Multi Replacer
20.10.2013

1. Introduction

Multi replacer program is a powerful file browser, text search, replace and data extraction program. This program runs in windows environment and supports regular expressions which is a very useful feature.

With Multireplacer, the process is done step by step. Each step is designed in an easy understandable way. Various reports can be taken from a lot of pages in the program.

Multi replacer has multiple file selection feature. Any number of files and directories can be added to the list of selection. İn addition to its **multiple file selection feature**; multiple text and information search / replace operations can be easily done. By establishing some links between information search proceses and the others; **advanced info/text search / replace techniques** are supported. After the search; the undesirable files can be removed from the list before starting to replace process. Thanks to these features, to prevent faulty replace process, no questions asked like 'CHANGE IT YES/NO' before any replace process as well as other repalce programs. Multi Replacer supports the Drag&Drop and Clipboard features of Windows.

Search results are displayed as text outputs and provides the abilty of data extraction. All e-mail adresses can be easily obtained by searching in a batch file and the results can be saved to the files.

All search tasks can be saved under a Project file; in this way, the same or similar search / replace operations can be easily repeated the next periods.

Text / info search can be operated with MS Office and PDF files. And Multi Replacer Program has the ability of making transactions from the command line.

2. Getting Started

PLEASE! READ THE FOLLOWING SUBJECTS CAREFULLY TO LEARN HOW TO USE MultiReplacer Program…

Select Project Type: choose a replacement method or text / information search.

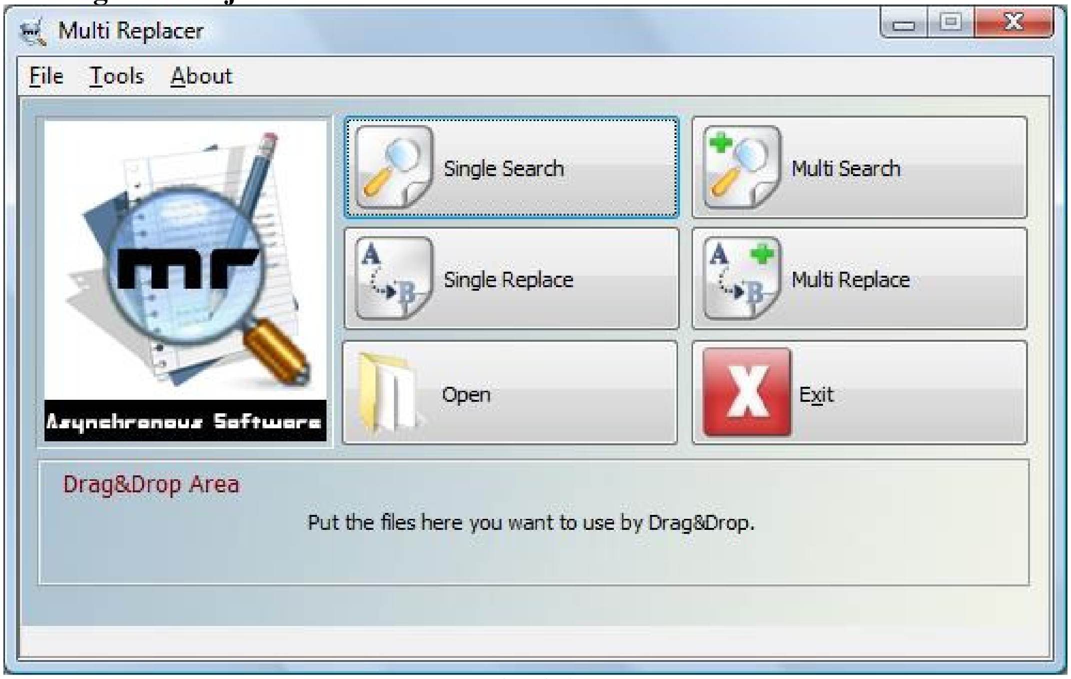
File Selection : Select the files for search operations

Search and Replace Texts: enter the search and replace texts.

Search results/Data extraction: the results are displayed after searching operations.

Replace Operation: The replace processes are applied after examining the searching results. (during Replace operations).

3. Startig New Project



Main Window

While starting a new Project; firstly it is necessary to select the type of transaction. The program makes some screen arrangements according to the type of transaction that will be selected. The processes and these transactions are very similar to each other but there are some important differents also.

The processing options are described below:

Single Search: This option is used to search only a text in a file or in a directory.

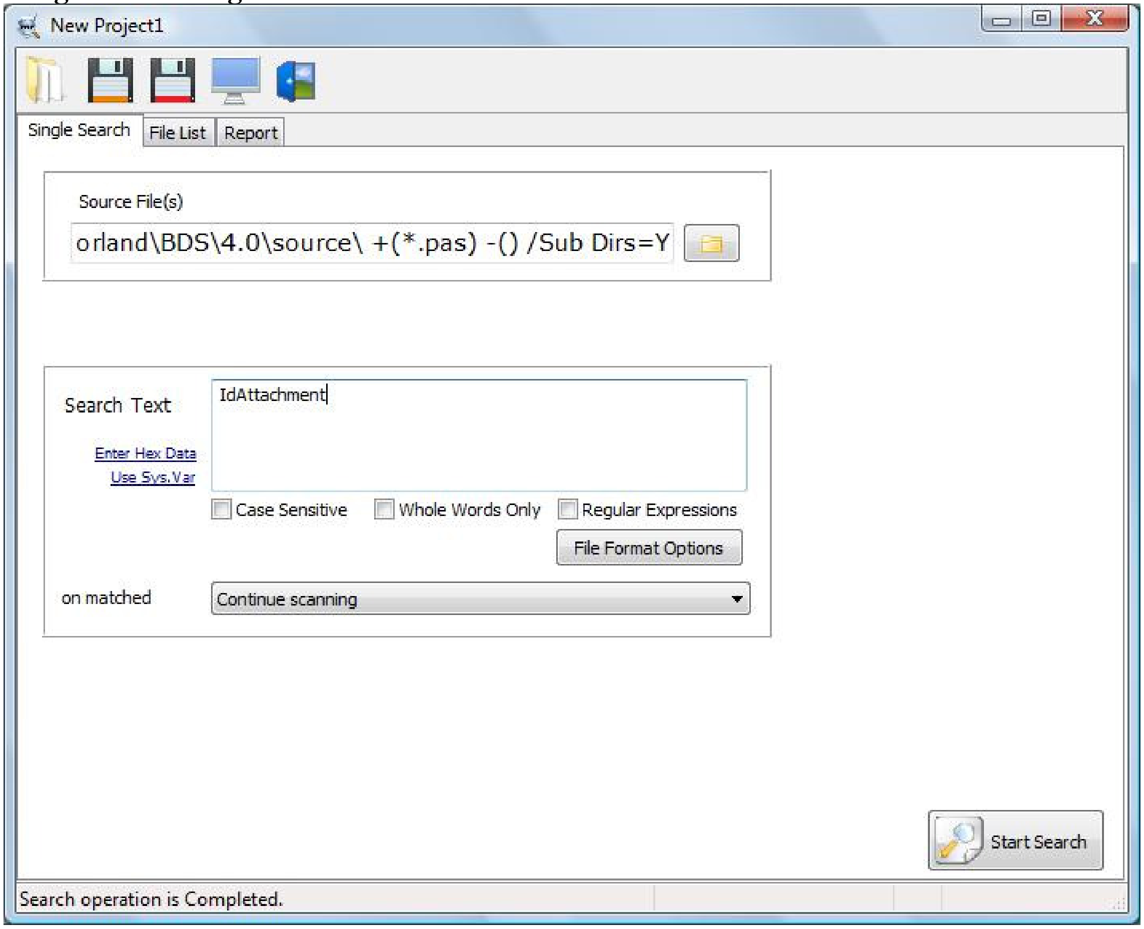
Single Replace: This option is used to search only a text in a file or in a directory and if prompted, it is used to replace.

Multi Search: This option is used to search one or more information on one or more files and directories.

Multi Replace: This option is used to search and replace one or more information on one or more files and directories.

At the end of the transaction type selection; a form consisting of several pages is displayed on the screen for the Project.

4. Single Search Page



New Project1

Source File(s): orland\BDS\4.0\source\ +(*.pas) -() /Sub Dirs=Y

Search Text: IdAttachment

□ Case Sensitive □ Whole Words Only □ Regular Expressions

on matched: Continue scanning

Search operation is Completed.

Single Search Page

This page is to search through a file or a directory or a page is used for some input operations.

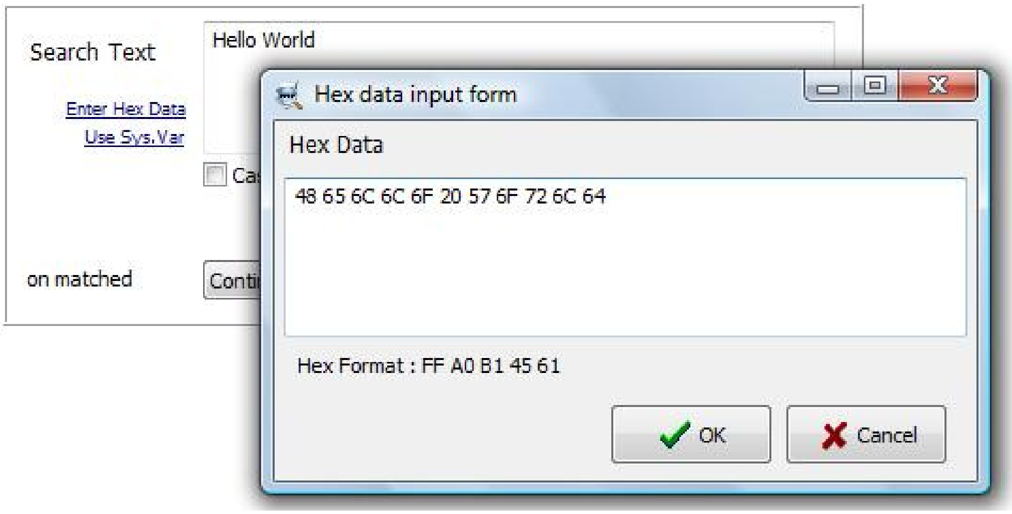
4.1 Source File(1)

This is a button box on the page and stays up side of the page, if clicked , opens the form of **Select File Source.** By using this form; the files or directories that search process made transferred to the **source File(s)** box. By using Source Files Box, to make selection of transaction type can also be used the Drag&Drop and clipboard functions of Windows. When a file or folder on the desktop is left on the Souce Files Box by dragging, the program brings up a form screen to use these files; and by clicking the **OK** button on the form, the transfer operation is completed. Also in the same way; by copying a file on the desktop or from another environment can be pasted to the Source Files Box (Short-cut is ‘Shift+insert)

4.2 Search Text

Enter the searching text here.

4.3 Enter Hex Data



Search Text: Hello World

Hex Data: 48 65 6C 6C 6F 20 57 6F 72 6C 64

Hex Format : FF A0 B1 45 61

Hex data input form

Enter Hex Data can be used to enter hex data to the Search Text / Replace Text etc. Boxes. If clicked to the **enter hex data link,** hex data entrance form is displayed to the screen. The Hex info is filled to this box on the form and using **ok** button; the filled text info are applied. These inputs displays as ASCII characters on the search text box.

4.4 Use Sys.Var

It is used to add some system variables to Search Text, Replace Text, etc. boxes. Among them, variables under File Informations option show the information belonging to the file on which search is performed. Variables under Current DateTime option show the information belonging to the start time of the search.

