Merging Source Files

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

Part No.: 801-7321-10
Revision A, August 1994

Contents

Preface

This manual explains how to use FileMerge in typical software engineering projects. FileMerge is an integrated part of the SPARCworks™/ProWorks™ toolset, which also includes:

- Analyzer (available only on ProWorks/SPARCworks running Solaris™ 2.x)
- Debugger
- MakeTool
- Manager
- SourceBrowser

Before You Begin

This manual is written for software developers who write and test programs coded in text (ASCII) source, including ANSI C, FORTRAN, Pascal, C++, and Assembler.

This manual assumes you are familiar with

- Sun® operating system commands and concepts
- The OPEN LOOK® interface and the OpenWindows™ 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 *Managing the Toolset*.

For more information on the OpenWindows environment, see the *OpenWindows Developer's Guide: User's Guide.*

Solaris 2.x and 1.x

From a usage point of view, almost all of the aspects of the SPARCworks and ProWorks toolsets under Solaris 2.x and 1.x are the same. This includes functionality, behavior, usage, and features. For the very few details that are different, the documentation calls out those differences.

Operating Environment

FileMerge runs under the Solaris 2.x and 1.x operating environments.

For SPARCworks, Solaris 2.x implies:

- Solaris 2.2 (or later) operating environment
- SunOS™ 5.2 (or later) operating system
- A SPARC® 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

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

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

The SunOS 5.x operating environment is based on the System V Release 4 (SVR4) UNIX[1] operating system, and the ONC™ family of published networking protocols and distributed service. SunOS 4.1.x is based on the UCB BSD 4.3 operating system.

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

How This Book Is Organized

This manual is organized as follows:

Chapter 1, "Introduction to FileMerge," presents an overview of FileMerge.

Chapter 2, "Starting and Loading FileMerge," describes how to start FileMerge and load files.

Chapter 3, "Examining Differences," contains a simple but detailed example of how to use FileMerge.

Appendix A, "Troubleshooting," describes what to do if you have trouble running FileMerge.

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 `.login` file. Use `ls -a` to list all files. system% You have mail.

`AaBbCc123`	What you type, contrasted with on-screen computer output	`system%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type `rm` *filename*.
AaBbCc123	Book titles, new words or terms, or words to be emphasized	Read Chapter 6 in *User's Guide*. These are called *class* options. You *must* be root to do this.
♦	A single-step procedure	♦ **Click on the Apply button.**

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

Table P-1 Typographic Conventions (Continued)

Typeface or Symbol	Meaning	Example
`%`	C shell prompt	`system%`
`$`	Bourne and Korn shell prompt	`system$`
`#`	Superuser prompt, all shells	`system#`

How to Get Help

SPARCworks tools include the following on-line help facilities:

- **AnswerBook**® system displays all SPARCworks/ProWorks tools manuals. You can read this manual on line and take advantage of dynamically linked headings and cross-references.

  To start the AnswerBook system, type: `answerbook &`
- **Magnify Help**™ 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. To access the man page for FileMerge, type:

  ```
  man filemerge
  ```

Related Documentation

This manual is part of the SPARCworks 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*

- *Managing the Toolset*
- *Performance Tuning an Application*

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

Introduction to FileMerge 1

FileMerge loads and displays two text files for side-by-side comparison, each in a read-only text pane. FileMerge marks lines that differ between the two files and displays a merged version in a third text pane. The merged version contains two types of lines:

- Lines that are common to both input files (these lines always appear in the output file)
- Marked lines that are different in each file (these lines appear as the result of the default automerge process)

You can edit the merged version and save it as an output file.

At the time you load the two files to be merged, you can also specify a third file, called the *ancestor* of the two files (which are called its *descendants*). When you have specified an ancestor file, FileMerge marks lines in the descendants that are different from the ancestor and produces a merged file based on all three files. In order to automatically merge (automerge) the two input files, you must specify an ancestor file.

This chapter is organized into the following sections:

1.1 Differences Defined

FileMerge operates on *differences* between files. Although you probably have a good intuitive grasp of what a difference is, the following describes how FileMerge recognizes and classifies differences.

1.1.1 Difference

When FileMerge discovers a line that differs between the two files to be merged (or between either of the two files and the ancestor), it marks with glyphs the lines in the two files. Together, these marked lines are called a *difference*. While FileMerge is focusing on a difference, it highlights the glyphs.

1.1.2 Current, Next, and Previous Difference

The difference on which FileMerge is focusing at any given time is called the *current* difference. The difference that appears immediately later in the file is called the *next* difference; the difference that appears immediately earlier in the file is called the *previous* difference.

1.1.3 Resolved and Remaining Difference

A difference is *resolved* if either you or FileMerge accept the changes to a line. Differences are resolved one of two ways:

While focusing on a difference, you can accept a line from one of the original files, or you can edit the merged version by hand. When you indicate that you are satisfied with your changes (by clicking on a command button), the current difference is then resolved.

If the Auto Merge feature is on, FileMerge resolves differences automatically. For more information on how FileMerge resolves differences, see the discussion in Section 1.2.2, “FileMerge Glyphs,” on page 1-4.

After a difference is resolved, FileMerge identifies it by changing its associated glyphs from solid to outline font. FileMerge then automatically advances to the next difference (if the Auto Advance property is on) or moves to the difference of your choice.

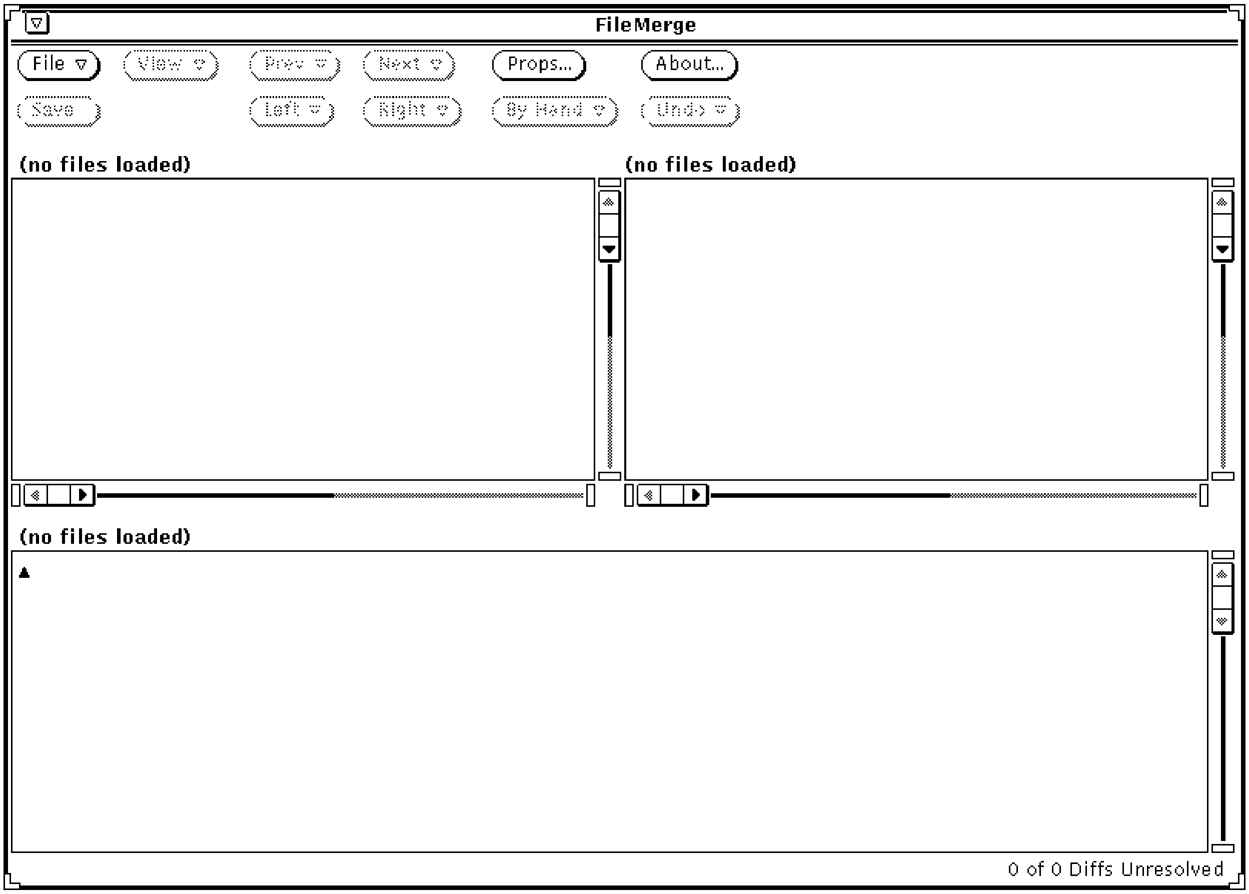
A *remaining* difference is one that has not yet been resolved.

1.2 Graphical Overview

The graphical interface for FileMerge consists of one main window, in which you do most of your work, and pop-up windows for handling files and setting properties.

1.2.1 FileMerge Window

The FileMerge window at startup is shown in Figure 1-1. The left and right text panes at the top display the files to be compared; the text pane at the bottom displays a merged version of the two files that you can edit.



(no files loaded)

(no files loaded)

(no files loaded)

0 of 0 Diffs Unresolved

Figure 1-1 FileMerge Window at Startup

1.2.2 FileMerge Glyphs

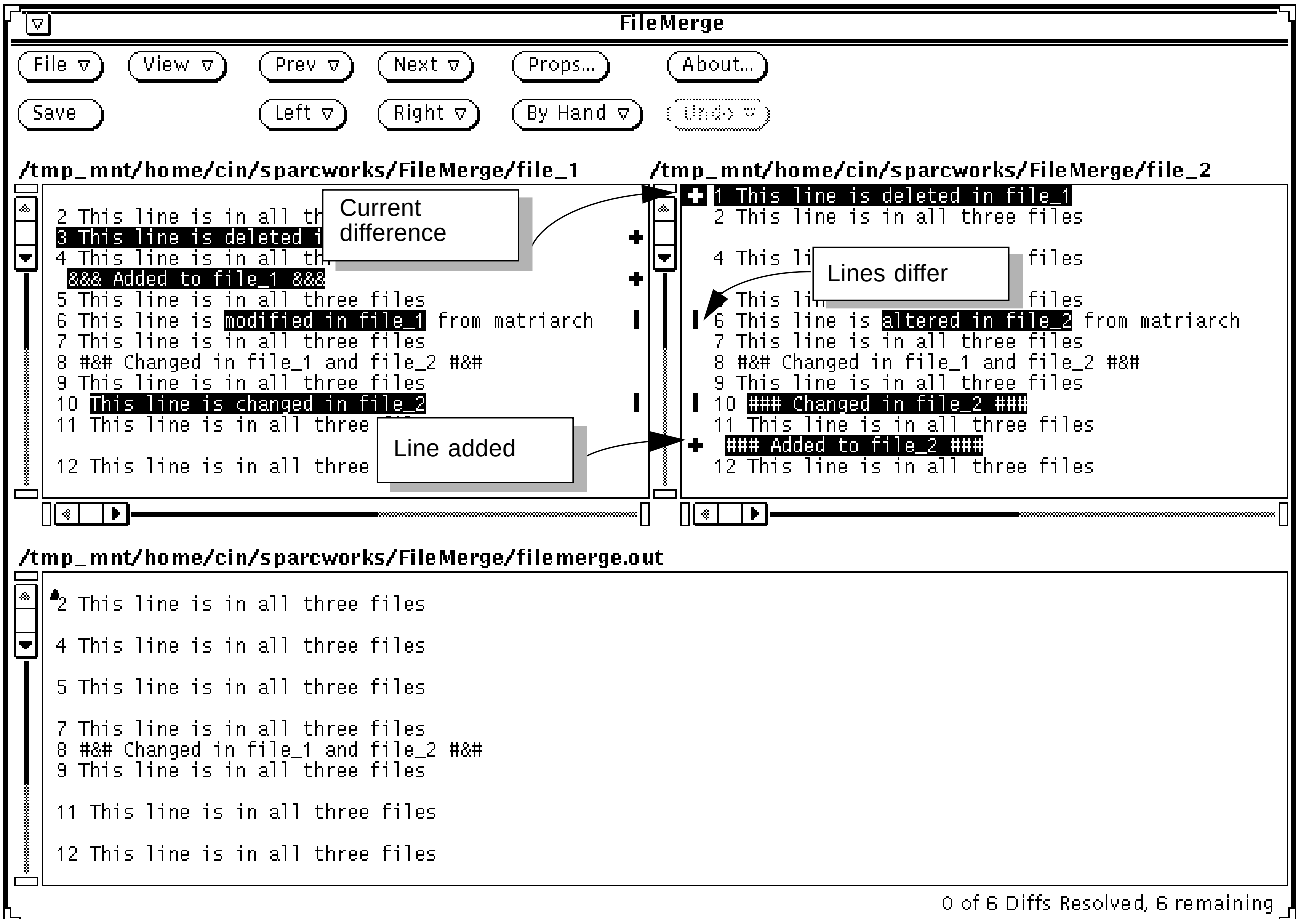
Two versions of the FileMerge window are shown in Figure 1-2 and Figure 1-3. The left and right panes at the top display the files to be merged; the text pane at the bottom is a merged version of the two files that you can edit. The windows have been loaded with example files that are explained in detail in Section 3.4, “An Example,” on page 3-34.

