Manager

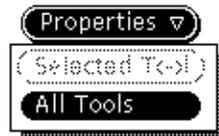
This section discusses methods for modifying the Manager and the tools. You will learn how to

- Modify the Manager
- Save Manager customizations
- Rename and recall a customized Manager
- Modify individual tools
- Duplicate tools
- Delete tools and restore deleted tools
- Add tools
- Start up a non-window-based tool by linking an editor to the Editor icon

### 3.2.1 Changing the Properties of the Manager

You can change the properties of the Manager such as the working directory of the Manager and the directory that the tools start from without quitting the Manager.

To change the properties of the Manager:



1. Choose All Tools from the Properties menu. Figure 3-2 shows the Properties window.

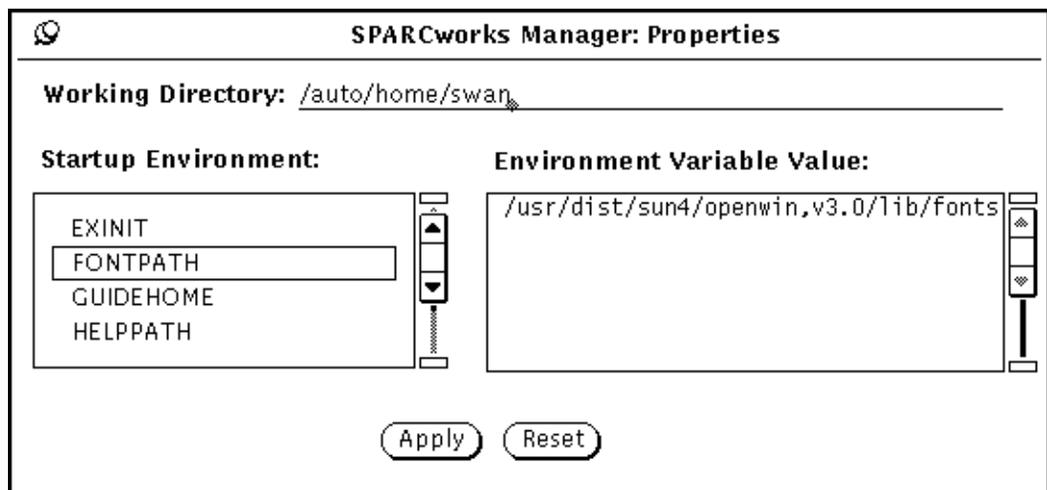


Figure 3-2 Manager Properties Window

2. To change the working directory of the Manager, change the Working Directory path name at the top of the property sheet. The Working Directory is the directory in which the tools are started.
3. To change any of the environment variables, select the desired variable from the Startup Environment column. You can then alter the value displayed for the variable in the Environment Variable Value column.
4. Set the changes by clicking on Apply. You can reset the properties to the last applied state by clicking on Reset.

---

**Note** – When you save the customized Manager, you actually save the properties of each tool, not the working directory or the startup environment. Changes to the working directory and the startup environment are only in effect during the current session. To save the Manager customizations permanently, see Section 3.2.3, “Saving Manager Tool Customizations,” on page 3-34.

---

### 3.2.2 *Changing the Properties of a Tool*

You can change the properties of the tools, customizing them to your personal specifications.

To customize tool properties:



- ♦ **Select the tool you want to customize from the palette and choose Selected Tool from the Properties menu.**

A Tool Properties window is displayed in Figure 3-3 on page 34. The properties are indexed according to the name of the tool. Each Tool Properties window contains the following fields:

**Tool Name** — name of the tool. This name is used to distinguish this tool’s properties in the configuration file.

**Command** — command line used to start the tool.

**Icon File** — file name for the icon glyph of the tool displayed in the palette.

**Icon Label** — name displayed on the icon in the palette.

To change the properties of a tool:

- ♦ **Type your changes in the fields and set the changes by clicking on Apply.** You can reset the properties to their last applied state by clicking on Reset.

Once you have customized a toolset, you can save the customizations permanently (see Section 3.2.3, “Saving Manager Tool Customizations,” on page 3-34).

**Note** – For information on environment variables, see the operating system documentation for the `sh` and `setenv` commands.

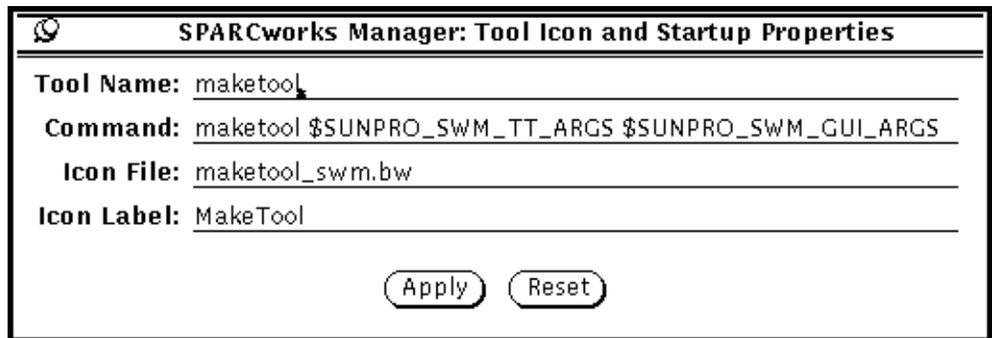


Figure 3-3 Tool Icon and Startup Properties Window

### 3.2.3 Saving Manager Tool Customizations

Once you configure the Manager tools, you may want to save the customizations to a new configuration file.

