

---

SilverAge Software, Inc.

# HandyFile Find And Replace: Office Edition

*(also applies to Text Aid Kit)*

Version 3.3

User Guide

---

# CONTENTS

<b>GENERAL INFORMATION .....</b>	<b>3</b>
<b>GETTING STARTED WITH HANDYFILE FIND AND REPLACE.....</b>	<b>4</b>
<b>FILES AND TEXT TAB.....</b>	<b>6</b>
COMMON CONTEXT.....	6
ADVANCED CONTEXT .....	7
<b>DATE TAB.....</b>	<b>9</b>
<b>PROPERTIES TAB .....</b>	<b>10</b>
<b>STORAGE FOLDERS TAB .....</b>	<b>12</b>
<b>COLLECTOR TAB .....</b>	<b>15</b>
<b>REPORT AREA .....</b>	<b>17</b>
<b>TOOLBAR .....</b>	<b>18</b>
<b>PROGRESS INDICATOR.....</b>	<b>19</b>
<b>FOLDER MANAGER DIALOG .....</b>	<b>20</b>
<b>SCENARIO EDITOR DIALOG.....</b>	<b>21</b>
CREATING SCENARIOS .....	21
DIALOG FIELDS.....	22
<b>SCENARIO EXPORT DIALOG .....</b>	<b>24</b>
EXPORTING SCENARIOS .....	24
DIALOG FIELDS.....	24
<b>MULTILINE EDITOR DIALOG.....</b>	<b>26</b>
<b>CONFIGURATION EXPORT DIALOG .....</b>	<b>28</b>
<b>REGULAR EXPRESSION LABORATORY.....</b>	<b>30</b>
HOW TO USE THE LABORATORY .....	32
<b>OPTIONS .....</b>	<b>34</b>
OPTIONS DIALOG - PROCESSING.....	34
OPTIONS DIALOG - INTERFACE .....	37
OPTIONS DIALOG - ALERTS.....	39
<b>REGULAR EXPRESSIONS .....</b>	<b>42</b>
REMARKS.....	44
<b>USING REGULAR EXPRESSIONS .....</b>	<b>46</b>
WHAT ARE REGULAR EXPRESSIONS? .....	46
WHY WOULD YOU USE REGULAR EXPRESSIONS.....	46
MATCHING OPERATORS .....	46
EXPRESSIONS.....	48
<b>REGULAR EXPRESSION EXAMPLES .....</b>	<b>50</b>
<b>HOWTO: PERFORM SIMPLE FILE SEARCH.....</b>	<b>53</b>
<b>HOWTO: SEARCH FOR TEXT IN FILES .....</b>	<b>54</b>
<b>HOWTO: SEARCH FOR TEXT IN FILES IN MULTIPLE FOLDERS .....</b>	<b>55</b>
<b>HOWTO: SEARCH FOR TEXT USING REGULAR EXPRESSIONS .....</b>	<b>56</b>
<b>HOWTO: FIND AND REPLACE TEXT IN WORD AND EXCEL FILES .....</b>	<b>57</b>
<b>HOWTO: FIND AND REPLACE TEXT IN FILES.....</b>	<b>58</b>
<b>HOWTO: SEARCH AND REPLACE TEXT USING REGULAR EXPRESSIONS .....</b>	<b>59</b>
<b>HOWTO: USE SCENARIOS TO AUTOMATE WORK.....</b>	<b>60</b>
<b>HOWTO: USE COMMAND LINE SWITCHES .....</b>	<b>62</b>
FIND AND REPLACE SIMPLE TEXT IN FILES .....	62
FIND AND REPLACE TEXT IN FILES USING REGULAR EXPRESSIONS .....	62
FIND AND REPLACE TEXT IN FILES USING A SCENARIO .....	63
<b>COMMAND LINE .....</b>	<b>64</b>
<b>CONFIGURATION FILE FOR USE IN COMMAND LINE.....</b>	<b>67</b>

# General Information

**HandyFile Find and Replace** is an assistant simple to use tool to help you process text in a number of files with a single click.

**HandyFile Find and Replace: Office Edition** supports the following file types.

- **Plain text files:** SGML documents (HTML, XML, etc.), source code and any other plain text files.
- **Microsoft® Office™ files:** **Microsoft Word** documents (.doc, .rtf) and **Microsoft Excel** documents (.xls). This feature requires that these components of the **Microsoft Office** are installed on your computer.

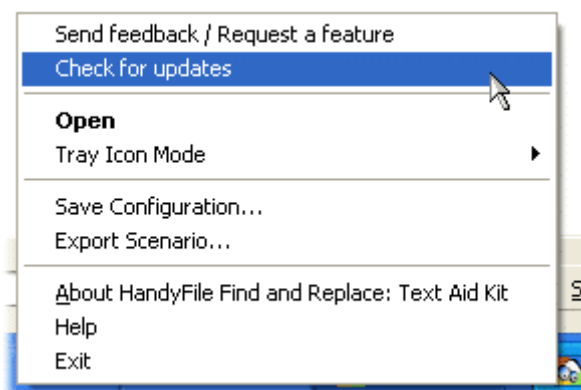
**HandyFile Find and Replace: Text Aid Kit** does not support the **Microsoft® Office™** files.

## Major advantages

- Works fast, even with regular expressions.
- Supports regular expressions with extended syntax.
- Supports multiline searches and replacements.
- Supports various command line parameters.
- Comprehensive and clear user interface.
- Text viewer to browse through the results!
- Binary viewer to display the binary files!
- Intelligible diagnostic messages.
- Keeps your files safe as the process of replacement is undertaken in memory, not in files.
- Ability to modify read-only files and restore their attributes after processing.
- Custom back-up and target folder modes.
- Supports environment variables expansion in paths.
- Search and replace *Scenarios* to deliver the utmost operability!


## Checking for updates

You can easily check for product updates by right-clicking the application icon in the system tray and selecting command **Check for updates**. Please ensure that you are connected to the Internet before doing so.




# Getting Started with HandyFile Find And Replace

This is the short description of operating the **HandyFile Find And Replace**. After you have read it, you can start working with the software. If you need more information on any particular feature, please read the corresponding topics. Many frequently used operations are described in the **Common Tasks** section.

1. Set the path to the **folder** containing the files to process. You can click the **Browse** button  to locate the folder.
2. Check the **Recurse subfolders** box if you want to change files in the recurred subfolders of the folder you have specified.
3. Set the one or more **file types** (masks).

## Examples:

```
*.txt  
*.txt;*.html  
prod*.html;prod*.asp
```

4. If you need to use the regular expressions, check the **Regular expressions** box.
5. Type in the text to search for (**Find What** field).  
*(Optional)* If your text contains multiple lines, you can click button  to open the **Multiline Editor** to enter the text.
6. Type in the text to replace the found text (**Replace with** field). You can leave this field empty to delete the found text.
7. *(Optional step)* **Additional search criteria** can be set if needed:

## Date Tab

1. Specify whether change all files or those having the definite date only. You can set the range implicitly, or in days from the current day, or in months from the current month. The day range includes the first and the last day.

## Properties Tab

2. Specify whether change all files or those having the definite size only. When setting the size, please note that when kilobytes are chosen as the measurement unit, this value is multiplied by 1024.
  3. Specify should the application process read-only files. If needed, you may choose to set this attribute back after the file is processed.
8. Click **Search** to just search for files, or **Replace** to replace. The **Search only** box forces the search mode even if **Replace** is clicked.

The process of search (and replace) began. You can always interrupt it by pressing **Stop**.

After the process of search and/or replacement is over, you may open the desired file with the HandyHTML Studio by double-clicking it in the list.

# | User Interface

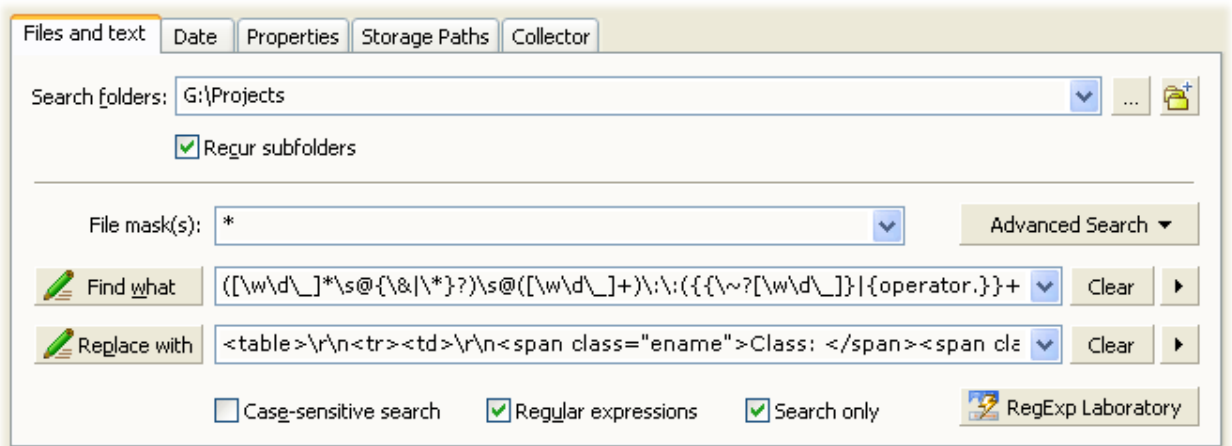
# Files and Text Tab

This is the main page that allows you to set preferences and parameters for the search and replace process.


This page can have two visual contexts - common and advanced.



## Common Context

The *common* visual context is active by default and is most commonly used. The illustration below shows how the tab looks in this mode.



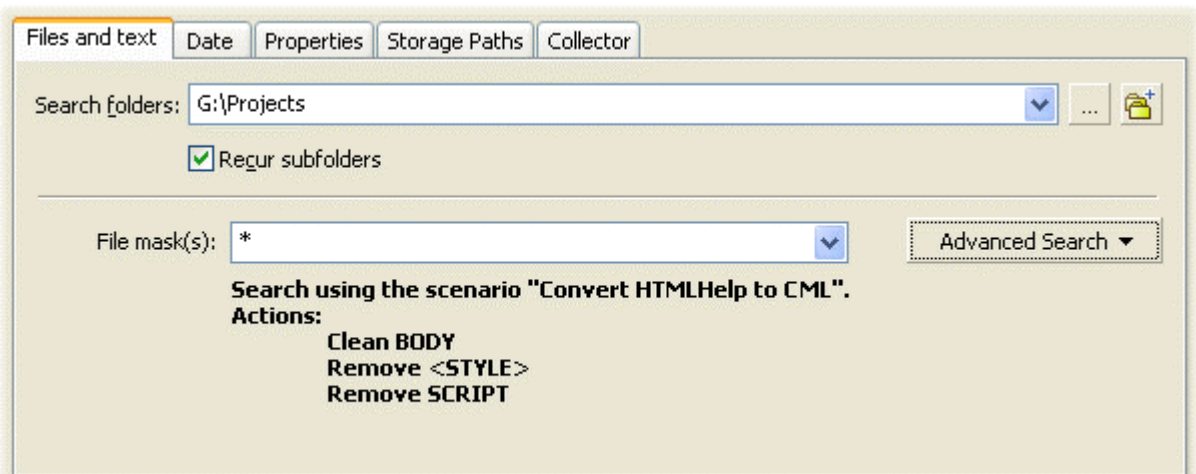
The table below describes the tab fields.

Field	Description
Search folder	Allows to set the folder that the application will scan for files. You can specify here multiple folders separated with semicolon (;). To facilitate assigning multiple folders, you can open the <a href="#">Folder Manager</a> by clicking the button  . The path that you specify here may contain not only canonic path. You can include system environment variables here. For example, you can specify path similar to the following: <code>%WINDIR%\TEMP\</code>
Recurse subfolders	With this option checked, the HFFR will scan all subfolders of the specified search folder(s) recursively.
File mask	One or more semicolon-separated wildcard masks to match against files in folder(s) to search. For example, the following mask set will match <code>htm</code> , <code>html</code> and <code>aps</code> files: <code>*.htm;*.html;*.asp</code>
Advanced search button	Clicking this button invokes a menu containing the Scenario options. If you select a scenario or open the <a href="#">Scenario Manager</a> dialog and activate a scenario, the <b>Files and Text</b> tab switches to the <b>Advanced</b> mode (see below).
Find What button	Click this button to open the <b>Multiline Editor</b> dialog. This dialog allows entering text with multiple lines in normal mode, converting line breaks to <code>\r</code> and <code>\n</code> as appropriate.