4.5 Case Sensitive

When this box is checked, the search is done case-sensitivity.

4.6 Whole Words Only

When this box is checked the condition of being whole word is added to the search process.

4.7 Regular Expressions

This box should be checked for searching by using Regular Expressions.

Notice: while using the searching for UNİCODE files, Regular Expressions option is active and unchangeable.

4.8 On Matched

This step is a answer of the question ‘How does The program decide to continue the process in the case of finding text info while searching operation’. This box has the following options:

4.8.1 Continue Scanning

This option is selected as default and the program runs from the previous one.

4.8.2 Search within one file only once

If this option is selected, then only one text information is searched. When text information is found within the file, search operation passes to the next file, if any.

4.8.3 Stop the search operation after the first file found

When this option is selected, search operation is stopped after the first file in which text information is found. But the search continues until the end of file.

4.8.4 Search within one file only once. Stop searching when found.

When this option is selected, search operation is stopped with the first file in which text information is found. The rest of the file in which the text information found isn't searched. This must not be confused with the previous option.

4.8.5 Use search starter

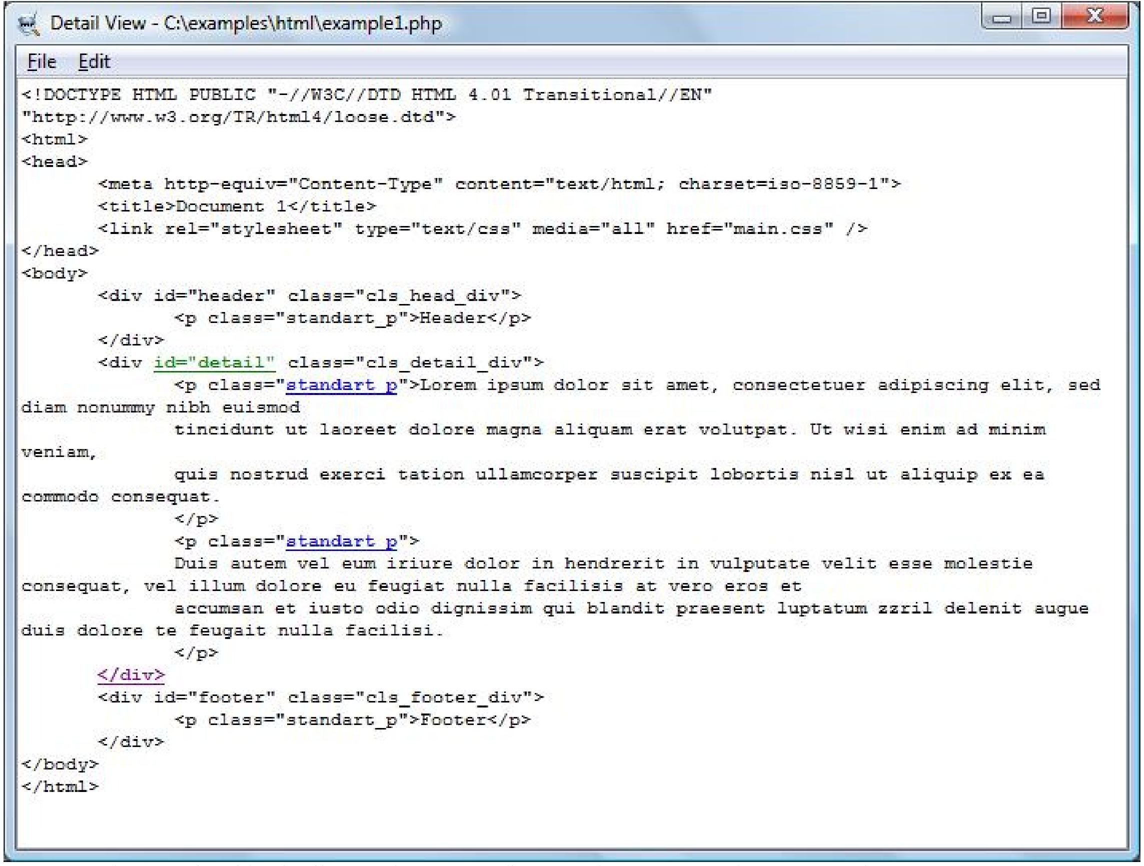
During the search operation in a file for more than one information, it may be desired to start the search beginning from text information. To specify text information as a search starting point in Multi Replacer program, this option is checked from On Matched box.

Note: This option is found in Multi Search and Multi Replace operations.

4.8.6 Use search stopper

During the search operation in a file for more than one information, it may be desired to perform the search until text information. To specify text information as a search ending point in Multi Replacer program, this option is checked from On Matched box.

Note: This option is found in Multi Search and Multi Replace operations.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
	<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
	<title>Document 1</title>
	<link rel="stylesheet" type="text/css" media="all" href="main.css" />
</head>
<body>
	<div id="header" class="cls_head_div">
		<p class="standart_p">Header</p>
	</div>
	<div id="detail" class="cls_detail_div">
		<p class="standart_p">Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed
diam nonummy nibh euismod
		tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim
veniam,
		quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea
commodo consequat.
		</p>
		<p class="standart_p">
		Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie
consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et
		accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue
duis dolore te feugait nulla facilisi.
		</p>
	</div>
	<div id="footer" class="cls_footer_div">
		<p class="standart_p">Footer</p>
	</div>
</body>
</html>
```

Example Operation: id="detail":Search Starter, </div>:Search Stopper, standart_p: Normal.

4.9 File Format Options

When this button is clicked, a form with the record format of the file, in which the text information will be searched, and related options appears. Following options exist in the form:

4.9.1 Normal

If the files searched contain plain text lines or there isn't any format, then this option should be checked (default option).

4.9.2 Within a particular character range

This option can be used if the character range of the file within which the statement should be searched is known (especially if the search will be performed in the title section of the file). Limiting the search section may allow for a quick search. When this option is selected, a number is entered in "First Char Position" box to specify the character position as starting point of the search from the beginning of the file. In Length box, a number is entered to specify the character count of section in which the search will be performed.

4.9.3 Within Range of Two Lines

This option is used if the line range of the file in which the statement will be searched is known or to limit the search to the range of two lines.

4.9.4 Search in Column Range

This option can be used mostly with the files having lines with fixed width. Search operation is performed on each line starting from the column number written in "First col number" box and for the length of character count written in "Field length" box.

Example: 5 and 7 have been entered in First col number box and Field Length box, respectively.
If the line being searched is "123456ABCDE00000000000000000000" then the search operation starts from "A" which is the 7th character and is performed for 5-character length, i.e. up to "E" character. Then, search will be carried out on other lines in the same way.

4.9.5 By Specifying Line and Column Range

Combination of two lines range and column range mentioned in the previous options and search methods can be used with this option.

4.9.6 Search in a column of CSV format file

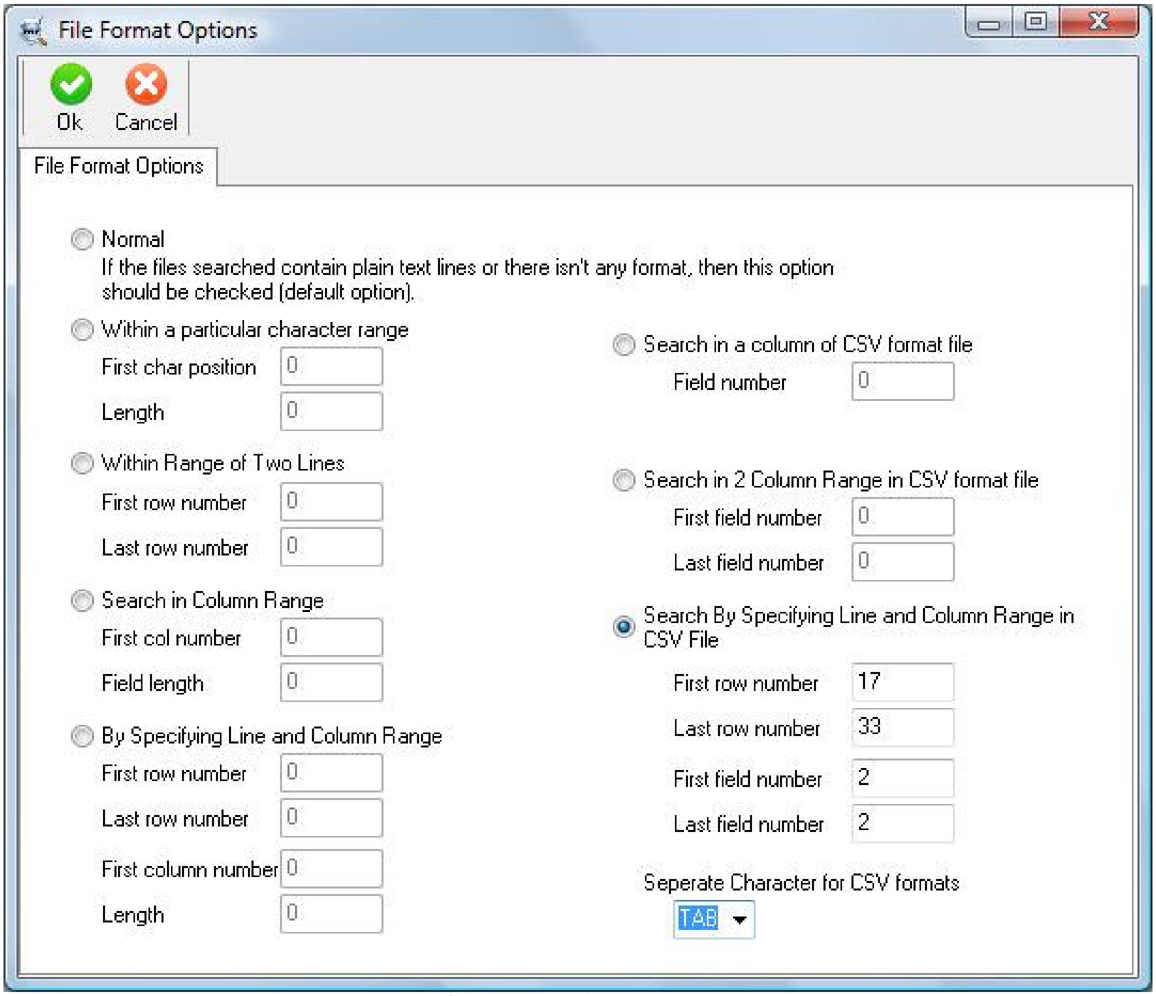
Some text files are divided into columns by using characters like comma, period, etc. This option can be used to search in a column of files saved with this format. Column separator used in the file is entered in "Separate Character For CSV Formats" box. Sequence number of the column to be searched is entered in "Field Number" box.

Example: A line within the file to be searched contains the information "abcd;1234; xyz;mm;". "Value of 3 is entered in Field Number" box. "Separate Character For CSV Formats" box contains ";" character. In this case, search within the line is performed only on "xyz;" information.

4.9.7 Search in 2 Column Range in CSV format file

It allows search within the range from the column number entered in "First Field Number" box of the file in CSV format to the column number entered in "Last Field Number" box. Search continues on all lines of the file.

4.9.8 Search By Specifying Line and Column Range in CSV File



○ Normal
If the files searched contain plain text lines or there isn't any format, then this option should be checked (default option).

○ Within a particular character range
First char position: 0
Length: 0

○ Within Range of Two Lines
First row number: 0
Last row number: 0

○ Search in Column Range
First col number: 0
Field length: 0

○ By Specifying Line and Column Range
First row number: 0
Last row number: 0
First column number: 0
Length: 0

○ Search in a column of CSV format file
Field number: 0

○ Search in 2 Column Range in CSV format file
First field number: 0
Last field number: 0

● Search By Specifying Line and Column Range in CSV File
First row number: 17
Last row number: 33
First field number: 2
Last field number: 2

Seperate Character for CSV formats: TAB

File Format Options

This option can be used if it is desired to search on the file with CSV format both within the range of two columns and a particular range of line.

> Note: If the line with the information to be searched is known approximately, specifying the ranges of line may provide faster searches.

4.9.9 Separate Character For CSV Formats

Column separator used in the file with CSV format is entered in this box. Default column separator is ";" character. Default character can be changed in the program.

After making changes in File Format Options form, "OK" button should be clicked to apply the changes. To exit without applying the changes, "Cancel" button can be clicked.

4.10 Start Search

This button starts the search operation.

5. File List Page

When the search is started, Process Viewer Form appears and search is performed on the files and directories specified previously. Files found are listed on File List page so as to appear within the container folders.

Columns in the list are:

File: Folder name containing the file or file name.

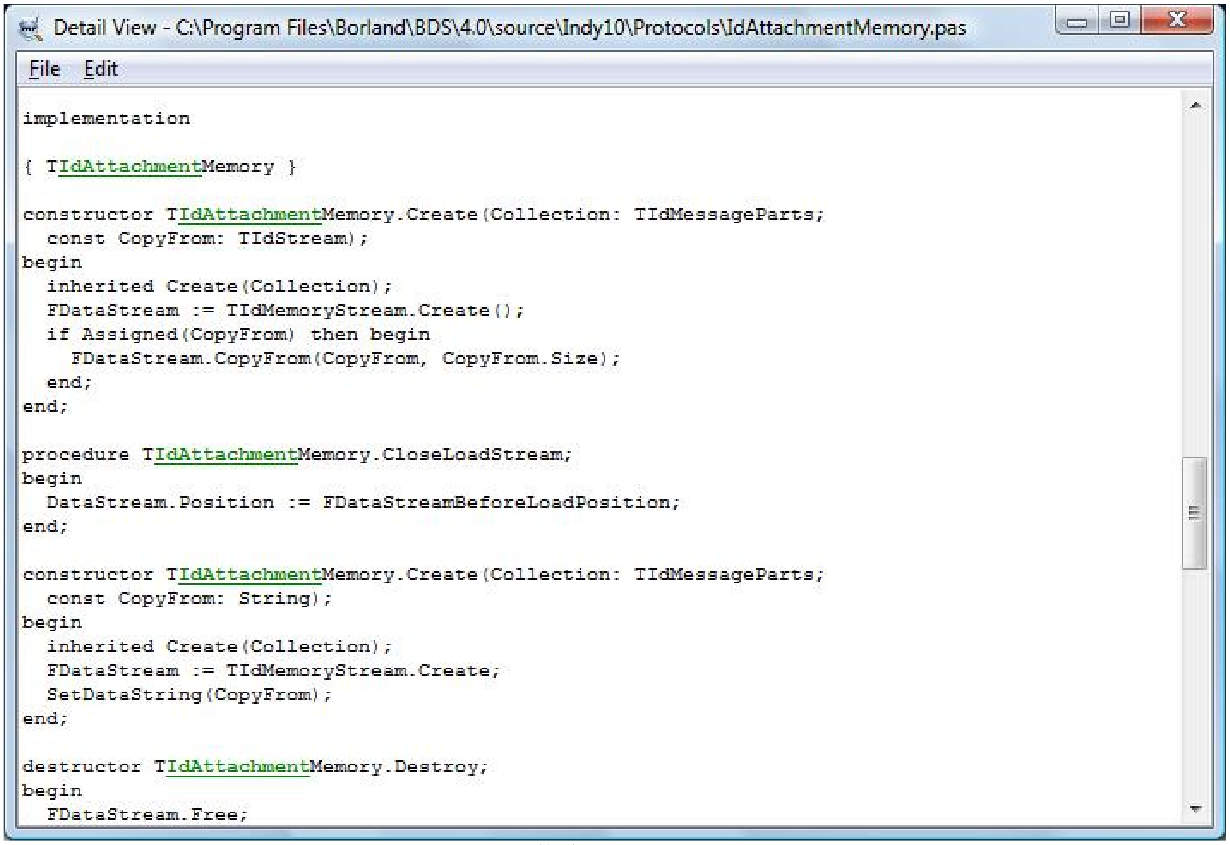
Matches: Matching count found in the file.

Size: Size of the file.

Modify Date: Last modification date of the file.

File Type: Type of the file

5.1. View Detail


Detail View - C:\Program Files\Borland\BDS\4.0\source\Indy10\Protocols\IdAttachmentMemory.pas
File
Edit

