The FileMerge Application

FileMerge lets you easily compare two plain text (ASCII) files, two rich text (RTF) files, or two directories. The files are displayed side-by-side, and their differences are highlighted. FileMerge also lets you merge the two files or directories, creating a third file containing passages from either of the original files.

Note: FileMerge is a demonstration application in Release 4.0. It will eventually take its place in as a fully supported NEXTSTEP application, but for now consider it (and this documentation) preliminary.

Specifying the Files or Directories to Compare

To use FileMerge, you first specify two files or directories to be compared. You can do this in three different ways:

- In FileMerge, choose Compare Files from the File menu. This displays a panel that lets you specify the two files to be compared. You specify the files by dragging them into the icon wells from the File Viewer, by typing their names in the text fields, or by clicking the Left and Right buttons adjacent to the text fields and using the standard Open panel to navigate to each file.

  CompareFiles.tiff ¬

- In applications that let you select files (such as Workspace Manager), select two files or directories and then choose the Compare Files command from the Services menu. FileMerge will place the older of the two files in the left panel and the newer in the right.

- From the command line, invoke FileMerge by issuing the the **opendiff** command (**/usr/bin/opendiff**), specifying the two files to be compared:

  `opendiff` ***file1 file2***

file1 will appear in the left pane.

Once you've specified the files to be compared, FileMerge opens a Comparison window and highlights the differences.

Ancestors

Frequently, you want to simply view the differences between two files. For example, if you have made changes to a file and want to review the changes before releasing the modified copy. In this situation, you might use the merge facility to simply omit some of the modifications that were made. The merge file initially appears identical to the modified ("right") file and you can iterate through each change, deciding whether to keep the change or not.

Another, perhaps less common, but more difficult situation is one in which a file has been modified twice, independently. For example, you and a colleague both make copies of a file and each modifies the copy independently. This is often referred to as branching. In this

situation you will want to specify an ancestor file. The ancestor file should be the file as it existed before it was modified. One of the modified files is specified as the "left" file and the other as the "right" file. "So, what does the ancestor file do?", you might ask.

The ancestor file allows FileMerge to selectively take changes from the left file and the right file. For example: if 3 lines are removed from the right file (relative to the left) file, how can one tell whether they were specifically deleted from left or whether they were added to right. By verifying whether those lines existed in the ancestor file the system is able determine the answer to these types of questions.

When merging two files with an ancestor file available, the system will "suggest" whether to choose "left" or choose "right" for each difference. Note that if some series of lines is changed in both the "left" and "right" files, FileMerge will not be able to determine which choice is correct. This situation is referred to as a conflict and requires that you specifically choose. A diff that is a conflict is shown with a red border.

Specifying Ancestor Files

To specify an ancestor file, resize the Compare Files panel (which you bring up by choosing Compare Files from the File menu). Initially, this panel only shows two fields. If you make it longer, two additional fields appear. One of these fields is the path name for the ancestor. If you're specifying directories, then the ancestor should also be a directory. If you're specifying files, then the ancestor should be a file.

The second field is a "merge" path. Specifying the merge path is most useful when merging a number of files in a directory. FileMerge will compute the name of individual merged files relative to the path given here.

CompareFilesWithAncestor.tiff ¬

If you're using the **opendiff** command, use the **-ancestor** and **-merge** flags to specify the ancestor and merge files, respectively. For example:

```
opendiff version1 version2 -ancestor version0 -merge version3.
```

Comparing Files

When you compare two files, the Comparison window appears to show you the changes. The Comparison window is divided into two panes; the top one displays the two files being compared, and the bottom one displays a file that would result from merging the top two files:

Comparison.tiff ¬

The differences between the two files are highlighted in the top pane. Where a line has been added, the highlighting tapers from a fine gray line to a wider one that encloses the addition. Where a line has been removed, the highlighting tapers in the opposite direction. Each difference (known as a *diff*) is also given a number, which is optionally (based on your preference) displayed in the central column.

As you move the cursor over the central column, the cursor changes to an arrow. Using the arrow cursor, you can drag the column to the left or right, revealing more of one or the other file. If you move the cursor over a diff clicking the mouse will not move the central column but rather will select the diff.

Selecting *diffs*

You can select a diff by clicking it. A selected diff displays a black outline (as illustrated by diff 2 in the figure above). You can select the next or previous diff by using the arrow keys or by choosing the Next Difference or Previous Difference commands from the Find menu. To select multiple diffs, Shift-click each one in turn.

Merging

The bottom pane (called the *merge view*) of the Comparison window displays the file that is the result of merging the two files under comparison. Notice that there's a splitview bar at the bottom of the window. Dragging the splitview displays this bottom pane. When an ancestor file is not specified, the merge file will initially be identical to the "right" (typically newer) file. In either case, you can review each diff and select whether the diff (or change) should be applied in the merge file.

You select which side of a diff to merge using the commands in the Actions pull-down menu:

Choose left	The merged file should contain the left side of the diff.
Choose right	The merged file should contain the right side of the diff.
Choose both (left first)	The merged file should contain both sides of the diff, with the left appearing above the right.
Choose both (right first)	The merged file should contain both sides of the diff, with the right appearing above the left.
Choose neither	The merged file should not contain the diff.

You can also use the left and right arrow keys to choose the left or right side of a diff (this is especially useful when you use the up and down arrow keys to navigate to the previous or next diff).

The merge view can be edited manually to fix up minor problems. (Currently, major changes to the number of newlines will cause the synchronized scrolling of the merge view to be incorrect).

Save the merged file by choosing the Save Merge or Save Merge As command from the File menu.