To save Manager customizations in a configuration file, follow these steps:



1. Choose **Save As** from the **File** menu.  
Figure 3-4 shows the Save As worksheet.

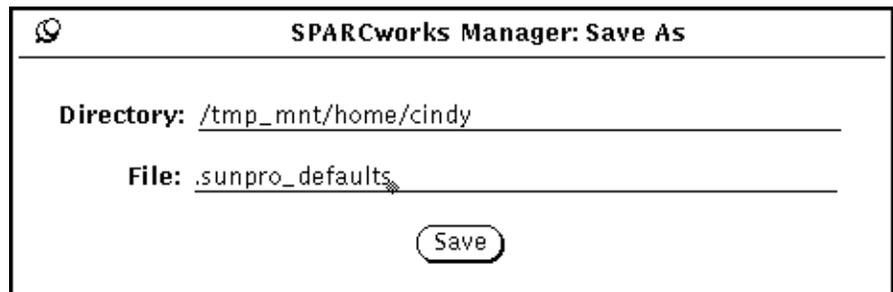


Figure 3-4 Worksheet for the Save As Option

2. Enter the directory for the new configuration file. The default directory is the current working directory of the Manager.
3. Enter the file name for the new configuration file.
4. Click on Save.  
The tool properties of all the tools in the palette are saved in the configuration file.

---

**Note** – Since the Manager looks for a configuration file named `.sunpro_defaults` in your home directory, it is recommended that you use this default file name and directory when saving.

---

### 3.2.4 Starting a Customized Version of the Manager

It is easy to call up a customized version of the Manager once you have created the customized file and saved it to a unique file name.

To start a customized version of the Manager:

- ♦ For SPARCworks, type `sparcworks` followed by the configuration file name, as follows:

```
sparcworks filename1 &
```

- ◆ **For ProWorks, type `proworks` followed by the configuration file name, as follows:**

```
proworks filename1 &
```

You can specify multiple file names separated by spaces. This allows you to create a palette from multiple saved configuration files. Without the file-name specification, the `.sunpro_defaults` configuration file in your home directory is used. If that file is not found, the default Manager configuration is used.

To create a palette that is a composite of multiple configuration files:

- ◆ **For SPARCworks, type `sparcworks` followed by multiple file names separated by spaces, as follows:**

```
sparcworks filename1 filename2 filename3 &
```

- ◆ **For ProWorks, type `proworks` followed by multiple file names separated by spaces, as follows:**

```
proworks filename1 filename2 filename3 &
```

### ***3.2.5 Duplicating and Adding Tools***

The Manager provides a fast and easy method for duplicating and adding tools. Once a tool is duplicated, you can modify the properties of one and have two different versions. For information on how to customize tools, see “Changing the Properties of a Tool” on page 3-33.

#### ***Duplicating a Tool***

For example, if you frequently debug programs A and B, you might want to create two tools called `debug_A` and `debug_B`, which automatically load programs A and B into the Debugger.

To duplicate a tool:



- ◆ **Select the tool you want to duplicate from the palette. Choose Duplicate from the Edit menu.**

---

**Note** – The duplicated tool has all the properties and functionality of the original tool with the exception of a unique label.

---

When you choose a tool for duplication, the icon display highlights by changing to reverse video. The duplication process automatically gives the new tool a unique label by adding a number to the end of the original tool name (for example, *Debugger37*).

Use the Selected Tool option under the Properties menu to give the modified tool a new label. For more information, see “Changing the Properties of a Tool” on page 3-33. Once you have created a tool through duplication, you can use it for the duration of a session. If you want to keep the tool permanently, you must save the Manager configuration (see “Saving Manager Tool Customizations” on page 3-34).

### *Adding a Tool*

To add a tool to the Manager, you use a combination of processes covered earlier in this section. First, duplicate an existing tool and then modify the properties. If you want to permanently add the tool to the Manager Palette, then you must save the configuration. For more information on these processes, see the following sections:

- “Changing the Properties of a Tool” on page 3-33
- “Saving Manager Tool Customizations” on page 3-34

### *3.2.6 Deleting a Tool*

You can delete any tool on the tool panel as follows:



- ◆ **Select the tool you want to delete. Choose Delete from the Edit menu.**  
The selected tool is automatically deleted from the palette.

### 3.2.7 Restoring a Deleted Tool

The Manager allows you to restore deleted tools at any time during a session. A record of deleted tools is displayed at the end of the Edit menu. Any of the listed tools can be restored until the time you quit the Manager. If you quit the Manager after deleting a standard tool, you can restore the tool by restarting the Manager from the default configuration file.

To restore a deleted tool:



- ◆ **Press MENU on the Edit button.**  
The pull-down menu lists the tools deleted during the current session. Choose the tool you want restored.

### 3.2.8 Starting a Non-window-based Tool

You are not limited to using only window-based tools with the Manager. It is also possible to start up non-window-based tools by following the instructions below.

The following example illustrates how you can link a tool icon to a script that starts up a shell tool and brings up `vi`. A similar process can be used to start other non-window-based tools. Adding a tool makes it launchable from the Manager Palette. If you want the application to be able to respond to the session control protocol, then it must be modified first. Refer to “Session Protocol” on page B-60.

To link a tool icon to a script invoking `vi`:



1. Choose Selected Tool from the Properties menu.

Modify the text fields in the Tool Icon and Startup Properties Window as follows:

SPARCworks Manager: Tool Icon and Startup Properties	
Tool Name:	<code>vi</code>
Command:	<code>cmdtool -I vi \$SUNPRO_SWM_GUI_ARGS</code>
Icon File:	<code>vi.icon</code>
Icon Label:	<code>VItool</code>
<input type="button" value="Apply"/> <input type="button" value="Reset"/>	

Figure 3-5 Tool Icon and Startup Properties Window Example

2. Change the Tool Name to `vi`.
3. Delete the default command line from the Command field and type:

```
cmdtool -I vi $SUNPRO_SWM_GUI_ARGS
```

**4. Change the Icon File name.**

This example uses `vi.icon` as the icon file name.

**5. Change the Icon Label field.**

This example uses `VItool` as the icon label.

**6. Click on Apply to save the tool property customizations.**

---

**Note** – Since the command tool does not support the ToolTalk protocol, this `VItool` will not respond to the Manager session controls.