```
implementation

{ TIdAttachmentMemory }

constructor TIdAttachmentMemory.Create(Collection: TIdMessageParts;
  const CopyFrom: TIdStream);
begin
  inherited Create(Collection);
  FDataStream := TIdMemoryStream.Create();
  if Assigned(CopyFrom) then begin
    FDataStream.CopyFrom(CopyFrom, CopyFrom.Size);
  end;
end;

procedure TIdAttachmentMemory.CloseLoadStream;
begin
  DataStream.Position := FDataStreamBeforeLoadPosition;
end;

constructor TIdAttachmentMemory.Create(Collection: TIdMessageParts;
  const CopyFrom: String);
begin
  inherited Create(Collection);
  FDataStream := TIdMemoryStream.Create;
  SetDataString(CopyFrom);
end;

destructor TIdAttachmentMemory.Destroy;
begin
  FDataStream.Free;
```

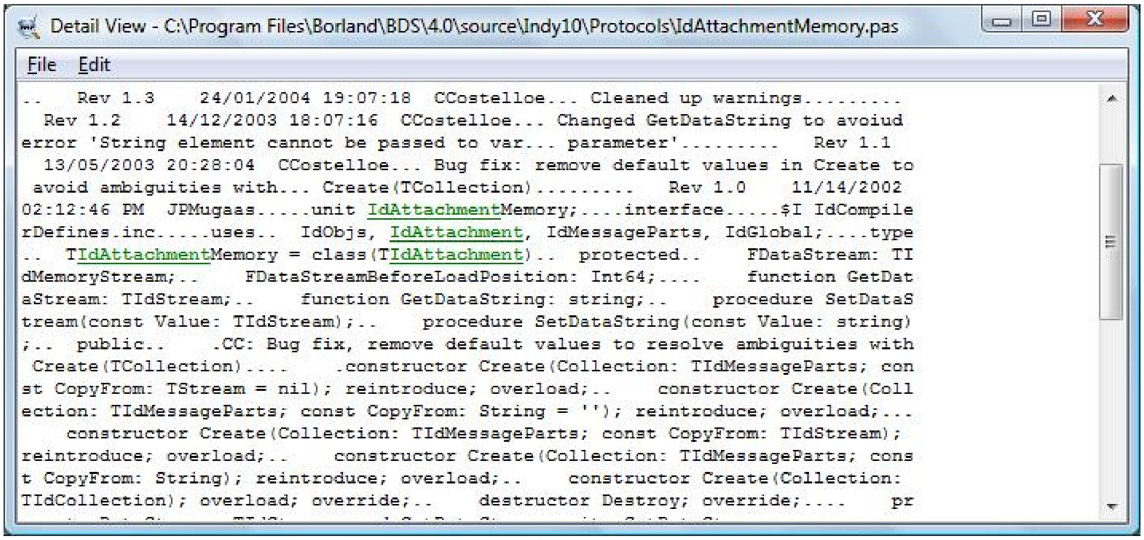
Detail View

This button displays the content of the file selected from the file list. When the button is clicked **Detail View** form is displayed. File content can be viewed in 3 formats. When View Detail button is clicked, normal text file view is displayed. To see other views, arrow-shaped button to the right of this button is clicked.

Views:

View Normal: It is recommended to open text-based files with this option.

View 80 Column Text:

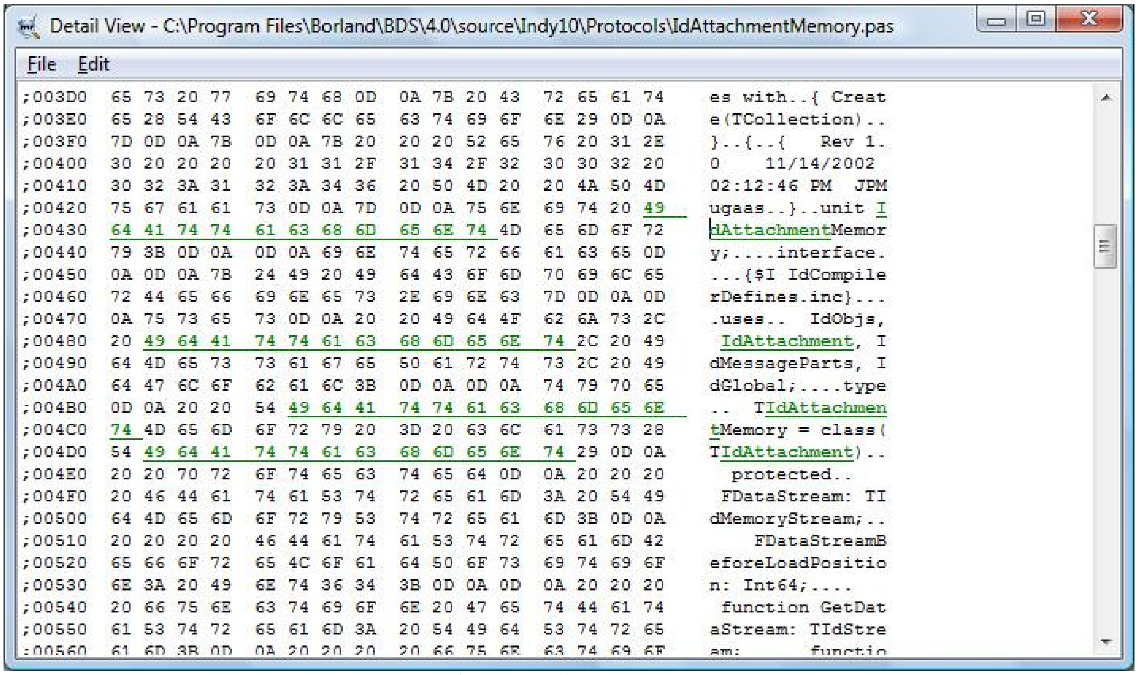

Detail View - C:\Program Files\Borland\BDS\4.0\source\Indy10\Protocols\IdAttachmentMemory.pas
File
Edit

