SourceGear DiffMerge

User Manual

SourceGear DiffMerge: User Manual

Table of Contents

Chapter 1. Introduction

DiffMerge is an application to visually compare and merge files and folders on Windows, OS X, and Linux.

- **Diff:** Graphically shows the changes between two files. Includes intra-line highlighting and full support for editing.
- **Merge:** Graphically shows the changes between 3 files. Allows automatic merging (when safe to do so) and full control over editing the resulting file.
- **Folder Diff:** Performs a side-by-side comparison of 2 folders, showing which files are only present in one file or the other, as well as file pairs which are identical, equivalent or different.
- **Windows Explorer/Shell Integration:** Right-click on any two files or folders in Windows Explorer to diff them immediately.
- **Configurable:** Rulesets and options provide for customized appearance and behavior.
- **International:** DiffMerge is a UNICODE-based application and can import files in a wide range of character encodings.
- **Cross-platform:** Identical feature set on Windows, OS X, and Linux:

Chapter 2. File Diff/Merge Windows

File Window Types

There are two types of file windows – **File Diff Windows** and **File Merge Windows**.

File Diff Windows

The first is a two-way view showing the differences between two files or two versions of the same file. You can use this window to see the changes side-by-side. And you can edit the file on the right, either interactively as you would with a normal editor or by applying patches/changes from the version on the left. You can use this window, for example, when looking at the revision history of a file to see the changes between two revisions.

File Merge Windows

The second is a three-way window that shows the differences between three files; typically this is a common ancestor (in the middle) and two branches that have independently evolved (on the left and right). You can use this window to see the changes side-by-side-by-side. You can edit the center file and merge in changes from both branches, creating a common/unified version.

Opening File Windows

You can open a File Diff Window by selecting **File | Open Diff Window** from the menu. This raises the **Select Files to Compare Dialog** and allows you to type or browse to the pathnames of the files you wish to compare:

If a **Swap** button is provided, it will swap the left and right pathnames. Keep in mind that only the right/bottom file will be editable.

You can open a File Merge Window by selecting **File | Open Merge Window** from the menu. This raises the **Select Files to Merge Dialog**:

Note: in both cases, if the set of files selected are already open in an existing window, that window will be raised rather than creating a new, duplicate window.

Parts of a File Window

Here is an example of a File Merge Window.

File Panels

In the center of File Diff and Merge Windows DiffMerge displays 2 or 3 files in a series of **File Panels**. Each panel looks like a basic text editor window with a few differences.

- **A Panel Label** is drawn above the panel. This may contain the pathname of the file or a symbolic name describing the version of the file.
- **Line Numbers** are optionally displayed next to each line of the file.
- **Invisible characters** such as whitespace and line terminators are optionally displayed using special symbols.
- **Bold Underlined Text** is use to indicate edited text.
- **An Edited Line Indicator (*)** is drawn in the line number area for edited lines.
- **A Void** is drawn using a shaded, diagonal hatch pattern to indicate missing lines. This is shown in Showing Changes.
- **Changes** are drawn in color. This is explained in Showing Changes.
- **A Gap** is drawn to indicate hidden lines. This is illustrated in the Display Mode and Hiding Omitted Lines [7].

The Glance Bar

The **Glance Bar** gives a visual summary of the "shape" of the changes in the files. The bar is a scaled representation of the files.

- Colored lines/blocks are drawn to indicate changes in the files; the coloring and the shape blocks reflect the color and shape of the actual changes.
- Black bars are drawn on the sides of the glance bar to show the current scroll position of the files.
- As the mouse moves over the Glance Bar, the corresponding line numbers from the files are shown in the Status Bar. A star * is used when the corresponding line is a Void.
- You may click on the glance bar to scroll the file panels to that position.

The View Selector

The **View Selector** enables you to select between the **Reference View** and the **Edit View**.

The **Reference View** is a static view of the changes in the files as originally read from disk. In this view the files are considered read-only. You may scroll around and view the differences and you may copy things to the clipboard -- but you cannot modify the files.

This mode can be useful in various situations, such as when you want to see the changes between two historical versions of a file.

The **Edit View** is an interactive editor and live difference viewer. It enables you to make changes to the center panel in a File Merge Window and the right panel in a File Diff Window. As you edit, your changes are highlighted, the files are re-compared in real-time, and their changes are displayed.

The View Selector is only present when editing is permitted on the files. When not present, the window only shows the Reference View and treats all files in the window as read-only.

Ruleset Indicator

The **Active Ruleset Indicator** shows the current **Ruleset** in use in this window. The Ruleset provides customizable rules for comparing and displaying file content (such as whitespace handling and character encodings) that are based upon file type.

Generally, the Ruleset is automatically selected based upon the file suffix. It may be changed using the File | Change Ruleset... command.

Character Encoding Indicator

The **Character Encoding Indicator** shows the character encoding of the files.

If both or all 3 files have the same character encoding, the value is listed once.

If each file has a different encoding, each is listed separately. DiffMerge compares the files **after** the files have been converted from the original character encoding into UNICODE. So files that are in different encodings can be compared.

If a file contained a UNICODE Byte Order Mark (BOM), a `(BOM)` will be shown.

The Splitter

By default, DiffMerge shows the file panels side-by-side with vertical “splitters” between them. This mode allows you to easily see and match up the corresponding lines in the files. However, you may have to scroll horizontally to see parts of a line.

DiffMerge also lets you view the files panels one above another with horizontal “splitters” between them. In this mode, it may be a little difficult to see the line correspondence, but you should be able to see the entire line in each file. In this mode, the order of the files is the same: editing takes place in the bottom panel in a File Diff Window and the center panel in a File Merge Window.

You may drag the splitter to change the relative size of the file panels. Double-click on it to restore it to the center.

The Toolbar

The toolbar above the File Diff/Merge Windows presents the following commands. Some appear only in File Merge Windows.

From left to right, these are:

View | Show All, View | Show Differences Only, and View | Show Differences with Context set the line display mode.

View | Show Line Numbers and View | Show Invisibles control the visibility of line numbers and whitespace and line terminator characters.

View | Split Windows Vertically and View | Split Windows Horizontally control whether the file panels are side-by-side or above-and-below.

File | Save File

Edit | Cut, **Edit | Copy**, and **Edit | Paste**

Edit | Undo and **Edit | Redo**

Apply Change from Left and Apply Change from Right apply the current change to the editable file.

Edit | Next Change [10] and Edit | Previous Change [10] move the cursor to the next and previous changes.

Edit | Next Conflict [10] and Edit | Previous Conflict [10] move the cursor to the next and previous conflicts.

Edit | Merge to Center performs Auto-Merge.

Showing Changes

Detail Level

DiffMerge has two Detail Levels: Lines Only and Lines and Characters.

Lines Only

In **Lines Only** mode, DiffMerge only performs difference analysis line-by-line. Changes are indicated based upon equality of the entire line.

Lines are colored without intra-line highlighting.

For example, in a File Diff Window:

And in a File Merge Window:

Lines and Characters

In **Lines and Characters** mode, DiffMerge performs line-by-line difference analysis and then within each change block it performs intra-line difference analysis.

Lines are colored using the overall line status color and then intra-line highlights are added.

For example, in a File Diff Window:

And in a File Merge Window:

Highlighting Changes

Highlighed Changes are surrounded by a dotted line. When a change is highlighted, it may be acted upon by the Apply Change... [5] commands and the custom right-mouse context menus. You can also use the Next/Previous Change [5] commands to jump between changes.

Aggregating Changes

As a precaution, adjacent or overlapping changes are aggregated into a single change. Usually this produces a conflict (depending on the individual types of changes in the grouping). When you're merging multiple files, these usually require your attention because DiffMerge cannot automatically determine what to do with them.

For example, in the figure below, "Line 1" was added to both branches (in the left and right panels) and "Line 2" was deleted from the right panel. So what should the merge result look like?

To access an individual line change within a block of changes, hold down the Control key while using the left or right mouse buttons to select the desired line.

Manual Alignment Markers

You can add **Manual Alignment Markers** to force the alignment of a set of lines in the difference analysis. Markers are drawn as a pair of dashed lines between two lines of text.

For example, in the following example a block of text has been moved. DiffMerge chose to match up the `BEGIN_EVENT_TABLE...` block and treat the `#define...` block as a delete and an insert.

If you want to force the `#define...` blocks to line up, you could insert a marker as shown here.

This causes the other block of code to appear as a delete and an insert.

You can use the **View | Delete All Manual Alignment Markers** menu command to to delete all of the Manual Alignment Markers in the window.

The Line Display Mode

The **Line Display Mode** allows you to hide lines that are not currently of interest. When lines are hidden, a single line **Gap** is drawn.

The Line Display Mode can be set to one of the following:

View | Show All. This mode shows all lines in the files.

View | Show Differences Only. This mode shows only the changes in the files. Identical lines are hidden. For example:

View | Show Differences with Context. This mode shows changes with upto 3 lines of context around each. Identical lines not adjacent to a change are hidden.

Note

The last 2 modes are only available in the Reference View.