Field	Description
<b>Find What</b> text field	<p>This field is used to enter the text to search for. You can click the drop-down button to select the previously specified text. The recent text list can contain up to 15 lines.</p> <p>You can leave this field empty to only search for <i>all</i> files matching the supplied <i>mask</i> in the <i>search folder(s)</i>.</p>
<b>Replace With</b> button	Same purpose as for <b>Find What</b> button, but applies to the replacement text.
<b>Replace With</b> text field	Same purpose as for <b>Find What</b> text field, but applies to the replacement text.
<b>Clear</b> buttons	Clears text in the <b>Find What</b> or <b>Replace With</b> field.
<b>Control Escape</b> buttons 	<p>If you have the option <b>Regular Expressions</b> selected, shows a menu with regular expressions that you can select and insert.</p> <p>If the <b>Regular Expressions</b> option is not selected, shows a menu with common control escapes.</p>
Case-sensitive search	<p>Checking this button allows restrict your searches to those phrases that match the sought text exactly.</p> <p>If using Regular Expressions, you can control case sensitivity with the regular expression switches(\L, \U, \c and \C). See <a href="#">more information on Regular Expressions</a>.</p>
Regular expressions	Check this button to allow searches with <a href="#">Regular Expressions</a> .
Search only	This option overrides the behaviour of the <b>Replace</b> button. If checked, no replacements will occur even if the sought text is found.
Regular Expression Laboratory button 	Opens the Regular Expression Laboratory window, where you can test your regular expressions.

## Advanced Context

The Advanced visual context is active in case any search Scenario is active. In this mode, all standard text options are disabled and the corresponding controls hidden. The scenario name is displayed instead:



# Date Tab

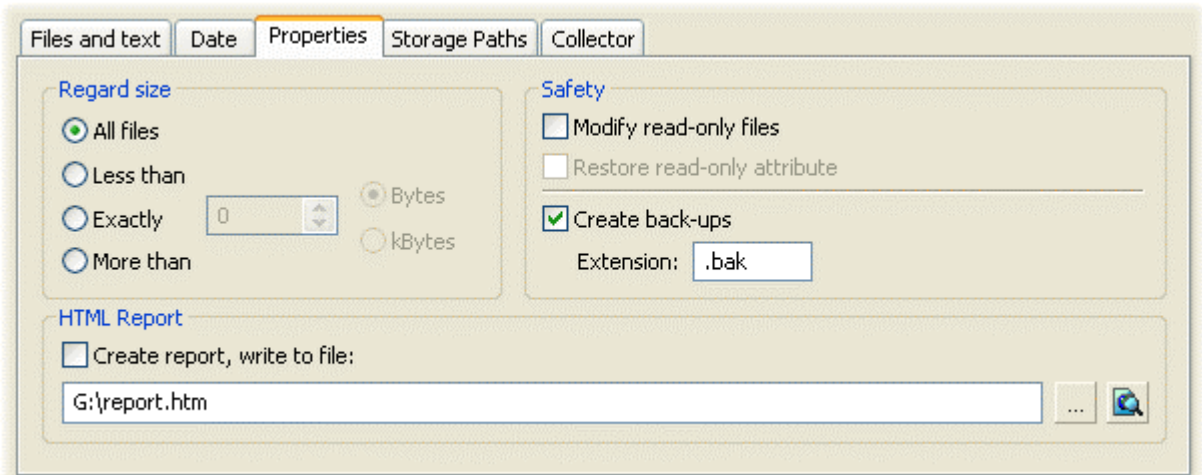
This tab allows you to specify the date or range of dates that the matching files should have been created, opened or changed.

The table below describes the tab fields.

Field	Description
All files	Default selection. Denotes that all found files will match, disregarding time stamp.
Only files	Enables the drop-down list allowing to select the type of date to match: <ul style="list-style-type: none"> <li>• Changed</li> <li>• Created</li> <li>• Opened</li> </ul> <p>After you have selected the desired date type, you should select the date using the following control groups.</p>
<b>between</b> dates	Select this option if you know the exact dates of file creation and want to find them. Choose dates from the drop-down calendars. The search includes files that match the starting and ending dates as well.
<b>within</b> days	Select this option if you want to find files changed, created or opened within the specified number of days. Set the number of days to 0 (null) to find files with today's file stamp.
<b>within</b> months	Select this option if you want to find files changed, created or opened within the specified number of months.
Retain time stamp	Tells to leave the times of the modified file as-is, without change.

# Properties Tab

This page contains various options to control file matching and perform user-defined operations.



The tables below describe the tab fields.

## Regard Size Group


Field	Description
All files	Default selection. Denotes that all found files will match, disregarding the size.
Less than	Denotes that files sized less than specified value will only match.
Exactly	Denotes that files with the specified size will only match.
More than	Denotes that files with the size more than specified value will only match.
Bytes or kBytes	Select the units of file size measure. To precisely set the size, select <b>Bytes</b> .  <b>Note</b> Please keep in mind that <i>1 kByte</i> equals <i>1024 bytes</i> .

## Safety Group

Field	Description
Modify read-only files	By default, this option is disabled. If checked, the HandyFile will attempt to modify files with the R/O attribute.
Restore read-only attribute	If the <b>Modify read-only files</b> option is on, denotes that the R/O attribute will be restored for such files after they are modified.
Create back-ups	If enabled, directs creating copies of files to be modified.  By default, back-up copies are created in the same directory as the modified file and have the <code>.bak</code> extension. To alter this behavior, open the <a href="#">Storage Paths Tab</a> .
Extension	Default extension to append to back-up copies. Files with this extension are excluded from search.

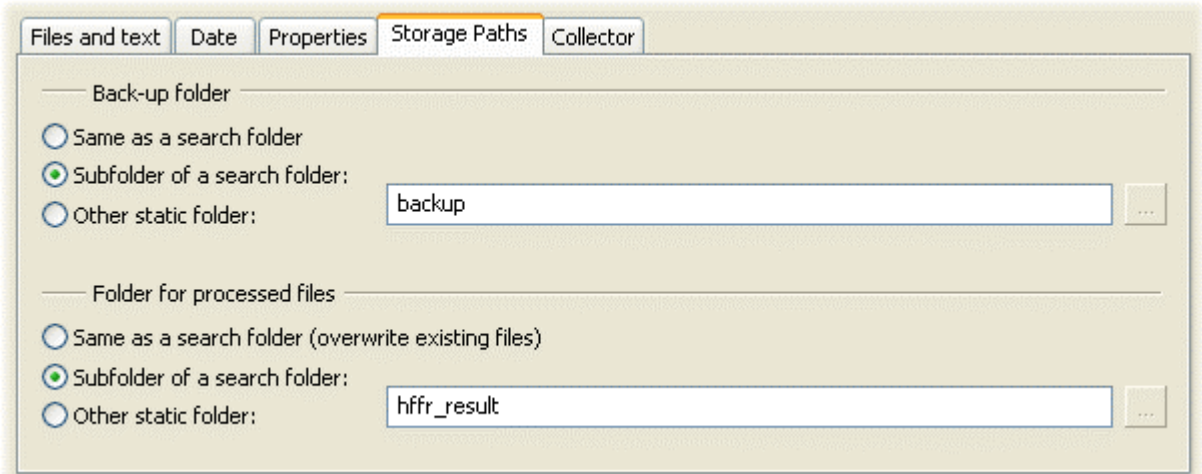
## HTML Report

---

<b>Field</b>	<b>Description</b>
Create report	If enabled, the report will be created.
File path field	Specify path and name of the HTML report file here.
<b>View report</b> button 	Click to open the report with system default browser.

# Storage Folders Tab

The **HandyFile Find And Replace** is capable of storing the back-up copies and processed files in folders other than specified source folder(s). This tab allows to specify rules that the **HandyFile** will use when creating and copying files.



The tables below describe the tab fields.

## Back-up Folder Group

Any of the following options is only applicable if the **Create back-ups** option is set on the [Properties Tab](#). Otherwise, no backup copies are created.

Field	Description
Same as a search folder	<p>Default selection. The following rules apply.</p> <ul style="list-style-type: none"> <li>• Back-up files are stored in the same directory as the source files.</li> <li>• Back-up files have the extension supplied on the <a href="#">Properties Tab</a>.</li> <li>• If the back-up file already exists, it will be overwritten.</li> <li>• Files having the extension supplied on the <a href="#">Properties Tab</a> are excluded from search.</li> </ul>
Subfolder of a search folder	<p>Denotes that the back-up files will be created in a folder created in the search folder. You can set the folder name in the text field. The following rules apply.</p> <ul style="list-style-type: none"> <li>• The back-up subfolder is created in the root search folder (or folders, if multiple folders are specified).</li> <li>• The back-up subfolder structure mirrors the search folder structure recursively.</li> <li>• Back-up files do not change the extension; they are simply copied.</li> <li>• Back-up folders are excluded from the search.</li> </ul>

Field	Description
Other static folder	<p>Denotes that the back-up files will be created in a supplied static folder. The folder path may be a fully qualified path:</p> <p style="text-align: center;"><code>c:\My files\Backup</code></p> <p>or they may include environment variables:</p> <p style="text-align: center;"><code>%WINDIR%\TEMP</code></p> <ul style="list-style-type: none"> <li>• Back-up files do not change the extension; they are simply copied.</li> <li>• Static back-up folders are excluded from the search.</li> </ul>

### Folder for Processed Files Group

These options control where the processed files are stored. Any of the following options is only applicable if the **Find What** field is not empty and the **Search only** option is *not* checked on the [Files and Text Tab](#) - that is, any replacement may have occurred.

Field	Description
Same as a search folder	<p>Default selection.</p> <p style="text-align: center;"><b>Note</b> The existing files will be overwritten.</p>
Subfolder of a search folder	<p>Denotes that the processed files will be stored in a folder created in the search folder. You can set the folder name in the text field. The following rules apply.</p> <ul style="list-style-type: none"> <li>• The target subfolder is created in the root search folder (or folders, if multiple folders are specified).</li> <li>• The target subfolder structure mirrors the search folder structure recursively.</li> <li>• Target folders are excluded from the search.</li> </ul>
Other static folder	<p>Denotes that the back-up files will be created in a supplied static folder. The folder path may be a fully qualified path:</p> <p style="text-align: center;"><code>c:\My files\Backup</code></p> <p>or they may include environment variables:</p> <p style="text-align: center;"><code>%WINDIR%\TEMP</code></p> <ul style="list-style-type: none"> <li>• The target subfolder structure mirrors the search folder structure recursively.</li> <li>• Static target folders are excluded from the search.</li> </ul>

### Remarks

General rules for creating names for directories (and files) on the Windows platform include the following:

- Use any character in the current code page for a name, but do not use a path separator, a character in the range 0 through 31, or any character explicitly disallowed by the file system. A name can contain characters in the extended character set (128–255).
- Use the backslash (\), the forward slash (/), or both to separate components in a path. No other character is acceptable as a path separator.
- Use a period (.) as a directory component in a path to represent the current directory.
- Use two consecutive periods (..) as a directory component in a path to represent the parent of the current directory.

- Use a period (.) to separate the base file name from the extension in a directory name or file name.
- Do not use the following characters in directory names or file names, because they are reserved:

<> : " / \ |

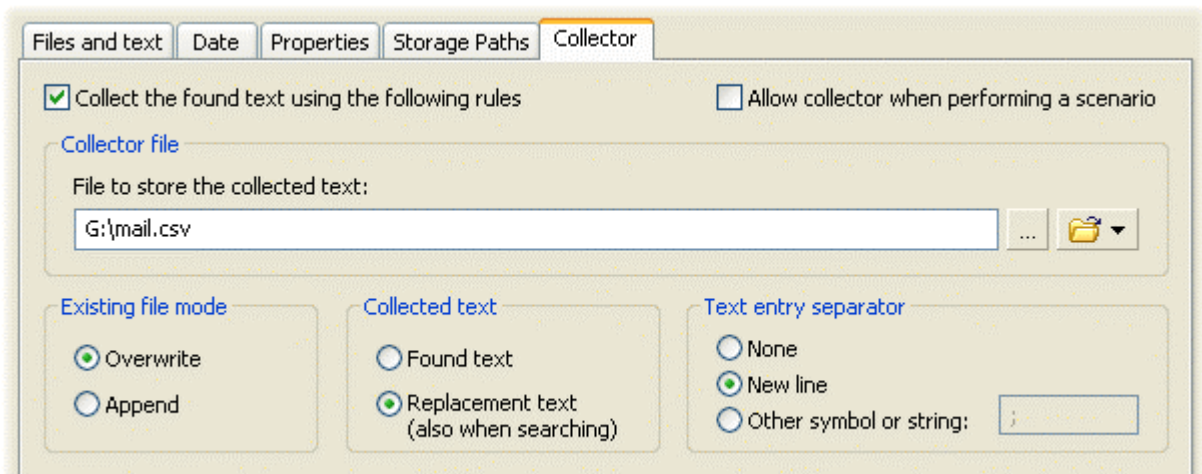
- Do not use device names, such as *aux*, *con*, *lpt1*, and *prn*, as file names or directory names.
- Do not assume case sensitivity. Consider names such as *OSCAR*, *Oscar*, and *oscar* to be the same.
- The following reserved words cannot be used as the name of a file: CON, PRN, AUX, CLOCK\$, NUL, COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, and LPT9. Also, reserved words followed by an extension—for example, NUL.tx7—are invalid file names.

(Source: *Microsoft Developer Network* <http://msdn.microsoft.com>)

# Collector Tab

This page allows you to store the found text (or the text you want to replace the found text with) in an arbitrary file. Such collector function allows to:

- generate CSV files with e-mail addresses from a number of HTML pages;
- form a source code documentation draft by collecting the function prototypes;
- perform any other operations that require data gathering.