---

# The OPEN LOOK GUI



SPARCworks and ProWorks conform to the OPEN LOOK Graphical User Interface (GUI) standard. This chapter is a summary of the OPEN LOOK GUI basics.

The OPEN LOOK Graphical User Interface is a standardized interface shared by most Sun Microsystems® desktop applications. It promotes ease-of-use across application programs with a standard set of buttons, menus, controls and operations.

---

**Note** – If you are already familiar with OPEN LOOK, you can skip this chapter.

---

This chapter includes the following sections:

<i>OPEN LOOK GUI Basics</i>	<i>page 4-42</i>
<i>OPEN LOOK Base Window</i>	<i>page 4-44</i>
<i>Resizing and Scaling a Window</i>	<i>page 4-49</i>
<i>Getting Help in the OPEN LOOK GUI</i>	<i>page 4-50</i>

## 4.1 OPEN LOOK GUI Basics

The basic paradigm of the OPEN LOOK user interface is that of *select-and-operate*. You first select an object, button, menu, command, or item and then perform an operation upon it. You carry out this select-and-operate model on the screen through the use of a pointer and a pointer control device, most commonly a mouse. Moving the mouse moves the pointer on the screen.

### 4.1.1 Mouse Buttons

The OPEN LOOK user interface allows you to use either a one-button, two-button, or three-button mouse. Three standard functions are assigned to the buttons on a mouse. These functions are SELECT, ADJUST, and MENU.

- SELECT specifies objects on which to operate and manipulates objects and controls.
- ADJUST extends or reduces a selection.
- MENU displays a menu associated with the pointer location or with a selected object.

On a three-button mouse the left button is SELECT, the middle button is ADJUST, and the right button is MENU.

---

**Note** – You can change the mouse button assignments with an OPEN LOOK GUI implementation.

---

A mouse with only one or two buttons uses a combination of mouse buttons and keyboard equivalents to carry out the three functions. The keyboard equivalents are user-defined and vary from system to system.

On a one-button mouse the button is SELECT. ADJUST and MENU are carried out with keyboard equivalents.

On a two-button mouse the left button is SELECT and the right button is MENU. ADJUST is carried out with a keyboard equivalent.

Figure 4-1 illustrates the mouse button functions.

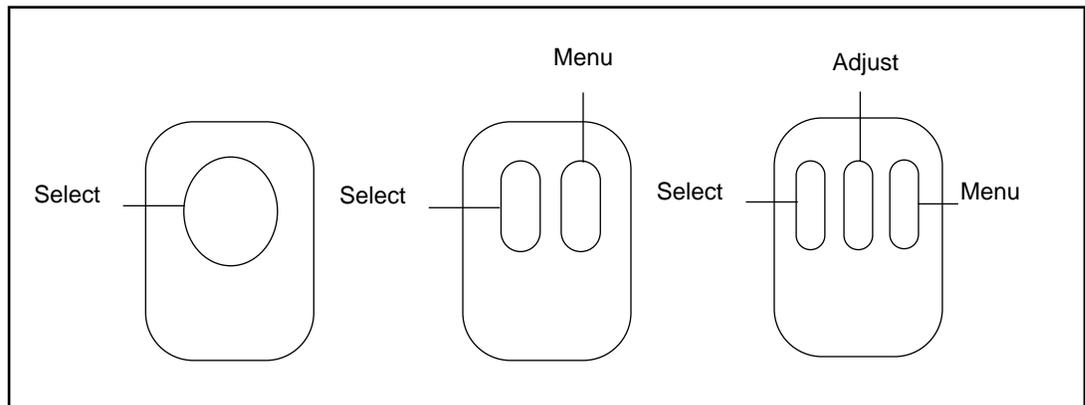


Figure 4-1 Mouse Button Functions

### 4.1.2 Mouse Actions

The following terms describe ways you use the mouse buttons and pointer with the OPEN LOOK user interface.

- **Press** — Push a mouse button and hold it.
- **Click** — Push and release a mouse button.
- **Double-click** — Push and release a mouse button twice in quick succession.
- **Move** — Slide the pointer without pushing any mouse buttons.
- **Drag** — Push a mouse button and hold it down while moving the pointer.

### 4.1.3 Pinned and Unpinned Menus

Menus in the OPEN LOOK GUI can either be *pinned* or *unpinned*. An unpinned menu is dismissed as soon as an operation is carried out; a pinned menu remains on the screen until you unpin it.



With a pinned menu, the pushpin is pushed in. In an unpinned menu the pushpin is pulled out, as shown here.

To pin or unpin a menu, place the pointer on the pin and click SELECT.

#### 4.1.4 Moving Windows and Menus

In the OPEN LOOK user interface, you can move and reposition windows and menus. To move and reposition a window or menu, place the pointer in the header or on a corner of the window or menu, click SELECT, and drag the window to the new position. When the window or menu is in position, release SELECT.

### 4.2 OPEN LOOK Base Window

A base window is an application window that contains many of the characteristics of the OPEN LOOK GUI. In general, a base window is the main window that is displayed when an application is started. The base window can be thought of as a framework within which the contents of an application are displayed.

A standard base window is divided into four areas, each of which may have several elements within it. The four main areas of a base window are: header, control area, pane, message area.

The figure below is a sample base window. It shows the four main areas of the window, as well as other elements within each area. The four main areas and the elements within them are described in greater detail in the following sections.