Omitted Lines

A Ruleset can declare that certain lines be **omitted** from the difference analysis. These lines are treated as if they were not present in the file during the analysis. For example, you might use this feature to ignore differences caused by RCS Keyword Expansion.

Normally, these lines are drawn in a special gray color. For example, when we add the Regular Expression `\$Revision:.*\$`, we can see that the revision lines were omitted from the analysis.

The **View | Hide Omitted Lines** menu option causes these lines to be hidden and a **Gap** drawn instead.

Note

The Hide Omitted Lines feature is only available in the Reference View when you have the **Display Mode** set to **Show All**.

Hiding Unimportant Differences

A Ruleset can also describe the various **Contexts** in a file and classify them as **Important** or **Unimportant**. For example, changes within a source code comment are not as important as changes in the code itself or in a string literal. Furthermore, whitespace changes in C source code are not as important as whitespace changes within a string literal.

When the Detail Level is set to Lines and Characters, DiffMerge classifies all changes as either important or unimportant using the information in the Ruleset. Normally, DiffMerge draws these unimportant changes in a dimmer color. In the following example, compare the coloring of the “CHANGED” and “YOUR/MY” changes in the file panel and in the glance bar. Also, the Change Stats shows that there are 3 changes.

When the Display Mode is set to **Show All**, the **View | Hide Unimportant Differences** command causes DiffMerge to draw them as if they were identical (and with a slightly dimmer color for intra-line character changes). In the following example, the 2 comment lines are no longer highlighted. And the word “CHANGED” is dimmed slightly. The glance bar no longer shows a block for the change. The whitespace before “sz” is no longer highlighted. The Change Stats shows that there are 2 important changes and 1 unimportant change (not being shown).

If we set the Display Mode to **Show Differences Only**, we see only the 3 changes with Gaps between them.

If we now turn on **View | Hide Unimportant Differences**, the unimportant changes are not drawn at all. Again, the Change Stats indicates that 1 unimportant change is being hidden.

Change Ruleset

The **File | Change Ruleset...** menu command allows you to select a different Ruleset for the current window. This raises the **Choose Ruleset Dialog**.

The currently selected Ruleset is displayed in the status bar.

Exporting File Differences

The commands under the **Export | File Diffs** menu will export the differences between the 2 file versions to a file.

There are several output options, such as "Text" vs "HTML" mode and "Unified" vs "Traditional" vs "Side-by-Side" formats, but in all cases:

- The **current view** (Reference or Edit) is exported and if the Edit View is chosen and it has unsaved edits, those changes will be included in the output.
- The **current Ruleset** and the settings for: **Show/Hide Unimportant**, **Show/Hide Omitted**, and **Tab Size** are respected.
- When exporting in "HTML" mode, intra-line highlighting will be shown if the **Detail Level** is set to Lines and Chars.
- These Export commands always write to the destination, even if there are no differences to report. *This differs from the command line interface which for historical reasons only writes to the destination file when there are differences to report.*

Export | File Diffs | Unified

Writes the file differences to a "Text" or "HTML" file in a format *similar* to the industry standard "unified" format. For example:

Note: DiffMerge extends the industry standard format to allow for hidden and omitted lines, so our "Text" mode output, while providing you with additional information, may confuse tools like "patch".

Export | File Diffs | Traditional

Writes the file differences to a "Text" or "HTML" file in the traditional "diff" format. For example:

Export | File Diffs | Side-by-Side

Writes the file differences to a "HTML" file in a side-by-side format similar to how DiffMerge displays them in a window. For example:

Side-by-side HTML output also respects the **View | ShowView All**, **View | Differences Only**, and **View | Differences with Context** settings, so you can see as much of the 2 files as you want.

File Editing

File Diff and Merge Windows only allow 1 of the files in the set to be edited. This is called the **Editable Panel**. This is the file in the right panel in a File Diff Window and the center panel in a File Merge Window on the Edit View.

DiffMerge enables two types of editing: **Interactive Editing** and **Patching**.

Interactive Editing

The term **Interactive Editing** refers to the normal editing operations found in most text editors and includes inserting/deleting text with the keyboard and mouse and the standard cut/copy/paste clipboard operations.

Patching

The term **Patching** refers to the various commands that let you apply a change to the Editable Panel using the content of the other panel(s).

These commands are based upon concept of a Highlighted Change.

With a change highlighted, you can use the right-mouse context menu to select an action.

Patching Verbs

The content of the context menu will vary depending on the type of the change or conflict, which panel you click on, and whether you have a File Diff or File Merge window. All context menu patching commands are variations of one of the following verbs:

Insert	This option inserts content from the source panel into the **Void** in the editable panel.
Replace	This option replaces the content in the editable panel with the content from the source panel.
Delete	This option deletes the content in the editable panel.
Prepend	This option inserts the source content before the existing content in the editable panel.
Append	This option inserts the source content after the existing content in the editable panel.

The Default Action

Most of the time, the first item in the context menu will be labeled with "`(default)`". This is the **Default Action**.

> To apply the default action, you can **shift-right-click** on the change or use one of the **Apply Change...** commands.

For conflicts, there may not be a default action.

Merge to Center Panel (Auto-Merge)

> The **Merge to Center Panel** command automatically applies the Default Action [9] for all of the non-conflicting changes in a File Merge Window. The

goal of Auto-Merge is to automatically take care of the easy changes so that you can focus on the conflicts.

The complete auto-merge is performed as a batch (in 1 transaction) and may be undone using the **Undo** command.

After the Auto-Merge command completes, the **Auto-Merge Results** dialog displays a report on what changes were made, from where they came, and finally a notification about what conflicts could not be merged.

Note

It is strongly recommended that if you wish to use Auto-Merge you use it **BEFORE** making any other edits in the window. This prevents Auto-Merge from possibly reverting some of your edits.

Also, after Auto-Merge has been applied to a window, it is disabled. This is to prevent Auto-Merge from being used twice and possibly reverting some edits made by the first Auto-Merge.

Undo/Redo

DiffMerge features an unlimited Undo/Redo so that it is possible to undo changes all the way back to the initial state.

Moving Around within Files

DiffMerge offers a number of ways to move around within files.

Edit | Find... and **Edit | Go To Line...**

These commands cause the **Find Panel** to appear at the top of the File Window. The Find Panel will let you search for text or jump to a particular line in the file. This panel can stay visible or can be dismissed at any time.

This panel replaces the **Find Dialog** and **Go To Line Dialog** found in previous versions of DiffMerge.

Edit | Use Selection for Find

This command can be used within a file window to directly set the *find-string* to the current selection for a subsequent **Find Next** or **Find Previous** commands without raising the Find Panel.

Edit | Find Next" and **Edit | Find Previous"**

These commands repeat the last search without raising the Find Panel.

Edit | Next Change and **Edit | Previous Change**

These commands enable you to quickly jump between changes in the files.

Edit | Next Conflict and **Edit | Previous Conflict**

These commands are only available in File Merge Windows. They enable you to quickly jumpt to the next/previous conflict in the files.

Using the Glance Bar

The Glance Bar can be used to find changes and scroll the file panels to a specific line or change.

Simply move the mouse over the Glance Bar and click. The corresponding line numbers are displayed in the status bar as you move the mouse.

Chapter 3. The Folder Diff Window

The **Folder Diff Window** enables you to compare two folders and quickly see all of the differences between them. It shows you which files and folders are present in both or only present in one. It also shows you which files are different and optionally which are equivalent or identical.

You can use this, for example, to compare two versions of a software project and quickly see an overview of the changes between them. You can then click on an individual pair files to open a File Diff Window and see how the files have changed.

The Folder Diff Window is a recursive listing; it examines all files contained within the given root folders and all sub-folders regardless of how deeply the folders are nested. Producing a listing in a large folder tree can take a significant amount of time.

There are various Show/Hide commands to let you limit the clutter in the window and concentrate on the important items more easily.

Filtering options are available in the Options Dialog to filter out generated and other uninteresting files and sub-folders.

Opening Folder Diff Windows

You can open Folder Window by selecting **File | Open Folder Diff** from the menu. This raises the **Select Folders to Compare Dialog** and allows you to type or browse to the pathnames of the folders you wish to compare:

If a **Swap** button is provided, it will swap the left and right pathnames.

Note: if the set of folders selected are already open in an existing window, that window is raised rather than creating a new, duplicate window.

Parts of a Folder Diff Window

Here is an example of a Folder Diff Window.

Folder Listing

In the center of the window is a 2 column listing of the files and sub-folders that are present in the 2 folders being compared. This is a recursive listing; all sub-folders are completely expanded. Lines are colored and an icon is attached to each line based upon the status of the files or sub-folders on the line. In the following example, the files `C:\examples\ancestor\subfolder1\Preview_LineMode_3.txt` and `C:\examples\branch_1\subfolder1\Preview_LineMode_3.txt` are different and the file `version.h` only exists in the `C:\examples\ancestor\` folder.

The status bar contains a summary of the 2 folders. This is independent of the various show/hide settings.

From left to right the fields are the sum of the number of:

- Files: equal equivalent quickmatch different peerless
- Folders: pairs peerless
- Shortcuts: equal different peerless *(Windows only)*
- Symlinks: equal different peerless *(Linux/Mac only)*

Using The Mouse

On rows with a pair of files or folders, double-clicking will open a new DiffMerge file or folder window on the pair.