The difference between the two figures is that Figure 1-2 shows two files being merged having no common ancestor, and Figure 1-3 shows two files that have a common ancestor (this case is actually a three-way merge). The meaning of the glyphs in each case is slightly different, as explained below.

Two Input Files

When only two files have been loaded into FileMerge (as shown in Figure 1-2), lines in each file are marked by glyphs to indicate when they differ from corresponding lines in the other file:

- If two lines are identical, no glyph is displayed.
- If two lines are different, a vertical bar (|) is displayed next to the line in each input text pane, and the different characters are highlighted.
- If a line appears in one file but not in the other, a plus sign (+) is displayed next to the line in the file where it appears, and the different characters are highlighted.



/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
2 This line is in all th
3 This line is deleted i
4 This line is in all th
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three fi

12 This line is in all three
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files

4 This li                    files

  This li                    files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files

4 This line is in all three files

5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files

11 This line is in all three files

12 This line is in all three files
```

0 of 6 Diffs Resolved, 6 remaining

Figure 1-2 Loaded Files with No Common Ancestor

Three Input Files

When an ancestor file has been specified for the two files to be merged (as shown in Figure 1-3), lines in each descendant are marked according to their relationship to the corresponding lines in the common ancestor:

- If a line is identical in all three files, no glyph is displayed.

- If a line is not in the ancestor but was added to one or both of the descendants, a plus sign (+) is displayed next to the line in the file where the line was added, and the different characters are highlighted.
- If a line is present in the ancestor but was removed from one or both of the descendants, a minus sign (-) is displayed next to the line in the file from which the line was removed, and the different characters are highlighted and in strikethrough.
- If a line is in the ancestor but has been changed in one or both of the descendants, a vertical bar (|) is displayed next to the line in the file where the line was changed, and the different characters are highlighted.

Resolved differences are marked by glyphs in outline font.

The following table summarizes the automerging algorithm. Ancestor is the version of a text line that is in the ancestor file; Change 1 is a change to that line in one of the descendants; Change 2 is another change, different from Change 1. Only when a line is changed differently in the left and right descendants does automerging fail.

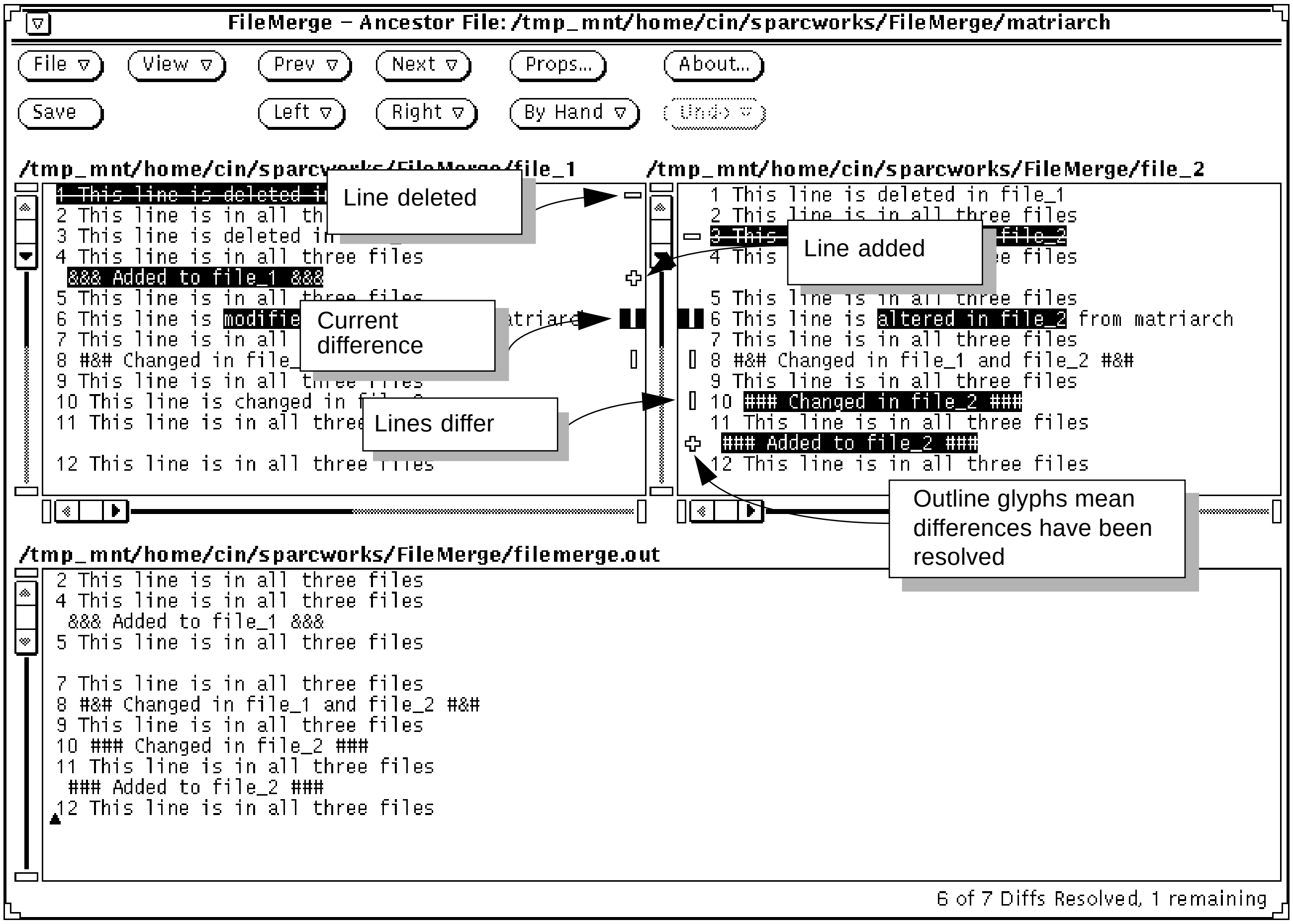
Table 1-1 Automerging Rules Summary (*see explanation in text*)

Left Descendant	**Right Descendant**	**Automerged Line**
Ancestor	Ancestor	Ancestor
Change 1	Ancestor	Change 1
Ancestor	Change 2	Change 2
Change 1	Change 1	Change 1
Change 1	Change 2	No automerge

When FileMerge automatically resolves a difference, it changes the glyphs to outline font. FileMerge lets you examine automatically resolved differences in order to be sure that it has made the correct choices.

You can disable automatic merging in the Properties window. When automatic merging is disabled, FileMerge constructs a merged file using only lines that are identical in all three files and relies on you to resolve the differences.

If you do not specify an ancestor file, FileMerge has no reference with which to compare a difference between the two input files. Consequently, FileMerge cannot determine which line in a difference is likely to represent the desired change. The result of an automerge with no ancestor is the same as disabling automatic merging: FileMerge constructs a merged file using only lines that are identical in both input files and relies on you to resolve differences.



FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch

/tmp_mnt/home/cin/sparcw

```
1 This line is deleted i
2 This line is in all th
3 This line is deleted in
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modifie                atriarc
7 This line is in all
8 #&# Changed in file_
9 This line is in all three files
10 This line is changed in f
11 This line is in all three

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
 1 This line is deleted in file_1
 2 This line is in all three files
 3 This                  file_2
 4 This                  e files

 5 This line is in all three files
 6 This line is altered in file_2 from matriarch
 7 This line is in all three files
 8 #&# Changed in file_1 and file_2 #&#
 9 This line is in all three files
 10 ### Changed in file_2 ###
 11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