```
..   Rev 1.3    24/01/2004 19:07:18  CCostelloe... Cleaned up warnings.........
  Rev 1.2    14/12/2003 18:07:16  CCostelloe... Changed GetDataString to avoiud
error 'String element cannot be passed to var... parameter'.........   Rev 1.1
  13/05/2003 20:28:04  CCostelloe... Bug fix: remove default values in Create to
 avoid ambiguities with... Create(TCollection).........   Rev 1.0    11/14/2002
02:12:46 PM  JPMugaas.....unit IdAttachmentMemory;....interface.....$I IdCompile
rDefines.inc.....uses..  IdObjs, IdAttachment, IdMessageParts, IdGlobal;....type
..  TIdAttachmentMemory = class(TIdAttachment)..  protected..    FDataStream: TI
dMemoryStream;..    FDataStreamBeforeLoadPosition: Int64;....    function GetDat
aStream: TIdStream;..    function GetDataString: string;..    procedure SetDataS
tream(const Value: TIdStream);..    procedure SetDataString(const Value: string)
;..  public..    .CC: Bug fix, remove default values to resolve ambiguities with
 Create(TCollection)....    .constructor Create(Collection: TIdMessageParts; con
st CopyFrom: TStream = nil); reintroduce; overload;..    constructor Create(Coll
ection: TIdMessageParts; const CopyFrom: String = ''); reintroduce; overload;...
    constructor Create(Collection: TIdMessageParts; const CopyFrom: TIdStream);
reintroduce; overload;..    constructor Create(Collection: TIdMessageParts; cons
t CopyFrom: String); reintroduce; overload;..    constructor Create(Collection:
TIdCollection); overload; override;..    destructor Destroy; override;....    pr
```

View 80 Column Text

It can be used to display text-based files or files containing text information. This option, unlike normal view, displays the files in the form of 80-column lines. End of line characters in the file are ignored. Special character code like "A-Z","a-z","0-9" and "?,*,+,-..." apart from normal characters are displayed as "." character.

View Hex:


Detail View - C:\Program Files\Borland\BDS\4.0\source\Indy10\Protocols\IdAttachmentMemory.pas
File
Edit

```
;003D0  65 73 20 77  69 74 68 0D  0A 7B 20 43  72 65 61 74   es with..{ Creat
;003E0  65 28 54 43  6F 6C 6C 65  63 74 69 6F  6E 29 0D 0A   e(TCollection)..
;003F0  7D 0D 0A 7B  0D 0A 7B 20  20 20 52 65  76 20 31 2E   }..{..{   Rev 1.
;00400  30 20 20 20  20 31 31 2F  31 34 2F 32  30 30 32 20   0    11/14/2002 
;00410  30 32 3A 31  32 3A 34 36  20 50 4D 20  20 4A 50 4D   02:12:46 PM  JPM
;00420  75 67 61 61  73 0D 0A 7D  0D 0A 75 6E  69 74 20 49   ugaas..}..unit I
;00430  64 41 74 74  61 63 68 6D  65 6E 74 4D  65 6D 6F 72   dAttachmentMemor
;00440  79 3B 0D 0A  0D 0A 69 6E  74 65 72 66  61 63 65 0D   y;....interface.
;00450  0A 0D 0A 7B  24 49 20 49  64 43 6F 6D  70 69 6C 65   ...{$I IdCompile
;00460  72 44 65 66  69 6E 65 73  2E 69 6E 63  7D 0D 0A 0D   rDefines.inc}...
;00470  0A 75 73 65  73 0D 0A 20  20 49 64 4F  62 6A 73 2C   .uses..  IdObjs,
;00480  20 49 64 41  74 74 61 63  68 6D 65 6E  74 2C 20 49    IdAttachment, I
;00490  64 4D 65 73  73 61 67 65  50 61 72 74  73 2C 20 49   dMessageParts, I
;004A0  64 47 6C 6F  62 61 6C 3B  0D 0A 0D 0A  74 79 70 65   dGlobal;....type
;004B0  0D 0A 20 20  54 49 64 41  74 74 61 63  68 6D 65 6E   ..  TIdAttachmen
;004C0  74 4D 65 6D  6F 72 79 20  3D 20 63 6C  61 73 73 28   tMemory = class(
;004D0  54 49 64 41  74 74 61 63  68 6D 65 6E  74 29 0D 0A   TIdAttachment)..
;004E0  20 20 70 72  6F 74 65 63  74 65 64 0D  0A 20 20 20     protected..   
;004F0  20 46 44 61  74 61 53 74  72 65 61 6D  3A 20 54 49    FDataStream: TI
;00500  64 4D 65 6D  6F 72 79 53  74 72 65 61  6D 3B 0D 0A   dMemoryStream;..
;00510  20 20 20 20  46 44 61 74  61 53 74 72  65 61 6D 42       FDataStreamB
;00520  65 66 6F 72  65 4C 6F 61  64 50 6F 73  69 74 69 6F   eforeLoadPositio
;00530  6E 3A 20 49  6E 74 36 34  3B 0D 0A 0D  0A 20 20 20   n: Int64;....   
;00540  20 66 75 6E  63 74 69 6F  6E 20 47 65  74 44 61 74    function GetDat
;00550  61 53 74 72  65 61 6D 3A  20 54 49 64  53 74 72 65   aStream: TIdStre
;00560  61 6D 3B 0D  0A 20 20 20  20 66 75 6E  63 74 69 6F   am;.    functio
```

View Hex Window

It is used especially to display binary files (exe,dll, etc.). This option has completely different view from normal view. Each line displayed on the screen corresponds to 16-character section of the file. Each line consists of 3 parts.

Line examples:

```
;02810  0D 0A 20 20  69 66 20 41  73 73 69 67  6E 65 64 28    ..  if Assigned(
;02820  46 4F 6E 52  6F 6C 6C 62  61 63 6B 29  20 74 68 65    FOnRollback) the
; 02830  6E 0D 0A 20  20 20 20 46  4F 6E 52 6F  6C 6C 62 61 n..    FOnRollba
```

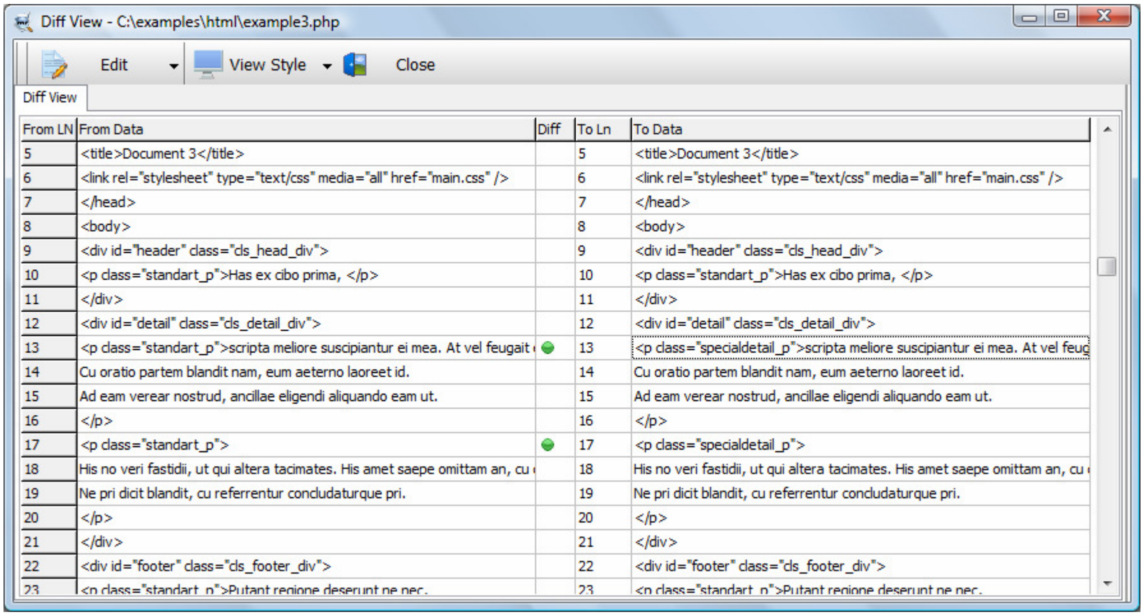
Part 1: In 8-character section of the line, it shows the address of the line displayed on the screen in hex.

Part 2: Shows hexadecimal form of 16-character line of the file.

Part 3: 16-character line of the file is displayed as normal characters. But end of line characters are displayed as special character codes "." like end of page characters.

Note: View 80 Column Text and View Hex features can't be used with Unicode file operations.

5.2. View Diff



From LN	From Data	Diff	To Ln	To Data
5	<title>Document 3</title>		5	<title>Document 3</title>
6	<link rel="stylesheet" type="text/css" media="all" href="main.css" />		6	<link rel="stylesheet" type="text/css" media="all" href="main.css" />
7	</head>		7	</head>
8	<body>		8	<body>
9	<div id="header" class="cls_head_div">		9	<div id="header" class="cls_head_div">
10	<p class="standart_p">Has ex cibo prima, </p>		10	<p class="standart_p">Has ex cibo prima, </p>
11	</div>		11	</div>
12	<div id="detail" class="cls_detail_div">		12	<div id="detail" class="cls_detail_div">
13	<p class="standart_p">scripta meliore suscipiantur ei mea. At vel feugait	●	13	<p class="specialdetail_p">scripta meliore suscipiantur ei mea. At vel feug
14	Cu oratio partem blandit nam, eum aeterno laoreet id.		14	Cu oratio partem blandit nam, eum aeterno laoreet id.
15	Ad eam verear nostrud, ancillae eligendi aliquando eam ut.		15	Ad eam verear nostrud, ancillae eligendi aliquando eam ut.
16	</p>		16	</p>
17	<p class="standart_p">	●	17	<p class="specialdetail_p">
18	His no veri fastidii, ut qui altera tacimates. His amet saepe omittam an, cu		18	His no veri fastidii, ut qui altera tacimates. His amet saepe omittam an, cu
19	Ne pri dicit blandit, cu referrentur concludaturque pri.		19	Ne pri dicit blandit, cu referrentur concludaturque pri.
20	</p>		20	</p>
21	</div>		21	</div>
22	<div id="footer" class="cls_footer_div">		22	<div id="footer" class="cls_footer_div">
23	<p class="standart_p">Putant regione deserunt ne nec.		23	<p class="standart_p">Putant regione deserunt ne nec.

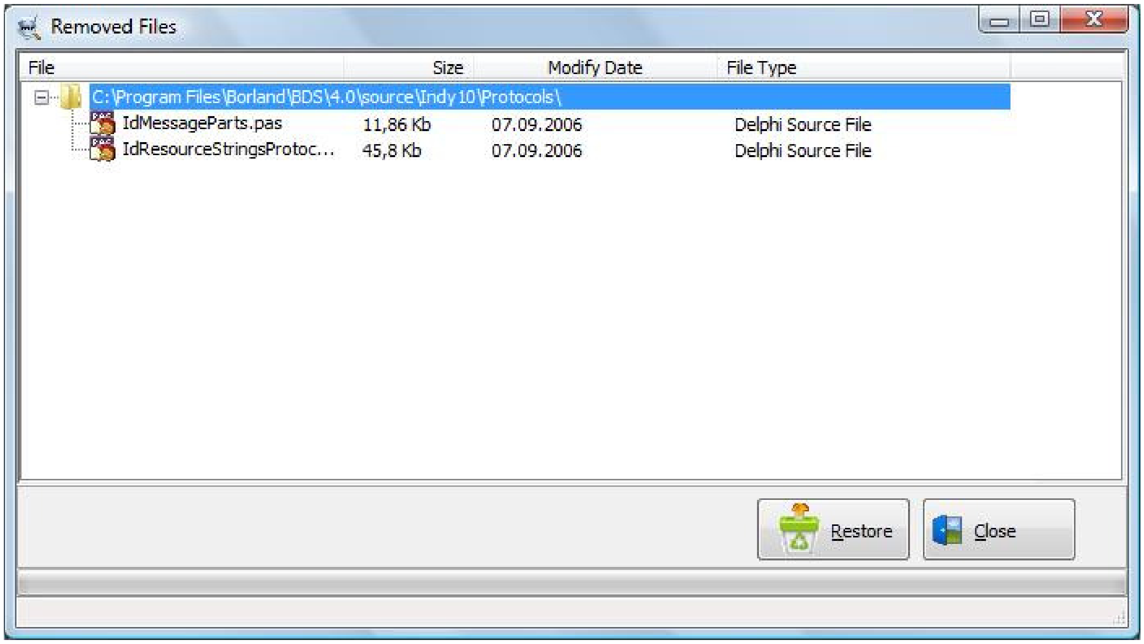
View Diff Window

This button can be used only with replace operations. It opens Diff Utility form that shows the status of the file selected in the file list with details before and after replace operation.

There are arrow-shaped button to the right of View Diff button that shows the options of using View Hex and View 80 Column Text. These options can be used mostly for the purpose of checking binary files.

Note: View 80 Column Text and View Hex features can't be used with Unicode file operations.

5.3. Remove



File	Size	Modify Date	File Type
C:\Program Files\Borland\BDS\4.0\source\Indy10\Protocols\			
IdMessageParts.pas	11,86 Kb	07.09.2006	Delphi Source File
IdResourceStringsProtoc...	45,8 Kb	07.09.2006	Delphi Source File

Removed Files Window

This button removes the selected file from File List. Replace operation can't be performed on the files removed from the list. When a file is removed from the list, "Show Removed Files" button appears.

5.4. Show Report

This button can be used to switch to report page.

5.5. Print

This button can be used to print the information in the file list.

5.6. Other Actions

When this button is clicked, a menu is opened next to it.

Menu Options

Show Matches Form: It opens Matches Form that allows detailed view of the files found with the search.

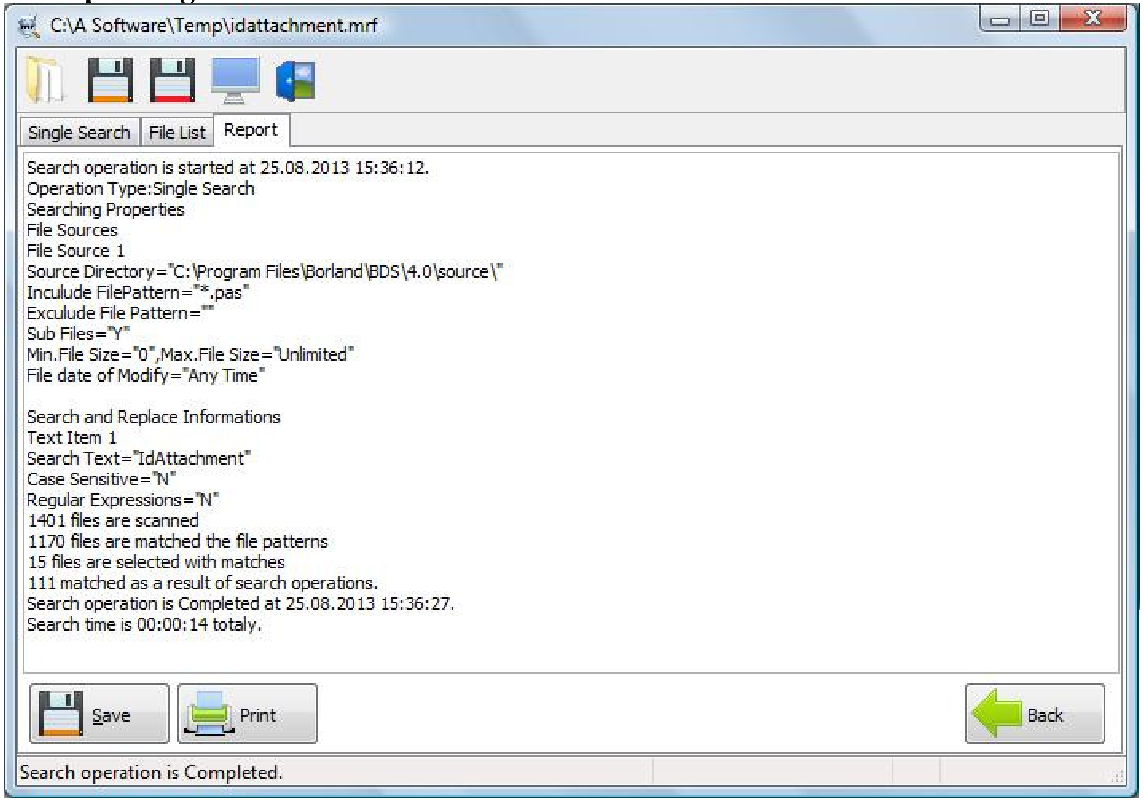
Show Removed Files: This option opens "**Removed Files"** form that shows the list of files that have been removed from File List with "Remove" button previously.

Export File List Options: File List content can be saved as a file in several formats.

Show Extracted Words: It opens a form that displays all information found during the search. File - Save As option on this form allows saving the information to a file.

Show Extracted Lines: It opens a form that displays the lines found during the search. File - Save As option on this form allows saving the information to a file.

6. Report Page