On rows with a single or pair of shortcuts, double-clicking will open a dialog showing information about the shortcut(s). *(Windows only)*

On rows with a single or pair of symlinks, double-clicking will open a dialog showing information about the symlink(s). *(Linux/Mac only)*

Right-clicking on a row will raise a context menu with additional commands.

The Toolbar

The toolbar above the Folder Diff Window presents the following icons.

From left to right, these are:

View | Compare Selected Files and **View | Compare Selected Folders** open the selected pair of items in a File Diff or Folder Diff Window.

View | Show Equal shows/hides (byte-for-byte) identical files and shortcuts.

View | Show Equivalent Files shows/hides files that have minor differences (such as in whitespace), but are otherwise effectively “equivalent”. Equivalence settings are set in the Options Dialog.

View | Show Quick-Match Files shows/hides items that were marked equivalent using Quick Match Mode. This mode only looks at file size.

View | Show Peerless shows/hides items (files, folders, and shortcuts) that are only present in one of the 2 folders.

View | Show Folders shows/hides sub-folders present in both folders.

Other Menu Commands

There are a few Folder Diff Window commands that are only in the Menu and not on the Toolbar.

**File	Reload**	Force a rescan the folders. This is useful if you have automatic reloading turned off in the Options Dialog.

View | Shortcut Details...	On Windows, this command raises the **Shortcut Details** dialog on the selected row. For each shortcut this dialog shows the target pathname, associated icon, description, working directory, and any command line arguments. For shortcut pairs that have file or folder targets, this dialog contains a button to open a file or folder window to compare them.
View | Symlink Details...	On Linux and Mac, this command raises the **Symlink Details** dialog on the selected row. For each symlink this dialog shows the target pathname.
View | Show Errors	Show/Hide items that had a filesystem error of some type. Double-click on an error row to see the error message.

Context Menus

There are a few right-mouse context menu items worth mentioning here:

Open Pair in New DiffMerge Window	Opens the pair of files or folders in a new window.
Copy Left Pathname to Clipboard	Copies the full pathname of the item in the left tree to the clipboard.
Copy Right Pathname to Clipboard	Copies the full pathname of the item in the right tree to the clipboard.
Copy Left to Right	Copies the file, symlink, shortcut or folder in the left tree to the right tree. If the right side does not exist, it will be created. **If the right side does exist, it will be overwritten.** The new item will be placed in the same relative location in the right tree as it has in the left tree. If the parent directory on the right does not exist, it will be created first. **Caution: This will create or overwrite the item in the right tree. If this window was opened by a version control tool, you may still need to tell it about the new files/folders. If the version control tool asked DiffMerge to show an historical view, the right side may refer to a temporary location (that may be deleted by the version control tool after DiffMerge exits).** If the item is a folder, only the folder itself will be created in the right tree. To copy the folder and its contents use the **Copy Left to Right Recursively** command.
Copy Right to Left	As above but reversed.

	Caution: This will create or overwrite the item in the left tree. If this window was opened by a version control tool, you may still need to tell it about the new files/folders. Also, version control tools tend to populate the left side with an historical view of the tree and that should be treated as read-only. Use caution when making changes to the left side.
Shortcut Details...	On Windows, this command will raise the Shortcut Details dialog for the single or pair of shortcuts.
Symlink Details...	On Linux and Mac, this command will raise the Symlink Details dialog for the single or pair of symlinks.

Export Commands

Export | File Diffs

The menu commands under here are available when a file pair with differences is highlighted in the Folder Window. These commands will export the differences in the highlighted file pair but without opening a File Diff Window on them.

See the section on Exporting File Differences for more information on the commands under this menu.

Export | Folder Summary

The menu commands under here menu will export the contents of the current Folder Window to a file.

There are several output options, such as "Text" vs "HTML" vs "CSV", but in all cases:

- The output will contain the same information as is currently on screen.
- The current state of the various Show/Hide commands on the View menu and/or the toolbar will be respected.

For example, the HTML view looks like this:

Chapter 4. The Options Dialog

The **Options Dialog** allows you to fine-tune DiffMerge to better suit your needs.

All of the various settings are grouped by category into a series of **Pages** whose titles are listed on the left side of the dialog. You can click on these titles to visit each of the various Pages.

Each Page is somewhat self-contained and has a **Restore Defaults** at the bottom. This button will restore all of the fields on that Page to their original factory defaults.

The Options Dialog can be accessed using **Tools | Options...** on Windows and Linux and **DiffMerge | Preferences...** on OS X.

The File Windows Page

The **File Windows Page** contains basic settings for File Diff and Merge Windows. It lets you to select the font used in File Panels on screen and when printed. It also contains a series of miscellaneous settings explained below.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

Fonts

The **Window Font** and **Printer Font** fields determine the screen and printer fonts used to draw and print the files. The **Choose...** buttons raise the standard system font chooser.

Window Options

The **Window Options** box contains a collection of miscellaneous settings:

DiffMerge can optionally test and display a warning if a file has been modified by another application. This warning also asks you if you would like to reload the file(s). This test is performed each time that a File Diff or Merge Window comes to the foreground and becomes the active window.

When printing File Diff and Merge Windows, DiffMerge actually prints 2 (or 3) files simultaneously. When this option is checked, DiffMerge prints all the page 1s first, then all the page 2s, and so on. When this option is off, DiffMerge prints all of the pages from the left file, then all of the pages from the center file, and then all of the pages from the right file.

When this option is enabled, DiffMerge adds the appropriate end of line character(s), if necessary, to the end of the final line of the file before saving.

When **Auto-Save** is enabled, DiffMerge periodically saves the edited file in a special **Temporary Auto-Save File**. In the event of a program or system crash, you can recover your work from this temporary file. Auto-Save does not modify the original file.

The **Edit Interval** determines the frequency of the auto-saves. This is an edit count. For example, if the Edit Interval is set to 100, then the Auto-Save File is rewritten after every 100 changes.

When this option is enabled, DiffMerge automatically advances to and highlights the next Change after you apply a patch/change using one of the various toolbar or menu commands.

This allows you to rapidly apply a series of changes without having to manually advance using the Next Change command.

If not checked, DiffMerge leaves the caret where it was after the patch/change was applied and does not highlight a change.

The Rulesets Page

The **Rulesets Page** contains all of the settings for controlling how Rulesets are used by DiffMerge. For information about the contents of an individual Ruleset, see Rulesets.

Rulesets allow you to fine-tune DiffMerge to handle different types of files. This includes things such as character encodings, end of line conventions, and whitespace handling.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings. **This will also delete all of your Rulesets and restore the original set of Rulesets that were originally shipped with the program.**

Default Ruleset

The **Default Ruleset** is a built-in Ruleset that is applied to a File Diff or Merge Window when no Custom Ruleset is appropriate or when Custom Rulesets are disabled.

The Default Ruleset cannot be deleted, but you can edit it.

Custom Rulesets

The **Custom Rulesets** box contains a collection of overall settings:

This option turns on Custom Rulesets. When disabled, the Default Ruleset is used for all File Windows.

This option lets DiffMerge try to automatically select the proper Custom Ruleset using the suffixes of the files and the list of suffixes in each Custom Ruleset. For example, if you open `foo.cpp` and `bar.cpp` in a File Diff Window, DiffMerge will automatically select the **C/C++/C# Source** Custom Ruleset. If you open `foo.py` and `bar.py`, DiffMerge will automatically select the **Python Source** Custom Ruleset.

If this option is disabled, DiffMerge will ask you to select a Ruleset when each File Window is opened.

Ignores the case of the file suffixes when matching.

You probably want this turned on.

When the files in the set have different suffixes, it's possible to take the first match, or to require all of the files to match the same Ruleset.

You probably don't want this turned on. Temp files often get a system-defined temp name rather than the proper suffix. Enabling one file to match enables you to ignore the temp file suffix when it's being compared against a properly named file in your workspace.

When DiffMerge cannot find a suffix match, it can either use the Default Ruleset or it can ask you to select one using the Choose Ruleset Dialog.

The list box contains all of the currently defined Custom Rulesets. DiffMerge ships with Rulesets predefined for C/C++/C# Source, VB Source, and several others. This list is ordered – DiffMerge searches for a match in the order listed here.

The **Add** / **Delete** / **Clone** buttons let you create/destroy Rulesets.

The **MoveUp** / **MoveDown** buttons let you re-order them.

You can double-click an item or click the **Edit...** button to edit or view the Ruleset. See the chapter on Rulesets for more information.

The Detail Level Page

The **Detail Level Page** contains all of the parameters for the controlling the Analysis Detail Level. This is a global setting and affects all File Diff and Merge Windows.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

Analysis Detail Level

The **Analysis Detail Level** determines how the files are compared:

- In Lines Only mode, DiffMerge compares the files line-by-line and does not perform any intra-line highlighting/analysis.

  ### Tip

  This is the fastest mode. Switch to it if you're working with large files and/or having performance problems.