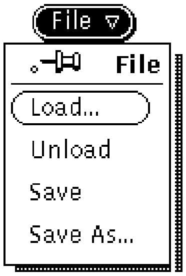
```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

6 of 7 Diffs Resolved, 1 remaining

Figure 1-3 Loaded Files with Common Ancestor

1.2.3 File Menu

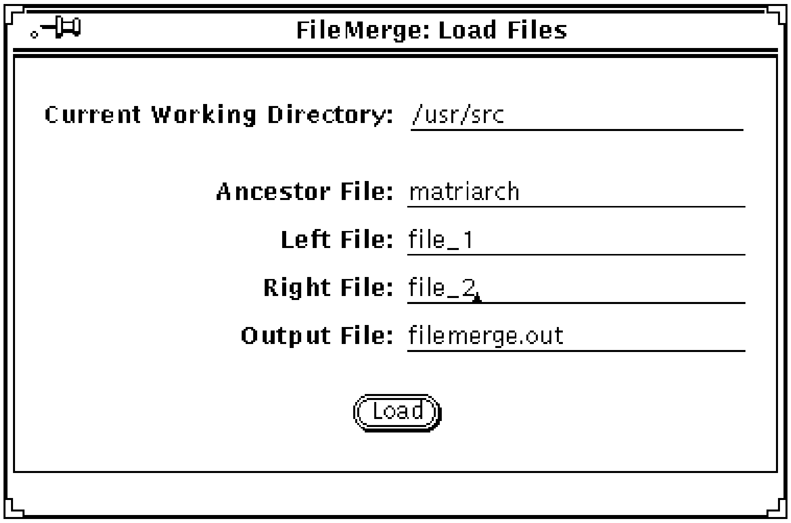


The File menu lets you load files and specify the name of the output file. See Section 2.2, “Loading Files from the FileMerge Window,” on page 2-22 for details on how to load files.

Load

Opens a pop-up window (shown in Figure 1-4) in which you enter the names of files to be loaded. You can also set these values in this window:

- Default name to use when you save the output file
- Current working directory



FileMerge: Load Files

Current Working Directory: /usr/src

Ancestor File: matriarch

Left File: file_1

Right File: file_2

Output File: filemerge.out

Figure 1-4 Load Files Pop-Up Window

Unload

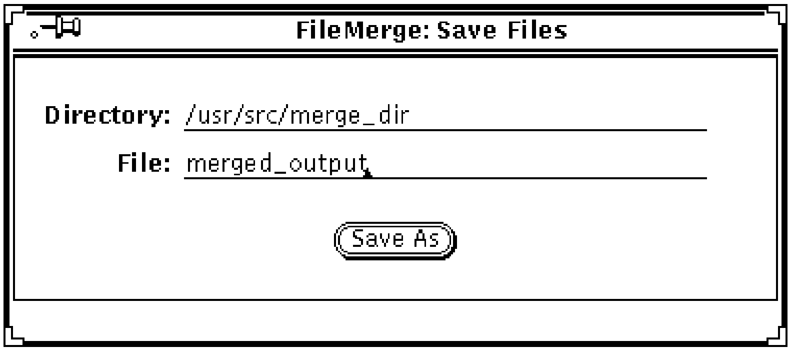
Lets you clear loaded files without saving the output file.

Save

Stores the output file under the name you specify in the Load Files pop-up window (Figure 1-4). If you do not enter a different name, FileMerge uses the name `filemerge.out`. Choosing Save is equivalent to clicking on the Save button.

Save As

Lets you save the output file under a name different from the name shown in the Load Files pop-up window, or in a directory other than the current working directory. Save As opens the following pop-up window:



Directory: /usr/src/merge_dir

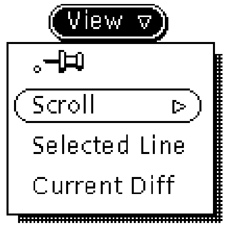
File: merged_output

Figure 1-5 Save Files Pop-Up Window

1.2.4 Save Button

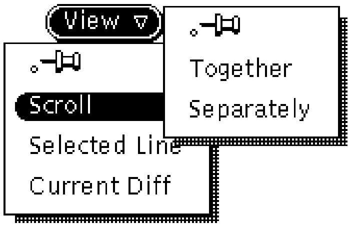
Stores the output file under the name you specify in the Load Files pop-up window (Figure 1-4). If you do not enter a different name, FileMerge uses the name `filemerge.out`. Clicking on the Save button is equivalent to choosing Save from the File menu.

1.2.5 View Menu



The View menu lets you control the behavior of the text panes and quickly move through the text files they display.

Scroll



Scroll allows you to set the text panes so they scroll together (so that corresponding lines are always aligned in each window) or separately. By default, the windows are set to scroll together.

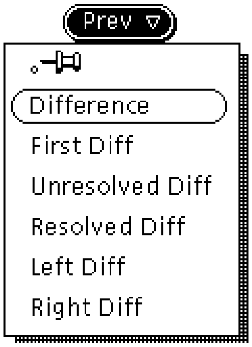
Selected Line

Causes a line that has been selected in one text pane to automatically scroll into view in the other two panes. This is useful when the text panes have been set to scroll separately.

Current Diff

Displays the current difference in all text panes. This is useful when the current difference is not visible in all windows.

1.2.6 Prev Menu



Prev menu lets you move the current difference up the list of differences (that is, toward the beginning of each file).

Difference

Moves to the previous difference, whether resolved or unresolved, making it the current difference.

First Diff

Moves to the first difference between the two files, making it the current difference.

Unresolved Diff

Moves to the previous unresolved difference, making it the current difference.

Resolved Diff

Moves to the previous resolved difference, making it the current difference.

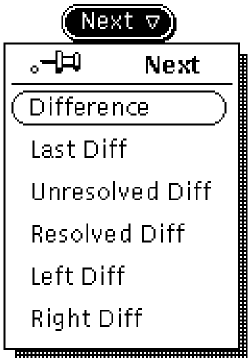
Left Diff

Moves to the previous difference in the file displayed in the left text pane, making it the current difference.

Right Diff

Moves to the previous difference in the file displayed in the right text pane, making it the current difference.

1.2.7 Next Menu



Next menu lets you move the current difference down the list of differences (that is, toward the end of each file).

Difference

Moves to the next difference, whether resolved or unresolved, making it the current difference.

Last Diff

Moves to the last difference between the two files, making it the current difference.

Unresolved Diff

Moves to the next unresolved difference, making it the current difference.

Resolved Diff

Moves to the next resolved difference, making it the current difference.

Left Diff

Moves to the next difference in the file displayed in the left text pane, making it the current difference.

Right Diff

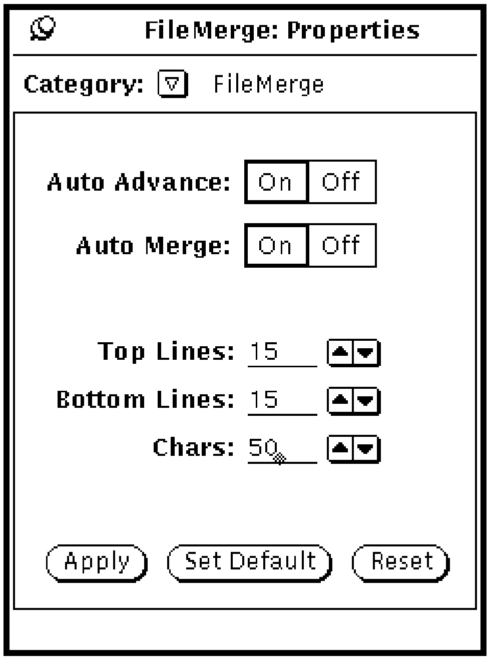
Moves to the next difference in the file displayed in the right text pane, making it the current difference.

1.2.8 Props Button

The Props button displays the Properties window. There are two categories: Filemerge (if you are using a monochrome monitor, this is your only category) and Colors. If you have a color monitor, FileMerge allows you to select colors from a palette to highlight the various line differences (see “Color Palette” on page 1-15).

FileMerge Properties

The values you set in the Properties window affect the behavior of FileMerge and the way files are displayed in the text panes.



Category: FileMerge

Auto Advance: On

Auto Merge: On

Top Lines: 15

Bottom Lines: 15

Chars: 50

Figure 1-6 FileMerge Properties Window with the FileMerge Category Selected

These are the properties you can edit when the FileMerge category is selected:

Auto Advance

Sets the default choices for the Left and Right menus to Next After Apply (the Left/Right menu is shown on page 1-18). When off, the default choices are set to Stay After Apply. The Next After Apply choice causes FileMerge to move to the next unresolved difference after the current difference has been resolved; the Stay After Apply choice causes FileMerge to stay at the current difference. In either case, you can override the default choice by choosing a different item from the Left or Right menus. The default setting is On.

Auto Merge

Constructs in the output text pane an automatically merged version of the input files. When Auto Merge is off, FileMerge does not attempt to resolve differences on its own, but places in the output pane only those statements that are identical in all input files. The default setting is On.

Top Lines

The number of text lines displayed in the Left and Right panes. Change the number by editing directly or by clicking on the arrow buttons next to the number entry field. The default value is 20.

Bottom Lines

The number of text lines displayed in the bottom pane, which displays the merged version of the files. Change the number by editing directly or by clicking on the arrow buttons next to the number entry field. The default value is 20.

Chars

The number of character columns displayed in the left and right panes. Change the number by editing directly or by clicking on the arrow buttons next to the number entry field. The default value is 70.

The buttons work as follows:

Apply

Transfers the settings in the Properties window to FileMerge but does not save the settings for use the next time you start FileMerge.

Set Default

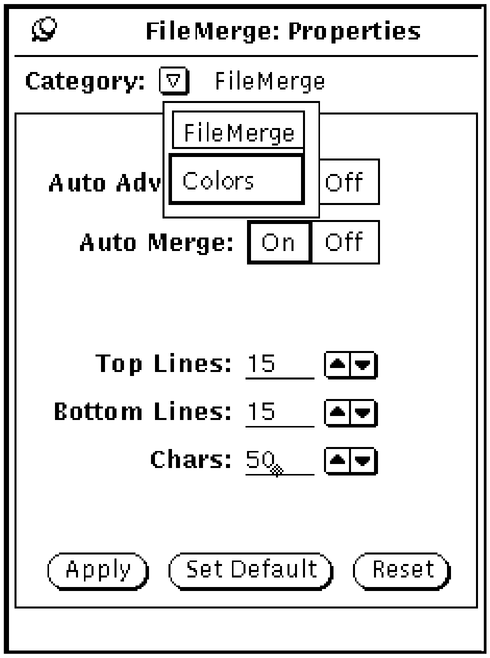
Applies the settings, saves them in a file named `.filemergerc` in your home directory, and automatically applies them the next time you start FileMerge.

Reset

Changes the settings to those that were saved the last time you clicked on Set Default. It does *not* change the values to the settings that were last applied (unless you clicked on Set Default as well as Apply).

Color Palette

If you have a color monitor, FileMerge allows you to select colors from a palette to highlight the various line differences. Select the Colors category from the FileMerge Properties window to bring up the color palette (see Figure 1-8 on page 1-17).



FileMerge: Properties

Category: ▽ FileMerge

Auto Adv… Off

Auto Merge: On Off

Top Lines: 15

Bottom Lines: 15

Chars: 50

Figure 1-7 FileMerge Properties Window with Colors Selected

To use the Color Palette:

1. **Click on a line type, such as Added Lines.**
2. **Choose a color from the palette by clicking on it.**
3. **Check your choice in the Preview Area.**
4. **Click Apply.**

The buttons work as follows:

Apply

Applies the color you've selected. The settings remain in effect for the current session of FileMerge only. Use Set Default if you want to save your choices.

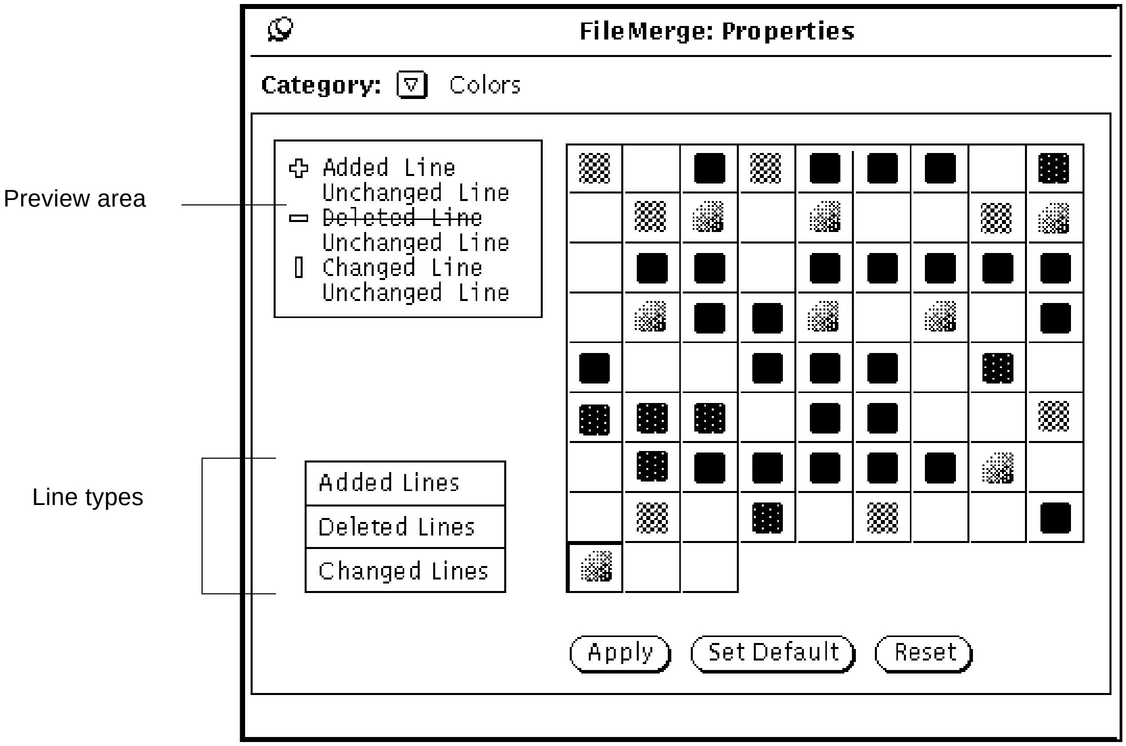
Set Default

Sets the default to always use a particular color.

Reset

Restores default settings.

Refer to the discussion of Properties buttons on page 1-14 for more detailed information about these buttons.



FileMerge: Properties

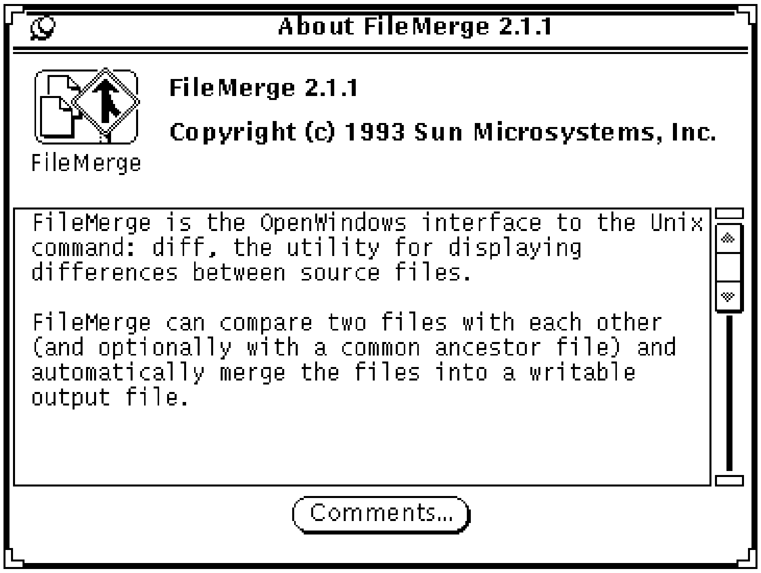
Category: Colors

Added Line
Unchanged Line
~~Deleted Line~~
Unchanged Line
Changed Line
Unchanged Line

Figure 1-8 Color Palette

1.2.9 About Button

The About button displays a pop-up window that gives the version number, copyright, and a brief description of FileMerge (see Figure 1-9 on page 1-18). Clicking on the Comments button displays another window from which you can send comments about the tool to SunPro.



FileMerge 2.1.1

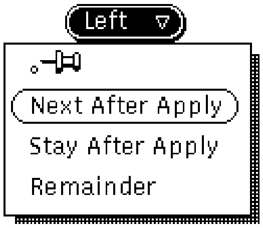
Copyright (c) 1993 Sun Microsystems, Inc.

FileMerge is the OpenWindows interface to the Unix command: diff, the utility for displaying differences between source files.

FileMerge can compare two files with each other (and optionally with a common ancestor file) and automatically merge the files into a writable output file.

Figure 1-9 About Box

1.2.10 Left and Right Menus



The Left and Right menus are identical; one applies to the left text pane, the other to the right.

The default items are determined by your settings in the Properties window: if the Auto Advance property is on, Next After Apply is the default; if Auto Advance is off, Stay After Apply is the default.

Next After Apply

Resolves the current difference by placing the line from the text pane into the merged output file and automatically advancing to the next unresolved difference (making that difference the current difference).

Stay After Apply

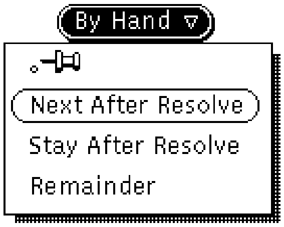
Also resolves the current difference by placing the line from the text pane into the merged output file. However, instead of automatically advancing to the next unresolved difference, FileMerge maintains its display of the resolved difference (which remains the current difference).

Remainder

Resolves all remaining differences (whether actually resolved or not) by placing the lines from the text pane into the merged output file and marking all unresolved differences as resolved.

1.2.11 By Hand Menu

To resolve a difference by hand, first edit the marked line in the merged file. The merged file is displayed in an OpenWindows text pane, so all the standard OpenWindows text editing commands are available, including the ability to paste text from other windows. If you need help with OpenWindows text editing conventions, see the *OpenWindows Developer's Guide: User's Guide*.



After you have edited the file, choose an item from the By Hand menu to resolve the current difference.

Next After Resolve

Acknowledges your edits by marking the current difference as resolved and moving on to the next unresolved difference, which becomes the current difference.

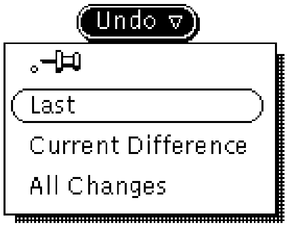
Stay After Resolve

Also acknowledges your edits by marking the current difference as resolved. However, instead of automatically advancing to the next unresolved difference, FileMerge maintains its display of the resolved difference (which remains the current difference).

Remainder

Resolves all remaining differences by accepting the lines shown in the lower text pane (the merged file) into the output file.

1.2.12 Undo Menu



The Undo menu lets you undo the effects of your actions by remarking differences you have resolved during a session and undoing edits you made by hand in the merged file.

Last

Undoes the edits that you applied to the previous difference (not the current difference).

Current Difference

Undoes the edits that you applied to the current difference.

All Changes

Undoes the edits you applied to all differences, enabling you to start the entire FileMerge session over. You may also start over using Load.

Starting and Loading FileMerge 2

This chapter explains how to start FileMerge, load it with files, and save the output file. The chapter is organized into the following sections:

You can start FileMerge from the Manager or from a shell command line. Each method has its advantages. The command line provides flexible access to all FileMerge options such as loading files at startup time. However, the Manager provides the following additional capabilities:

- An easy-to-use visual interface for starting all tools
- The ability to set properties, such as a default working directory, for all tools
- A properties sheet specific to each tool that lets you specify command-line options and save the command line for use in a later session

2.1 Starting FileMerge from the Manager

To start FileMerge from the Manager (shown in Figure 2-1):

♦ **Select the FileMerge icon in the Manager window and drag it onto the workspace, or double-click on it.**

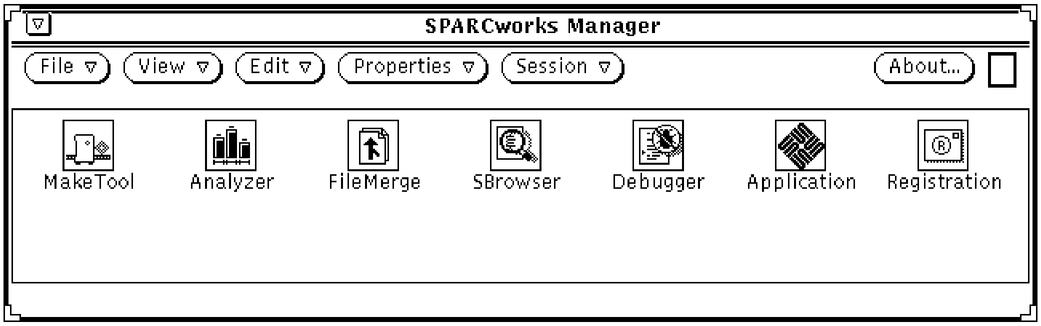


Figure 2-1 Manager Window

When FileMerge starts successfully, the FileMerge window opens, as shown in Figure 1-1 on page 1-3.

See *Managing the Toolset* for information on setting startup properties for tools.

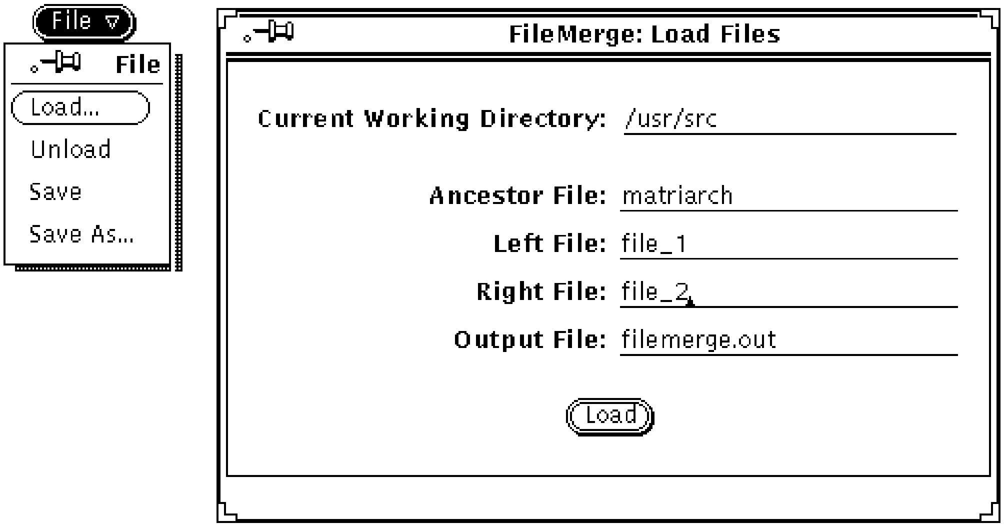
Note – The ProWorks and SPARCworks Manager palettes are identical except for the product name in the header.

2.2 Loading Files from the FileMerge Window

To load files in FileMerge:

♦ **Select Load from the File menu.**

The resulting pop-up window provides text fields in which to enter file names. As an example, from the directory`/usr/src, matriarch` has been entered as the ancestor file, `file_1` as the left file, and `file_2` as the right file.



Current Working Directory: /usr/src

Ancestor File: matriarch

Left File: file_1

Right File: file_2

Output File: filemerge.out

Figure 2-2 Specifying Files to Load and Save

Current Working Directory

This text field shows the current working directory whenever you start FileMerge from the Manager or from the command line with no arguments. You can edit this field.

FileMerge interprets the file names you specify in the window as relative to the current working directory. Therefore, you can use such constructs as *subdir/filename* to specify a file in a subdirectory and *../filename* to specify a file in a parent directory. Any file name you specify that begins with a "/" character is interpreted as an absolute path name, not as relative to the current working directory.

Ancestor File

If you enter the name of an ancestor file, FileMerge will compare it to the files to be merged and identify lines in those files that differ from the ancestor. The automerged file will be based on the ancestor file, but the ancestor file itself is not displayed in any FileMerge window.

If you do not enter an ancestor file name, FileMerge compares only the left and right files and derives the output file from them. Automerging is not possible without an ancestor file.

Left File

The file you specify in this text field appears in the left text pane.

Right File

The file you specify in this text field appears in the right text pane.

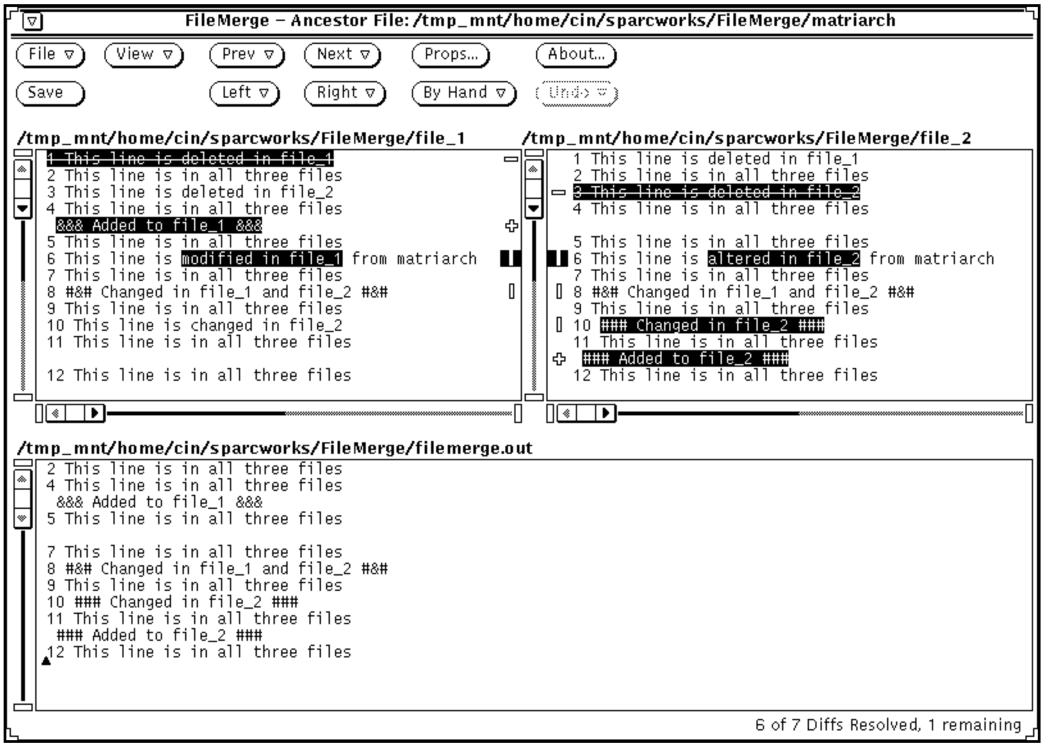
Output File

The merged output file takes the name you specify in this text field. FileMerge uses the name `filemerge.out` unless you specify a different name, and stores the file in the current working directory.

Load Button

Click on the Load button to load the files you have specified in the text fields.

Figure 2-3 illustrates a loaded FileMerge window. Note that the names of the left file, right file, and output file are displayed above each text pane. The name of the ancestor file (for a three-way diff only) is displayed in the window header.



FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

6 of 7 Diffs Resolved, 1 remaining

Figure 2-3 FileMerge Window with Loaded Files and Common Ancestor

2.3 Starting FileMerge from the Command Line

Starting FileMerge from the command line allows you to

- Determine which files (if any) are loaded into FileMerge at startup
- Specify the name of the merged output file at startup
- Specify whether or not FileMerge should consider leading white space (tabs and spaces) when it identifies differences
- Specify whether FileMerge should produce a writable merged output file or function in read-only mode
- Load lists of files from specified directories
- Specify input file display names, which are displayed over the left and right text panes in the FileMerge base window (This feature is especially useful when you merge entire directories of files from a list.)

2.3.1 Basic Startup Command

To start FileMerge from the command line without loading any input files (assuming that the FileMerge executable is in your search path):

♦ **Type the following command at a command-line prompt:**