```
Search operation is started at 25.08.2013 15:36:12.
Operation Type:Single Search
Searching Properties
File Sources
File Source 1
Source Directory="C:\Program Files\Borland\BDS\4.0\source\"
Inculude FilePattern="*.pas"
Exculude File Pattern=""
Sub Files="Y"
Min.File Size="0",Max.File Size="Unlimited"
File date of Modify="Any Time"

Search and Replace Informations
Text Item 1
Search Text="IdAttachment"
Case Sensitive="N"
Regular Expressions="N"
1401 files are scanned
1170 files are matched the file patterns
15 files are selected with matches
111 matched as a result of search operations.
Search operation is Completed at 25.08.2013 15:36:27.
Search time is 00:00:14 totaly.
```

Search operation is Completed.

Show Report Page

This page shows the brief information related to Search and Replace operations.

Save: This button can be used to save report information in a file.
Print: This button can be used to print report information.

7. Single Replace Page

This page is used to perform some input operations in order to replace an information within a file or directory with another one.

7.1 Source File(s)

This box exists at the top of the page. When the buttons next to this box are clicked, "Select File Source" form opens. By using this form, file or directory on which the search operation will be performed is transferred to Source File(s) box. An important point to be taken into account on "Select File Source" form is to determine the directory for saving the new files to be created as a result of Replace operation (Destination Directory). A destination directory to be used frequently can be adjusted by parameters. If saving to a different directory is desired, firstly Destination Directory Options is selected as "Specific Destination Directory" and then the preferred directory is selected by using the button next to Destination Directory box. To perform file selection by using Source File(s) box, Drag&Drop and Clipboard functions of Windows can also be used. If a file or folder on the desktop is dragged and dropped on Source File(s) box, program displays a form that allows the use of these files. If OK button on the form is clicked, then transfer operation is completed. Again in the same way, a file on the desktop can be copied and then pasted (Shift+Insert) with Source File(s) box checked.

7.2 Sub Match Text

When carrying out replacing operation, it allows searching another information within the one being searched. Replacing operation is performed by searching "Sub Match Text" information within each "Search Text" information found, and then replacing with "Replace Text" information. If sub match text hasn't been entered, text field found as a result of the search is replaced completely with the text entered in Replace Text box.

Example: There is an information like "Indiana Jones" in the file to be searched.
Search Text: "I.* Jones"
Sub Match Text: "Jones"
Replace Text: "Smith"
Regular Expression:On

In this manner, "Indiana Jones" is changed as "Indiana Smith" with a replace operation.

7.3 Replace Text

Enter the replace text here.

7.4 Keep Case Options

While the text found with a replace operation is being changed with Replace Text, this option is checked to find lowercase and uppercase compliance according to the text found. To use this option in normal text searches without using Regular Expressions, "Case Sensitive" option must also be disabled.

Example:

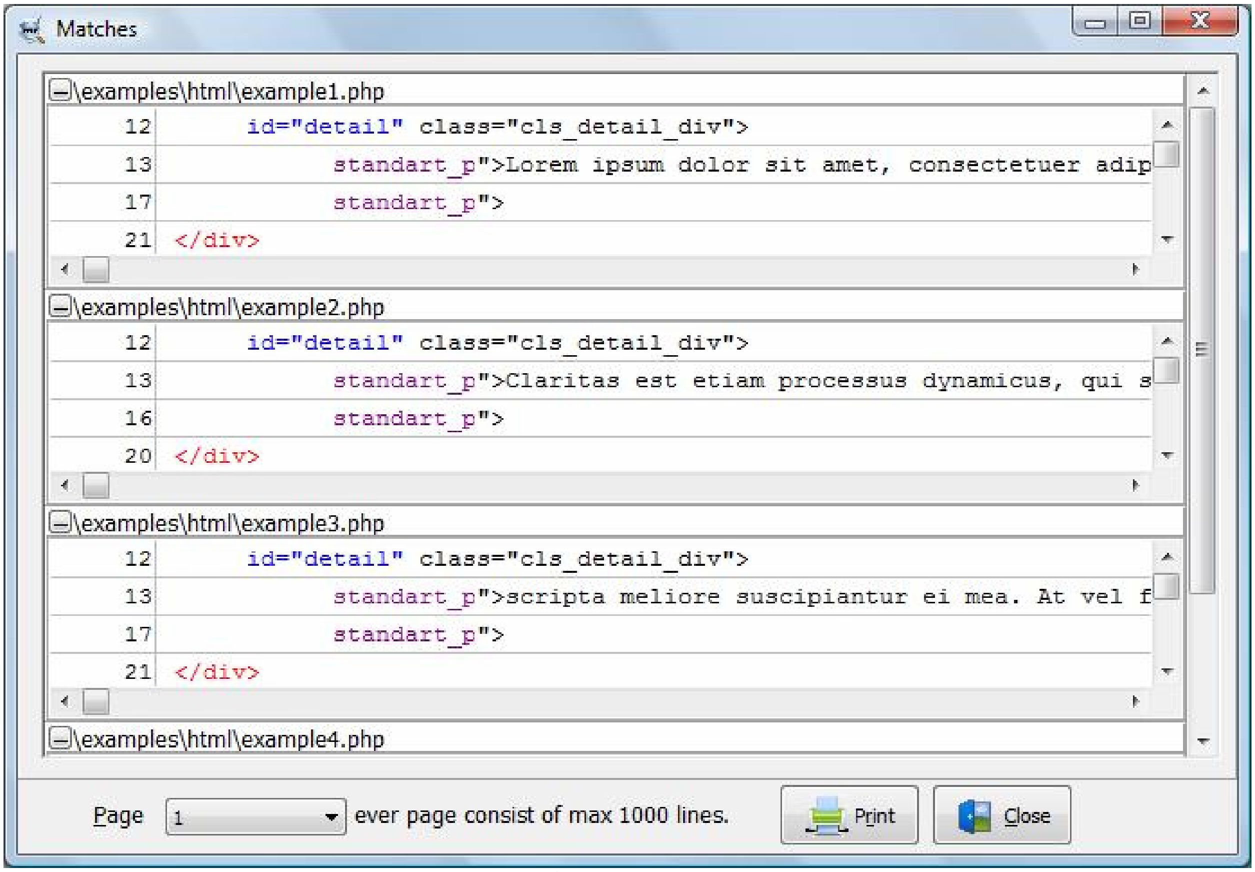
Search Text: "TOM"
Replace Text: "JERRY"
Case Sensitive: Off
Keep Case: On

A line in the file: "Tom started to run immediately when it saw the dog."
After the replace operation: "Jerry started to run immediately when it saw the dog."

7.5 The following topics are described in the previous pages

Search Text
Use Sys.Var
Enter Hex Data

8. Matches Form