- In Lines and Characters mode, DiffMerge performs the line-by-line analysis and then performs intra-line analysis within each of the changes. Intra-line analysis is performed over the body of each change and is governed by the following setting.

Multi-Line Intra-Line Analysis Detail Level

The **Multi-Line Intra-Line Analysis Detail Level** determines how hard DiffMerge works to match up characters within a multiple line change when in Lines and Characters mode.

- In **Complete** mode, the entire body of the change (spanning all of the lines in the change) are essentially joined into a single line before doing the character-level analysis. This allows changes introduced by re-wrapping text to be separated from actual changes to the text.

  ### Tip

  This mode can be **very** expensive if the files have lots of changes and/or have very low correlation. Disable it or switch to Simple mode if you experience performance problems.

- In **Simple** mode, lines are joined like in Complete mode, but this is limited to short change blocks to avoid using an excessive amount of time and memory on highly uncorrelated blocks.

- If **Disabled**, DiffMerge does not attempt to join lines before doing the character level analysis.

For example, the following images show where a line of source code was broken across several lines in one version. When in Simple or Complete mode, only the whitespace is highlighted because DiffMerge matches up the words by crossing the line boundaries.

When Disabled, DiffMerge sees this as a five line change. There is a one line change (where text was deleted from the end of the line) followed by an insert of four lines of new text. Since DiffMerge does not combine the lines in this mode, the two parts are seen simply as adjacent, but unrelated, changes.

Intra-line Smoothing Threshold

The **Intra-line Smoothing Threshold** causes small spans of equal text between 2 intra-line changes to be marked as part of one combined change. This helps eliminate “noise” within the line.

In the following, the threshold was set to 0 (disabled). Notice how the “x” characters are matched up between the changes on the line.

In the following, the threshold was set to 3. Notice how the “x”, “xx”, and “xxx” sequences are combined with the surrounding changes. But the “xxxx” sequence is not affected.

The intra-line threshold value doesn't change any difference analysis results; it only affects the coloring of short spans of text within the line.

The intra-line threshold is only used when the Detail Level is set to Lines and Characters.

Inter-line Smoothing Threshold

The **Inter-line Smoothing Threshold** causes small groups of equal lines between two changes to be marked as part of one combined change. Inter-line smoothing can be thought of as “clumping” nearby changes into a single change.

In the following, when the smoothing is set to 0 (disabled), we see 2 changes:

When the smoothing is set to 3 or more, we see that they have been combined into 1 change that includes the identical lines between them:

This feature is initially set to 0 (disabled) because in File Merge Windows this smoothing can cause adjacent changes (such as in the above example) to be considered a conflict when the individual (unclumped) changes would not be. When you increase the threshold, you increase the likelihood that Auto-Merge will require manual follow-up. This may or may not be a good thing -- it's somewhat questionable how far apart nearby changes should be to be considered independent changes. By increasing the setting to 1 or 2, you may produce a few conflicts that will cause Auto-Merge to complain, but these are areas that you probably want to examine anyway.

Also, you might use this feature to reduce the number of annoying little changes reported. For example, in source code a function frequently consists of several lines of text, a blank line, several lines of text, a blank line, and so on. If someone inserts or changes a big chunk of code, the blank lines can sync up and the unrelated chunks of code can appear as a sequence of little changes. Setting the threshold to 1 should cause these individual changes to appear as 1 large change rather than numerous (annoying) little ones.

This field is used in both Detail Levels.

The Line Colors Page

The **Line Colors Page** lets you set the various colors used to draw text with a File Window. These colors are used when the Detail Level is set to Lines Only. These colors are also used as the line base colors when the Detail Level is set to Lines and Characters.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

The Intra-Line Colors Page

The **Intra-Line Colors Page** lets you set the various colors used to draw intra-line changes on top of the line base colors. These colors are only used when the Detail Level is set to Lines and Characters.

The overall line colors (from the Line Colors Page) are shown for reference in the first and last columns. The buttons in the Foreground (Unimportant) column enable you to change the de-emphasized foreground colors which are used when an intra-line change is marked as unimportant, such as text within a comment. The buttons in the Intra-line Background column enable you to change the background colors of the intra-line highlight.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

The Other Colors Page

The **Other Colors Page** lets you to set the various colors used to draw miscellaneous items in File Diff and Merge Windows.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

The Folder Windows Page

The **Folder Windows Page** contains basic settings for Folder Diff Windows.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

Fonts

The **Window Font** and **Printer Font** fields determine the screen and printer fonts used to draw and print the Window. The **Choose...** buttons raise the standard system font chooser.

Window Options

The **Window Options** box contains miscellaneous settings:

DiffMerge can optionally automatically rescan the filesystem and look for files that have been modified by other applications. This rescan is performed each time a Folder Diff Window comes to the foreground and becomes the active window.

Tip

This option can be **very** expensive and annoying if you are comparing large folders. Disable it if restacking windows causes an unacceptable delay.

The Folder Filters Page

The **Folder Filters Page** contains the settings for controlling how DiffMerge filters out uninteresting files and sub-folders from the Folder Diff Window display.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

Tip

Scanning the file system is **very** expensive and time consuming. The use of Folder Filters can usually speed this up considerably by eliminating the need to look at various types of files and entire sub-folders. It will also help reduce on-screen clutter.

Filename Filters

DiffMerge can filter out temporary, binary, and other machine-generated files that are normally produced as a by-product of software development. This section handles filtering out various files.

The first text field contains a list of suffixes that should be excluded, such as `EXE` and `DLL`. When DiffMerge sees a file with one of these suffixes during the file system scan, it skips it completely.

The second text field contains a list of filenames that should be excluded. Items in this list can contain simple glob-style wildcards, such as "*.dll".

The first field is preserved for historical reasons; the wildcard matching now provided by the second field completely eliminates the need for the first field.

Items in both lists are delimited by whitespace, commas, or semicolons. If you need to add a pattern with one of those characters, surround the entire pattern in double quotes. To include a double quote within a filename, double it.

Sub-folder Filters

DiffMerge can filter out entire sub-folders (and everything contained within them). This is useful for compiler output directories as well as for the various folders of state files that many Version Control Systems leave in your source tree.

The text field contains a list of sub-folder names that should be excluded. When DiffMerge sees a sub-folder with one of these names during the file system scan, it skips it (and everything contained within the sub-folder).

Items within this list can use wildcards too. And have the same delimiters and quoting details as described in the previous section.

Ignore Case

The **Ignore Case in Patterns** controls whether case is ignored or respected when using all of the above set of filter patterns.

The **Ignore Case when Matching Rows** controls whether filename case is ignored or respected when the contents of the 2 directories are matched up. For example, when enabled a file "FOO.TXT" would be assumed to match up with a file "foo.txt" in the other directory. When disabled, they would appear as 2 peerless items. This option also causes the folder window to be sorted with a case-insensitive sort.

These options are primarily intended for Windows and Mac users which have case-insensitive filesystems.

The Folder Colors Page

The **Folder Colors Page** lets you set the various colors used to draw rows within a Folder Diff Window.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

The Equivalence Mode Page

The **Equivalence Mode Page** lets you configure how Folder Diff Windows compares files.

When a Folder Diff Window scans the file system, it compares each pair of files using one of these 2 techniques:

- **Quick Match Mode:** When Quick Match is enabled and the file suffix is in the associated list, DiffMerge will just compare the sizes of the 2 files and report "qm" or "different". This is intended to help speed up scans when there are large media files in the folders. *This method does not require either file to be read, so the results are an approximation.*

- **Normal Mode:** The contents of both files are read and compared. Pairs that are byte-for-byte identical are marked **equal**. Non-equal pairs are marked **different**.

  When a difference is detected, DiffMerge can optionally rescan non-equal file pairs and ignore things like changes in line termination, changes in whitespace, and etc. in order to avoid reporting trivial differences. Files which only have ignorable differences as marked **equivalent** rather than **different**.

  Please note that this second step is expensive and will cause DiffMerge to take longer to complete the scan. For this reason, there are 2 levels of equivalence testing, each with different levels of complexity:

Simple File Equivalence

Simple equivalence mode attempts to address the most trivial reasons for differences that are common to many types of text files:

- It will only be applied to files with one of the listed suffixes.

- During the rescan it will ignore differences in line termination characters and/or whitespace.
- It operates as if the Detail Level is set to Lines Only mode.
- It assumes that files are in an 8-bit encoding compatible with US-ASCII.

Simple equivalence mode **DOES NOT** look at upper/lower case, address character encoding issues, nor import the files into UNICODE.

Ruleset-based File Equivalence

Rulesest-based equivalence is a more thorough attempt to determine if the files only have trivial changes. It uses *most* of the settings in the corresponding Ruleset for each pair of files; this includes:

- ignoring differences in character encoding by using the settings on the Character Encodings Page of the Ruleset Dialog and importing the files into UNICODE;
- ignoring differences in line termination, whitespace, and letter case by using the settings on the Equivalence Mode Page of the Ruleset Dialog; and
- stripping out lines matched by the Lines to Omit settings on the Ruleset.

Ruleset-based equivalence **DOES NOT** use the settings on the Line Handling Page of the Ruleset Dialog.