## Using The Collector

Follow the steps below to gather text from files and store it in a collector file.

1. Specify the search location, file mask(s), text to find (and optionally to replace - this mostly applies to regular expressions) as usually.
2. On the **Collector** tab, check the box **Collect...**
3. Specify the collector file path and name.
4. If you want to create a new collector file, select the **Overwrite** file mode. Otherwise, select the **Append** mode.
5. Set what kind of text you want to collect - **Found** text of **Replacement** text. Choosing the **Replacement text** option allows you to alter the found text in any way, even if you do not perform the actual replace.
6. Choose the kind of **separator** to insert between each added text entry.
7. Click the **Search** button (or **Replace** if you need to simultaneously perform the replace operation).



The tables below describe the tab fields.

### General Options

Field	Description
Collect the found text using the following rules	If checked, creates (or reuses) the collector file and stores the text in it.  If unchecked, no collecting occurs.
Allow collector when	If this options checked, the collector function is allowed if you

Field	Description
performing a scenario	are performing one of the scenarios.

### Collector file

Field	Description
File to store the collected text	Path and name of the file to collect the text. If the file and/or path do not exist, they will be created.
<b>Browse for file button</b> 	Click this button to locate and select the file to store the collected text.
<b>Open file button</b> 	Clicking this button will display a menu with the following commands: <ul style="list-style-type: none"> <li>▪ <b>Open File</b> - opens the file using the associated application.</li> <li>▪ <b>Open File Folder</b> - opens the Windows Explorer and locates the file in its right pane.</li> </ul>

### Existing file mode

Field	Description
Overwrite	Select this option to replace the data in the existing file with the new contents. In the file does not exist, it will be created.
Append	Select this option to append the collected data to the file contents. In the file does not exist, it will be created.

### Collected text

Field	Description
Found text	If selected, the found text is stored in the collector file.
Replacement text (also when searching)	If selected, the text that you've specified in the <b>Replace With</b> field is stored in the collector file.  You can store the replacement text even if simply <i>searching</i> files for text. This option is generally useful for searches with Regular Expressions, as it allows you to alter the found text in any aspect.

### Text entry separator

Field	Description
None	The text is collected as-is, no additional data is written to a file.
New line	Finalizes each write operation to a collector file with a CR+LF pair.
Other symbol or string	Allows to insert arbitrary text between each write. Useful for creating CSV files.

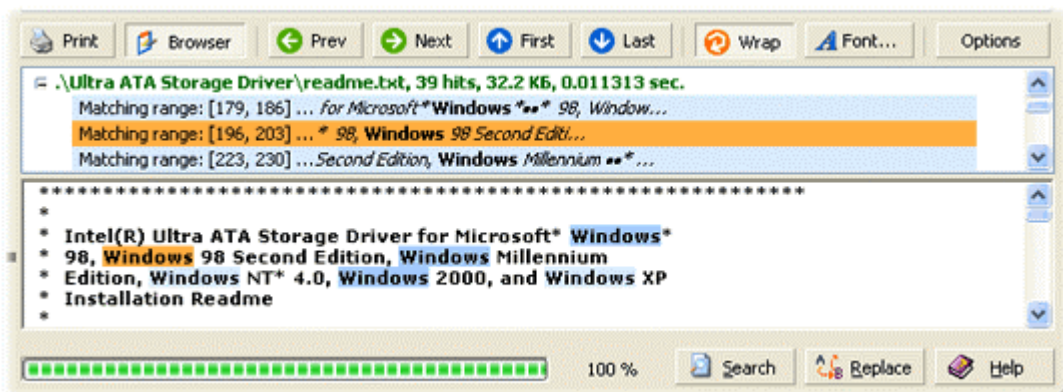
# Report Area

The **Report Area** displays information on the find (and replace) process. The **Report Area** consists of the two main areas: the **File Match Tree** and the **Text Viewer**.

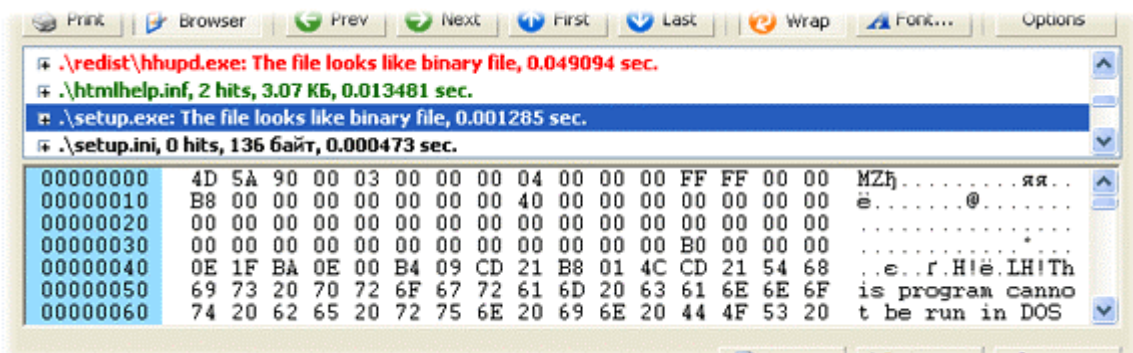
The **File Match Tree** contains information on files that has been processed and matches. Root elements of the tree displays basic information: relative file path, number of hits and the time spent for the operation.

- If any match(es) has been found in the file, they are displayed as the *child elements of the file*.
- Click the *plus icon* to expand the file element and view matches.
- Right-click on any file to invoke a context menu with options to open the file and explore the folder that contains this file.

The **Text Viewer** loads the currently selected file and highlights the matching text so you can easily browse through it. To facilitate browsing, you can select a match in the tree and view it in the text viewer.



If the viewed file is considered binary (see Options for more information), the text view is replaced with hexadecimal binary view:



Since version 3.1, there are 2 buttons to start processing: **Search** and **Replace**. Behaviour of the **Replace** button may be overridden by using the [Search only](#) option.

# Toolbar

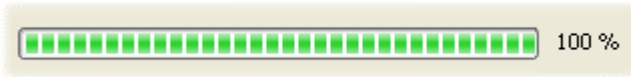


The **Toolbar** contains various useful commands. The command description is provided below.

Button	Description
Print	Click to print the current file tree report.
Browser	Shows or hides the text view area.
Prev	Click to browse to the previous match.
Next	Click to browse to the next match.
First	Click to go to the first match file in the report.
Last	Click to go to the last file in the report.
Wrap	Toggles the word wrapping mode in the text view area.
Font	Click to select the font for the text view area.
Options	<p>Click to display the <a href="#">Options</a> dialog, which allows you to customize various useful options.</p> <p>The drop-down button displays a menu with the following commands:</p> <ul style="list-style-type: none"> <li>▪ <b>Load Configuration</b> - opens the standard Windows file dialog, allowing you to select and apply the previously saved configuration file (.hfcfg) .</li> <li>▪ <b>Save Configuration</b> - opens the Configuration Export Dialog, allowing you to save the current configuration and optionally send it via the e-mail.</li> </ul>

# Progress Indicator

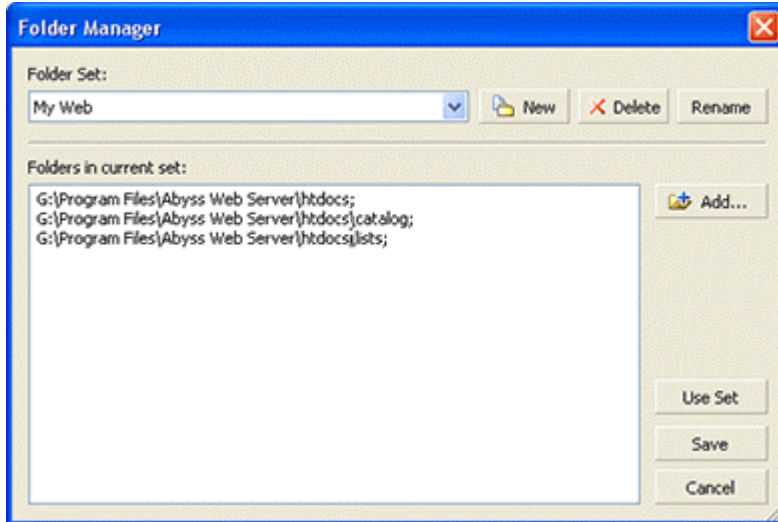
The progress indicator shows the relative amount of files processed and to be processed.



In some circumstances, the indicator may show value less than 100% after the processing completes. This means that not all files in the search folder satisfy the search parameters.

# Folder Manager Dialog

The **Folder Manager** dialog is the convenient front-end for specifying and storing sets of multiple folders in which the search process is to be performed.



To create a new folder set, click the **New** button. After the new set is created, you can add folders to it by clicking the **Add...** button.

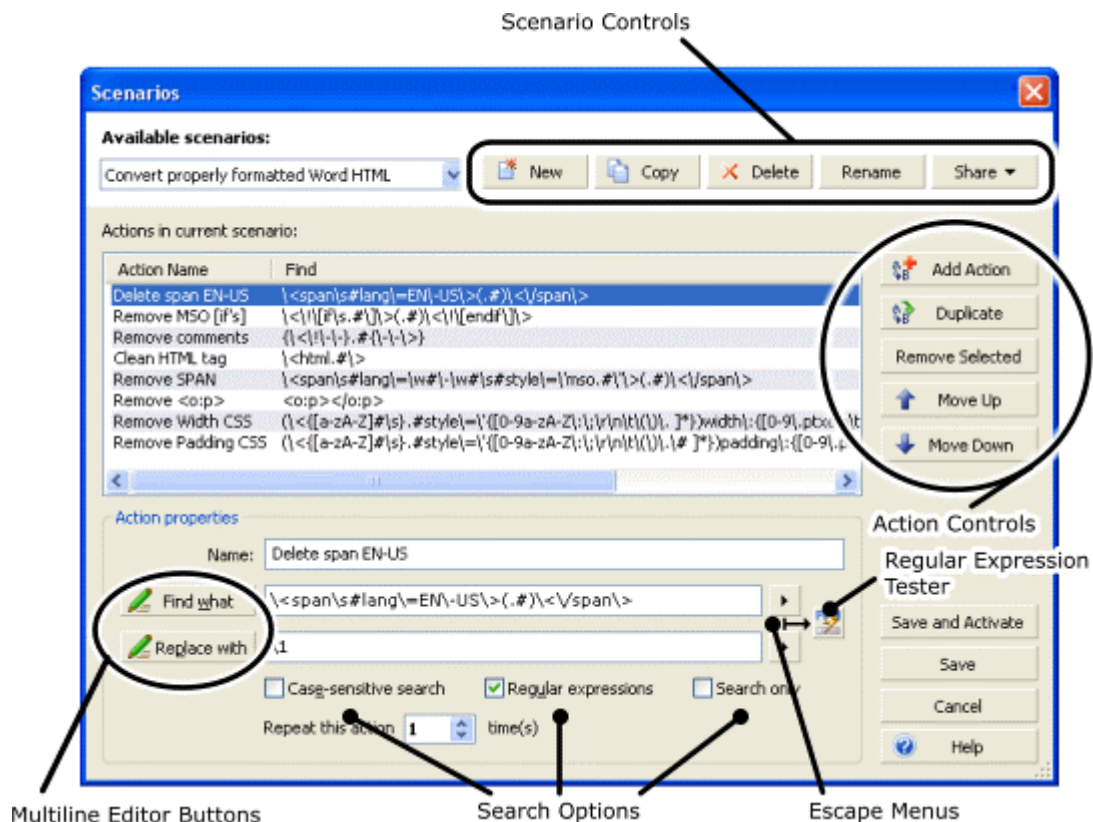
The table below describes the dialog fields.

Field	Description
<b>Managing Sets</b>	
Folder Set	Displays a list of existing folder sets. You can select the set from the list by clicking on a drop-down arrow.
New	Click to create a new set.
Delete	Click to delete the selected set.
Rename	Click to rename the selected set.
<b>Managing Folders in Set</b>	
Folders in current set	Edit field displaying the folders in the current set. You can edit them manually here. Folders are separated by semicolon.
Add	Click to open the folder selection dialog. Select a folder and click <b>OK</b> to add it to the set.
Use Set	Click this button to save the set and apply it; that is, copy the folder paths to the <b>Search folders</b> field of the <a href="#">Files and Text Tab</a> .
Save	Click this button to save changes and close the dialog.
Cancel	Click this button to close the dialog without saving changes.

# Scenario Editor Dialog

Very often the search and replace operation requires that you replace different multiple strings in a file or files. You can do it by specifying the search and replace text consequently. This is a monotonous annoying routine. The **HandyFile Find And Replace** offers a neat solution to this problem - the **Scenarios**.

A **Scenario** is a set of search and replace text and conditions applied consequently to a file or files. Such a set of text and conditions is referred to as an **Action**. You can create a scenario using the **Scenarios** dialog.



## Creating Scenarios

Follow the steps below to create and use a scenario.

1. Click the **New** button to create a scenario. In the **Name** dialog box, provide a name for the scenario.
2. Click the **Add Action** button to create a new action.
3. Specify the sensible name for the action. This is essential if you want to edit your scenario some time later.
4. Specify text and options for the action.
5. Repeat steps 4 and 5 to create more actions.
6. Click either **Save and Activate** or **Save** button. If you click **Save and Activate**, the new scenario will be activated.