```
\examples\html\example1.php
   12        id="detail" class="cls_detail_div">
   13             standart_p">Lorem ipsum dolor sit amet, consectetuer adip
   17             standart_p">
   21   </div>
\examples\html\example2.php
   12        id="detail" class="cls_detail_div">
   13             standart_p">Claritas est etiam processus dynamicus, qui s
   16             standart_p">
   20   </div>
\examples\html\example3.php
   12        id="detail" class="cls_detail_div">
   13             standart_p">scripta meliore suscipiantur ei mea. At vel f
   17             standart_p">
   21   </div>
\examples\html\example4.php
```

Page 1 ever page consist of max 1000 lines.

Matches Form

On this form, files found after the search operation and lines matched within the file are listed.

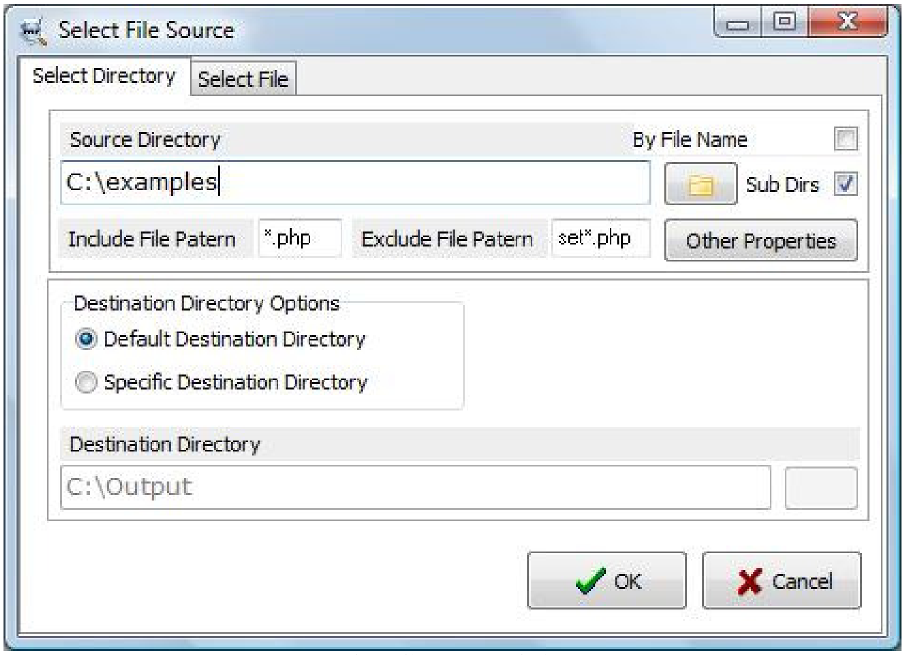
Way of listing:

File name and directory are displayed in the first line. There is a button that opens or closes the detailed information on the left of the line. Lines where searched words are found in the file are listed in order in the subsequent lines. Lines are displayed in the form of 2 columns: Line number in 1st column and line itself in 2nd column.

Page: This combo box is used to select a page. Hundreds of files and thousands of lines may be found as a result of search. It takes time to load this data to the memory of the computer and also it may cause some problems. Thus, Multi Replacer Program uses paging system to display this information. Each page consists of maximum 1000 lines. Page combo box is used to switch between pages.

Print: When this button is clicked, preview form is displayed for printing the information on Matches Form. Report printout can be checked on this form and it can be printed if desired.

9. Select File Source Form



Source Directory: C:\examples

By File Name ☐

Sub Dirs ☑

Include File Patern: *.php

Exclude File Patern: set*.php

Destination Directory Options

● Default Destination Directory

○ Specific Destination Directory

Destination Directory: C:\Output

Select File Source Window

This form helps in the selection of file sources on which the searching/replacing operations will be performed. File source adding operation can be performed in two ways. Two pages have been created for these options.

a) Select Directory
b) Select File

If it is desired to add only one file, then "Select File" page is used. To add a directory, "Select Directory" page is used.

9.1 Select File

Name along with the path of the file on which the search will be performed are entered in source file box on this page. If desired, file selection can be made easily with a dialog form by clicking the button next to the box.

If being worked on a replace project, directory to which the new file will be written after the replacing operation needs to be determined. If "Destination Directory Options" option is set as "Default Destination Directory", destination directory that has been set with the general parameters of the program is used. In this case, nothing is entered into "Destination Directory" box. If it is desired to select the directory to which the new file will be written, then "Specific Destination Directory" option is checked. Then, destination directory can be entered in "Destination Directory" box. Directory selection can also be done with a dialog form by clicking the button next to "Destination Directory" box.

9.2 Select Directory

On this page, the directory to be searched is entered in Source Directory box. Button next to the box can be used to perform this operation easily. When the button is clicked, an auxiliary dialog form for directory selection appears on the screen. If it is desired to perform the search in subdirectories under the selected directory, then "Sub Dirs" box must be checked. After the directory selection, it may also be needed to make file filtering. To perform a search on all files within the selected directory, "*.*" is entered in Include File Pattern box. If it is wanted to perform only on the files with "txt" extension, then "*.txt" is entered in this box. There may be files for which the search operation isn't desired. Exclude File Pattern box is used to specify these files. For example, files with "exe" extension are program files and searching/replacing operations are undesirable. To prevent searching on these files, "*.exe" can be entered to this box. More than one definition can be entered in File Pattern boxes and ";" should be used to separate the definitions. For example, "*.exe;*.dll" can be entered in Exclude File Pattern.

If being worked on a replace project, directory to which the new files will be written after the replacing operation needs to be determined. Destination Directory is explained in "Select File" subject.

Note: Within the program, default source directory, destination directory, include file pattern, exclude file pattern, sub dirs information can be adjusted on the form accessed via Tools – Options.

9.2.1 By File Name

If information search operation will be performed in the file names rather than content, then this option is checked.

9.2.2 Other Properties

This button found on Select Directory page can be used to specify additional file properties. It is used especially when it is needed to perform a search on too much files. "File Search Criteria" form opens when this button is clicked.

9.2.2.1 File Modified Dates

If it is known in which interval the files to be searched change, then related time period is selected from this box. If the desired time period doesn't exist in the options, then "Custom" is selected. When Custom option is selected, "Custom File Modified Dates" boxes appear. Time period can be selected by using these boxes. If it isn't desired to specify the date on which the files changed, then Any Time option must be selected.

9.2.2.2 File Sizes

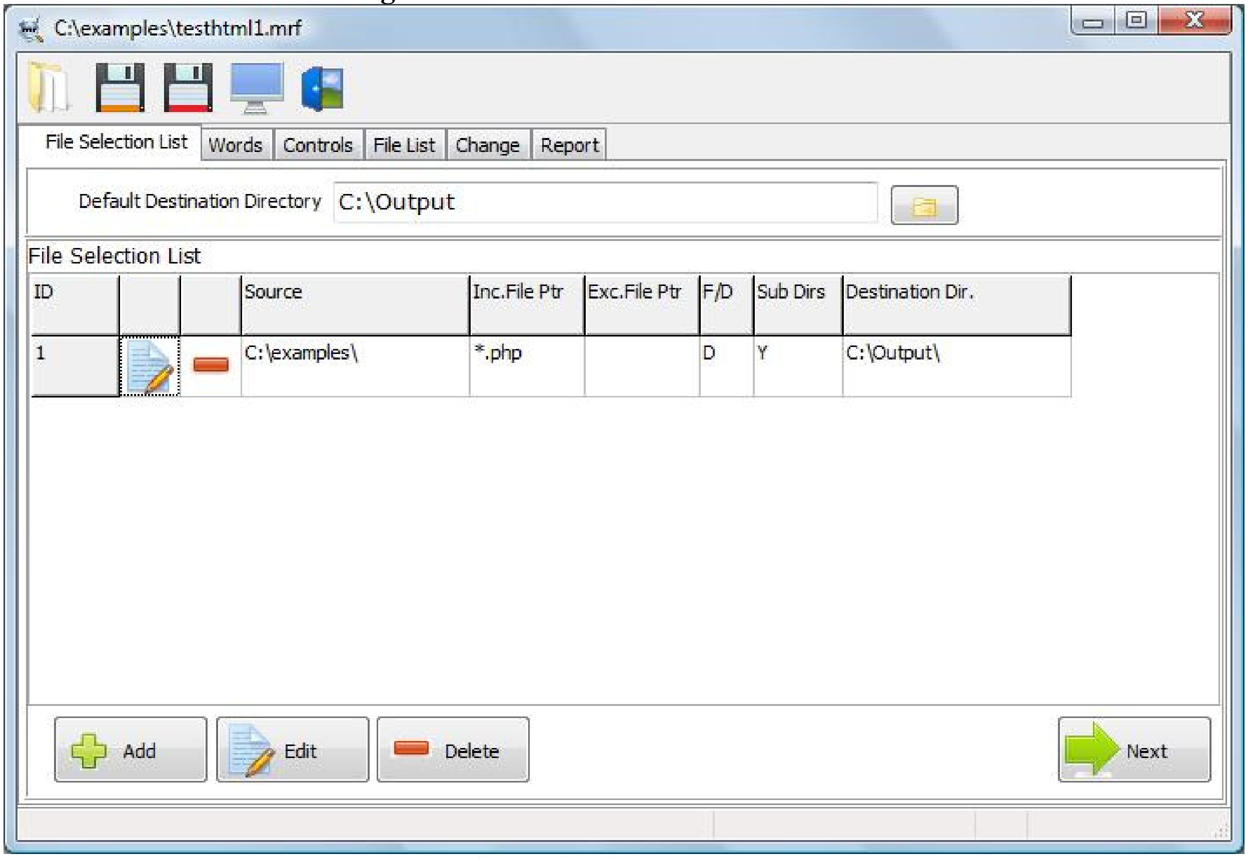
If approximate size of the file on which the search will be performed is known, then appropriate file size option is selected from this box. If it is wanted to enter the file size manually, then Custom option is selected. When Custom option is selected, "Custom File Sizes" boxes appear. These boxes help in entering file size within desired interval. If it isn't wanted to specify the file size, then "Any Size" option is selected.

After making changes in "File Search Criteria" form, "OK" button should be clicked to apply the changes. To exit without applying the changes, "Cancel" button is clicked.

9.2.4 Examples

To perform search in a network folder; e.g. "\\fileserver\documents\"
To perform search on a CD or disk drive; e.g. "E:\"
If a home user wants to make search on the files saved on his/her computer but doesn't know the exact search location, then he/she can enter "C:\".

10. File Selection List Page



Default Destination Directory C:\Output

File Selection List

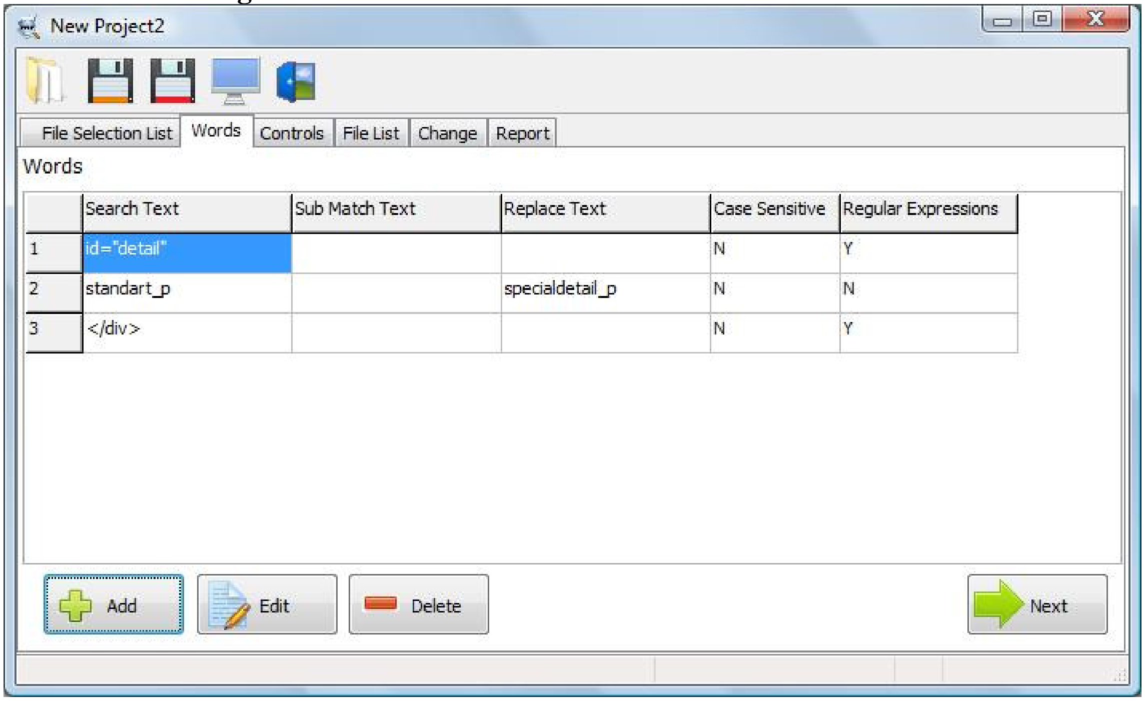
ID			Source	Inc.File Ptr	Exc.File Ptr	F/D	Sub Dirs	Destination Dir.
1	✎	➖	C:\examples\	*.php		D	Y	C:\Output\

File Selection List Page

Operation on multiple file sources can be performed with Multi Search and Multi Replace operations. File sources are added by clicking Add button found on File Selection List page. When Add button is clicked, "**Select File Source**" form appears on the screen. Information of added file is listed within "File Selection List". Item numbers of file sources are displayed on the first column in the list. Search operation on file sources is performed in order. If it is needed to make a change in the order, left button of the mouse is clicked on item number column of the line of which item number will be changed and it is dropped on the desired item number. In this way, line is moved to the desired order.

Edit button that can be used to change the file source is found on 2nd column in the list. Delete button that can be used to remove the file source from the list is found on 3rd column in the list. Basic information of file sources are displayed on the remaining columns of the list. Edit button on 2nd column in the list can be used to view in detail.

11. Words Page



	Search Text	Sub Match Text	Replace Text	Case Sensitive	Regular Expressions
1	id="detail"			N	Y
2	standart_p		specialdetail_p	N	N
3	</div>			N	Y

Words Page

More than one information to be searched can be entered with Multi Search and Multi Replace operations. Added search information appears in Words list. When Add button is clicked, "Add Word" form appears on the screen. Searching and replacing information is added to "Words" list with Add Word form.

Item Number, Search Text, Replace Text, Case Sensitive and Regular Expressions columns exist in Words list. Program uses the order found in words list when performing search and replace operations.