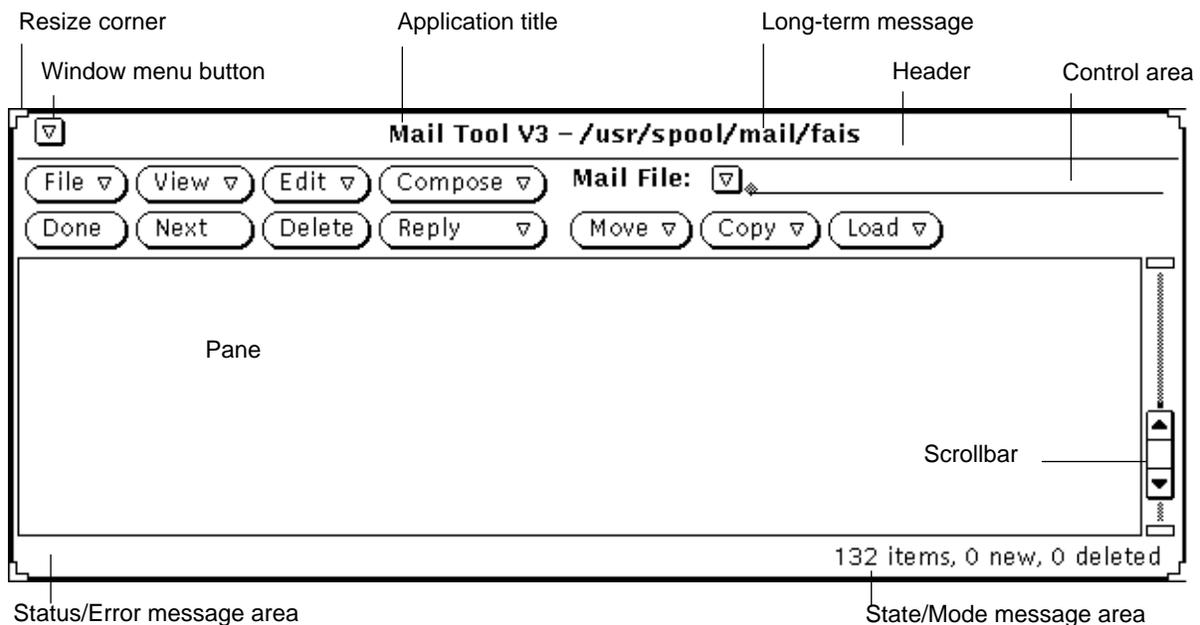


Figure 4-2 Base Window Elements

Since many of the characteristics within the OPEN LOOK GUI can be customized by application developers, the base window for one application may be slightly different from the base window for another. However, it will be easy for you to see the relationships between applications.

### 4.2.1 Header

The Header is the area at the top of a base window. It contains the Window menu button, the Application title, and a long-term message, such as the path name for the file currently being displayed.

#### *Window Menu*

The Window menu items activate window operations, such as closing the window, enlarging it, displaying a Properties sheet, moving an application into the background, refreshing the screen, or quitting the application.



To display the Window menu, place the pointer anywhere in the Header and press MENU.

Since Close is the default option on the Window menu, you can quickly close the base window to an icon without displaying the menu. Simply place the cursor on the Window menu button and click SELECT.

To reopen the base window, place the pointer on the icon and double-click SELECT.

## 4.2.2 Control Area

In the Control Area, buttons and pop-up menus are used to carry out a variety of operations.

### *Buttons and Pop-up Menus*

A *button* is a graphical equivalent of a keyboard button. Clicking or pressing the mouse on it initiates the action named on the button, but does not display an additional menu. A *menu button*, on the other hand, displays a pop-up menu when you click or press the mouse on it. Menu buttons are distinguished by the triangle on their right side, called the *menu mark*. A pop-up menu may display additional submenus.

### *Menu Button Items*

Three types of items are available from a menu button:

- **Command items** — carry out the command named by the item. A command item has no distinguishing marks except the name of the command. In the previous figure, Delete is a command item.
- **Menu items** — display an additional menu. Menu items have a menu mark on the right side.
- **Window items** — display an additional pop-up window, into which you can type text. Window items are distinguished by three dots after the item name.

### 4.2.3 Pane

The Pane is the area of the base window in which you work. It contains the main information about the application and is the central portion of the base window. Some OPEN LOOK applications may have more than one pane in the base window.

The pane frequently has a *scrollbar*, so that you can scroll the contents for better viewing. Each pane in the base window may have its own scrollbar.

#### *Scrolling*

A pane in the base window can have both horizontal and vertical scrollbars. Figure 4-3 shows a vertical scrollbar. In all of its parts and operation, it is exactly like a horizontal scrollbar.

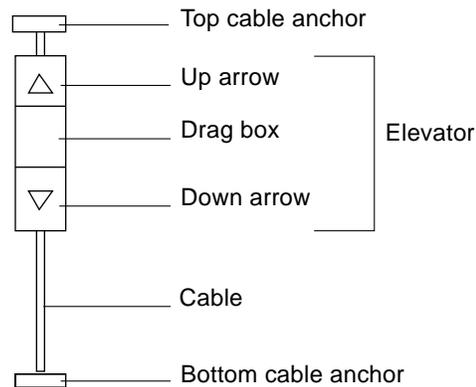


Figure 4-3 Vertical Scrollbar

To move the contents of a pane using the scrollbar:

- Click on the *top cable anchor* to view the contents at the top of the pane.
- Click on the *bottom cable anchor* to view the contents at the bottom of the pane.
- Click on the *up arrow* or *down arrow* to move the contents incrementally.
- Move the pointer to the *drag box* in the middle of the *elevator*; press and drag the elevator up or down to see the desired contents.

### ***Base Window Menu***

The base window menu is a *pop-up menu* displayed within the pane. To display the base window menu, place the pointer anywhere within the pane and press MENU.

The items on the base window menu often duplicate commands found on the menu button menus. The base window menu provides you quick access to the most commonly used commands.

### *Other Pop-up Menus*

Each area of an OPEN LOOK GUI base window may have its own pop-up menu. For example, you can scroll the contents of a pane by choosing scrolling options from a pop-up menu displayed when the pointer is placed on the scrollbar.

Pop-up menus of OPEN LOOK applications can be summoned from various areas of the base window. To see what pop-up windows are available in your current application, place the pointer in different areas of the base window and press MENU.