```
filemerge &
```

The command starts FileMerge (in background) without loading any files (as shown in Figure 1-1 on page 1-3).

Note – All of the command-line constructs discussed in this section can also be used to start FileMerge from the Manager. To specify a startup command, select the FileMerge icon in the Manager palette, choose Selected Tool from the Properties menu, and then enter the command in the resulting property sheet. See *Managing the Toolset* for details.

2.3.2 Command-Line Synopsis

The complete `filemerge` command is summarized below, with command options enclosed in square brackets.

`filemerge [-b] [-r] [-a` *ancestor*`] [-f1` *name1*`] [-f2` *name2*`]`
`[-l` *listfile*`] [` *leftfile rightfile* `[`*outfile*`] ]`

Options Defined

`-b`

Causes FileMerge to ignore leading blanks and tabs when comparing lines.

`-r`

Starts FileMerge in read-only mode. When you specify this option, only the input file text panes are displayed, and the output text pane is absent.

`-a` *ancestor*

Specifies an existing ancestor file of the two files to be merged (called *descendants* of the ancestor file). The merged file is based on this ancestor file and the changes to it that have been made in the descendants.

When used with the `-l` *listfile* option, *ancestor* is a directory of files, which you can load in succession from the File menu.

`-f1` *name1*

Sets the file name displayed for the first (left) file. This option is useful when a list of files is being loaded (with the `-l` option), and you want to display a name for reference only in the FileMerge window.

For example, if you are loading files from two directories that correspond to two different revisions of a product, you could use the `-f1` option to display the name `Rev1` above the left pane and the `-f2` option to display the name `Rev2` above the right pane.

`-f2` *name2*

Sets the file name displayed for the second (right) file.

`-l` *listfile*

Specifies a file that contains a list of individual file names. This option is useful for merging entire project directories.

FileMerge uses the names in *listfile* to successively load files from directories you name with the *leftfile* and *rightfile* arguments, placing the output files in the directory you name with the *outfile* argument. The names in *listfile* must match file names in the *leftfile* and *rightfile* directories. When used with the `-a` *ancestor* option, the *ancestor* argument must be a directory: FileMerge will look in the *ancestor* directory for files that have the same names as those in *listfile* and use those with matching names as ancestor files for each merge.

If you start FileMerge with the `-l` option, the Load item in the File menu changes to Load Next From List. To load successive files named in *listfile*, choose this menu item. See “Loading Files from a List File” on page 2-29 for an example.

If you specify the character “`-`” for *listfile*, FileMerge reads the list of files from standard input.

leftfile

The name of the left file to be loaded for comparison. When used with the `-l` *listfile* option, *leftfile* is a directory of files, which you can load in succession from the File menu.

rightfile

The name of the right file to be loaded for comparison. When used with the `-l` *listfile* option, *rightfile* is a directory of files, which you can load in succession from the File menu.

Note – If you use the `-l` *listfile* option, then all three input file names (*ancestor*, *leftfile*, and *rightfile*) must be directories. If you do *not* use the `-l` *listfile* option, then any two input file names can be directories, but one of the three must be a simple file name. In this case, FileMerge uses the file name to find a file with the same local name in each directory.

outfile

Specifies the name of the merged output file. If you do not specify an *outfile*, the output file is given the default name `filemerge.out`. If you want to specify a different name when you save the file, use Save As from the File menu.

When used with the `-l` *listfile* option, *outfile* names the directory to be used when each merged output file is saved. Individual file names in the *outfile* directory are the same as the names listed in *listfile*.

2.3.3 Loading Two Files at Startup

To load two files at the time you start `filemerge`, change to the directory in which the files are stored and specify the file names on the command line. To merge two files named `file_1` and `file_2`, use the following command:

```
filemerge file_1 file_2 &
```

The first file listed appears in the left text pane; the second file appears in the right pane.

2.3.4 Loading Three Files at Startup

To merge the same two files and at the same time compare them to a common ancestor named ancestor_file, change to the directory in which the files are stored and use the following command:

```
filemerge -a ancestor_file file_1 file_2 &
```

The ancestor file is not displayed, but differences between the ancestor file and the two descendants are marked, and the merged output file is based on the ancestor file.

2.3.5 Loading Files from a List File

You can sequentially load files from a list of file names. For example, suppose ancestor versions of a project's source files are stored in a directory named /src. You have been editing the files file_1, file_2, and file_3 in your directory /usr_1, and another developer has been simultaneously editing the same files in the directory /usr_2. You have been given the responsibility of merging the changes to both sets of files, and you want to place the merged versions in a directory named /new_src.

To merge the /src, /usr_1, and /usr_2 directories, you first create a list file that contains only the names of the three files to be merged with each name on a separate line, as follows:

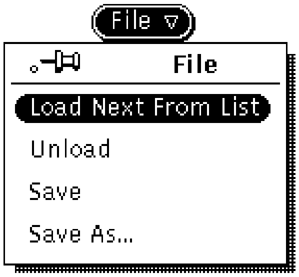
```
file_1
file_2
file_3
```

Name the file sourcelist and place it in the working directory where you plan to start FileMerge. Change to that directory (with the cd(1) command) and start FileMerge with the following command:

```
filemerge -a /src -l sourcelist /usr_1 /usr_2 /new_src &
```

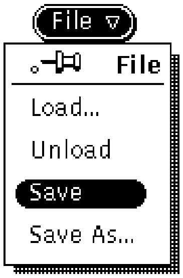
This command causes FileMerge to load /usr_1/file_1 into the left text pane, /usr_2/file_1 into the right text pane, and compare both files to the common ancestor /src/file_1. Note that the command could alternatively be placed in the Selected Tool property sheet in the Manager, which would cause FileMerge to start with the same options when started from the Manager (see *Managing the Toolset* for more information).

After you have resolved the differences between the two files, choose Save from the File menu to automatically save the output file as `/new_src/file_1`.



To load the next file from the list, choose Load Next From List from the File menu. Repeat the process until all three pairs of files have been merged.

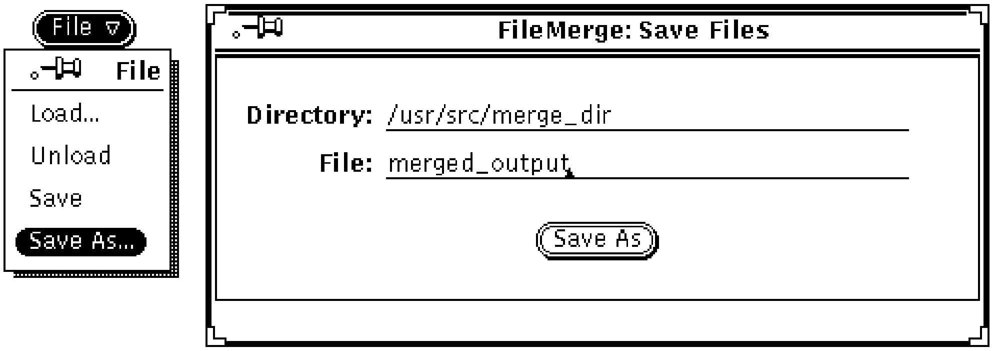
2.4 Saving the Output File



Save the output file by clicking on the Save button or choosing Save from the File menu. The name of the output file is the name you specify in the Output File text field of the Load pop-up window, as explained in Section 2.2, “Loading Files from the FileMerge Window,” on page 2-22.

As explained in Section 2.3.5, the Load item on the File menu changes depending on whether or not you are loading files from a list; however, Save and Save As are available in either case.

To change the name of the output file while saving, choose Save As and fill in the new file and directory names in the resulting pop-up window, as shown in the following figure.



Directory: /usr/src/merge_dir

File: merged_output

Figure 2-4 Saving an Output File Under Another Name

Examining Differences 3

This chapter further explains some features and presents an example of how to use FileMerge. The chapter is organized into the following sections:

3.1 Moving Between Differences

When Auto Advance has been set in the Properties pop-up window, FileMerge automatically moves to the next unresolved difference immediately after you resolve the current difference.

To move manually from one difference to another, use the Next and Prev menus. For details, see the sections “Next Menu” on page 1-12 and “Prev Menu” on page 1-11.

3.2 Resolving Differences

A difference is represented by a blank line in the merged (output) file in the lower text pane. To resolve a difference, you edit the line displayed there in one of the following ways:

- Accept the line displayed in the left pane and incorporate it into the merged file by choosing an item from the Left menu.
- Accept the line displayed in the right pane and incorporate it into the merged file by choosing an item from the Right menu.
- Edit the line in the merged file by hand and choose an item from the By Hand menu.

3.3 Automatic Merging

If you have loaded a common ancestor file, FileMerge is often able to resolve differences automatically, based on the following rules:

- If a line has not been changed in either descendant (it is identical in all three files), it is placed in the merged file.
- If a line has been changed in only one of the descendants, the changed line is placed in the merged file. A change could be the addition or removal of an entire line, or an alteration to some part of a line.
- If identical changes have been made to a line in both descendants, the changed line is placed in the merged file.
- If a line has been changed differently in both descendant files so that it is different in all three files, FileMerge places no line in the merged file. You must then decide how to resolve the difference — either by using a line from the right or left file, or by editing the merged file by hand.

3.4 An Example

This example merges two files that have a common ancestor. The files are `file_1` and `file_2`, and the ancestor file is named `matriarch`. The descendant files `file_1` and `file_2` were derived from `matriarch` by editing. The edits are intended to show all varieties of changes that could occur in the descendants: deleting lines, adding new lines, and changing lines.

The content of each line in the example helps to identify whether or not it was changed, and how. The ancestor file contains only twelve lines and is shown in Figure 3-1 on page 3-35.

Note – FileMerge does not number lines in the files it loads; the numbers are part of the example text and were placed there for clarity.

```
1 This line is deleted in file_1
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
5 This line is in all three files
6 This line is changed in descendants
7 This line is in all three files
8 This line is changed in descendants
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files
12 This line is in all three files
```

Figure 3-1 Ancestor File (`matriarch`)

Figure 3-2 shows the contents of `file_1`. This file is identical to `matriarch` with the following exceptions:

- The line numbered 1 in the `matriarch` file was deleted in `file_1`.
- A new line was added following the line numbered 4.
- The line numbered 6 was changed (a different change was made to this line in `file_2`).
- The line numbered 8 in the `matriarch` file was changed (an identical change was made to this line in `file_2.`)

```
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files
12 This line is in all three files
```

Figure 3-2 Descendant File (`file_1`)

Figure 3-3 shows the contents of `file_2`. This file is identical to `matriarch` with the following exceptions:

- The line numbered 3 in the `matriarch` file was deleted.
- The line numbered 6 was changed (a different change was made to this line in `file_1`).
- The line numbered 8 was changed (an identical change was made to this line in `file_1`).
- The line numbered 10 was changed (no change was made to this line in `file_1`).
- A new line was added following the line numbered 11.