## Dialog Fields

The tables below describe the dialog fields.

### Available Scenarios Group



Field	Description
<i>Scenario List</i>	A drop-down list containing the existing scenarios. Select the scenario by clicking on a drop-down arrow.
New	Click to create a new scenario.
Copy	Click to create a new scenario containing the same set of commands as the selected scenario; that is, <i>copy</i> the selected scenario.
Delete	Click to delete the selected scenario.
Rename	Click to assign a new name to a scenario.
Share	This feature explores the teamwork functions. Clicking this button will display a menu with the following commands: <ul style="list-style-type: none"><li>▪ <b>Import Scenarios</b> - opens the standard Windows file dialog, allowing you to select and import the previously exported scenario file (.hfscc).</li><li>▪ <b>Export Scenarios</b> - opens the <b>Scenario Export Dialog</b>, allowing you to select, export and optionally send the exported scenarios via the e-mail.</li></ul>

### Scenario Actions Group

Field	Description
<i>Action List</i>	A list of actions in the current scenario. To display the action properties, simply select it.
Add Action	Click to add a new action to the scenario.
Duplicate	Click to create a new action and copy the currently selected action into the created one.
Remove Selected	Click to remove the selected action from the selected scenario.
Move Up	Click to move the selected action upwards. Press and hold the button down to start moving the action up.
Move Down	Click to move the selected action downwards. Press and hold the button down to start moving the action down.

### Action Properties Group

Field	Description
Name	A list of actions in the current scenario. To display the action properties, simply select it.
<b>Find What</b> button	Click this button to open the <b>Multiline Editor</b> dialog. This dialog allows entering text with multiple lines in normal mode, converting line breaks to <code>\r</code> and <code>\n</code> as appropriate.
<b>Find What</b> text field	This field is used to enter the text to search for.
<b>Replace With</b> button	Same purpose as for <b>Find What</b> button, but applies to the replacement text.
<b>Replace With</b> text field	Same purpose as for <b>Find What</b> text field, but applies to the replacement text.
<b>Control Escape</b>	If you have the option <b>Regular Expressions</b> selected, shows a menu with

Field	Description
buttons 	regular expressions that you can select and insert.  If the <b>Regular Expressions</b> option is not selected, shows a menu with common control escapes.
<b>Regular Expression Laboratory</b> button 	Opens the <b>Regular Expression Laboratory</b> window, where you can test your regular expressions.
Case-sensitive search	Checking this button allows restrict your searches to those phrases that match the sought text exactly.  If using Regular Expressions, you can control case sensitivity with the regular expression switches. See <a href="#">more information on Regular Expressions</a> .
Regular expressions	Check this button to allow searches with <a href="#">Regular Expressions</a> .
Search only	If you check this button, no replacements will occur even if the sought text is found. Useful for simply locating files with the required text.
Repeat this action	Specify here how many times the selected action will iterate. Allows to replace the repetitive blocks of text. The default repetition count is 1.

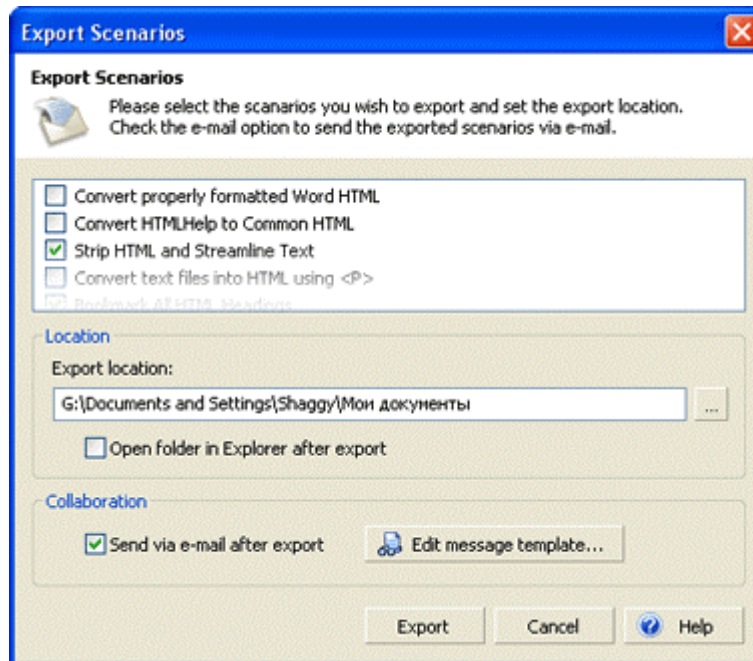
### Common Controls

Control	Description
Save and Activate	Saves changes and sets the current scenario active in the main application window.
Save	Saves changes and closes the dialog.
Cancel	Closes the dialog discarding changes.
Help	Shows this help.

# Scenario Export Dialog

The Scenario Export Dialog offers the user interface for exporting the HandyFile Find and Replace scenarios, which enables you with the teamwork options.

Each of the scenario is exported in an individual file in the user-supplied location.



## Exporting Scenarios

Follow the steps below to export a scenario (or a set of scenarios).

1. In the **Scenario Editor Dialog**, click the **Share** button and select the **Export Scenarios** menu item.
2. In the **Scenario Export Dialog**, select the scenarios you want to export by checking the corresponding scenario boxes.
3. Specify the folder in which you want to store the exported scenarios. If you want to view the folder in **Windows Explorer** after the export, check the **Open folder...** button.
4. If you want to send the exported scenarios to your colleagues right after the export is done, check the **Send...** option. This will open the standard message creation window of your default e-mail client, with the exported files attached.
5. Click the **Export** button.

## Dialog Fields

The tables below describe the dialog fields.

### Location

Field	Description
-------	-------------

<b>Field</b>	<b>Description</b>
Export location	A directory that will contain the exported scenarios.
Open folder in Explorer after export	Check this option if you want to open the folder in Windows Explorer after the export.

### **Collaboration**

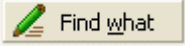

---

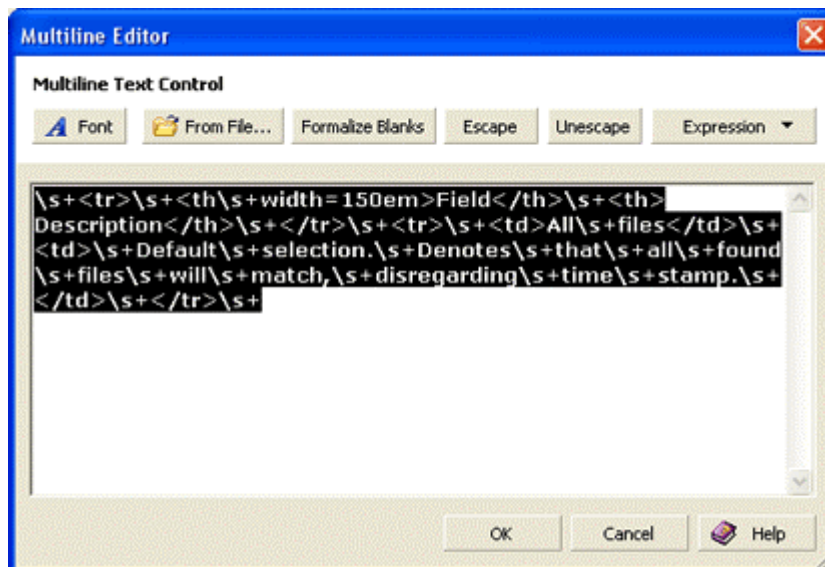
<b>Field</b>	<b>Description</b>
Send via e-mail after export	Check this option if you want to send the exported scenarios to your colleagues right after the export is done.
<b>Edit message template...</b> button	<p>This option allows you to save time while preparing your message to send. Click this button to open an editor window, where you can provide your custom e-mail text that is preserved between sessions and will be used for creating new messages later.</p> <p>The message can contain macro %%files%% that is replaced with the actual file names of the exported scenarios.</p>

# Multiline Editor Dialog

When entering text in the [Files and Text Tab](#), you can specify multiple lines of text by inserting control escapes `\r` and `\n`. This is rather inconvenient if there are a large number of text lines. Moreover, you cannot edit the multiline text as you usually do in text editors. This is why a **Multiline Editor** was created.

Multiline Editor allows you to supply multiple lines of text to search for or replace with. If the text in the Files and Text Tab contains line-breaking escapes (`\r` and `\n`), they will be converted to normal line breaks when the Editor is opened. When it closes, line breaks are converted back to the line-breaking escapes.

You can open this dialog box by clicking buttons  or .



The table below describes the Editor fields.

Field	Description
Font	Click to set the editor font most suitable for you.
Load From File	Click to load the text from any <i>plain-text</i> file in the editor.
Formalize Blanks	<p>This button, if clicked, converts all spaces, tabs and line breaks of the selected text to the regular expression operator <code>\s+</code> that would match the entire blank area. For example, the following text</p> <pre>&lt;tr&gt;   &lt;th width=150em&gt;Field&lt;/th&gt;   &lt;th&gt;Description&lt;/th&gt; &lt;/tr&gt; &lt;tr&gt;   &lt;td&gt;All files&lt;/td&gt;   &lt;td&gt;     Default selection. Denotes that all found files will match,     disregarding time stamp.   &lt;/td&gt;</pre>

Field	Description
	<p data-bbox="395 199 501 226">&lt;/tr&gt;</p> <p data-bbox="395 241 659 268">will be converted into</p> <pre data-bbox="395 286 1458 398">\s+&lt;tr&gt;\s+&lt;th\s+width=150em&gt;Field&lt;/th&gt;\s+&lt;th&gt;Description &lt;/th&gt;\s+&lt;/tr&gt;\s+&lt;tr&gt;\s+&lt;td&gt;All\s+files&lt;/td&gt;\s+ &lt;td&gt;\s+Default\s+selection.\s+Denotes\s+that\s+all\s+found\s+files\s+will \s+match,\s+disregarding\s+time\s+stamp.\s+&lt;/td&gt;\s+&lt;/tr&gt;\s+</pre> <p data-bbox="395 416 1458 479">This ensures that all text blocks similar to this one but differing in formatting only will be found.</p> <p data-bbox="395 488 1385 551">As an alternative to using <code>\s+</code> with the Regular Expressions, you can use the Ignore options when searching without the Regular Expressions.</p>
Escape	<p data-bbox="395 566 1369 629">Click to insert the escape character (\) before any non-alphanumeric symbol in the <i>selected</i> text.</p> <p data-bbox="395 667 1294 694">This button is only available if the <b>Regular Expressions</b> option is checked.</p>
Unescape	<p data-bbox="395 714 1437 777">Click to remove the the escape character (\) before any non-alphanumeric symbol in the <i>selected</i> text, if such escapes exist.</p> <p data-bbox="395 815 1294 842">This button is only available if the <b>Regular Expressions</b> option is checked.</p>
Expression	<p data-bbox="395 862 1410 889">Click to display a menu with control escapes that you can select and insert in the text.</p>

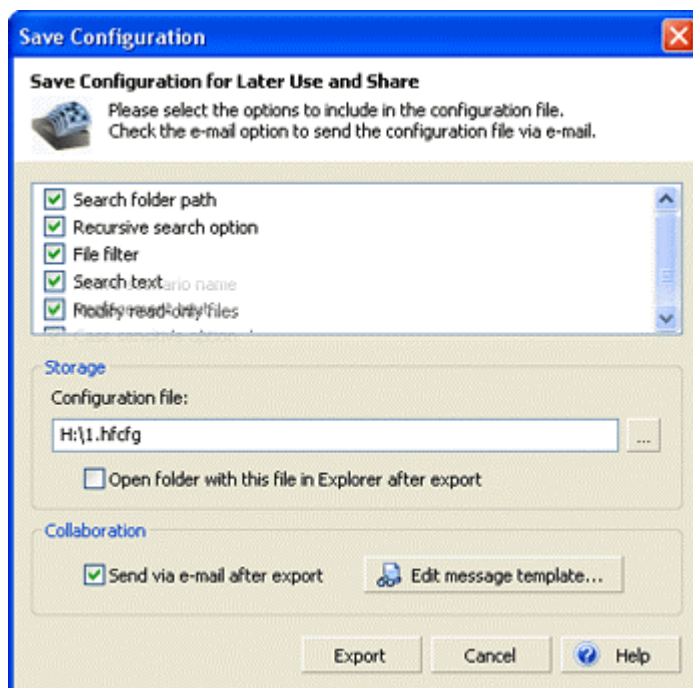
# Configuration Export Dialog

A *configuration* is a collection of preferences representing the current application state. Configuration can include the search path, search and replacement text, etc.

The **Configuration Export Dialog** enables you to save the HandyFile Find and Replace configurations for later use. You can select options to include in the configuration file. This window, together with **Scenario Export Dialog**, enables you with the teamwork options.

When loading a configuration, if any of settings are not found in the configuration file, their current state remain unchanged.

You can use the exported configuration file as the **command line option**.



## Exporting the Configuration

Follow the steps below to export the current configuration.