#### *4.2.4 Footer*

The Footer is the area of the base window where an application displays error messages or messages about the state or status of the application. The way in which messages are displayed can vary from application to application.

#### *4.2.5 Property Windows*

Most OPEN LOOK applications have a property window through which you can customize the characteristics of the application. To display the property window for an application, choose Properties on the Window menu or place the pointer in the application and press the Props key on the keyboard.

## *4.3 Resizing and Scaling a Window*

You can change the size of an OPEN LOOK base window or pop-up window by resizing and scaling.

### *4.3.1 Resizing a Window*

Resizing a window changes its dimensions without retaining the proportions of the window. To resize a window, place the pointer on a resize corner, press SELECT, and drag the resize corner until the window is the size you want.

### *4.3.2 Scaling a Window*

Scaling changes the size of a window while maintaining its proportions. To scale a window, choose Properties from the Window menu and use the Base Window Scale option. In applications without a Properties window, scaling may not be provided as a feature.

## *4.4 Getting Help in the OPEN LOOK GUI*

The OPEN LOOK GUI provides Magnify Help, a standard feature of the OpenWindows software environment. To access Magnify Help, place the pointer on the window, menu, or menu button, and then press the Help key.

# Troubleshooting



This appendix explains how to overcome problems that can arise while using the Manager. It is organized into the following sections:

<i>How the Manager Finds Things</i>	<i>page A-51</i>
<i>Installation Checklist</i>	<i>page A-54</i>
<i>Troubleshooting Checklist</i>	<i>page A-54</i>
<i>Reporting Problems</i>	<i>page A-55</i>
<i>Manager Messages</i>	<i>page A-56</i>

## A.1 How the Manager Finds Things

All products that include the Manager also include enough information so that the Manager palette is automatically extended the next time the Manager is invoked.

The Manager palette depends primarily on 2 types of files:

1. Configuration files that are included with each package and contain enough information to launch the tools supplied with that package. Configuration files are in `xdefaults` format. The information is essentially the same as that which can be set using the Tool Icon and Startup Properties pop-up window (see “Properties Menu” on page 2-15).
2. Icon files that the Manager can display on the palette.

The Manager locates configuration files by examining the `/opt/SUNWspro/SW3.0.1/lib/sunpro_defaults` directory. This directory contains a set of symbolic links to the actual configuration files supplied by each package that cooperates with the Manager. This extra level of indirection allows package providers freedom in the layout of their package; the actual configuration files can be anywhere so long as the links are properly installed.

The default palette is composed of the union of all of the package-supplied configuration files. Note that this means that it is essential for all cooperating packages to be installed in the same directory as the Manager if the palette is to be automatically configured.

The Manager locates icon files in a similar fashion. It examines the `/opt/SUNWspro/lib/icons` directory, which should contain a symbolic link to the icon for each tool to be displayed on the palette. As with configuration files, the icon files can be in any directory in the cooperating package as long as the symbolic links are properly installed. For a detailed discussion of icons files, see Section A.2, “Creating Icon Files”.

The Manager launches tools by following the `PATH` environment variable. While this is inconsistent with the convention that is used to locate configuration and icon files, it is the convention that most Solaris users are familiar with.

## A.2 Creating Icon Files

The Manager can load XPM (X Pixmap Format), XBM (X Bitmap Format), or Xview format icons. It selects the correct icon to load by concatenating suffixes to the specified string and determining the best fit for the display. The Manager looks for the icon in the following order:

1. Current directory
2. Directory in which the Manager was started
3. User’s home directory
4. A `lib` directory located relative to the Manager binary  
(`/opt/SUNWspro/lib/icons` in a normal installation)

The names of the icon have `<ToolTalk Ptype>_swm` as the file prefix. The suffix can be one of the following:

- `.xpm8` 8 bit XPM format color icon
- `.xpm1` 1 bit XPM Black&White icon
- `.xbm1` 1 bit XBM Black and White
- `.bw` Xview format

All icons should reside in the same directory. The XBM or the `.bw` icon is used as a cursor since Xview cannot use XPM format Drag & Drop cursors. However, the Manager uses a generic Drag & Drop cursor if you do not supply one.

### A.2.1 Creating Your Own Icons

To use your own icons, set up your own icon directory similar to the following example:

```
/myicondir/  
  MyTool.xpm8  
  MyTool.xpm1  
  MyTool.xbm1  
  MyTool.bw    (Alternate Drag & Drop cursor)
```

Then set up the SWM configuration file as follows (the easiest way to do this is to modify the Manager property sheet and save the changes):

```
swToolBox.toolPanel.margin: 6  
swToolBox.toolPanel.maxWidth: 600  
swToolBox.toolPanel.showLabels: True  
swToolBox.MyTool.glyph.file: /myicondir/MyTool.xpm8  
swToolBox.MyTool.glyph.label:MyTool  
swToolBox.MyTool.glyph.x: -1  
swToolBox.MyTool.glyph.y: -1  
swToolBox.MyTool.command: $SUNPRO_SWM_APP_FILE  
swToolBox.MyTool.guiArgsFormat: -display %s.%d -Wp %d %d  
swToolBox.MyTool.ptype:  
swToolBox.MyTool.TtVersion: 2.0  
swToolBox.tools: MyTool registration
```

For color icons, you must supply the icon suffix with a depth value if your icons do not reside in one of the default directories. You can install symbolic links in the `/opt/SUNWspro/lib/icons` directory that point to your installations custom icons. That is, execute the following command for an icon in `/myicondir`:

```
ln -s <icon> /opt/SUNWspro/lib/icons
```

Then remove all of the `.xpm` suffixes from the configuration file.

## A.3 Installation Checklist

If the Manager does not display all of the tools that you know are installed, check for a saved configuration file in `$HOME/.sunpro_defaults`. If one exists, rename it and restart the Manager.