Ruleset-based equivalence operates as if the Detail Level is set to Lines Only and therefore *does not* use any of the Content Handling “Context” settings.

For Ruleset equivalence to work, Rulesets and automatic suffix matching must be enabled. For an individual Ruleset, character encoding selection must be automatic. If a Ruleset or character encoding cannot be automatically chosen, equivalence testing will either be skipped or the Default Ruleset chosen instead.

NOTE: Do not enable the Default Ruleset if you have binary files in your folders since the attempt to import the files into UNICODE will generally fail and just waste time.

As a performance consideration, you may want to set an upper file size limit for Ruleset equivalence testing. No ruleset-based equivalence testing will be done on files that exceed this limit.

The Messages Page

The **Messages Page** lets you to choose which message dialogs that you do or do not want to see.

Each of the message dialogs has a “Do not show again!” option.

The following check boxes let you re-enable any message dialog that you turned off.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings.

The Explorer Integration Page

This Page is only available on Microsoft Windows.

The **Explorer Integration Page** lets you control the DiffMerge integration with Windows Explorer. This feature allows DiffMerge to add commands to the context menu of Windows Explorer. These commands allow Windows Explorer to launch DiffMerge with the selected file(s) and folder(s).

Personal Settings

The integration feature is provided by a DiffMerge-supplied library that is loaded by Windows Explorer. This library must be registered with the operating system before Windows Explorer will use it. This is a system-wide registration and will affect all users of the computer.

Once the library has been registered with the system, you may personally enable or disable the feature in your user account.

The **Restore Personal Defaults** button restores your personal settings to their original factory settings; it does not affect the system-wide registration of the library.

Using Explorer Integration

When Explorer Integration is enabled, DiffMerge will (when appropriate) add a pop-up context menu to the Windows Explorer context menu.

When 1 Item is Selected:

When 1 file of folder is selected, the context menu will look something like this:

DiffMerge shows the long version of the context menu. You can launch DiffMerge with the single item (and interactively fill in the other file or folder), remember the item's pathname in the list, or compare the item with something from the remembered list.

DiffMerge allows you to remember 5 file pathnames and 5 folder pathnames for use in future comparisons.

When you compare with something from the list, DiffMerge will put the **older** file/folder in the left panel; you may override this using the **Swap** button in the Select Files to Compare or Select Folders to Compare dialogs.

When 2 Items are Selected:

When 2 files or folders are selected, the context menu will look something like this:

DiffMerge shows the short version of the context menu. With it you can launch DiffMerge to compare the 2 items and choose which will appear on the left and right.

The External Tools Page

The **External Tools Page** lets you to direct DiffMerge to invoke external, third-party tools when comparing and/or merging various types of files.

The **Restore Defaults** button restores all the fields on this (and only this) page to their original factory settings. **This will also delete all of your External Tool configurations.**

Overall Configuration

This option turns of/off the entire External Tools feature.

Ignores filename case when matching file suffixes.

Indicates if the suffixes for all of the files in the set must match or if only one needs to match.

You probably don’t want this turned on. Temp files often get a system-defined temp name rather than the proper suffix. Turning this off allows you to ignore the temp file suffix when it is being compared against a properly named file in your workspace.

The list box contains all of the currently configured External Tools. By default, no External Tools are configured.

The **Add** / **Delete** / **Clone** buttons let you create/delete tool configurations.

The **MoveUp** / **MoveDown** buttons let you re-order them.

You can double-click an item or click the **Edit...** button to edit or view a configured tool. See the chapter on External Tools for more information.

Chapter 5. Rulesets

Overview

Rulesets allow you to fine-tune DiffMerge to handle different types of files. This includes things such as character encodings, end of line conventions, and whitespace handling.

DiffMerge uses the Ruleset settings and the list of Rulesets defined in the Options Dialog to control how a set of files are loaded from disk and compared in a File Diff or Merge Window.

DiffMerge includes a Default Ruleset and a list of Custom Rulesets. The Default Ruleset is used when the use of Custom Rulesets is disabled or when none of the Custom Rulesets is appropriate.

The Ruleset Dialog

The **New/Edit Ruleset Dialog** lets you create or modify the settings within a Ruleset. You can edit both the Default Ruleset and individual Custom Rulesets.

All of the various Ruleset settings are grouped by category into a series of **Pages** whose titles are listed on the left side of the dialog. You can click on these titles to visit each of the various Pages.

The Name Page

The **Name Page** of the **New/Edit Ruleset Dialog**.

This Page does not appear when you are editing the Default Ruleset; it is only present for Custom Rulesets.

Ruleset Name

This is a “human-readable” name for the Ruleset. This name is for information only; it is displayed in various places to help you identify the Ruleset.

File Suffixes

This is a list of suffixes that should be assigned to this Ruleset. Entries should be separated by spaces and should not contain dots.

If a Ruleset has a matching suffix for a particular file, then that Ruleset *might* be used when the File Window is created.

The Character Encodings Page

The **Character Encodings Page** of the **New/Edit Ruleset Dialog** lets you specify the character encoding for files handled by this Ruleset.

DiffMerge uses this information to convert the files from their on-disk encoding into UNICODE before comparing them. This allows, for example, DiffMerge to compare the effective content of 2 files that were created with different encodings.

After the files are loaded, DiffMerge displays the character encoding(s) of the files in the status bar.

Note

If a file is loaded in multiple File Diff or Merge Windows, it will only be read from disk once. Subsequent windows will share the in-memory copy of the file. Therefore, when the file is loaded into the first window, the character encoding settings for the Ruleset in that window will be used to convert the file into UNICODE. When the file is "referenced" by a subsequent window, no conversion is required, so the encoding settings in the (possibly different) Ruleset in the second window will not be used for that file.

Automatic Detection

When **BOM Detection** this is enabled, DiffMerge looks for a UNICODE Byte-Order-Mark (BOM) when reading the files. If a BOM is present, DiffMerge assumes the indicated encoding. If a BOM is not present, DiffMerge uses the Fallback Options below.

Fallback Options

The **Fallback Options** are used when no BOM is present.

DiffMerge assumes that files are encoding using the local system default character encoding.

Use this if all of your files use system default encoding.

DiffMerge will raise the **Choose Character Encoding Dialog** and ask you to select an encoding when the File Diff or Merge Window is created.

Use this if you have files of this type in multiple encodings on your system, but will only be comparing files that have the same encoding.

DiffMerge will raise the Choose Character Encoding Dialog once and ask you to select an encoding **FOR EACH FILE** when the File Diff or Merge Window is created.

Use this if you have files of this type in multiple encodings on your system and need to do arbitrary comparisons. This gives you maximum flexibility and lets you select an encoding on a file-by-file basis.

This option allows you to specify a primary and upto 2 alternate character encodings. DiffMerge will first try to use the primary character encoding. If that fails, DiffMerge will attempt to use the alternates.

Named Encoding

Each of the controls in the **Named Character Encoding** section have a list of all character encoding converters on the system.

The Lines to Omit Page

Everything on this page is optional.

The **Lines to Omit Page** of the **New/Edit Ruleset Dialog** lets you describe lines that should be completely omitted from the difference analysis. For example, lines containing RCS revision or date keywords.

The list box presents a list of the patterns already defined. You may create as many of these patterns as you want. You can think of these as a series of `grep -v` commands applied to the each file before DiffMerge compares them.

The Pattern Dialog

The **Add/Edit Pattern Dialog**.

Pattern

Here you can enter a regular expression that will be used to search for lines to omit.

For convenience, there are buttons to the right to supply patterns for some commonly omitted items.

Lines to Skip

This field indicates how many lines should be omitted with each match. You might use this, for example, if you know that there are 2 lines of page headers after each page break. Most of the time this field will be 1.

The Line Handling Page

The **Line Handling Page** of the **New/Edit Ruleset Dialog** lets you control how DiffMerge treats End of Line (EOL) characters and if trivial differences are ignored when matching up lines in File Diff and Merge Windows.

Note

Unless you have specific needs, everything on this Page should always be enabled.

Line Termination

Different platforms use different line termination characters. Such differences can cause two otherwise identical files to look like completely different files.

- When enabled, DiffMerge treats all CR, LF, and CRLF characters as a generic end-of-line (EOL) marker and excludes them from the line. Line termination differences are not detected or indicated in any way. This lets you compare files from different platforms with different EOL conventions.

  When enabled, EOL characters are grayed-out to indicate that they are being ignored:

- When disabled, the original CR, LF, and CRLF characters are preserved and used in the analysis. The EOL characters are treated as part of the line. So 2 otherwise identical files with different EOL characters will appear to be 2 completely different files (with 1 change spanning the whole file).

  When disabled, EOL characters are not grayed-out; they are part of the line and drawn just like other characters on the line.

Line Matching

These options let DiffMerge ignore whitespace and letter case when vertically aligning and matching up lines. Generally, this allows DiffMerge to achieve the best vertical text alignment.

Note

Changes in whitespace and letter case on a line will still be detected and indicated, but they will not throw-off the vertical alignment. This is very useful when source files are re-indented.