1. Click the drop-down arrow to the right of the [Options](#) button and select the **Save Configuration** menu item.
2. In the **Save Configuration** dialog, select the options you want to store in the configuration file by checking the corresponding boxes.
3. Specify the file in which you want to store the configuration. If you want to view the file folder in **Windows Explorer** after the export, check the **Open folder...** button.
4. If you want to send the exported configuration to your colleagues right after the export is done, check the **Send...** option. This will open the standard message creation window of your default e-mail client, with the configuration file attached.
5. Click the **Export** button.

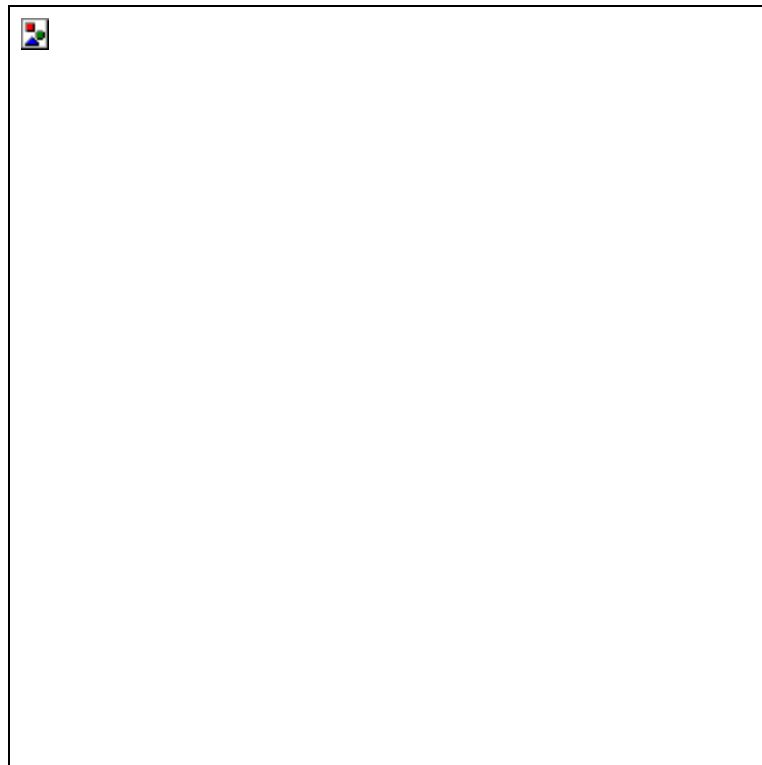
The table below describes the dialog fields.

<b>Field</b>	<b>Description</b>
Configuration file	A file that will contain the exported configuration.
Open folder with this file in Explorer after export	Check this option if you want to open the folder in Windows Explorer after the export.
Send via e-mail after export	Check this option if you want to send the exported configuration to your colleagues right after the export is done.
<b>Edit message template...</b> button	<p>This option allows you to save time while preparing your message to send. Click this button to open an editor window, where you can provide your custom e-mail text that is preserved between sessions and will be used for creating new messages later.</p> <p>The message can contain macro <code>%%file%%</code> that is replaced with the actual configuration file name.</p>

# Replacement Confirmation Dialog

This dialog allows you to review each replacement in each processed file and accept or decline it. You have an option to accept or reject the current replacement; skip the currently processed file entirely; accept changes made to the current file and go to the next file; work silently during the present replacement session or abort the processing.

You can enable or disable the replacement confirmations on the [Options Dialog - Processing](#) page.



The table below describes the dialog fields.

Field	Description
Existing text	The text originally stored in file. The text to be replaced is highlighted with different background colour.
New text	The text that will replace the old one according to the settings on the <a href="#">Files and Text Tab</a> .
<b>Buttons</b>	
Accept All	Click this button to accept the current and all further replacements. You will not be prompted of these changes during this run; the operation will be performed silently.
Replace	Click the <b>Replace</b> button to accept and confirm the current replacement.
Skip	Click this button to cancel the proposed replacement and continue searching.
Skip File	Click this button to reject changes made to the current file (if any) and go to the next file.
Next File	Click this button to accept changes made to the current file (if any) and go to the next file.
Abort	This button stops the processing.


<b>Field</b>	<b>Description</b>

# Regular Expression Laboratory

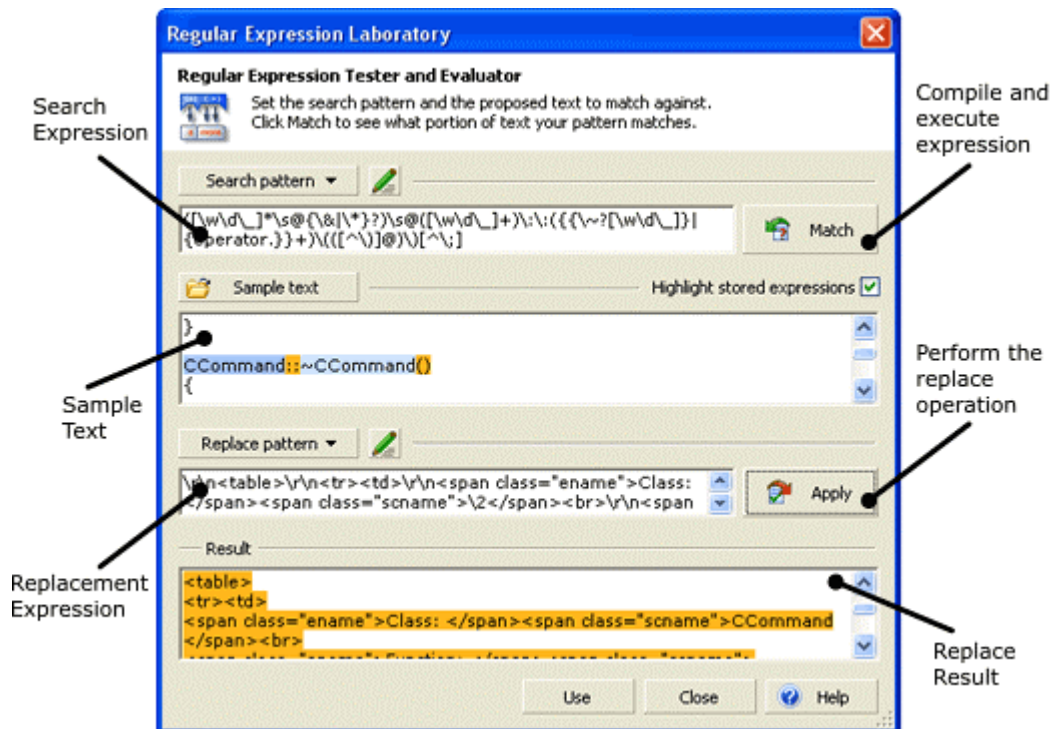
If you are not familiar with regular expression and want to learn how to use it, or want to test your own expressions, the **Regular Expression Laboratory** will help you much. Its main intent is to visualise both the internals of the expression compilation process and how the prepared expression matches the text portions.

At the heart of the regular expression engine is a concept of a *finite-state automata*. You can treat this automata as a tiny virtual computer working by a program. The program that the automata runs is the precompiled binary form of an expression. In other words, the textual expression that you provide is first compiled into a binary form and then executed against a text.

This is why you get regular expression errors sometimes. Some applications may even crash if the regular expression faults are not handled properly. Using the **Regular Expression Laboratory**, you will easily get a robust working expression.

You can open the Laboratory by clicking button  in the Files and Text main window tab or Scenario Editor.

The following picture illustrates the **Regular Expression Laboratory** front-end.






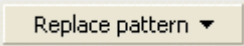

## How to Use the Laboratory

Using it is fairly simple. What is more, the Laboratory provides you with visual cues when you are mistaken with your expression. The below given sequence is usually the best practice.

1. Type your text to operate on in the **Sample text** field. As an alternative, you can load the text from file by clicking the *button* **Sample text**.

2. Type your expression in the **Search pattern** field. For example, if you want to match all HTML tags, you could type `\<[^\>]#\>`
3. Click the **Match** button. This will compile the expression and find the first occurrence of the matching text in the **Sample text** field. If the compilation fails, the erratic search expression symbol will blink with red.
4. Type the replacement text in the **Replace pattern** box. For example, if you want to convert the HTML tag to lowercase: `\L\0`
5. Click the **Apply** button. The result text box will contain the result of the replace operation.

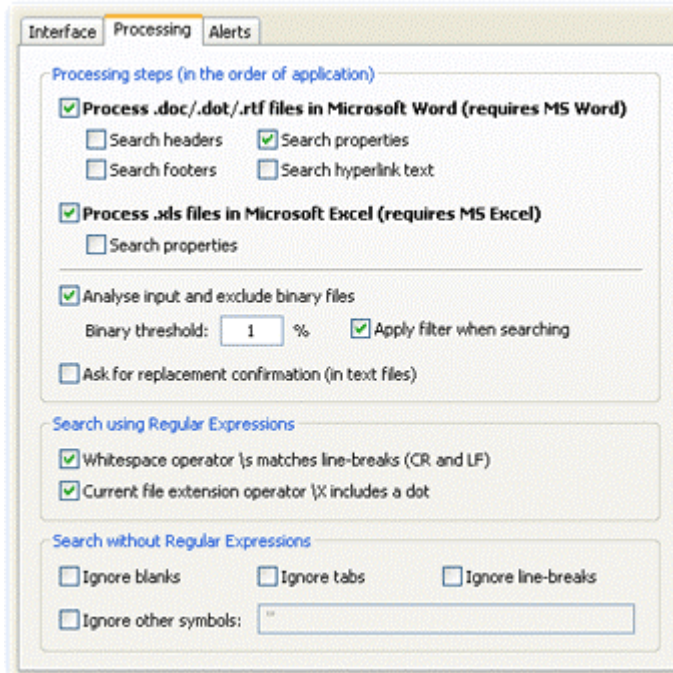
The table below describes the dialog fields.

Field	Description
<b>Search pattern</b> button 	Shows a menu with regular expressions that you can select and insert.
<b>Multiline editor</b> button 	Click this button to open the <b>Multiline Editor</b> dialog. This dialog allows entering text with multiple lines in normal mode, converting line breaks to <code>\r</code> and <code>\n</code> as appropriate.
<b>Match</b> button	Compiles the provided expression and tries to match it against the sample text. Click this button again to find the next match in the sample text. You can press and hold this button to iterate through the text.
<b>Sample text</b> box	Contains the sample text that the expression will match against.
<b>Sample text</b> button 	Allows to load the sample text from file.
Highlight stored expressions	If checked, the matching stored expressions, i.e. those enclosed in parentheses ( ) will have different background colour in the sample text box.
<b>Replace pattern</b> text box	Contains the replacement expression.
<b>Replace pattern</b> button 	Shows a menu with regular expressions that you can select and insert.
<b>Multiline editor</b> button 	Same purpose as for the <b>Multiline editor</b> button for the <b>Search pattern</b> .
<b>Apply</b> button	Given the replacement expression, performs the replace operation on the sample text.

# Options

## Options Dialog - Processing

The Processing option page contains settings that define how HandyFile Find And Replace treats files and text.



### Dialog Fields

The tables below describe the dialog fields.

#### Processing steps

Field	Description
Process .doc and .rtf files in Microsoft Word (requires MS Word)	<p>If this option is checked, all <b>.doc</b> and <b>.rtf</b> files will be processed using the installed <b>Microsoft Word</b> engine. This allows to modify these files correctly, ensuring their safety and integrity.</p> <p>The following restrictions apply when processing <b>.doc</b> and <b>.rtf</b> files in Microsoft Word.</p> <ul style="list-style-type: none"> <li>• Regular Expressions cannot be used.</li> <li>• Microsoft Word does not report the true match count; the only information that can be obtained is that the sought text is found (or not found) in a file.</li> <li>• <b>Collector</b> does neither apply to Word documents, nor it collects any text in such files.</li> <li>• As processing a Word file requires running a Microsoft Word instance, the processing speed is rather slow.</li> </ul>