If the Manager is unable to launch a tool that is displayed in the palette:

1. Check that the tool is reachable via the `PATH` variable by using `type toolname` in `sh` or `ksh` or `which toolname` in `csh`. If the tool isn't found, it should be installed in the same directory as the `sparcworks` or `proworks` command, so you should add that directory to your `PATH` variable. If the binary is not installed there, contact your system administrator.
2. If you do not wish to change your `PATH` variable, you can create a custom palette that specifies the full path name of each tool in the Command field of the Tool Icon and Startup Properties pop-up window. This is not recommended as it must be done for each tool and so is error prone.

If the Manager is unable to locate the icon for a tool:

1. Check the `/opt/SUNWspro/SW3.0.1/lib/icons` directory and make sure there is a symbolic link to the icon file named in the Tool Icon and Startup Properties pop-up window. Make sure that the symbolic link actually points to a valid icon file by using `ls -l`. If it doesn't, contact your system administrator.
2. If you know the full path name of the icon file, modify the entry for the selected tool in the Tool Icon and Startup Properties pop-up window and save the properties as `$HOME/.sunpro_defaults`.

## A.4 Troubleshooting Checklist

If you are having problems using the Manager, check for the following:

- Is the Manager installed correctly?  
If not, contact your system administrator.

Is `/opt/SUNWspro/bin` in your `PATH`?

If not, see the installation instructions in the *3.0.1 Installation AnswerBook* for information on how to add the SPARCworks or ProWorks directory to your `PATH`. The default directory for installation is `/opt/SUNWspro/bin`. A custom installation could go elsewhere, but this is not recommended.

Is the `HELPPATH` environment variable set?

The Manager relies on finding the file `manager.info` in the directory that contains `sparcworks`. The Manager Magnify Help provides on-line help for each control, window, pane, and error message displayed on the screen. See “Manager Messages” on page A-56 for a list of the Manager error messages and instructions on what to do next.

Do you have enough swap space?

If you receive a message stating “Request for xxx bytes of memory failed,” you have run out of swap space. Use the `mkfile(8)` and `swapon(8)` commands to create more swap space or abort some existing processes (windows) to free up swap space. To determine which processes occupy significant swap space, use the `ps -ale` command (`ps -alx` for Solaris 1.x) and look in the `SZ` column. To determine how much swap space you have, use the `swap -l` command (`pstat -s` for Solaris 1.x).

Does your window system have enough resources?

If the Manager cannot activate one of its pop-up windows, your window system may be running out of resources. Contact your system administrator for help.

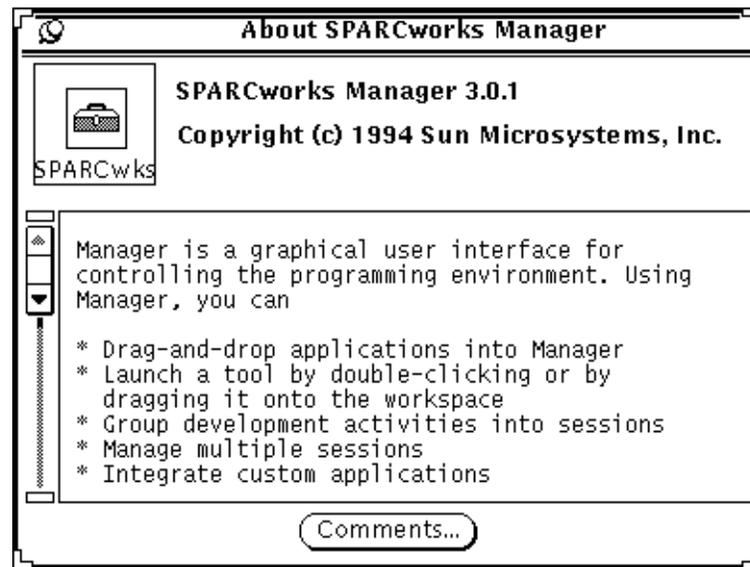
## A.5 Reporting Problems

If you have gone through the checklist and are still having problems using the Manager, call your local service office. Have the release number ready to give to the dispatcher. Also, be ready to provide the version number of your operating system and your hardware system configuration.

To display the Manager’s version number:

◆ **Click on the About button.**

The resulting pop-up window shows the Manager product release number. Clicking on the Comments button opens a window in which you can send comments about the tool to SunSoft.



## A.6 Manager Messages

The Manager displays messages to provide you with information or tell you about an error. This section lists the Manager messages and offers instructions about what to do next.

`couldn't initialize tootalk`

The ToolTalk `ttsession` program is not running, or is not properly configured. See the `ttsession` manual page for instructions (type `man ttsession`).

`can't write configuration file file`

The permissions on the specified file do not allow write access. Change the file permissions with the `chmod` command. See the `chmod` manual page for instructions (type `man chmod`).

`skipping startup configuration file file, can't open`

The startup configuration file cannot be opened. Change the file permissions with the `chmod` command. See the `chmod` manual page for instructions (type `man chmod`).

---

couldn't load the glyph for *application* from *file*

The icon for the named application could not be loaded from the named file. Either the file does not exist or the permissions do not allow the icon to be loaded. Change the file permissions with the `chmod` command. See the `chmod` manual page for instructions.

ignoring configuration file *file*

The specified configuration file could not be read correctly. Either the file does not exist or the permissions do not allow the file to be read. Change the file permissions with the `chmod` command. See the `chmod` manual page for instructions.

bad application file *file*

An application path name that does not support read-access was dropped on the Manager Drop Target. Change the file permissions of the application file with the `chmod` command. See the `chmod` manual page for instructions.

≡ A

---

# Integrating an Application into the Manager



This section describes the ToolTalk service messages that are used to connect individual tools that are part of the SPARCworks and ProWorks 3.x programming environments.