```
 1 This line is deleted in file_1
 2 This line is in all three files
 4 This line is in all three files
 5 This line is in all three files
 6 This line is altered in file_2 from matriarch
 7 This line is in all three files
 8 #&# Changed in file_1 and file_2 #&#
 9 This line is in all three files
 10 ### Changed in file_2 ###
 11 This line is in all three files
  ### Added to file_2 ###
 12 This line is in all three files
```

Figure 3-3 Descendant File (`file_2`)

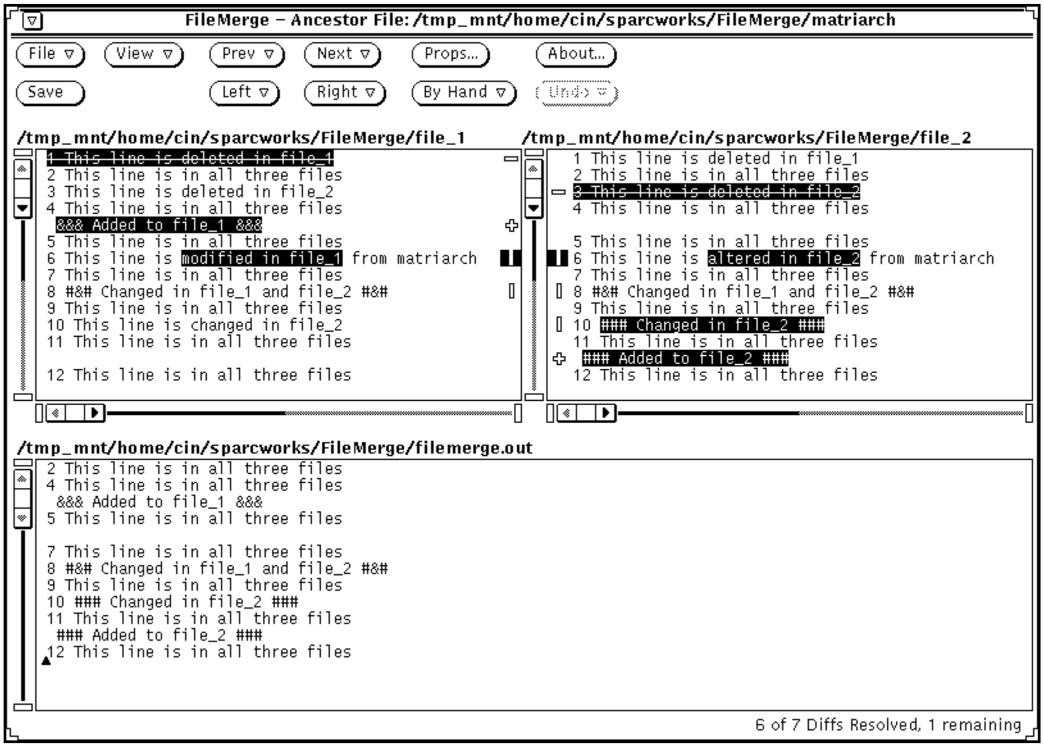
To simplify the example, we place all three files in one directory, and use this directory as the working directory where we will start FileMerge.

3.4.1 Starting FileMerge

♦ **Go to the directory in which** `matriarch`**,** `file_1`**, and** `file_2` **are stored. Type the following to start FileMerge in background mode and load the three files:**

```
filemerge -a matriarch file_1 file_2 &
```

FileMerge starts up with the Auto Merge feature turned on by default. The window that appears is shown in Figure 3-4 and displays an automerged output file.



FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

6 of 7 Diffs Resolved, 1 remaining

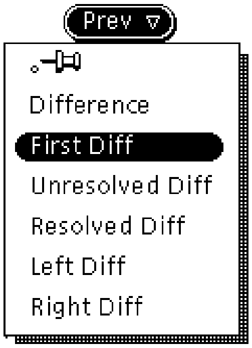
Figure 3-4 Loaded FileMerge Window at Startup

In the right footer (lower right corner of the window), FileMerge has reported finding seven differences, of which only one remains unresolved—six differences were resolved by automerging, and are marked by glyphs in outline font. The unresolved difference (line 6) is marked by a vertical bar. See Section 1.2.2, “FileMerge Glyphs,” on page 1-4 for an explanation of glyphs.

3.4.2 Examining Differences

FileMerge highlights the unresolved difference, which it identifies as the line numbered 6 in `file_1` and `file_2`. The remainder of this chapter examines each of the differences and resolves the single unresolved difference.

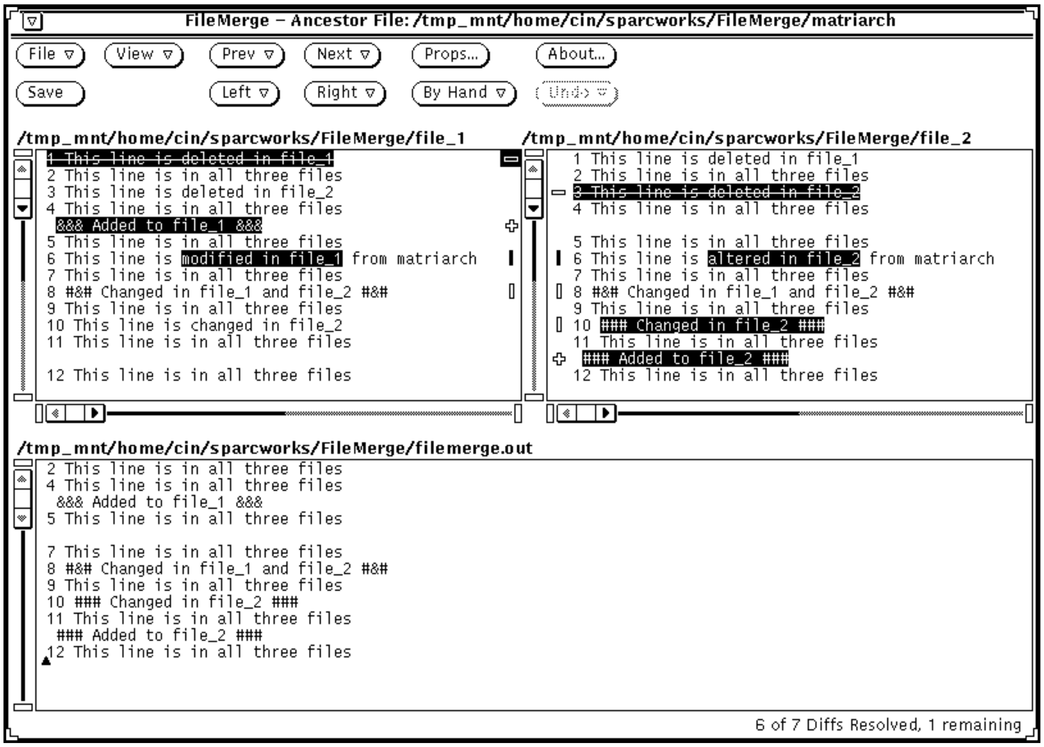
First Difference



To scroll to the first difference, select First Diff from the Prev menu.

The resulting FileMerge window (`filemerge.out`) is shown in Figure 3-5. The highlighted glyph in the `file_1` text pane indicates that the first difference is now the current difference. Next to the highlighted glyph in `file_1` is highlighted strikethrough text, which indicates that the difference occurred because the line numbered 1 in the `matriarch` file was deleted in `file_1`.

The line deleted from `file_1` is still in `file_2`, but FileMerge automerges the change so that the deleted line is not included in the output file. When using Auto Merge, a deleted line from either file is not included in the output file.


FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch
File ▽
View ▽
Prev ▽
Next ▽
Props...
About...
Save
Left ▽
Right ▽
By Hand ▽
Undo ▽

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

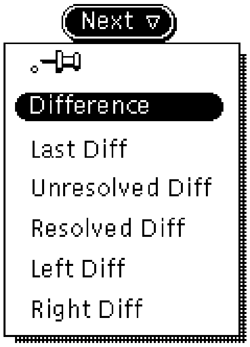
```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

6 of 7 Diffs Resolved, 1 remaining

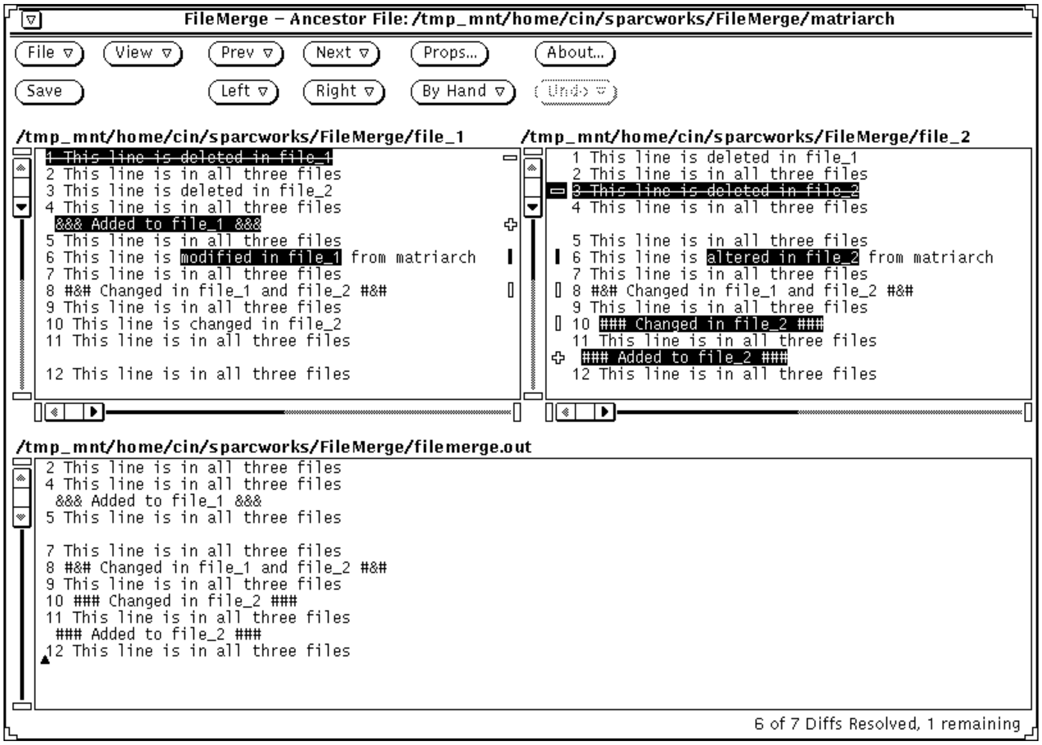
Figure 3-5 First Difference

Second Difference



Proceed to the next difference by choosing Difference from the Next menu (or simply by clicking on the Next button—Difference is the default). The next difference becomes the current difference.

The second difference is the result of deleting the line numbered 3 in the `matriarch` file from `file_2`. Auto Merge resolves the difference by deleting this line in the output file, just as in the first difference.



FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

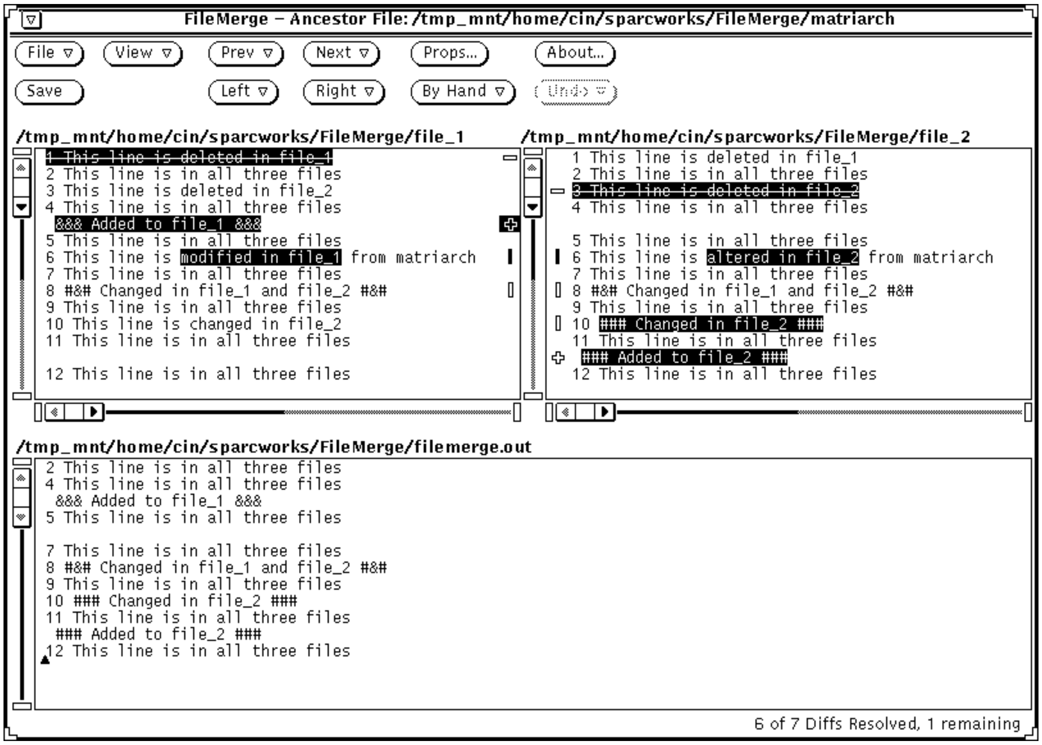
6 of 7 Diffs Resolved, 1 remaining

Figure 3-6 Second Difference

Third Difference

Proceed to the third difference by clicking on Next.

The third difference is the line that was added to `file_1`. No change was made at this point in `file_2`, and FileMerge resolves the difference by adding the line to the output file. Auto Merge preserves a change that was made to one file if no change was made in the other file.


FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch
File ▽
View ▽
Prev ▽
Next ▽
Props...
About...
Save
Left ▽
Right ▽
By Hand ▽
Undo ▽

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

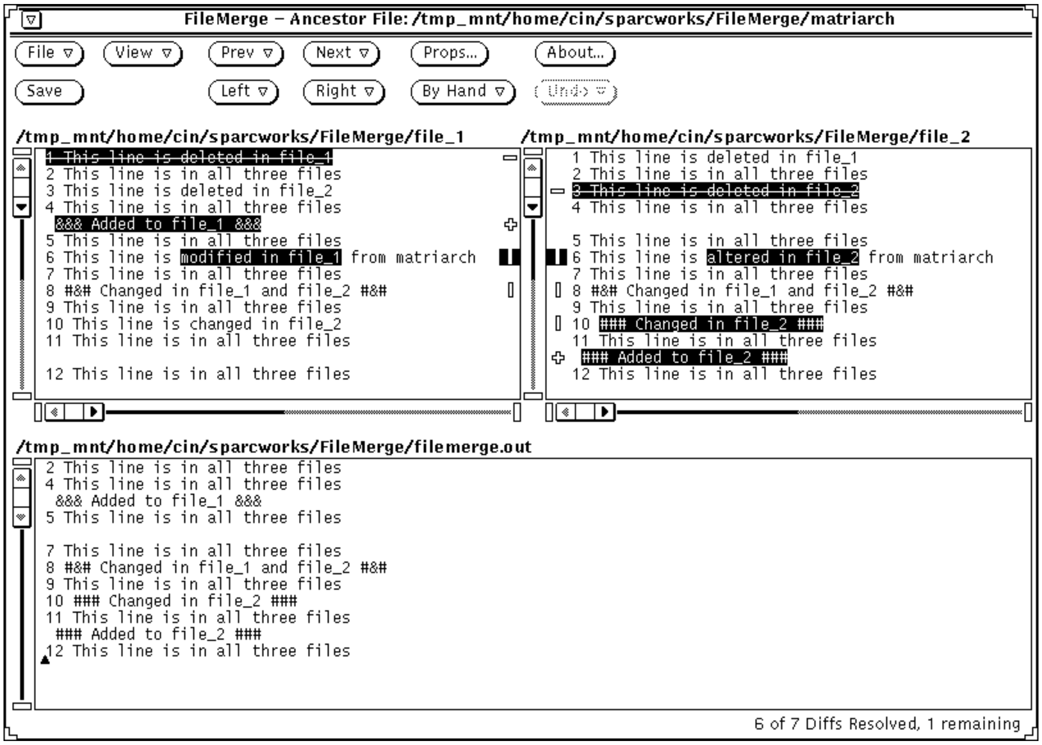
6 of 7 Diffs Resolved, 1 remaining

Figure 3-7 Third Difference

Fourth Difference (Unresolved)

Proceed to the fourth difference by clicking on Next.

The fourth difference has not been resolved by Auto Merge, as indicated by the solid highlighted glyph next to the lines involved in the difference. The vertical line indicates that the line has been changed (as opposed to added or deleted). In this case, Auto Merge failed because the same line was changed differently in the two files, and FileMerge could not decide which change was more valid.


FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch
File
View
Prev
Next
Props...
About...
Save
Left
Right
By Hand
Undo

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files

7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

6 of 7 Diffs Resolved, 1 remaining

Figure 3-8 Fourth Difference (Unresolved)

You could resolve this difference in one of the following ways:

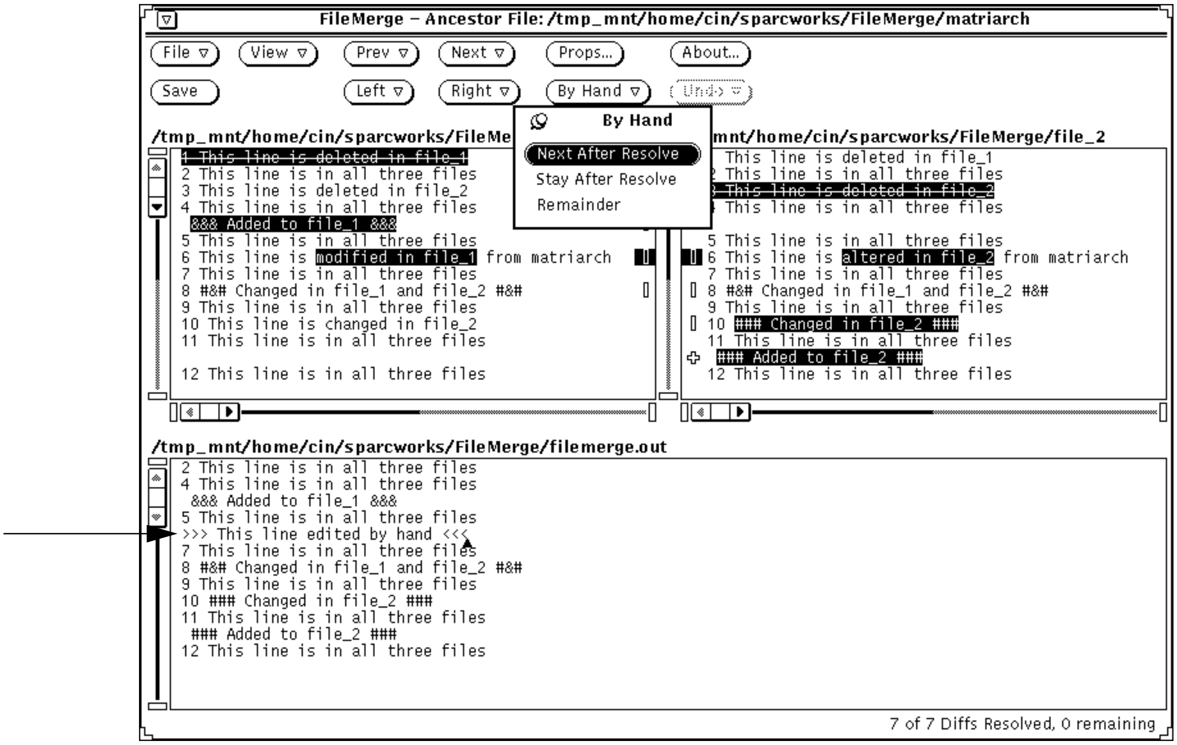
- Click on the Left button to place the line from `file_1` into the output file.
- Click on the Right button to place the line from `file_2` into the output file.
- Edit the output file by hand.

Editing the Output File

To edit the output file, move the pointer into the output file's text pane and place it in the line you want to change. In this example, the following line was typed in:

```
>>> This line edited by hand <<<
```

Choose the Next After Resolve item from the By Hand menu (or simply click on the By Hand button, since Next After Resolve is the default), as shown in the following figure. This menu item resolves the difference and would automatically scroll to the next unresolved difference if one existed. In this example there are no more unresolved differences, so the fourth difference remains the current one.