The Content Handling Page

The **Content Handling Page** of the **New/Edit Ruleset Dialog** lets you tell DiffMerge about the various contexts, such as string literals and comments, within a file. DiffMerge can use this information to classify changes as important or unimportant.

These settings are only used in File Diff and Merge Windows when the Detail Level is set to Lines and Characters.

Within source code there are usually three types of content. These are called **Contexts**:

Literal Context	Quoted strings and other such text where you want exact matching. For example, changing the whitespace or letter case within a string prompt may change the behavior of a progrem. We will mark changes in this context as **Important**.

Comment Context	Within a source code comment you may want to relax the rules some and disregard some of the changes. For example, suppose you have a large block of comment text and you re-wrap it in one version of the file. You might like to hide the change and pretend it's not there. We will mark changes in this context as **Unimportant**.
Default Context (Everything Else)	Any content that cannot be assigned to one of the above contexts is said to be in the Default Context. For source code, this is generally the bulk of the file. Generally, changes in this context should be marked as Important, but you may override that.

Matched Contexts

Literals and Comments are said to be **Matched Contexts** because we need to have a pattern to identify them.

The list box shows all of the defined context patterns. The **Add...** and **Edit...** buttons will let you create a new pattern or modify an existing one using the Context Dialog.

Default Context Guidelines

The following fields let you control how DiffMerge will mark changes that are found within this context. This discussion applies to the Default Context as well as Matched Contexts that are defined using the Context Dialog.

This option lets you declare that the overall context is either important or unimportant. If it is important, you can refine the context further to make commonly ignored things unimportant.

Generally, this should always be checked.

This option is only enabled if you elected to keep EOL characters in the analysis (not Ignore/Strip EOLs on the Line Handling Page).

This option will mark changes as Important/Unimportant that are caused by inconsistencies in the EOL characters within a file.

This option will mark changes in letter case as Important/Unimportant.

Generally, this option should be turned on -- unless you are in a case-insensitive language like VB.

This option will mark changes in whitespace as Important/Unimportant. this can mean SPACES-only or SPACES-and-TABS depending on the next field.

In languages like C/C++ you should turn this option off for the Default Context because whitespace is not significant within code.

In languages like Python you should turn it on for the Default Context because leading whitespace is significant in Python.

This option determines whether TABs are equivalent to SPACES.

Generally, this option should be turned on for most languages.

The Context Dialog

The **Define New/Edit Context Dialog** lets you define the boundaries of a Context. A Context is a mechanism for identifying portions of a document that should be specially handled such as a string literal or comment.

A Context is defined as a Start Pattern and an optional End Pattern. Patterns must be valid Regular Expressions. The End Pattern may be omitted if **Ends at EOL** is checked. Set the **Escape Character** if the context has a special character (such as a backslash) to prevent premature matching of the End Pattern or EOL.

The text that matches the Start Pattern is not considered part of the matched context; only the text following it (and including the text matching the End Pattern).

See the Default Context Guidelines for an explanation of the Context Guideline fields on this dialog.

Equivalence Mode

The **Equivalence Mode Page** of the **New/Edit Ruleset Dialog** lets you control what differences are ignored in Folder Diff Windows when the Equivalence Mode is set to Ruleset-based.

Note

The settings on Equivalence Mode Page are very similar to the settings on the Line Handling Page. The settings here are only used by Folder Windows; the others are only used for File Windows.

Chapter 6. External Tools

Overview

Whenever DiffMerge is asked to compare or merge a set of files and the External Tools feature is enabled, DiffMerge can either open the files in a File Diff or Merge Window or it can hand them to an external application and let it process the files.

You can use this feature, for example, to let other applications handle binary files types (such as word processor documents) or XML files and then use a DiffMerge Folder Diff Window to compare 2 folders of documents (using the built-in Exact Match Only Equivalence Mode). And when you double-click on a pair of files, the external application will be launched to actually show you the changes.

DiffMerge uses the suffixes of the files and searches the list of configured External Tools for a match. The tools are searched in the order listed in the Options Dialog. DiffMerge then tries to use the first matches. If no tool matches, DiffMerge creates a normal File Diff or Merge Window.

Launching an External Tool

When DiffMerge gets a match for files given on the command line, DiffMerge will invoke that tool directly and wait for it to exit before exiting; DiffMerge will then exit with that tool's exit status. If Messages are disabled, DiffMerge will do this without even creating a DiffMerge window.

When DiffMerge gets a match for an interactive request, DiffMerge just launches it as a peer application and does not wait for it to complete.

The External Tools Dialog

The **Add/Edit External Tool Dialog** lets you create or modify the settings for an External Tool.

For each External Tool that you configure, you can decide if the tool is enabled for File Diffs, File Merges, or both. You can select different executables and command line argument templates for each mode.

All of the various tool settings are grouped by category into a series of **Pages** whose titles are listed on the left side of the dialog. You can click on these titles to visit each of the various Pages.

The Name Page

The **Name Page** of the **Add/Edit External Tool Dialog**.

Name

This is a “human-readable” name for the tool. This name is for information only; it is displayed in various places to help you identify the tool.

File Suffixes

This is a list of suffixes that should be assigned to this External Tool. Entries should be separated by spaces and should not contain dots.

The Diff Page

The **Diff Page** of the **Add/Edit External Tool Dialog** lets you configure the tool to be used for File Diffs.

2-Way File Diffing

This check box determines if this tool should be used for File Diffs.

Pathname

This field contains the pathname of the executable that should be used for File Diffs.

Tip

On OS X, you can give the path to either the application bundle (the `foo.app` folder) or the name of the actual executable inside the application bundle (`foo.app/Contents/MacOS/foo`).

Command Line Arguments

Since each external application seems to have its own unique argument ordering and command flags, we define the command line arguments using a template containing place-holder tokens. When DiffMerge is ready to launch the external application, it will substitute pathnames and labels into the template and build the actual command line.

DiffMerge defines the following substitution tokens for File Diffs:

`%LEFT_LABEL%` `%RIGHT_LABEL%`	The labels for the left and right files. In a DiffMerge File Diff Window, these labels are displayed above the panel; other applications may display them differently or not use them at all. When DiffMerge is handing off the initial set of files received on the command line to an external tool, the values for these tokens come from the `/title1` and `/title2` arguments given to DiffMerge. Otherwise, they default to the pathnames of the corresponding files.
`%LEFT_PATH %RIGHT_PATH%`	The pathnames of the corresponding files.

The **Left** and **Right** buttons are for your convenience and can be used to insert the corresponding token (with quotes) at the current insertion point into the text field.

Tip

It is highly recommended that you enclose each token in quotes so that whitespace in the titles and/or pathnames are properly received by the external application.

The Merge Page

The **Merge Page** of the **Add/Edit External Tool Dialog** lets you configure the tool to be used for File Merges.

The descriptions for most of these fields are identical to the descriptions on the previous page for File Diffs.

DiffMerge defines the following substitution tokens for File Merges:

`%WORKING_LABEL%` `%OTHER_LABEL%`	The labels for the “working” (left) and “other” (right) files. When DiffMerge is handing off the initial set of files received on the command line to an external tool, the values for these tokens come from the `/title1` and `/title3` arguments given to DiffMerge. Otherwise, they default to the pathnames of the corresponding files.
`%DEST_LABEL%`	The label for the “destination” (center) file. When DiffMerge is handing off the initial set of files received on the command line to an external tool, the value for this token come from the `/title2` argument given to DiffMerge. Otherwise, it defaults to the pathname of the destination result or the common ancestor baseline file.
`%WORKING_PATH%` `%OTHER_PATH%`	The pathnames of the “working” (left) and “other” (right) files.
`%BASELINE_PATH%`	The pathname of the “baseline” (center) file. This is the common ancestor.
`%DEST_PATH%`	The pathname where the merge result should be written. This comes from the `/result` argument given to DiffMerge. It falls back to the baseline pathname.

Chapter 7. Printing

Printing File Windows

You can print the contents of File Diff and Merge Windows and see the differences highlighted as they are on the screen.

Tip

With the addition of HTML Export in 4.2, if you want to print or share file differences, you might want to use the new HTML Export feature to export the differences to a .html file and let the browser handle the details of print and/or share the .html file with co-workers, rather than using the native print feature described in this section.

I feel that the look of the HTML output is much nicer than the output produced by the original native DiffMerge print commands. For example, the HTML output supports line wrapping and 3 different layouts whereas the native printing does not wrap and only handles side-by-side on facing pages.

The native printing output is due for overhaul; until then you might consider the HTML option.

DiffMerge uses all of the current display settings, such as the Detail Level, Display Mode, various hiding options, and etc. So for example, if you are have selected Show Differences Only in the window, the output will also only contain the differences.

Changes are printed in color using a bold font. Conflicts are printed in color using a bold and underlined font. DiffMerge uses the same coloring rules for text as on the screen; however, it does not use the background colors.

The page footer gives a summary of the settings that affected the output.

When DiffMerge prints a File Window, it actually prints both/all of the files in the window simultaneously. Each File Panel is printed on a separate page, 2 or 3 pages across. The page numbers are augmented with the letters a, b, and c to help you track the printed pages. Vertically, files are synchronized like they are on screen so that content lines up on each page. For example, when printing a File Merge Window, you should be able to place pages 10a, 10b, and 10c side-by-side and have everything line up just like on the screen.