	<p>If this option is unchecked, any <b>.doc</b> file will be reported as a binary file if the binary filter is switched on. <b>RTF</b> files may report that the text is found, but we <i>do not recommend</i> using simple text replace on <b>RTF</b> files to avoid data corruption.</p>
<p>Process .xls files in Microsoft Excel (requires MS Excel)</p>	<p>If this option is checked, all <b>.xls</b> files will be processed using the installed <b>Microsoft Excel</b> engine. Checking this option is the only way to find and replace text in your <b>XLS</b> files. This allows to modify these files correctly, ensuring their safety and integrity.</p> <p>The following restrictions apply when processing <b>.xls</b> files in Microsoft Word.</p> <ul style="list-style-type: none"> <li>• Regular Expressions cannot be used.</li> <li>• Microsoft Excel does not report the true match count; the only information that can be obtained is that the sought text is found (or not found) in a file.</li> <li>• <b>Collector</b> does neither apply to Excel documents, nor it collects any text in such files.</li> <li>• As processing an Excel file requires running a Microsoft Excel instance, the processing speed is rather slow.</li> </ul> <p>If this option is unchecked, any <b>.xls</b> file will be reported as a binary file if the binary filter is switched on.</p>
<p>Analyse input and exclude binary files</p>	<p>If this options is checked, each file that is about to be processed is checked for presence of non-printable bytes. The main intention of this option is to ensure the safety of binary files if the file filter mask is set to * (match all file names and extensions) .</p> <p>Normal text files do not contain non-printable characters. The only allowed non-printable symbols are blanks (\x20), tabs (\x09), carriage returns (\x0D) and line feeds (\x0A). If the file contains any other non-printable characters (with the code less than \x20), it is considered <i>suspect</i>.</p> <p>You can adjust the suspect value by using the <i>Binary threshold</i> parameter.</p> <p>Enabling this option slows down the processing speed.</p>
<p>Binary threshold</p>	<p>This parameter defines the maximum quota of non-printable characters allowed in the suspicious file. If the file exceeds this value, it is rejected and is not processed.</p> <p>For example, you can set the value to 1% to allow the incorrectly formatted text files to be processed. As researches show, this is the best value.</p> <p>A value of 0% rejects all suspicious files.</p> <p>A value of 100% is similar to unchecking the option <i>Analyse input and exclude binary files</i>.</p>
<p>Apply filter when searching</p>	<p>If checked, files will be checked for binary content when searching and replacing. This allows to find binary or invalid text files.</p> <p>If unchecked, files will be checked for binary content when replacing only.</p>

Ask for replacement confirmation (in text files)	<p>If this option is checked, the Replacement Confirmation Dialog will appear asking you to accept or decline the replacement of every occurrence of the sought text in each processed file.</p> <p>This option is only effective for text files. This does not apply to Microsoft Word or Excel documents.</p>
--	---

### Search using Regular Expressions

Field	Description
Whitespace operator \s matches line-breaks (CR and LF)	<p>Normally, the regular expression operator \s matches line-breaks in addition to whitespace and tabs. This allows to find the irregular text blocks (that differ in formatting) easily.</p> <p>You may choose to turn this option off for some reason.</p>
Current file extension operator \X includes a dot	<p>The operator \X that is used in the <i>Replace</i> expressions inserts an extension of the file under process - for example, .html. Uncheck this option to not include the leading dot in the extension: html.</p>

### Search without Regular Expressions

Field	Description
<p><b>These options affect the search exactness. <i>Spanning</i> some or all of the formatting characters allows to find blocks of text similar to the target string but differentiating in formatting. See the <a href="#">Remarks</a> below.</b></p>	
Ignore blanks (\x20)	Skip all blanks (character code \x20) in both the processed text and the search string when searching. That is, number of blanks and their position in the text do not affect the match/no match result.
Ignore tabs (\x09)	Skip all blanks (character code \x09) in both the processed text and the search string when searching.
Ignore line-breaks (CR and LF)	Skip all carriage returns (\x0D) and line feeds (\x0A) in both the processed text and the search string when searching.
Ignore other symbols	Allows to specify and skip the user-defined symbols in both the processed text and the search string when searching.

## Remarks

### Spanning options

These options are extremely useful if you need to find some heterogeneous blocks of text and do not want to use Regular Expressions. For example, if you use a WYSIWYG HTML editor, you notice that it formats the code in a higgledy-piggledy fashion. The formatting is generally performed using blanks, tabs and line-breaks. Say, you need to find the following code:

```
<a href="http://www.mysite.com">

</a>
```

The WYSIWYG formatter might write it like this:

```
<a href="http://www.mysite.com"></a>
```

or like this:

```
<a href="http://www.mysite.com"></a>
```

or even like this:

```
<a href="http://www.mysite.com"></a>
```

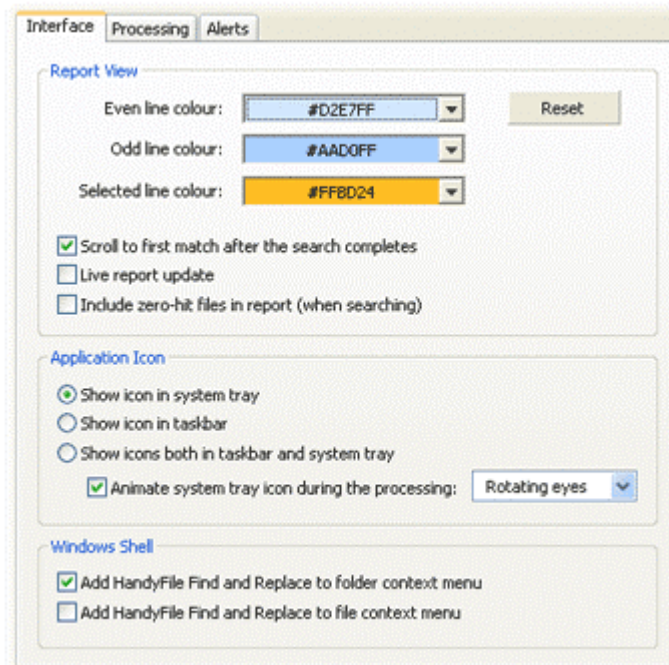
You can easily handle all cases using Regular Expressions, but if you do not want bother yourself or not familiar with them, you can use the spanning options. Simply check all the three boxes (*to match the text in this example*) and the HFFR will find the string.

### Note

Use of the spanning options results in search speed degradation.

## Options Dialog - Interface

The Interface option page contains settings that define the display mode and style of HandyFile Find And Replace.



### Dialog Fields



The tables below describe the dialog fields.

#### Report View

This group of controls defines the appearance and behaviour of the report areas: the match tree and the file viewer.

Field	Description
Even line colour	Defines the colour of the first (zero-based) and the consecutive alternate matches in the file match report and the file viewer.
Odd line colour	Defines the colour of the second (zero-based) and the consecutive alternate matches in the file match report and the file viewer.
Selected line colour	Defines the colour of the selected match in the file match report and the file viewer.
Reset	Click to reset colours to their default values.
Scroll to first match after the search completes	If checked, the first file in the tree will be selected and its contents loaded in the viewer.
Live report update	If this option is unchecked, the search results are added to the <a href="#">file match tree</a> and no scrolling to the last added item occurs. If this option is checked, the search results are added to the file match tree and the tree scrolls every time a new item is added. Enabling this option slows down the processing speed.
Include zero-hit files in report (when searching)	The HFFR versions prior to 3.1 did not include files not containing the target text in the report when performing the search (unlike the Replace). This option allows to include files not containing the sought text in the report.

### Application icon

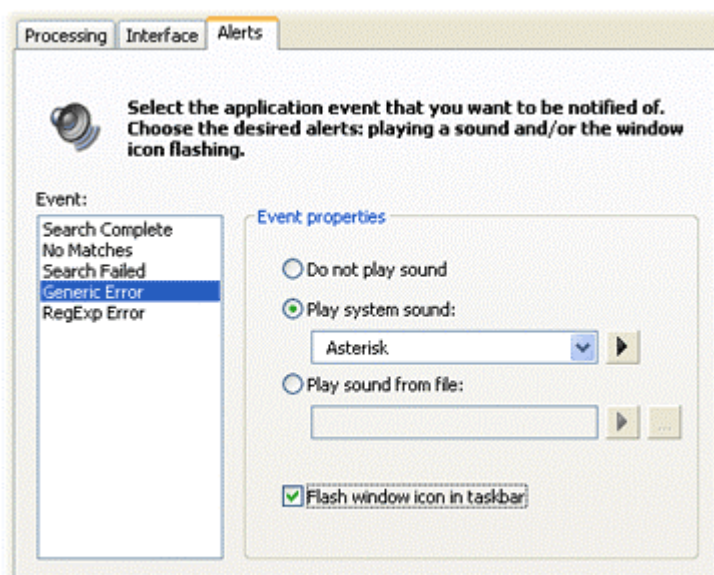
Field	Description
Show icon in system tray	This option tells to place the application icon in the system tray (area that contains the system clocks). The application icon will not be listed on the Windows Taskbar when the HandyFile Find And Replace window is minimized, thus allowing to save space.
Show icon in taskbar	This option tells to list the application icon on the Windows Taskbar (the default behavior).
Show icons both in taskbar and system tray	This option combines the previous two modes: tells to place the application icon in the system tray and on the Windows Taskbar.
Animate system tray icon during the processing	If you select to display an icon in the tray, you can check this option to enable the icon animation during the process time. The animation had been optimized to be not resource-consuming, so you can use it as an indicator of any long-lasting operation.  You can choose between the two animation types: <ul style="list-style-type: none"> <li>rotating: </li> <li>and blinking: </li> </ul>

### Windows Shell

Field	Description
Add HandyFile Find and Replace to <b>folder</b> context menu	If checked, folder context menus in Windows Explorer contain command <b>HandyFile Find And Replace in this folder</b> . This provides quicker access to the HandyFile Find And Replace functions from within Windows shell.
Add HandyFile Find and Replace to <b>file</b> context menu	If checked, file context menus in Windows Explorer contain command <b>HandyFile Find And Replace on this file</b> .

## Options Dialog - Alerts

The **Alerts** option page contains settings that allow you to configure how HandyFile Find And Replace will notify you of the process completion and/or errors.



## Dialog Fields

The tables below describe the dialog fields.

### Events

This box contains a list of available application events. To assign any alert type to an event, select it in the list and configure it using the Event properties controls.

Field	Description
Search Complete	This event fires when the search (and/or replace) operation completes without any error and one or more matches was found.
No Matches	This event fires when the search (and/or replace) operation completes without any error and no matches was found.
Search Failed	This event fires if the operation fails due to any Microsoft Office error when processing the MS Office files, or a file cannot be accessed.
Generic Error	This event fires when the operation fails due to any system or preprocessing error.
RegExp Error	This event fires if the provided regular expression contains errors.

### Event properties

Field	Description
Do not play sound	Does not emit any sound on event.
Play system sound	Plays a system-defined sound (configured in the Control Panel) on event.
Play sound from file	Plays a user-supplied sound (WAV file) on event.
Flash window icon in taskbar	Tells to flash the application icon in the Windows Taskbar if an event fires.



# | Regular Expressions

# Regular Expressions

This topic contains information on regular expression control switches and operators, as well as important notes on using regular expressions.

## IMPORTANT!

Non-alphanumeric symbols in search expressions **must** be escaped with a backslash. For example, if you need to find a symbol # using regular expressions, you must type \#. Otherwise, symbols are treated as regular expression operators.

Please note that literal switches are case-sensitive! For example, \s matches whitespace, while \S matches non-whitespace.

Repeat qualifiers (@, #, \*, +) can be applied (for example: .+) to match text segments as large as 2 Mb or less.

*Greedy* means matching this condition as much as possible, disregarding further coinciding or alike conditions. See the more detailed description of the greedy and non-greedy modes in topic [Using Regular Expressions](#).

## Expressions for searching

Expression	Description
^	Match the beginning of file.  <b>HandyFile Find And Replace</b> searches the file as if it is a single line. This allows to search for the line-breaking characters and text blocks, manipulate and replace them without any hassle.
\$	Match the end of file.
.	Match any character.
[ ]	Match characters in set. Specify set (e.g. [aghet2]) or range (e.g. [A-Z]).
[ ^ ]	Match characters not in set (e.g. [^A-Z]).
?	Match previous pattern 0 or 1 times ( <a href="#">greedy</a> ).
	Match previous or next pattern.
@	Match previous pattern 0 or more times ( <a href="#">non-greedy</a> ).
#	Match previous pattern 1 or more times (non-greedy).
*	Match previous pattern 0 or more times (greedy).
+	Match previous pattern 1 or more times (greedy).
{ }	Group characters to form one pattern.
( )	Group and remember for further referencing and use.
\	Quote next character (only of not a-z; e.g. ">" designates symbol ">").
<	Match beginning of a word.
>	Match end of a word.

Expression	Description
\t	Match 0x09 (tab).
\e	Match escape (^E).
\s	Match whitespace (tab, space, CR, LF). <a href="#">More information...</a>
\S	Match non-whitespace.
\w	Match word character.
\W	Match non-word character.
\d	Match digit character.
\D	Match non-digit character.
\U	Match uppercase.
\L	Match lowercase.
\C	Match case sensitively from here on.
\c	Match ignoring the case from here on.
\#	Match contents of previously remembered group; #=[1-9].
\xNN	Character with hexadecimal value of NN, where N=[0-9A-F].
\dNNN	Character with decimal value of NNN, where N=[0-9].
\oNN	Character with octal value of NNN, where N=[0-7].