This chapter includes the following sections:

<i>Messages Description Syntax</i>	<i>page B-59</i>
<i>Session Protocol</i>	<i>page B-60</i>
<i>Manager</i>	<i>page B-60</i>

## B.1 Messages Description Syntax

The ToolTalk service messages that are used are described below. (Note that a BNF-like syntax is used for clarity).

message ::= '(' **Tt\_message** <attribute>\* [<args>] ')'

attribute ::= '(' <field\_name> <value> ')'

args ::= '(' **Tt\_args** <arg>+ ')'

arg ::= '(' <direction> [<type>] <value> ')'

direction ::= TT\_IN | TT\_OUT | TT\_INOUT

type ::= **string** | **int**

Values in upper case are ToolTalk enumeration constants, those in bold are ToolTalk keywords, and those enclosed by brackets (<>) are variable names.

In addition to the above, two additional notations are defined:

1. Optional arguments are enclosed in square brackets ([]).
2. Variable length lists are indicated by bracketing ({ }\*) groups of arguments that may be repeated.

## B.2 Session Protocol

The tools are started and controlled in two ways, either:

- Directly by the user, or
- By the Manager

## B.3 Manager

The tools can be started and manipulated by means of the Manager. When the Manager is used to start your tool, the Manager and your tool must exchange information about each other in order to subsequently communicate properly when using the ToolTalk service. The Manager must know your tool's *procid*. Your tool must know:

- The Manager's *procid*
- Its own unique ID *key*
- The session number

When the Manager itself is started, it sets the value of the environment variable `SW_TT_STARTUP_PROCID` to the value of its own *procid*.

1. When the Manager launches your tool, it sends a unique ID string (*key*) as the argument to the `-swtm` option.
2. Your tool receives the *key* and sends a ToolTalk startup message back to the originating Manager. The message contains the *key* and the tool's own *procid*. It determines the Manager's *procid* by reading the `SW_TT_STARTUP_PROCID` shell environment variable.

```
(Tt_message (Tt_class TT_REQUEST) (Tt_op "PEI_STARTED")
(Tt_address TT_HANDLER) (Tt_handler procid)
(Tt_args (TT_IN key) (TT_OUT sw_session)))
```

3. The Manager replies to the startup message and supplies your tool with the session number (*sw\_session*).

A tool that has been started by the Manager may, in turn, start additional tools; the automatic pass-down of the environment variable `SW_TT_STARTUP_PROCID` will connect them with a Manager.

---

**Note** – The `-swtm` argument is not required for proper operation, but is provided so that the Manager can track the tool it starts.

---

Alternatively, a tool may acquire more control over tools it starts by changing the environment variable and receiving the `PEI_STARTED` message itself.

A set of messages control sessions. The first is a simple ping:

```
(Tt_message (Tt_class TT_REQUEST) (Tt_op "PEI_NOP")
(Tt_args (TT_IN sw_session)))
```

The following messages allow coordinated control of a collection of tools. These are commands are sent by the Manager to the tools running in the session.

```
(Tt_message (Tt_class TT_NOTICE) (Tt_op "hide")
(Tt_args (TT_IN sw_session)))
```

- Hide the tool's window.

```
(Tt_message (Tt_class TT_NOTICE) (Tt_op "expose")
(Tt_args (TT_IN sw_session)))
```

- Expose the tool's window (if it is hidden).

```
(Tt_message (Tt_class TT_NOTICE) (Tt_op "PEI_OPEN")
(Tt_args (TT_IN sw_session)))
```

- Open the tool's window.

```
(Tt_message (Tt_class TT_NOTICE) (Tt_op "PEI_CLOSE")
(Tt_args (TT_IN sw_session)))
```

- Close the tool's window to its icon.

```
(Tt_message (Tt_class TT_NOTICE) (Tt_op "quit")
(Tt_args (TT_IN sw_session)))
```

- Quit the tool.

When a tool is about to exit, it informs the Manager by means of:

**≡ B**

---

**(Tt\_message (Tt\_class TT\_NOTICE) (Tt\_op "departed")  
(Tt\_args (TT\_IN *sw\_session*)))**

# *Index*

---

## **A**

About button, 2-21  
adding a tool, 3-37  
ADJUST, mouse button, 4-42  
All Tools, menu option, 2-17  
Analyzer, definition, 1-6  
AnswerBook® system, x

## **B**

background menu, 2-23  
base window, 4-44  
    control area, 4-46  
    footer, 4-49  
    header, 4-45  
    menu, 4-48  
    pane, 4-47  
    resizing and scaling, 4-49  
buttons, 4-46  
    control, 2-11  
    Edit, 2-11, 2-14  
    File, 2-11, 2-12  
    Properties, 2-11, 2-15  
    Session, 2-11, 2-18  
    View, 2-11, 2-13

## **C**

cable anchor, 4-48  
CallGrapher, definition, 1-5  
ClassBrowser, definition, 1-5  
ClassGrapher, definition, 1-5  
Click, mouse action, 4-43  
Close, menu option, 2-18  
CodeBrowser, definition, 1-5  
command items, 4-47  
Compact Palette  
    viewing, 2-25  
compact palette, 2-24, 2-25  
    definition, 2-13, 2-24  
    menu option, 2-13  
    redisplay full palette, 2-26  
    reshaping, 2-26  
control area, 4-44, 4-46  
customizations, 3-34

## **D**

Debugger, definition, 1-5  
Delete Tool, menu option, 2-14  
deleting a tool, 3-37  
development process, 1-6  
Double-click, mouse action, 4-43  
Drag, mouse action, 4-43

---

Drop Target, 2-11, 2-21  
Duplicate Tool, menu option, 2-14  
duplicating a tool, 3-36