For your convenience, DiffMerge can print the files sequentially (1a, 2a, ..., 1b, 2b, ...) or interleaved (1a, 1b, 2a, 2b, ...). This is controlled in the Options Dialog.

Currently, DiffMerge does not line-wrap long lines. Lines that do not fit across a single page are truncated. If long lines are a problem, try printing in landscape mode and/or using a smaller font.

Printing Folder Windows

You can also print the contents of Folder Diff Windows.

Tip

With the addition of HTML Export in 4.2, if you want to print or share the summary of differences in the folder pair, you might want to use the new HTML Export feature to export the differences

to a .html file and let the browser handle the details of print and/or share the .html file with co-workers, rather than using the native print feature described in this section.

I feel that the look of the HTML output is much nicer than the output produced by the original native DiffMerge print commands. For example, the HTML output has a single column layout similar to "unified" diffs which is helpful when there are long pathnames.

The native printing output is due for overhaul; until then you might consider the HTML option.

DiffMerge uses all of the current Show/Hide settings and the same line icons and text coloring, so the folder listing on paper should match the listing on screen.

The folder summary statistics are printed in the page footer.

Chapter 8. Command Line Arguments

DiffMerge allows a variety of different command lines formats. These allow DiffMerge to launch with different types of initial windows and to be invoked by other applications. Each usage is summarized in the following sections.

DiffMerge Command Flags/Options have a long and short form and may either be prefixed by a '`/`' or a '`-`' depending upon your platform. For example: `-h`, `-help`, `--help`, `/h`, or `/help`.

For options that require a value, they must have a COLON, EQUAL, or SPACE separator character between the keyword and the value. And the value should be quoted if it contains spaces or special characters. For example, `/caption="Hello World!"`.

Generally, DiffMerge will exit with 0 status when there are no errors and a 3 status when there is a command line syntax error.

Opening an Empty Window

Synopsis

`diffmerge`

Opens an empty DiffMerge Window.

Exit Status

Always exits with exit status OK (0) or SYNTAX-ERROR (3).

Opening a Diff Window

Synopsis

`diffmerge` [-caption=*`caption`*] [-ro2] [-shex] [-t1=*`title1`*] [-t2=*`title2`*] {*`file1`*} [*`file2`*]

Opens a File Diff Window with the given files. If only one file is given, you will be prompted for the second. *File1* will appear in the left panel; *file2* will appear in the right panel.

If you do not specify `/ro2`, *file2* will be editable and you can apply patches to it from *file1*.

`diffmerge` [-caption=*`caption`*] [-shex] [-t1=*`title1`*] [-t2=*`title2`*] {*`folder1`*} [*`folder2`*]

Opens a Folder Diff Window with the given folders. If only one folder is given, you will be prompted for the second. *Folder1* will appear in the left panel; *folder2* will appear in the right panel.

Flags

-c, **-caption**=*message*

A caption for the application title bar of the first window.

-ro2

Treat *file2* as read-only.

-shex *(Windows only)*

This is used by the Windows Shell/Explorer Integration feature. This option forces the Open Files or Folders dialog to appear (even when not necessary) before opening the first window to allow you the opportunity to swap the pathnames if necessary.

-t1, **-title1**=*message*

Sets the File Panel Lable for the first file/column.

-t2, **-title2**=*message*

Sets the File Panel Lable for the second file/column.

Exit Status

Always exits with exit status OK (0) or SYNTAX-ERROR (3).

Opening a Merge Window

Synopsis

`diffmerge` [-caption=*`caption`*] [-merge] [-result=*`resultfile`*] [-ro2] [-shex] [-t1=*`title1`*] [-t2=*`title2`*] [-t3=*`title3`*] {*`file1`*} {*`file2`*} {*`file3`*}

Open a File Merge Window with the 3 given files. *File1* will appear in the left panel and is traditionally set to your version of the file. *File2* will appear in the center panel and should be the common ancestor of the other 2 files. *File3* will appear in the right panel and is traditionally set to the repository version of the file.

If you do not specify `/ro2`, *file2* will be editable and you can apply patches to it from *file1* or *file3*.

Flags

-m, **-merge**

Runs Auto-Merge immediately after loading the files.

-r, **-result**=*pathname*

Specifies a pathname for saving the merge result. It omitted, the merge result (if saved) will overwrite the center file. **This flag causes DiffMerge to return an exit status based upon the merge result in the window.**

-t3, **-title3**=*message*

Sets the File Panel Lable for the third file/column.

The meanings of the other flags are described in the Flags for Opening a Diff Window [38].

Exit Status

When `/result` is not used, DiffMerge exits with exit status OK (0) or SYNTAX-ERROR (3) as usual.

When `/result` is used, DiffMerge sets the exit status to reflect the state of the merge:

0: MERGE-RESOLVED - you saved the merge result. *This does not mean that there are no conflicts remaining; it only means that you looked at the files, made whatever edits were necessary, and saved your edits into the result file.*

1: MERGE-ABORTED - you gave up the merge by not saving your changes. *This means that you either never saved your edits into the result file or that you closed the window with unsaved edits (possibly after a checkpoint save).*

2: FILE-ERROR - there were file errors/problems that prevented the merge window from being shown.

3: SYNTAX-ERROR - there were syntax errors with the command line arguments.

Showing Help

Synopsis

`diffmerge` {-help}

Show the DiffMerge Help Dialog and exit.

Exit Status

Always exits with exit status OK (0) or SYNTAX-ERROR (3).

Diffing Files to a File

Synopsis

`diffmerge` {-d=*`diffoutput`*} [-u] [-i] {*`file1`*} {*`file2`*}

Compare 2 files and write the differences to a file (without opening a window).

DiffMerge will use the Ruleset settings and the file suffixes to *try* to automatically select a Ruleset; if it cannot automatically pick one, DiffMerge will fall-back to the Default Ruleset.

DiffMerge will use the Ruleset's settings from the Character Encoding, Lines to Omit, and Line Handling Pages. When appropriate, it will use the last interactively selected tab size.

Since the output will be reported in lines, the Detail Level and the Ruleset's settings on the Content Handling Page are not relevant.

Flags

-d, **-diff**=*pathname*

Compare 2 files and write the results to this pathname. The file is only created if there are differences to report.

-u, **-unified**

If there are differences, write them in Unified Format rather than Tranditional Format.

-i, **-ignore_unimportant**

Suppress "Unimportant" changes by enabling "Hide Unimportant". *Note that this feature can cause the reported diffs to appear slightly differently which may confuse programs like "patch". So it is best to only use this option for personal use.*

Exit Status

When diffing files to a file, DiffMerge sets the exit status to:

0: IDENTICAL - the input files are identical *(and the output file is not created)*.

1: DIFFERENT - the input files are different *(and the differences are written to the output file)*.

2: FILE-ERROR - there were file errors that prevented the files from being compared; this includes file I/O errors and problems detecting the character encoding of the input files.

3: SYNTAX-ERROR - there were syntax errors with the command line arguments.

Diffing Folders to a File

Synopsis

`diffmerge` {-diff=*`diffoutput`*} {*`folder1`*} {*`folder2`*}

Compare 2 folders and write the differences to a file (without opening a window).

Flags

-d, **-diff**=*pathname*

Compare 2 folders and write the results to this pathname. The file is always created even if there are no changes to report.

Exit Status

When diffing folders to a file, DiffMerge sets the exit status to:

0: IDENTICAL - the 2 folders are identical.

1: DIFFERENT - there are changes to report.

2: FILE-ERROR - there were filesystem errors that prevented the folders from being compared.

3: SYNTAX-ERROR - there were syntax errors with the command line arguments.

Chapter 9. Integration with Third-Party Software

DiffMerge can be configured as an external compare / merge tool with various third-party version control systems. Usually this just involves formatting the correct command line template string in the package's options dialog or configuration file. Here are instructions for some popular packages.

Git

The `git difftool` and `git mergetool` commands can be used to launch a variety of external diff and merge tools. For background information on configuring Git, see:

https://www.kernel.org/pub/software/scm/git/docs/git-difftool.html [https://www.kernel.org/pub/software/scm/git/docs/git-difftool.html]

https://www.kernel.org/pub/software/scm/git/docs/git-mergetool.html [https://www.kernel.org/pub/software/scm/git/docs/git-mergetool.html]

Settings for OS X

First confirm that `/usr/local/bin/diffmerge` is present. If you used the PKG Installer, this was installed when /Applications/DiffMerge.app was installed. If you used the DMG file, refer to the instructions for installing the Extras.