## Expressions for replacing

Expression	Description.
\0	Place whole found text.
\#	Place contents of the stored group numbered by the "#" [groups are defined by braces (...) in the search expression]; #=[1-9].
\t	Place tab character.
\u	Make next character uppercase.
\l	Make next character lowercase.
\U	Force further output uppercase.
\L	Force further output lowercase.
\E	Turn off case transformation (after the use of \U or \L)
\f	Insert current file name and extension (file.ext).
\F	Insert current file name only, without extension (file).
\X	Insert current file extension (.ext). <a href="#">More information...</a>
\R	Insert a random 8-digit number. For example: 267124D2.
\S	<p>Insert a random 8-digit number that remains constant within a single replace expression.</p> <p>For example, the expression</p> <pre>id="\S" onClick="openPageId('\S')"</pre> <p>might insert</p> <pre>id="267124D2" onClick="openPageId('267124D2')"</pre>

Expression	Description.
<code>\Zhe</code>	Converts the next symbol to numeric HTML entity.
<code>\ZhE</code>	Turns on the <b>ASCII - to - HTML Entity</b> conversion. Makes all further output converted to numeric HTML entities. For example, the operator <code>\ZhEMyString</code> will insert <code>&amp;#77;&amp;#121;&amp;#83;&amp;#116;&amp;#114;&amp;#105;&amp;#110;&amp;#103;</code>
<code>\Zhx</code>	Turns off the <b>ASCII - to - HTML Entity</b> conversion previously initialised by <code>\ZhE</code> .
<code>\P[h]:"file_path_name"</code>	Executes the path name resolver.  Very often your files (for example, HTML pages) are stored in different folders and use reference to one file in some other folder (for example, CSS file). This operator allows to insert a path name of the referenced file relative to a processed file, given a fully-qualified path and name of the referenced file.  <code>\P</code> operator uses backslash as the path part separator. <code>\Ph</code> operator ( <code>h</code> stands for <i>HTML mode</i> ) uses forward slash.  If the given path name cannot be converted to a relative path, the provided path name is inserted.  See <b>Remarks</b> for example.
<code>\Pc[h]:"directory_path"</code>	Similar to operator <code>\P</code> . Executes the path name resolver on the currently processed file. Inserts path to the current file relative to the provided directory.  For example, if the current file is <code>c:\dir1\dir2\dir3\file.ext</code> and the <i>directory_path</i> is <code>c:\dir1\dir2</code> , the operator <code>\Pc:"c:\dir1\dir2"</code> will insert <code>.\Dir3\file.ext</code>  <code>\P</code> operator uses backslash as the path part separator. <code>\Ph</code> operator ( <code>h</code> stands for <i>HTML mode</i> ) uses forward slash.  If conversion fails, the current file path name is inserted.
<code>\xNN</code>	Insert character with hexadecimal value of NN, where N=[0-9A-F].
<code>\dNNN</code>	Insert character with decimal value of NNN, where N=[0-9].
<code>\oNN</code>	Insert character with octal value of NNN, where N=[0-7].

## Remarks

### Example of using the path name resolve operator

For example, you are processing files and folders in `C:\MyWeb\HTML\Catalog`. You want to insert a reference to a CSS file `style.css` stored in the folder `C:\MyWeb\HTML`. You could write the following replace expression:

```
<LINK REL=stylesheet  
  HREF="\Ph:"C:\MyWeb\HTML\style.css\" TYPE="text/css">
```

This will insert the following text in HTML files in C:\MyWeb\HTML\Catalog:

```
<LINK REL=stylesheet HREF=" ../style.css" TYPE="text/css">
```

and the following text in HTML files in

C:\MyWeb\HTML\Catalog\dir1\dir2:

```
<LINK REL=stylesheet HREF=" ../../style.css"  
  TYPE="text/css">
```

**Note!**

Root parts of path names of both a referenced and processed file must be the same. It means that they must have a common drive letter.

# Using Regular Expressions

## What are Regular Expressions?

Regular expressions are a way to search for substrings ("matches") in strings. This is done by searching with "patterns" through the string.

### Example

You probably know the '\*' and '?' characters used in the `dir` command on the DOS command line. The '\*' character means "zero or more arbitrary characters" and the '?' means "one arbitrary character".

When using a pattern like `text?.*`, it will find files like

- `textf.txt`
- `text1.asp`
- `text9.html`

But it will not find files like

- `text.txt`
- `text.asp`
- `text.html`

This is exactly the way regular expressions work. While the '\*' and '?' are a very limited subset of patterns, regular expressions supply a much broader spectrum of describing patterns.

## Why would you use Regular Expressions

Example usages could be:

- remove all occurrences of a specific tag from an html file;
- check whether an e-mail address is well-formed;
- replace value of some tag attribute with a different one;
- many more other tasks.

## Matching Operators

Any operator or set of operators represent a *pattern*.

### Any Character

You will probably need to match some patterns containing symbols that may differ and vary in some way. For example, you want to find words starting with `tom` and having four characters in length. The operator that matches any character is dot (`.`). Thus, the following pattern would match all these words: `tom.`

This example will also find text like `tom.`, `tom>`, `tom!`, etc.

## Sets of Characters

To prevent the pattern `tom` from matching not meaningful phrases, we should narrow the search criteria to only alphabetic symbols. This can be done using character sets. A set is specified with square brackets. Sets may include individual symbols and ranges. For example, the following set will match any one symbol of a, t, z and 8: `[atz8]`. And this set will match all lowercase letters: `[a-z]`.

Thus, to limit the previous example to meaningful phrases, we could write a pattern: `tom[a-z]`.

## Negative Sets of Characters

Sometimes you need to find all symbols except some. Writing a large set including all possible symbols is ineffective. So we better use a negation operator in a set: `^`. For example, the following set will match any one symbol except @: `[^\@]`. Please note that the symbol @ is escaped as it is not alphanumeric.

## Repetition Qualifiers

Regular expressions would be of no use unless they might match any text of any length. To achieve this, repetition qualifiers were introduced, which allows matching nearly any text.

### Match 0 or 1 times

In the previous example, a pattern `tom[a-z]` would successfully find any word of four symbols in length except `tom` itself. To force the pattern to match `tom`, we should instruct it to do so. The qualifier `?` tells to match the preceding pattern 0 or 1 times. The following pattern will match `tom` as well: `tom[a-z]?`

## Greedy and Non-greedy Matches

Before we proceed with the other repetition qualifiers, we should understand one important thing about repetition modes.

Imagine a text that contains some occurrences of a character. For example, `one, two, three, four`. This text has 3 entries of a comma. Now we want to instruct the regular expression engine to **"match all characters but stop before a comma"**.

A greedy mode will match *all* characters and stop before the **last** comma:

```
one, two, three, four.
```

A non-greedy mode will match *all* characters and stop before the **first** comma:

```
one, two, three, four.
```

## Match previous pattern N or more times

Let us extend the previous example by introducing a new condition: match all text starting from `tom` but ending with full-stop. So we need to:

1. match `tom`;
2. match any character;

3. repeat the preceding condition 0 or more times until the first occurrence of the next match (4) is found;
4. match a full-stop (a dot).

The following table shows the corresponding operators:

Part	Operator	Comment
Match tom	tom	A simple text
Match any character	.	A dot-operator
Repeat the preceding condition until the first occurrence of the next match is found	@	Repeat qualifier: <i>Match previous pattern 0 or more times (non-greedy).</i>
Match a full-stop (a dot)	\.	A dot. Escape is added to instruct to process the dot as a common symbol, not operator.

Thus, the pattern would look like:

```
tom.@\.
```

## Patterns and Alternatives

Say you need to find one of the words: *macrocoding* and *macrocode*. There are several ways to do that. For example, we can split each word into **macrocod+ing** and **macrocod+e**. Now, we will need a pattern that would:

- match macrocod;
- match either ing or e.

When we say "or", we say "|". When a regular expression says "or", it says "|". Armed with this knowledge, we write: `macrocoding|e`.

Looks rather meaningless, doesn't it? What would this expression do: match `macrocoding` or `e` or match `macrocodin` and `g` or `e`? That's why a *pattern operator* had been developed.

A *pattern operator* concatenates several stand-alone symbols or patterns to form one pattern. For example, a single symbol `e` is a pattern. The first symbol (`i`) in the "ing" is a stand-alone pattern. To form a single pattern from "ing", we should enclose it in braces:

```
{ing}
```

Now, `ing` is a single pattern.

This allows us to write the following pattern:

```
{macrocod{ing}|{e}}
```

This is a correct well-formed single pattern.

## Expressions

In terms of semantics, expression and pattern operators are the same. The difference is that the text that matches the expression is stored and can be referenced further, for example, when replacing.

For example, we could alter the previous example to make an expression out of the ending `{ing}|{e}` by enclosing it in the round braces:

```
{macrocod({ing}|{e})}
```

Now we can reference the ending with the operator `\1`. 1 stands for the number of the expression. We can write the replace pattern that would insert a plus sign between `macrocod` and the ending:

```
macrocod+\1
```

# Regular Expression Examples

Below are given some examples of using regular expressions in various situations.

## Find any HTML colour constant.

Find pattern	<code>\#[0-9A-Fa-f]+</code>
--------------	-----------------------------

## Replace all HTML colour constants with the new one (#AFEEEE).

Find pattern	<code>\#[0-9A-Fa-f]+</code>
Replace pattern	<code>#AFEEEE</code>

## Replace all occurrences of full stop before the lowercase letter with comma, truncating separating spaces to only one (useful for scanned documents).

Find pattern	<code>{\.\s#}(\L[a-z])</code>
Replace pattern	<code>\, \1</code>

## Strip HTML tags.

Find pattern	<code>{\&lt;}{\/?}{.#{\&gt;}}</code>
Replace pattern	<code>empty</code>

## Change the value of all "background-color:" CSS properties everywhere.

Find pattern	<code>(background\-color\:){\#[0-9A-Fa-f]+}</code>
Replace pattern	<code>\1#FFE4E1</code>

## Replace <P> tags with <DIV> tags, preserving tag attributes.

Find pattern	<code>\&lt;P\s(.#)\&gt;(.#)\&lt;/P\&gt;</code>
Replace pattern	<code>&lt;DIV \1&gt;\2&lt;/DIV&gt;</code>

## Colorize HTML tags for use inside HTML page: symbols with blue, tag with red, convert < and > to entities.

Find pattern	<code>\&lt;({\//}?) (.#)\&gt;</code>
Replace pattern	<code>&lt;span style="color:blue"&gt;&amp;lt;\1&lt;/span&gt;&lt;span style="color:red"&gt;\2&lt;/span&gt;&lt;span style="color:blue"&gt;&amp;gt;\3&lt;/span&gt;</code>

## Find host address.

The pattern would match the following:

- <http://www.host.dom/>
- <http://host.dom/>

- ftp://203.131.69.75/
- www.host.dom/

<b>Find pattern</b>	<code>(\{ht\} \{f\}tp\\:\\/\/)?\{[0-9]#\.[0-9]#\.[0-9]#\.[0-9]#\} \{[\^\.\/]#\}?\{[\^\.\/]#\.[\^\.\/]#\}\}\/</code>
---------------------	---

**Find the whole paragraph; store the paragraph contents.**

<b>Find pattern</b>	<code>(.#\r\n \{\r\} \{n\})</code>
---------------------	------------------------------------

# Common Tasks

The examples given in this section is an excerpt from the on-line documentation. Please refer to the CHM on-line guide to find more examples.

# HOWTO: Perform Simple File Search

The HandyFile Find And Replace allows to perform simple file searches. To find find file(s) of any type, do the following.

1. Specify the folder in which you want to find files. For example: `C:\MyWebFiles\Projects`
2. Set mask(s) of the files that you want to find. For example: `*.htm*;*.asp`
3. *Empty* the **Find What** field. This is important to just find files, not text.
4. Click the **Search** button.

# HOWTO: Search For Text In Files

To find file(s) of the required type containing the specific text, do the following.

1. Specify the folder in which you want to find files. For example: `C:\MyWebFiles\Projects`
2. Set mask(s) of the files that you want to find. For example: `*.htm*;*.asp`
3. Type the text that you want to find in the **Find What** field.
4. Click the **Search** button. This is important to click this button rather than **Replace**, to not replace occurrences of the found text.

# HOWTO: Search For Text In Files In Multiple Folders

To find file(s) of the required type containing the specific text in more than one folder, do the following.

1. Specify the folders in which you want to find files. Separate multiple folders with semicolon. For example: `C:\MyWebFiles\Projects;C:\MyProjects\CPP;C:\MyBooks\Programming`

Alternatively, you can use the [Folder Manager](#) to specify multiple folders.

2. Set mask(s) of the files that you want to find. For example: `*.htm*;*.asp;*.cpp`
3. Type the text that you want to find in the **Find What** field.
4. Click the **Search** button. This is important to click this button rather than **Replace**, to not replace occurrences of the found text.

# HOWTO: Search For Text Using Regular Expressions

To find file(s) of the required type containing the text that matches a pattern, do the following.

1. Specify the folder in which you want to find files. For example: `C:\MyWebFiles\Projects`
2. Set mask(s) of the files that you want to find. For example: `*.htm*;*.asp;*.cpp`
3. Provide the pattern in the **Find What** field for the matching text that you want to find.

For example, you want to find HTML files containing the `<OBJECT>` tags, not simply word OBJECT. The corresponding pattern would include:

- the opening tag;
- tag name (OBJECT);
- as the tag may contain attributes, some wildcard operator and repetition operator that would match the remaining tag inner text;
- the closing tag.

The pattern in this case would look as follows:

```
\<OBJECT.#\>
```

4. Click the **Search** button. This is important to click this button rather than **Replace**, to not replace occurrences of the found text.

# HOWTO: Find and Replace Text in Word and Excel Files

To find the Microsoft Office file(s) of the required type and replace text in them, do the following.