/tmp_mnt/home/cin/sparcworks/FileMe

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

mnt/home/cin/sparcworks/FileMerge/file_2

```
  This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
>>> This line edited by hand <<<
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

7 of 7 Diffs Resolved, 0 remaining

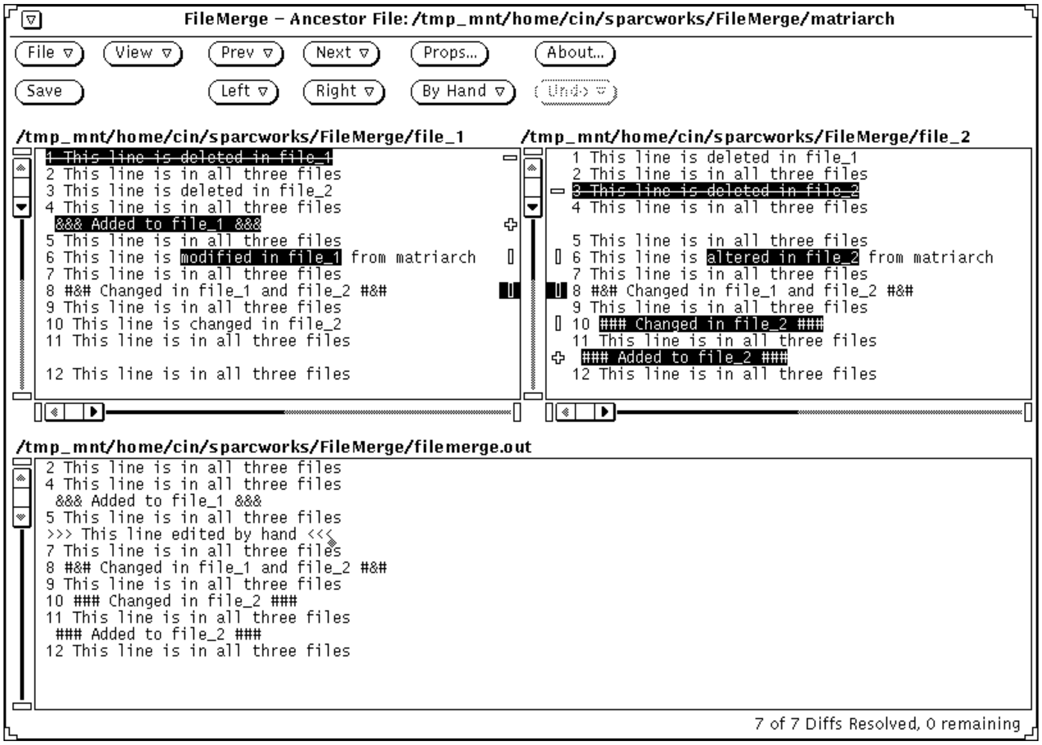
Figure 3-9 Resolving a Difference By Hand

Fifth Difference

Proceed to the fifth difference by clicking on Next.

Note that the message in the lower right corner of the window ("7 of 7 Diffs Resolved, 0 Remaining" as shown in Figure 3-10) now indicates that all differences have been resolved. Nevertheless, proceed to verify the automerged differences.

The fifth difference results from changes made to the line numbered 8 in both `file_1` and `file_2`. Unlike line 6, which was changed differently in both files, the changes to line 8 were identical. FileMerge accepts the difference and automerges the changed line into the output file.


FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch
File ▽
View ▽
Prev ▽
Next ▽
Props...
About...
Save
Left ▽
Right ▽
By Hand ▽
Undo ▽

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
>>> This line edited by hand <<<
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

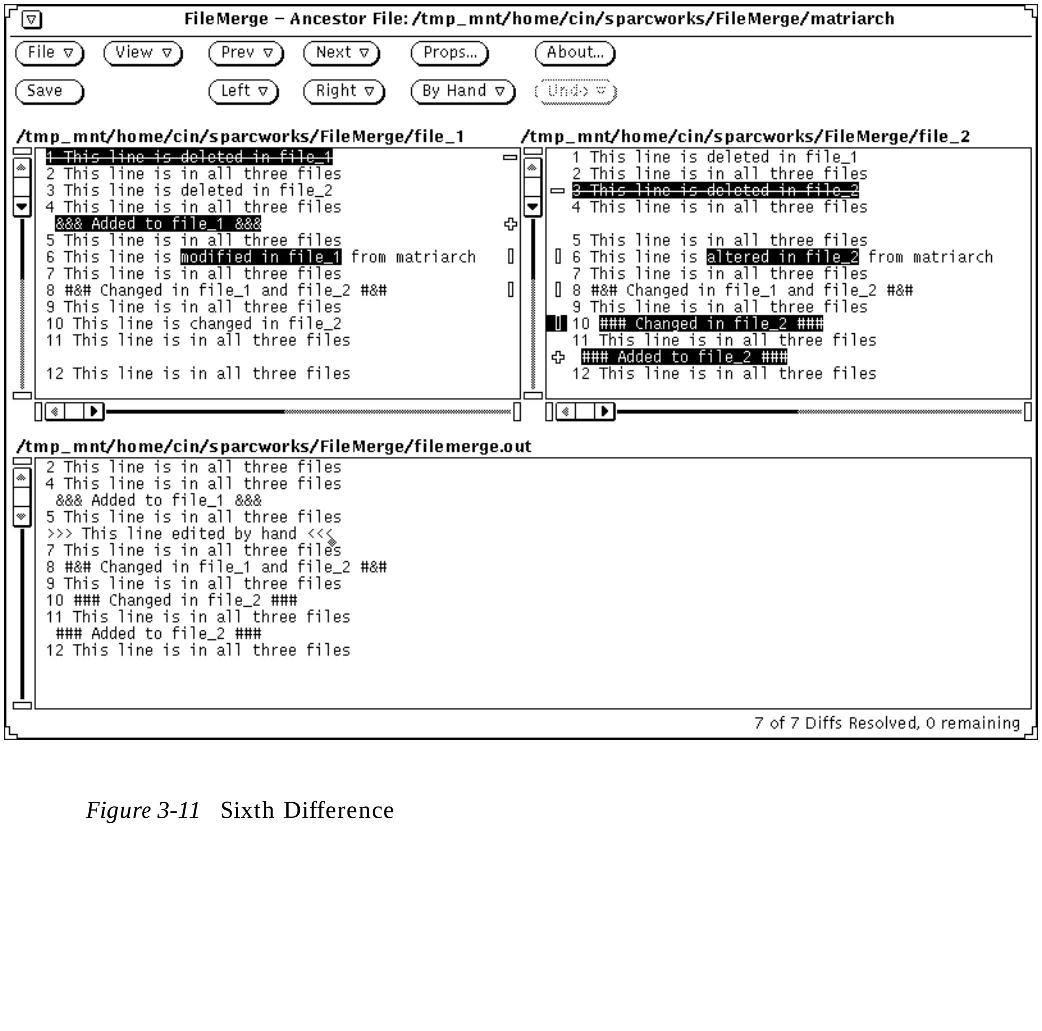
7 of 7 Diffs Resolved, 0 remaining

Figure 3-10 Fifth Difference

Sixth Difference

Proceed to the sixth difference by clicking on Next.

The sixth difference results from a change to the line numbered 10 that was made only in `file_2`. FileMerge places the changed line in the output file.



FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
>>> This line edited by hand <<<
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

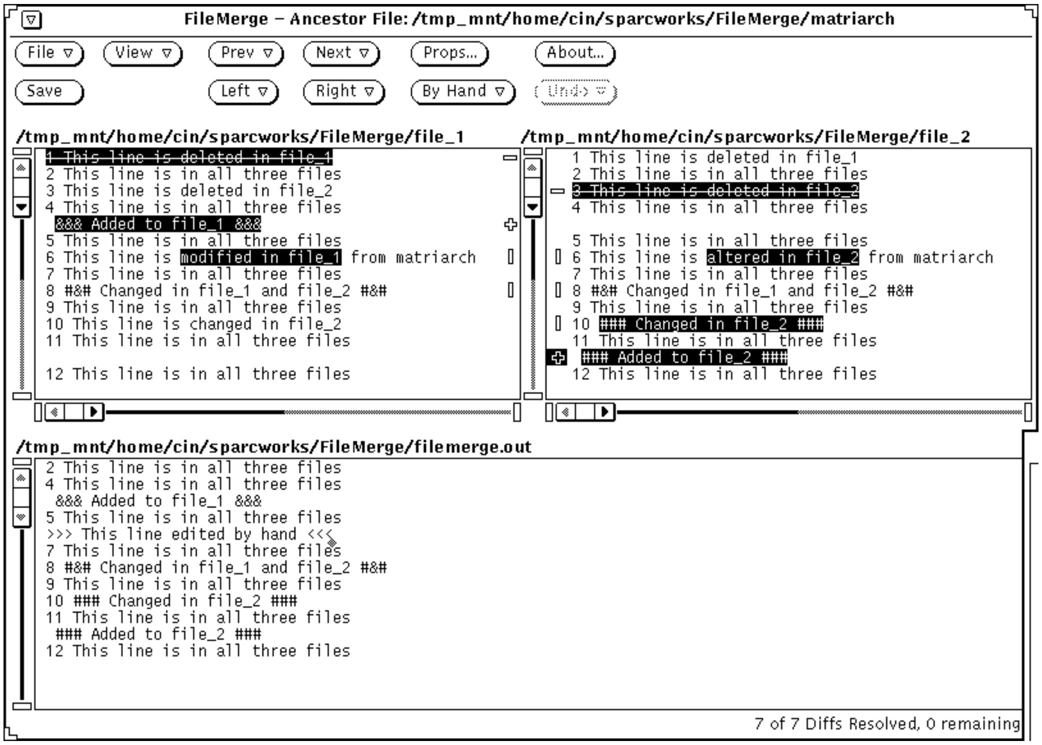
7 of 7 Diffs Resolved, 0 remaining

Figure 3-11 Sixth Difference

Seventh Difference

Proceed to the seventh difference by clicking on Next.

The seventh (and last) difference results from a line that was added only to `file_2`. FileMerge places the new line in the output file just as it did when a new line was added to `file_1`, which resulted in the third difference.



FileMerge – Ancestor File: /tmp_mnt/home/cin/sparcworks/FileMerge/matriarch
File ▽
View ▽
Prev ▽
Next ▽
Props...
About...
Save
Left ▽
Right ▽
By Hand ▽
Undo ▽

/tmp_mnt/home/cin/sparcworks/FileMerge/file_1

```
~~1 This line is deleted in file_1~~
2 This line is in all three files
3 This line is deleted in file_2
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
6 This line is modified in file_1 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 This line is changed in file_2
11 This line is in all three files

12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/file_2

```
1 This line is deleted in file_1
2 This line is in all three files
~~3 This line is deleted in file_2~~
4 This line is in all three files

5 This line is in all three files
6 This line is altered in file_2 from matriarch
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

/tmp_mnt/home/cin/sparcworks/FileMerge/filemerge.out

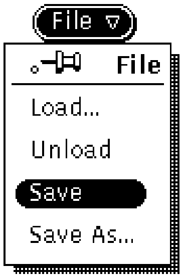
```
2 This line is in all three files
4 This line is in all three files
 &&& Added to file_1 &&&
5 This line is in all three files
>>> This line edited by hand <<<
7 This line is in all three files
8 #&# Changed in file_1 and file_2 #&#
9 This line is in all three files
10 ### Changed in file_2 ###
11 This line is in all three files
 ### Added to file_2 ###
12 This line is in all three files
```

7 of 7 Diffs Resolved, 0 remaining

Figure 3-12 Seventh Difference

3.4.3 Saving Output File

Now that all differences now have been resolved and the automerged differences verified, you can save the output file. The output file takes the name shown in the Load Files pop-up window, which by default is `filemerge.out`.



To write the file, choose Save from the File menu.To save the file under another name, use Save As.

Troubleshooting A

This appendix describes how to overcome problems when using FileMerge. It is organized into the following sections:

A.1 Troubleshooting Checklist

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

❏ `filemerge` is installed in the standard location. If you do not know where `filemerge` should be installed, contact your system administrator.

❏ `filemerge` can be found in your search path (as set by the `PATH` variable).

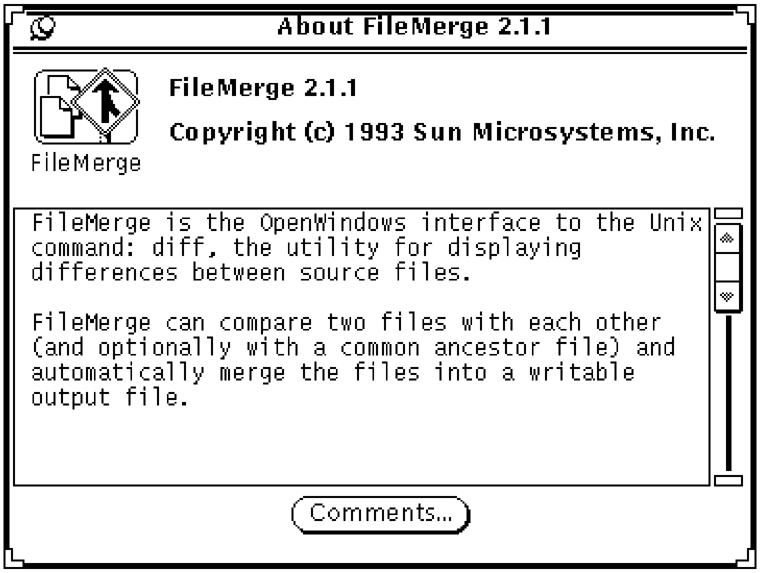
❏ You have chosen the desired settings in the Properties pop-up window.

A.2 Reporting Problems

If you have gone through the checklist and still have problems using FileMerge, call Sun Microsystems at 1-800-USA-4SUN or your local service office. Have FileMerge's version 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 FileMerge's version:

♦ **Click on the About button.**
The resulting pop-up window shows the product release number.



FileMerge 2.1.1

Copyright (c) 1993 Sun Microsystems, Inc.

```
FileMerge is the OpenWindows interface to the Unix
command: diff, the utility for displaying
differences between source files.

FileMerge can compare two files with each other
(and optionally with a common ancestor file) and
automatically merge the files into a writable
output file.
```

A.3 FileMerge Error Messages

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

`filemerge: too many differences`
FileMerge cannot handle the number of differences your files have generated. Try segmenting the files and merging them piece by piece.

`filemerge: trouble performing diff. Perhaps you are diffing a binary file?`
This message indicates that FileMerge encountered non-text characters while attempting to find file differences. Make sure your files are text files and that they have not been corrupted.

`Sorry, textedit Undo is not supported in filemerge`

This message appears when you attempt to use the Edit: Undo item in a text pane pop-up menu. Use the FileMerge Undo button instead.

Index