The following commands will update your `.gitconfig` to let Git use DiffMerge:

```
$ git config --global diff.tool diffmerge
$ git config --global difftool.diffmerge.cmd
    "/usr/local/bin/diffmerge \"\$LOCAL\" \"\$REMOTE\""

$ git config --global merge.tool diffmerge
$ git config --global mergetool.diffmerge.trustExitCode true
$ git config --global mergetool.diffmerge.cmd
    "/usr/local/bin/diffmerge --merge --result=\"\$MERGED\"
        \"\$LOCAL\" \"\$BASE\" \"\$REMOTE\""
```

Settings for Linux

The following commands will update your `.gitconfig` to let Git use DiffMerge:

```
$ git config --global diff.tool diffmerge
$ git config --global difftool.diffmerge.cmd
    "/usr/bin/diffmerge \"\$LOCAL\" \"\$REMOTE\""

$ git config --global merge.tool diffmerge
$ git config --global mergetool.diffmerge.trustExitCode true
$ git config --global mergetool.diffmerge.cmd
```

```
    "/usr/bin/diffmerge --merge --result=\"\$MERGED\"
        \"\$LOCAL\" \"\$BASE\" \"\$REMOTE\""
```

Settings for Windows

On Windows, the actual configuration settings depend upon which distribution of Git you are using.

GitHub for Windows or Git Bash Shell

The following is based upon the 1.0 release of GitHub for Windows or Git Bash Shell [http://windows.github.com/].

Since GitHub for Windows is built upon a version of the Git for Windows (MSysGit) package, the `Command Prompt` commands in the following section could also be used. The acutal configuration file settings are provided here to avoid various command line quoting issues in the various shells available to you.

Add the following lines to your `.gitconfig`. This file should be in your home directory in `C:\Users`:

```
[diff]
    tool = diffmerge
[difftool "diffmerge"]
    cmd = C:/Program\\ Files/SourceGear/Common/DiffMerge/sgdm.exe
        \"$LOCAL\" \"$REMOTE\"

[merge]
    tool = diffmerge
[mergetool "diffmerge"]
    trustExitCode = true
    cmd = C:/Program\\ Files/SourceGear/Common/DiffMerge/sgdm.exe
        -merge -result=\"$MERGED\" \"$LOCAL\" \"$BASE\" \"$REMOTE\"
```

Note that both of the `cmd = ...` lines were wrapped for this document and should appear as a single line in your `.gitconfig` file. You can verify your settings with the `git config --list` command.

The pathnames above assume that the MSI installer was used to install DiffMerge in the standard location. If you installed DiffMerge from the ZIP package, be sure to adjust the pathname to `sgdm.exe` accordingly.

Git for Windows (MSysGit) or Git Cmd

The following commands in a `Command Prompt` window will update your `.gitconfig` to configure Git use DiffMerge:

```
C:\> git config --global diff.tool diffmerge
C:\> git config --global difftool.diffmerge.cmd
    "C:/Program\ Files/SourceGear/Common/DiffMerge/sgdm.exe
        \"$LOCAL\" \"$REMOTE\""

C:\> git config --global merge.tool diffmerge
C:\> git config --global mergetool.diffmerge.trustExitCode true
```

```
C:\> git config --global mergetool.diffmerge.cmd
    "C:/Program\ Files/SourceGear/Common/DiffMerge/sgdm.exe
        -merge -result=\"$MERGED\" \"$LOCAL\" \"$BASE\" \"$REMOTE\""
```

This will also work for the Gitcmd that is offered on the Git website, but not the Git Bash shell.

Note that both of the `....diffmerge.cmd ...` lines were wrapped for this document. You can verify your settings with the `git config --list` command.

The commands above assume that the MSI installer was used to install DiffMerge in the standard location. If you installed DiffMerge from the ZIP package, be sure to adjust the pathname to `sgdm.exe` accordingly.

Git Under Cygwin or Git Bash Shell

The following instructions are for the version of Git included with Cygwin or the default Git Bash Shell. The following commands are run from a cygwin bash shell window.

This version of Git invokes external diff/merge tools using cygwin-style pathnames, such as `/tmp/foo` (which is relative to the root of the cygwin directory). These pathnames are not understood by DiffMerge. A shell script wrapper `sgdm_cygwin.sh` has been provided (in the same directory as the `sgdm.exe` executable) to translate these pathnames and launch DiffMerge.

The following commands will update your `.gitconfig` to let the Cygwin version of Git use DiffMerge:

```
$ git config --global diff.tool diffmerge
$ git config --global difftool.diffmerge.cmd
    "C:/Program\ Files/SourceGear/Common/DiffMerge/sgdm_cygwin.sh
        -p1=\"\$LOCAL\" -p2=\"\$REMOTE\""

$ git config --global merge.tool diffmerge
$ git config --global mergetool.diffmerge.trustExitCode true
$ git config --global mergetool.diffmerge.cmd
    "C:/Program\ Files/SourceGear/Common/DiffMerge/sgdm_cygwin.sh
        -merge -result=\"\$MERGED\"
            -p1=\"\$LOCAL\" -p2=\"\$BASE\" -p3=\"\$REMOTE\""
```

Note that both of the `....diffmerge.cmd ...` lines were wrapped for this document. You can verify your settings with the `git config --list` command.

Note that each of the input pathnames is associated with a `/pX=` option instead of being an unbound parameter.

The commands above assume that the MSI installer was used to install DiffMerge in the standard location. If you installed DiffMerge from the ZIP package, be sure to adjust the pathname to `sgdm.exe` accordingly.

Mercurial (Hg)

Information on configuring an external diff/merge tool can be found in:

http://www.selenic.com/mercurial/wiki/MergeToolConfiguration

http://www.selenic.com/mercurial/wiki/index.cgi/MergeProgram

http://www.selenic.com/mercurial/wiki/ExtdiffExtension

The command for comparing 2 versions of a file is `hg diff foo.c`. This will compare the baseline and working-folder versions of foo.c and print unified-diffs in the terminal window. If you add the settings below to your `mercurial.ini` or `.hgrc` file, you can type `hg diffmerge foo.c` and have DiffMerge launched to compare the 2 file versions.

The `hg merge` and `hg resolve` commands will merge files. When the settings below are added, Mercurial will try to use DiffMerge to perform the merge. There are lots of Mercurial options to control how external merge tools are selected (based upon priority and file type). Please see the above references for configuration options.

Settings for Linux

First confirm that `/usr/bin/diffmerge` is present. On Linux, this should have been installed for you.

Add the following lines to your `.hgrc`:

```
[extensions]
hgext.extdiff =

[extdiff]
cmd.diffmerge = /usr/bin/diffmerge

[merge-tools]
diffmerge.executable = /usr/bin/diffmerge
diffmerge.args = --result=$output  $local $base $other
diffmerge.binary = False
diffmerge.symlinks = False
diffmerge.gui = True
diffmerge.premerge = True
```

Settings for OS X

First confirm that `/usr/local/bin/diffmerge` is present. If you used the PKG Installer, this was installed when /Applications/DiffMerge.app was installed. If you used the DMG file, refer to the instructions for installing the Extras.

Add the following lines to your `.hgrc`:

```
[extensions]
hgext.extdiff =

[extdiff]
cmd.diffmerge = /usr/local/bin/diffmerge

[merge-tools]
diffmerge.executable = /usr/local/bin/diffmerge
diffmerge.args = --result=$output  $local $base $other
```

```
diffmerge.binary = False
diffmerge.symlinks = False
diffmerge.gui = True
diffmerge.premerge = True
```

Settings for Windows

On Windows, the actual configuration settings depend upon which distribution of Mercurial you are using.

Mercurial for Windows

Add the following lines to your `mercurial.ini` file:

```
[extensions]
hgext.extdiff =

[extdiff]
cmd.diffmerge = C:\Program Files\SourceGear\Common\DiffMerge\sgdm.exe

[merge-tools]
diffmerge.executable = C:\Program Files\SourceGear\Common\DiffMerge\sgdm.exe
diffmerge.args = -merge -result=$output  $local $base $other
diffmerge.binary = False
diffmerge.symlinks = False
diffmerge.gui = True
diffmerge.premerge = True
```

Mercurial Under Cygwin

The following instructions are for the version of Mercurial included with Cygwin. The following commands are run from a cygwin bash shell window.

This version of Mercurial invokes external diff/merge tools using cygwin-style pathnames, such as `/tmp/foo` (which is relative to the root of the cygwin directory). These pathnames are not understood by DiffMerge. A shell script wrapper `sgdm_cygwin.sh` has been provided (in the same directory as the `sgdm.exe` executable) to translate these pathnames and launch DiffMerge.

```
[extensions]
hgext.extdiff =

[extdiff]
cmd.diffmerge = C:/Program Files/SourceGear/Common/DiffMerge/sgdm_cygwin.sh

[merge-tools]
diffmerge.executable = C:/Program Files/SourceGear/Common/DiffMerge/sgdm_cygwin.sh
diffmerge.args = -merge -result=$output  -p1=$local -p2=$base -p3=$other
diffmerge.binary = False
diffmerge.symlinks = False
diffmerge.gui = True
diffmerge.premerge = True
```

Note that each of the input pathnames is associated with a `/pX=` option instead of being an unbound parameter.

Mercurial Tips

- It is **very important** that the `--result=$output` argument be present. Without this, DiffMerge will not set the exit status as Mercurial expects and it will save the merge result in the original file, which in this case is a Mercurial temporary file.
- If you want, you can add:

  ```
  -t1="Local Version" -t2=$output -t3="Other Version" --caption=$output
  ```

  to the `diffmerge.args` line. This will give you better window titles.