## E

Edit button, 2-11, 2-14  
    Delete Tool, 2-14  
    Duplicate Tool, 2-14  
    Selected Tool, 2-15  
    Snap Icons to Grid, 2-15, 2-27  
elevator, 4-48  
environment variables  
    SUNPRO\_SWM\_APP\_DIR, 2-16  
    SUNPRO\_SWM\_APP\_FILE, 2-16  
    SUNPRO\_SWM\_DIRECTORY, 2-16  
    SUNPRO\_SWM\_GUI\_ARGS, 3-30  
    SUNPRO\_SWM\_GUI\_ARGS, 2-16  
    SUNPRO\_SWM\_TT\_ARGS, 3-30  
    SUNPRO\_SWM\_TT\_ARGS, 2-16

## F

File button, 2-11, 2-12  
    Save, 2-12  
    Save As, 2-12  
FileMerge, definition, 1-6  
Full Palette, 2-24  
full palette  
    definition, 2-24  
    redisplaying, 2-26

## G

Graphical User Interface, GUI, 4-41

## H

header, 4-44, 4-45  
Help Facilities  
    getting help with OPEN LOOK, 4-50  
    Magnify Help, 4-50  
help facilities, x  
Hide, menu option, 2-18

## I

installation checklist, A-54

## L

Log window, 2-13, 3-30

## M

Magnify Help, 4-50  
Magnify Help™ messages, xi  
MakeTool, definition, 1-5  
man pages, xi  
Manager  
    changing properties, 3-31  
    compact palette, 2-24  
    control buttons, 2-11  
    formats, 2-11  
    Full Palette, 2-24  
    functions, 2-11  
    Log window, 3-30  
    modifying, 3-31  
    Properties Sheet, 2-17  
    quitting, 2-10, 2-11  
    saving customizations, 3-34  
    starting, 2-9  
    starting a tool, 3-29  
    working with, 3-29  
menu button, 4-46  
    command items, 4-47  
    definition, 4-46  
    menu items, 4-47  
    window items, 4-47  
menu mark, 4-46  
MENU, mouse button, 4-42  
menus  
    background, 2-23  
    moving, 4-44  
    pinned, 4-43  
    pop-up, 4-46, 4-49  
    unpinned, 4-43  
message area, 4-44  
mkfile command, A-55  
mouse actions, 4-43

---

Click, 4-43  
Double-click, 4-43  
Drag, 4-43  
Move, 4-43  
Press, 4-43  
mouse buttons, 4-42  
ADJUST, 4-42  
MENU, 4-42  
SELECT, 4-42  
Move, mouse action, 4-43

## N

non-window-based tool, 3-39  
notational conventions, x  
Notices, xi

## O

OPEN LOOK  
base window, 4-44  
buttons, 4-46  
getting help, 4-50  
GUI basics, 4-42  
menu button items, 4-46  
menu buttons, 4-46  
pop-up menus, 4-49  
property windows, 4-49  
scrolling, 4-47  
window menu, 4-45  
Open, menu option, 2-18

## P

palette  
compact, 2-25  
pane, 4-44, 4-47  
pinned menus, 4-43  
pop-up menu, 2-23  
pop-up menus, 4-46  
Press, mouse action, 4-43  
properties  
changing tool properties, 3-33  
Manager, 3-31

Properties button, 2-11, 2-15  
All Tools, 2-17  
SPARCworks Manager Properties  
Sheet, 2-17  
Tool Properties Window, 2-15  
Properties Window  
Manager, 2-17  
OPEN LOOK, 4-49  
Tool Properties, 2-15

## Q

quitting the manager, 2-11

## R

restoring a tool, 3-38  
roadmap, 1-6

## S

Save As, menu option, 2-12  
Save, menu option, 2-12  
scrollbar  
cable anchor, 4-48  
elevator, 4-48  
SELECT, mouse button, 4-42  
select-and-operate, 4-42  
Selected Tool, menu option, 2-15  
Session button, 2-11, 2-18  
Close, 2-18  
Hide, 2-18  
Open, 2-18  
Show, 2-18  
session, monitoring, 3-31  
Show, menu option, 2-18  
SIGUSR1, 2-11  
Snap Icons to Grid, menu option, 2-15,  
2-27  
Solaris, viii  
SourceBrowser, 1-5  
CallGrapher, 1-5  
ClassBrowser, 1-5  
ClassGrapher, 1-5

---

CodeBrowser, 1-5  
definition, 1-5  
SPARCworks Manager  
Properties Sheet, 2-17  
SunOS 4.1.X, viii  
SunOS 5.0, viii  
SUNPRO\_SWM\_APP\_DIR,  
definition, 2-16  
SUNPRO\_SWM\_APP\_FILE,  
definition, 2-16  
SUNPRO\_SWM\_DIRECTORY,  
definition, 2-16  
SUNPRO\_SWM\_GUI\_ARGS,  
definition, 3-30  
SUNPRO\_SWM\_GUI\_ARGS,  
definition, 2-16  
SUNPRO\_SWM\_TT\_ARGS,  
definition, 3-30  
SUNPRO\_SWM\_TT\_ARGS,  
definition, 2-16  
swap space, A-55  
System V Release 4 (SVR4), viii

## T

terminating a session, 2-11  
tool  
adding, 3-37  
deleting, 3-37  
duplicating, 3-36  
restoring, 3-38  
starting non-window-based, 3-39  
tool properties window, 2-15  
tool startup log, 2-13  
toolset  
roadmap, 1-6  
troubleshooting, A-54  
typographic conventions, x

## U

UCB BSD 4.3, viii  
unpinned menus, 4-43

## V

View button, 2-11, 2-13  
Compact Palette, 2-13  
compact palette, 2-13  
Tool Startup Log, 2-13

## W

window  
base, 4-44  
base window menu, 4-48  
cable anchor, 4-48  
control area, 4-44  
elevator, 4-48  
footer, 4-49  
header, 4-44  
main areas, 4-44  
menu, 4-45  
menu button items, 4-47  
moving, 4-44  
OPEN LOOK properties, 4-49  
pane, 4-44, 4-47  
resizing, 4-49  
scaling, 4-49  
scrolling, 4-47