1. Specify the folder in which you want to find files. For example:  
C:\MyOfficeFiles
2. Set mask(s) of the files that you want to find.
3. To search for the native Microsoft Word files (.doc), set the filter to \*.doc.
4. To search for .rtf files to be processed with Microsoft Word, set the filter to \*.rtf.
5. To search for both .doc and .rtf files to be processed with Microsoft Word, set the filter to \*.rtf;\*.doc.
6. To search for Microsoft Excel files (.xls), set the filter to \*.xls.
7. If you need to process both Microsoft Excel and Microsoft Word files, set the filter to \*.rtf;\*.doc;\*.xls.
8. Type in the text that you want to find in the Find What field.
9. Enter the text that you want to replace the found text in the Replace With field.
10. The Microsoft Office searches allows using the Case-sensitive search. If you need to find text exactly as you have typed, check this option.

## IMPORTANT!

Make sure the options that allow utilizing Microsoft Office facilities are on. To enable the processing, click the Options button, go to the Processing tab and check options Process .doc and .rtf files in Microsoft Word and/or Process .xls files in Microsoft Excel. If these options are inactive, the safety filter will reject such files (except .rtf because in fact they are text files) and will not allow processing them.

11. Make sure the option Search only is unchecked.
12. Click the Replace button.
13. To simply search for files containing the text without modifying them, click the Search button.

# HOWTO: Find And Replace Text In Files

To find file(s) of the required type and replace some text in them, do the following.

1. Specify the folder in which you want to find files. For example: `C:\MyWebFiles\Projects`
2. Set mask(s) of the files that you want to find. For example: `*.htm*;*.asp;*.cpp`
3. Type in the text that you want to find in the **Find What** field.
4. Enter the text that you want to replace the found text in the **Replace With** field.
5. Make sure the option **Search only** is *unchecked*.
6. Click the **Start** button.

# HOWTO: Search And Replace Text Using Regular Expressions

To find file(s) containing the text that matches a pattern and replace the text in them, do the following.

1. Specify the folder in which you want to find files. For example: `C:\MyWebFiles\Projects`
2. Set mask(s) of the files that you want to find. For example: `*.htm*;*.asp;*.cpp`
3. Provide the pattern in the **Find What** field for the matching text that you want to find.

For example, you want to replace the `CENTER` value of the `ALIGN` attributes with `LEFT` in all `<TABLE>` tags in HTML files. The corresponding pattern would include:

- the opening tag - `\<`;
- tag name (`TABLE`);
- as the tag may contain other attributes, some wildcard operator and repetition operator enclosed in the *expression* operator for further reference - `(. #)`;
- `ALIGN` attribute and a value - `ALIGN\=CENTER`;
- as the tag may contain other attributes after `ALIGN`, some wildcard operator and repetition operator enclosed in the *expression* operator for further reference - `(. #)`;
- the closing tag - `\>`.

The pattern in this case would look as follows:

```
\<TABLE(. #)ALIGN\=CENTER(. #)\>
```

4. Provide the replacement expression, where `\1` and `\2` stands for the previously remembered parts of text `(. #)`:

```
\<TABLE\1ALIGN=LEFT\2\>
```

5. Make sure the option **Search only** is unchecked.
6. Click the **Replace** button.

# HOWTO: Use Scenarios To Automate Work

We introduced **scenarios** in the **HandyFile Find And Replace** so that you can get rid of the monotonous operations that involve searching and replacing many different portions of text in multiple files. Let us consider an example of cleaning HTML files saved from within Microsoft Word.

We will want to:

- remove comments containing intrinsic information specific to the Word;
- delete `SPAN` tags containing `LANG` attributes and `msO` style definitions;
- remove `<o:p>` tags;
- clean the HTML tag;
- remove the Office `[if]` conditions.

There are much more Word dirt in the HTML but we shall limit our example to these searches.

We start by opening the [Scenario](#) editor by clicking the **Advanced Search** button and selecting **Manage Scenarios** from the menu.

In the **Scenario** dialog, create a new scenario by clicking the **New** button. Type the name of the new scenario, for example, *Clean up Word HTML*.

After the new scenario is created, start adding actions to it.

1. Click the button **Add Action** and give it a name for the first action, for example: *Remove comments*.
2. We shall need regular expressions to cover all comments in a file. So we check the **Regular expressions** option and type the following text in the **Find what** field:

```
\<\!\-\-\.#\-\-\>
```

Here:

- `\<\!\-\-` is the comment opening tag;
- `.\#` (any character + match 1 or more times non-greedy) means "match all further text but stop before the next symbol in the expression (non-greedy)";
- `\-\-\>` is the comment closing tag.

Please note that you should escape all non-alphanumeric symbols when searching with regular expressions.

As we want to remove comments, we leave the Replace with field blank.

3. To delete `SPAN` tags, we create a new action and also use regular expressions but more complex for the **Find what** field:

```
\<span\s#lang=\w#\-\w#\s#style=\\'mso.\#\'\>(.\#)\</span\>
```

Here:

- `#lang=\w#\-\w#` stands for the `LANG` attribute with the language abbreviation;
- `style=\\'mso.\#\'` matches any style attribute starting from any `msO` style attribute;
- `(.\#)` is the tag inner text that we must preserve.

We replace the found text with the tag inner text:

\1

We shall omit operations on adding actions for being short.

4. To delete tags <o:p>, we do not need regular expressions as they always have only one form:

Find what: <o:p></o:p>

Replace with: *empty*

5. To clean the HTML tag, we need regular expressions again:

Find what: \<HTML.#\>

Replace with: <HTML>

Please note that escapes are optional in the replace field.

And lastly, we shall remove the Office [if] conditions.

Find what: \<\!\[if\s.#\]\>(.#)\<\!\[endif\]\>

Replace with: \1

Please note that escapes are optional in the replace field.

We have now created a new scenario. Click the **Save and Activate** button. The [Files and Text Tab](#) will change to the **Advanced** mode displaying the scenario name.

Now do the common operations:

- Specify the folder you want to search for files. For example: C:\MyWebFiles\Projects
- Set mask(s) of the files that you want to find. For example: \*.htm\*
- Click the **Replace** button.

### Note

Please do not forget that you can run the **HFFR** from the command line and specify the name of a scenario to activate by using the parameter

/scn:"scenario\_name".

# HOWTO: Use Command Line Switches

The **HandyFile Find And Replace** has an extensive set of command-line switches allowing you to fully configure the tool when running it from the command line or batch file.

We shall consider some examples of running the **HFFR** from the command line.

## Note

Line-breaks in the command line samples are for readability only.

## Find And Replace Simple Text In Files

### Task

Replace all occurrences of the phrase *Silver Age Software* with *SilverAge Software* in HTML files in folder `C:\MyWebFiles\Projects`, including subfolders. The text requires case-sensitive search. Close the HFFR after the processing is done.

### Command line

```
hffr.exe /folder:"C:\MyWebFiles\Projects" /recurse:on  
/type:"*.htm*"  
  
/xwhat:"Silver Age Software"  
  
/xwith:"SilverAge Software"  
  
/case:on /rx:off /replace:yes /run /close
```

## Find And Replace Text In Files Using Regular Expressions

### Task

Replace all occurrences of the tag `SPAN`, preserving the tag inner text in HTML files in folder `C:\MyWebFiles\Projects`, *not* including subfolders. The text search is not case-sensitive. Do *not* close the HFFR after the processing is done.

### Command line

```
hffr.exe /folder:"C:\MyWebFiles\Projects" /recurse:off  
/type:"*.htm*"  
  
/xwhat:"\<span.#\>(.#)\</span\>"  
  
/xwith:"\1"  
  
/case:off /rx:on /replace:yes /run
```

## Find And Replace Text In Files Using a Scenario

### Task

Apply the previously created scenario *MyScenario* to HTML and ASP files in folder `C:\MyWebFiles\Projects`, including subfolders. Do *not* close the HFFR after the processing is done. Create the HTML report *report.html* in the folder `C:\MyWebFiles\Projects`.

### Command line

```
hffr.exe /folder:"C:\MyWebFiles\Projects" /recurse:on  
/type:"*.htm*;*.asp"  
/scn:"MyScenario"  
/list:"C:\MyWebFiles\Projects\report.html"  
/run
```

# Command Line

HandyFile Find and Replace is equipped with a set of command line parameters which enable you to perform the find/replace tasks from within any batch file and/or external-tool-aware application.

**All switches and their values are case-sensitive.**

**Please note that switches have no default values. The values that had been set during the last attended runtime session are considered default.**

Switch	Description
<b>File and Path Control</b>	
<code>/folder:"quoted_path"</code>	The folder in which the operation will be performed. Quote the string if it has long names.
<code>/recurse:on off</code>	Recurse subfolders.
<code>/type:"quoted_type"</code>	Type(s) (filters) of the files to search for.
<code>/romod:yes no</code>	Modify read only files.
<code>/rorest:yes no</code>	Restore read-only attribute.
<code>/retaintime:yes no</code>	Tells to leave the times of the modified file as-is, without change.
<code>/bakmode:{d s p}:"folder_or_path"</code>	<p>Back-up creation mode and path to the back-up folder.</p> <p><b>Modes (required)</b></p> <p>d - default; s - subfolder of the search folder; p - static path.</p> <p><b>Folder or path (required)</b></p> <p>If mode is <code>d</code>, folder name is ignored. If mode is <code>s</code>, name of the subfolder. If mode is <code>p</code>, path to the back-up folder; the path may include environment variables (<code>%variable_name%</code>).</p> <p>Command line path for <code>/bakmode</code> may not contain spaces. To overcome this limitation, declare any environment variable and specify it here with <code>p</code> mode.</p> <p><b>Examples</b></p> <pre>/bakmode:s:"backup" /bakmode:p:"%BackupFolder%"</pre> <p>These modes fully reflect the <a href="#">Storage Folders Tab</a> options for the <b>Back-up Folder</b>.</p>

Switch	Description
<pre>/tgtmode:{d s p}:"folder_or_path"</pre>	<p>Target (processed) file storage mode and path to the folder to store the processed files.</p> <p><b>Modes (required)</b></p> <p>d - default; s - subfolder of the search folder; p - static path.</p> <p><b>Folder or path (required)</b></p> <p>If mode is <b>d</b>, folder name is ignored. If mode is <b>s</b>, name of the subfolder. If mode is <b>p</b>, full path to the target folder; the path may include environment variables (<i>%variable_name%</i>).</p> <p>Command line path for <code>/tgtmode</code> may not contain spaces. To overcome this limitation, declare any environment variable and specify it here with <b>p</b> mode.</p> <p><b>Example</b></p> <pre>/tgtmode:s:"result" /tgtmode:p:"%TargetFolder%"</pre> <p>These modes fully reflect the <a href="#">Storage Folders Tab</a> options for the <b>Folder for Processed Files</b>.</p>
Text Control	
<pre>/xwhat:"quoted_string"</pre>	<p>Find What. Quote the string if it contains space(s).</p> <p>To use the quotation marks in the text, switch regular expressions on and use operator <code>\x22</code>.</p>
<pre>/xwith:"quoted_string"</pre>	<p>Replace With. Quote the string if it contains space(s).</p> <p>To use the quotation marks in the text, switch regular expressions on and use operator <code>\x22</code>.</p>
<pre>/case:on off</pre>	Case sensitivity.
<pre>/rx:on off</pre>	<a href="#">Regular Expressions</a> .
<pre>/replace:yes no</pre>	Whether perform replace or not.
<pre>/scn:"scenario_name"</pre>	<p>Designates the use of a <a href="#">scenario</a> to execute.</p> <p><b>Note</b> If the scenario name is correct, it will be activated upon start thus all other text control switches will have no effect if specified.</p>
Report Control	
<pre>/list:"quoted_path_name"</pre>	Generate HTML listing.

Switch	Description
<b>Runtime Control</b>	
<code>/cfg:"configuration_file"</code>	Name of the <a href="#">configuration file</a> .  <b>Note</b> This option erases and cancels <i>all</i> other options if they are specified on the command line. All options are presumed to be stored in the configuration file.
<code>/run</code>	Start automatically when launched.
<code>/close</code>	Close right after automatic run.
<code>/hidden</code>	Do not show the application window when running. This switch is only applicable if <code>/run</code> and <code>/close</code> switches are both specified.

# Configuration File for Use in Command Line

Instead of supplying a long tail of options on the command line, you can create and use the configuration file. The configuration file is a text file containing command-line options for use with the **HandyFile Find And Replace**.

Each option must reside within a single line of text. Options can start from the new line or can be merged into a single line. In the latter case, options must be separated with a space or tab.

The simplest way to create a configuration file is using the **Configuration Export Dialog**.

## Example of the configuration file

**File: c:\MyWeb\Projects\replace\_span.hffr.cfg**

```
/folder:"C:\MyWebFiles\Projects" /recurse:off
/type:"*.htm*"
/xwhat:"\<span.#\>(.#)\</span\>"
/xwith:"\1"
/case:off
/rx:on
/replace:yes
/run
```

## Running the HandyFile Find And Replace with a configuration file

```
hffr.exe /cfg:"c:\MyWeb\Projects\replace_span.hffr.cfg"
```