Comparing Rich Text Format Files

FileMerge can compare Rich Text Format (RTF) files, but this feature should be considered especially experimental. When comparing RTF files, the comparison is done on the textual contents of the file, not the formatting. Changing the font in some portion of the file won't be reflected as a change. Also, given that most paragraphs in a rich text file consist of single long lines (even though they're displayed as multiple lines due to wrapping), a single change in a paragraph will be displayed as a though the whole paragraph has been changed.

Comparing Directories

When comparing two directories, a window with a browser appears. The browser contains all files from both directories, that is, all of the files from the "left" directory plus all of the files from the "right" directory.

DirectoryComparison.tiff ¬

Each file in the browser is in one of these states:

·€Gray titles denote a file that is identical in the "left" and the "right" directory.
·€Black titles denote a file that is contained in both directories, but that has different contents.
·€Italicized titles denote a file that exists in only one of the two directories.

Selecting a file displays information about the comparison in the status line at the bottom of the window.

The Exclude check boxes let you control which files are visible in the browser:

Identical	Excludes all files that are the same in both directories. Frequently it's easier to see what's changed if you exclude identical files.
Changed left	Excludes files that were only changed on the left (relative to the ancestor file)
Changed right	Excludes files that were only changed on the right.
Added/Deleted	Excludes files that were added or deleted in either directory.

You can double click a file name (or choose Comparison from the View pop-up list) to bring up the file Comparison window. You can also use the View pull-down list to open files:

Comparison file	Opens the Comparison window.
Left file	Opens the file from the left directory using the Workspace.
Right file	Opens the file from the right directory using the Workspace.

Ancestor file Opens the ancestor file using the Workspace.

Merge file Opens the merge file using the Workspace.

Merging Directories

You can merge directories without comparing each individual file in a file comparison window. The Merge pull-down list in the directory comparison window allows you to choose amongst several ways of merging individual files:

Combine files Merges the two files based on the ancestor file and writes the result to the merge directory.

Use left The merge directory should contain the file from the left directory. Use this if you know that the version of the file in the left directory is the correct one.

Use right	The merge directory should contain the file from the right directory. If the file has been deleted in the right directory, the file does not appear in the merge directory.
Remove	The merge directory should not contain the file.
Unmark	When a file is merged (using any of the 4 commands above) it gets a check mark in its display in the browser. Choosing Unmark will clear the checkmark. This is useful if you decide that you've made a mistake on a particular file and want to remind yourself to come back to it.
Show failures	Shows the alert panel that also appears if any of these commands fail.

If the file contains any conflicts, it will not be written and a special alert panel will come up. This alert panel shows all of the files that could not be written because they contained conflicts (or for other reasons such as insufficient permissions).

If no merge directory was specified when the comparison was performed, FileMerge will prompt you for the base directory of the merge. Individual files will be written relative to that base directory.

You can select more than one file and then click one of the merge operations. This will cause the merge operation to be applied to each selected file in succession. Note that if you apply one of the merge operations to a directory, it will apply the operation to each file in the directory, not just the files which are visible in the browser.

Preferences

FileMerge's Preferences panel contains two views: Options and Filters.

The Options view contains controls for the following:

Wrapping	When enabled, all files are displayed with lines wrapped. Lines wrap for RTF files regardless of this preference's setting.
Number differences	When enabled, each difference is numbered. This is

	convenient for remembering locations or describing locations to others ("look at difference 23..."). This is enabled by default.
Show merge direction	When enabled, each difference is marked with an arrow pointer either to the left or the right. The arrow indicates whether the change from the left file or the right file is selected in the merge file, respectively. This is enabled by default.
Show change in scrollbar	When enabled, each difference is shown as a tick mark in the scrollbar. This can be useful for finding all of the differences. This is enabled by default.
Highlight differences	When enabled, the differences between the two files are shown in the selected color. For example, if you added a semi-colon to a line in the right file, the line is highlighted, and the semi-colon appears in red (or the selected color).
Exclude Files	Initial controls for which files should appear in the browser for

directory comparisons. You can override these settings by using the Exclude controls on the directory comparison window itself.

The Filters view controls which filters are applied to different types of files. A filter is a program which takes as a file as input and produces a different version of the file to be compared. For example, **nibTool** is a filter that produces an ASCII representation of a nib file. Similarly, RTF files are filtered into their ASCII representations (where RTF controls are represented by a *xx*) before they are compared.
You could write a filter that would, for example, exclude comments from a source code file and have that filter applied before any comparison of code files by including it in this table.

Each filter specifies the following information:

Extension	The type of file to which this filter should be applied.
Filter	The path to the filter program.
Display	Controls whether or not FileMerge should display the input to the filter or the output from the filter in the Comparison

	window. Original means the Comparison window should display the file before the filter was applied. Filtered means the Comparison window should show the result of the filter command. For example, when you are comparing nib files, you'll want the Comparison window to show the filtered versions of the nib files because showing the original versions would be meaningless. However, when displaying the result of a comparison of RTF files, you'll want to see the original RTF versions rather than the ASCII versions.
Apply	Controls whether the filter applies to directory comparisons. If Yes, FileMerge will apply the filter to the files in both the left and the right directories and compare the output. If No, the filter is not used in directory comparisons.
Files to ignore	Files that match any of the patterns listed will be ignored when comparing directories. For example *~ indicates to not consider files which end with the character ~.
Compress whitespace	If enabled, FileMerge applies a whitespace filter to all files. All

whitespace (tabs, blank lines, and spaces) are compressed into a single space before the files are compared. This is useful when you are comparing source code files and want to ignore changes such as indentation. However, when this preference is enabled, the comparison takes longer to complete.