Change in Order: Left button of the mouse is clicked on the item number column of the line of which the order will be changed and it is dropped on the item number column of the desired line. In this way, line is moved to the desired order. Ordering may help in clearing the errors. (Please see **Controls** page.)

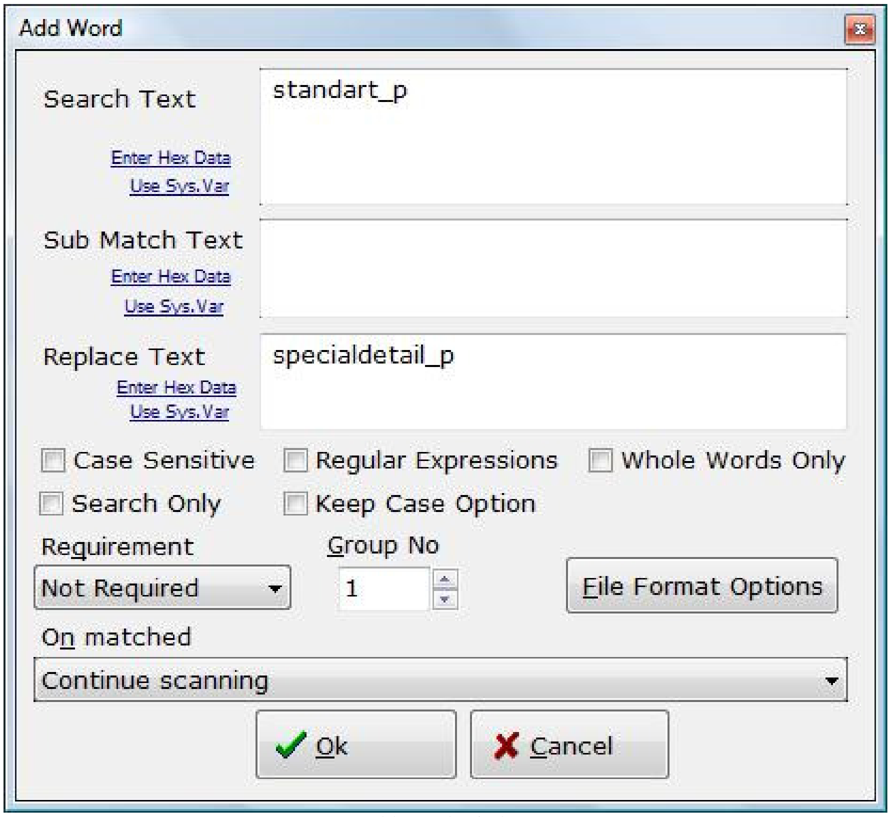
11.1 Edit

To make change on a line in Words list, firstly the line is checked, then Edit button is clicked.

11.2 Delete

To delete a line in Words list, firstly the line is checked, then Delete button is clicked.

12. Add Word Form



Add Word

Search Text: standart_p

Sub Match Text:

Replace Text: specialdetail_p

□ Case Sensitive □ Regular Expressions □ Whole Words Only

□ Search Only □ Keep Case Option

Requirement: Not Required

Group No: 1

On matched: Continue scanning

Add Word Window

The following topics are described in the previous pages

Search Text
Case Sensitive
Regular Expressions
Words Only
File Format Options
On Matched

12.1 Requirement

When many search operations are performed on a file, linking the search to inclusion or exclusion of a text may avoid unnecessary search operations. Default value of Requirement box is "Not Required ". In this case, "Requirement" box doesn't have any effect on the search. If it is set to "Match Required", then the condition for specifying search text for each file searched is added. Other search operations aren't performed on the files that don't have search information. If "No Match Required" is selected from Requirement box, condition for specifying "Search Text" information is added. Other search operations aren't performed on the files that have search text information. Requirement box uses grouping feature with **Group No** box. Some search operations may need to run dependently with each other, some of them may need to run independently. For this reason, the same number is assigned for the words to be run dependently with each other. Group number of the words that will run independently can be set as "0".

13. Controls Page

This page is used only in multi search and multi replace operations. After entering text information to be searched in Words page, Next button is clicked. Program switches to

Controls page. If Check button on this page is clicked, then program tries to find the lines that may be erroneous by comparing all search texts in order. If there are erroneous lines, these are added to "Wrong Rows" list. Erroneous record forms as follows:

If a search text information has been used on a latter line again as search text information, then the program marks this information as a line that may be erroneous. In this way, checking the errors by the user, especially for replace operations, is necessary.

Example:
1. Search Text: "John", Replace Text: "Eric"
...
8.Search Text: "John Smith", Replace Text: "Mark Williams"

If replace operation is performed in this way, then 8th line doesn't make a sense. Because the program changes all "John" words it found to "Eric". In this situation, "John Smith" words can't be found.

To clear this error, 8th line can be moved above 1st line (Please see Words page **/ Chang in Order** subject).

Note: In some cases, especially when Regular Expressions is used, program may indicate a correct line as erroneous. In such cases, user should decide by himself/herself whether the line is erroneous or not. Check operation that the program performs is provided to help in clearing only small errors. If there is no error, then one should pass to Search and Replace steps without making any changes.

Start Search: Starts the search operation.

14. Replace Page

This page can be used in replace projects. Start Replace button found within the page is clicked to complete replace operation. At the end of this operation, replace operation is carried out on the files listed on File List page. Results are written in specified "Destination Directory" location. Note: No changes are made on source files. If it is desired to replace the old files with those obtained as a result of replace operation, then users should paste the files saved in Destination Directories on source files manually.

15. Program Options

Program Options form is opened with Tools>Options menu on main form. Program parameters are displayed in pages on this form.

General Options:

On exit ask to save a new project: If this option is left checked, “Do you want to save the project?” is prompted when Close button is clicked without saving after a new project has started.

Using Case sensitive by default: Set this option so that new searches default using case sensitive.

Use regular expressions by default: Set this option so that new searches default using regular expressions.

Using Ole Automation: If it is wanted to perform search on MS Word and MS Excel files, then this option must be checked.
Note: Only search operation can be carried out on MS Word and MS Excel files with this option (replace operation can't be performed).

PDF Support: If it is desired to perform search operation on PDF files, then this option must be checked.

Note: Only search operation can be carried out on PDF files with this option (replace operation can't be performed).

Default Search Criteria

File modified dates: By using the options here, criteria can be defined in newly performed search options according to the last modification dates of the files. If it isn't desired to check the last modification dates of the files in the new search operations, then “Any Time” can be selected amongst the options.

File Sizes: By using the options here, criteria can be defined in newly performed search options according to the sizes of the files. If it isn't desired to check the sizes of the files in the new search operations, then “Any Size” can be selected amongst the options.

Default File Selection Options

Default Source and Destination Directory selections that will be used in new Search and Replace projects are defined in entry boxes on this page.

Default Source Directory:

On this page, the directory to be searched is entered in Source Directory box. Button next to the box can be used to perform this operation easily. When the button is clicked, an auxiliary dialog form for directory selection appears on the screen. If it is desired to perform the search in subdirectories under the selected directory, then "Sub Dirs" box must be checked. After the directory selection, it may also be needed to make file filtering. To perform a search on all files within the selected directory, "*.*" is entered in Include File Pattern box. If it is wanted to perform only on the files with "txt" extension, then "*.txt" is entered in this box. There may be files for which the search operation isn't desired. Exclude File Pattern box is used to specify these files. For example, files with "exe" extension are program files and searching/replacing operations are undesirable. To prevent searching on these files, "*.exe" can be entered to this box. More than one definition can be entered in File Pattern boxes and “;” should be used to separate the definitions. For example, “*.exe;*.dll” can be entered in Exclude File Pattern.

If being worked on a replace project, directory to which the new files will be written after the replacing operation needs to be determined. Destination Directory is explained in "Select File" subject.

Default Destination Directory: In Replace projects, directory in which the new files will be saved after replacing operation should be specified. Desired directory is entered in Default Destination Directory box. Directory selection can also be done with a dialog form by clicking the button next to "Default Destination Directory" box.

Regular Expressions

Some settings related to the use of Regular Expressions can be done. Greedy: The greedy setting controls whether an expression matches the maximum (greedy) or minimum it can (non greedy), e.g. With the text "123{abc}567{xyz}89" and the search text "\{.*\}" then greedy will only find one match (between the first and last bracket) and non greedy will find two matches.

MultiLine: The ^ (beginning of string) and $ (ending of string) regex operators will also match right after and right before a newline in the Subject string. This effectively treats one string with multiple lines as multiple strings. Equivalent to Perl's /m modifier.

SingleLine: Normally, dot (.) matches anything but a newline (\n). With preSingleLine, dot (.) will match anything, including newlines. This allows a multiline string to be regarded as a single entity. Equivalent to Perl's /s modifier.

Note that preMultiLine and preSingleLine can be used together.

Extended: Allows the use of Perl 5 regex extensions, such as \b word boundary, (?:) non-back referencing parenthesis, etc.

16. Command Line Parameters

Multi Replacer program has been prepared so that many functions can be carried out by using command line parameters. Command line usage is seen below:

```
MultiReplacer [<Multi Replacer File>] | [<Search files>] | [<Search folders>]
[-Subs] [-NoSubs] [-IncPtr=<pattern>] [-ExcPtr=<patterns>] [-DestDir=<destination>]
[-DMAnyTime]
[-DMWithinanhour] [-DMToday] [-DMYesterday] [-DMThisweek] [-DMThismonth]
[-DMThisYear]
[-CDMAfter=<date>] [-CDMBefore=<date>] [-MinFileSize=<bytes count>]
[-MaxFileSize=<bytes count>]
[-Search=<text>] [-Case] [-NoCase] [-Regex] [-NoRegex] [-SubMatchText=<text>]
[-ReplaceText=<text>]
[-StartSearch] [-StartReplace] [-AutoClose] [-StopAfterMatchThisFile] [-StopAfterMatchAll]
[-ExtractedWordsFile=<filename>]          [-ExtractedLinesFile=<filename>]          [-
ReportFile=<filename>]
```

The first parameter to be passed to the program must be file source or a Multi Replacer File. If file source isn't specified, then other commands will be invalid since the program can't perform any operation on the file.

-Subs(S): It is used to search also in subfolders of a folder specified as data source.

-NoSubs(NS): It is used not to search in subfolders of a folder specified as data source.

Note: Default value is NoSubs so it is not necessary to specify this again.

-IncPtr(IP): File pattern can be used to make search on specific files within a folder.

Example 1: To perform search on text files, parameter is entered as "-IncPtr=*.txt".

Example 2: To perform search on all files, parameter is entered as "-IncPtr=*.*".

-ExcPtr (EP): It can be used to prevent search on some files within a folder.

Example: To prevent the search on program files with exe extension, parameter is entered as "-ExcPtr=*.exe".

-DestDir (DD): For replace operations, folder name in which the files to be replaced will be saved is given with this parameter.

-DMAnyTime(DMANY),
-DMWithinanhour(DMH),
-DMToday(DMD),
-DMYesterday(DMY),
-DMThisweek(DMW),
-DMThismonth(DMM),
-DMThisYear(DMTY): It provides filtering properties according to the last modification date of the files on which search will be performed. For example, if –DMThisweek (DMW) is used, the search operation will be performed on the files written or replaced within the current week.

-CDMAfter(DMA): It can be used to perform search on the files that have been changed after a specific date.

-CDMBefore(DMB): It can be used to perform search on the files that have been changed before a specific date.

-MinFileSize(MIN): It can be used to provide a file size with a condition of specific minimum byte amount.

-MaxFileSize(MAX): It can be used to provide a file size with a condition of specific maximum byte amount.

-SearchText(STX): Information to be searched in files is given with this parameter.
-Case(C): Enables Case sensitive feature.

-NoCase(NC): Disables Case sensitive feature (Default).

-Regex(RX): Enables Regular Expressions feature.

-NoRegex(NRX): Disables Regular Expressions feature (Default).

-SubMatchText(SMT): When carrying out replacing operation, it allows searching another information within the one being searched.

-ReplaceText(RTX): Set the replace text.

-StartSearch(STS): Enables automatic start of search operation.

-StartReplace(STR): Enables automatic start of replace operation.

-AutoClose(AC): This parameter can be used with StartSearch or StartReplace parameters and it allows the program to close automatically after it performed the operations.

-ExtractedWordsFile(EXWF): File name in which information obtained as a result of search operation will be written (It provides data extraction feature.)

-ExtractedLinesFile(EXLF): File name in which the lines containing information will be written as a result of search operation.

-ReportFile(RF): It is used to write report information obtained by search and replace operations to a file. (If file extension is entered as “rtf”, then it is saved in Rich Text Format.)

17. Regular Expressions

17.1 Introduction

Regular Expressions are a widely-used method of specifying patterns of text to search for. Special meta characters allow You to specify, for instance, that a particular string You are looking for occurs at the beginning or end of a line, or contains n recurrences of a certain character.

Regular expressions look ugly for novices, but really they are very simple (well, usually simple ;)), handly and powerfull tool.

17.2 Simple matches

Any single character matches itself, unless it is a meta character with a special meaning described below.

A series of characters matches that series of characters in the target string, so the pattern "bluh" would match "bluh" in the target string. Quite simple, eh ?

You can cause characters that normally function as meta characters or escape sequences to be interpreted literally by 'escaping' them by preceding them with a backslash "\", for instance: meta character "^" match beginning of string, but "\^" match character "^", "\\" match "\" and so on.

Examples:

foobar	matchs string 'foobar'
\^FooBarPtr	matchs '^FooBarPtr'

17.3 Escape sequences

Characters may be specified using a escape sequences syntax much like that used in C and Perl: "\n" matches a newline, "\t" a tab, etc. More generally, \xnn, where nn is a string of hexadecimal digits, matches the character whose ASCII value is nn. If You need wide (Unicode) character code, You can use '\x{nnnn}', where 'nnnn' - one or more hexadecimal digits.

\xnn	char with hex code nn
\x{nnnn}	char with hex code nnnn (one byte for plain text and two bytes for Unicode)
\t	tab (HT/TAB), same as \x09
\n	newline (NL), same as \x0a
\r	car.return (CR), same as \x0d
\f	form feed (FF), same as \x0c
\a	alarm (bell) (BEL), same as \x07
\e	escape (ESC), same as \x1b

Examples:

foo\x20bar	matchs 'foo bar' (note space in the middle)
\tfoobar	matchs 'foobar' predefined by tab

17.4 Character classes

You can specify a character class, by enclosing a list of characters in [], which will match any one character from the list.

If the first character after the "[" is "^", the class matches any character not in the list.

Examples:

foob[aeiou]r	finds strings 'foobar', 'foober' etc. but not 'foobbr', 'foobcr' etc.
foob[^aeiou]r	find strings 'foobbr', 'foobcr' etc. but not 'foobar', 'foober' etc.

Within a list, the "-" character is used to specify a range, so that a-z represents all characters between "a" and "z", inclusive.

If You want "-" itself to be a member of a class, put it at the start or end of the list, or escape it with a backslash. If You want ']' you may place it at the start of list or escape it with a backslash.

Examples:

[-az]	matchs 'a', 'z' and '-'
[az-]	matchs 'a', 'z' and '-'
[a\-z]	matchs 'a', 'z' and '-'
[a-z]	matchs all twenty six small characters from 'a' to 'z'
[\n-\x0D]	matchs any of #10,#11,#12,#13.
[\d-t]	matchs any digit, '-' or 't'.
[]-a]	matchs any char from ']'..'a'.

17.5 Meta Characters

Meta Characters are special characters which are the essence of Regular Expressions. There are different types of meta characters, described below.

Meta characters - line separators

^	start of line
$	end of line
\A	start of text
\Z	end of text
.	any character in line

Examples:

^foobar	matchs string 'foobar' only if it's at the beginning of line
foobar$	matchs string 'foobar' only if it's at the end of line
^foobar$	matchs string 'foobar' only if it's the only string in line
foob.r	matchs strings like 'foobar', 'foobbr', 'foob1r' and so on

The "^" meta character by default is only guaranteed to match at the beginning of the input string/text, the "$" meta character only at the end. Embedded line separators will not be matched by "^" or "$".
You may, however, wish to treat a string as a multi-line buffer, such that the "^" will match after any line separator within the string, and "$" will match before any line separator. You can do this by switching On the modifier /m.
The \A and \Z are just like "^" and "$", except that they won't match multiple times when the modifier /m is used, while "^" and "$" will match at every internal line separator.

The "." meta character by default matches any character, but if You switch Off the modifier /s, then '.' won't match embedded line separators.

Multi Replacer works with line separators as recommended at www.unicode.org (http://www.unicode.org/unicode/reports/tr18/):

"^" is at the beginning of a input string, and, if modifier /m is On, also immediately following any occurrence of \x0D\x0A or \x0A or \x0D (if You are using Unicode version of Multi Replacer, then also \x2028 or \x2029 or \x0B or \x0C or \x85). Note that there is no empty line within the sequence \x0D\x0A.

"$" is at the end of a input string, and, if modifier /m is On, also immediately preceding any occurrence of \x0D\x0A or \x0A or \x0D (if You are using Unicode version of Multi Replacer, then also \x2028 or \x2029 or \x0B or \x0C or \x85). Note that there is no empty line within the sequence \x0D\x0A.

"." matchs any character, but if You switch Off modifier /s then "." doesn't match \x0D\x0A and \x0A and \x0D (if You are using Unicode version of Multi Replacer, then also \x2028 and \x2029 and \x0B and \x0C and \x85).

Note that "^.*$" (an empty line pattern) doesnot match the empty string within the sequence \x0D\x0A, but matchs the empty string within the sequence \x0A\x0D.

Multiline processing can be easely tuned for Your own purpose with help of Multi Replacer properties LineSeparators and LinePairedSeparator, You can use only Unix style separators \n or only DOS/Windows style \r\n or mix them together (as described above and used by default) or define Your own line separators!

17.6 Meta characters - predefined classes

\w	an alphanumeric character (including "_")
\W	a nonalphanumeric
\d	a numeric character
\D	a non-numeric
\s	any space (same as [\t\n\r\f])
\S	a non space

You may use \w, \d and \s within custom character classes.

Examples:

foob\dr matchs strings like 'foob1r', ''foob6r' and so on but not 'foobar', 'foobbr' and so on

foob[\w\s]r matchs strings like 'foobar', 'foob r', 'foobbr' and so on but not 'foob1r', 'foob=r' and so on

Multi Replacer uses properties SpaceChars and WordChars to define character classes \w, \W, \s, \S, so You can easely redefine it.

17.7 Meta characters - word boundaries

\b	Match a word boundary
\B	Match a non-(word boundary)

A word boundary (\b) is a spot between two characters that has a \w on one side of it and a \W on the other side of it (in either order), counting the imaginary characters off the beginning and end of the string as matching a \W.

17.8 Meta characters - iterators

Any item of a regular expression may be followed by another type of meta characters - iterators. Using this meta characters You can specify number of occurences of previous character, meta character or sub expression.

*	zero or more ("greedy"), similar to {0,}
+	one or more ("greedy"), similar to {1,}
?	zero or one ("greedy"), similar to {0,1}
{n}	exactly n times ("greedy")
{n,}	at least n times ("greedy")
{n,m}	at least n but not more than m times ("greedy")

*? zero or more ("non-greedy"), similar to {0,}?
+? one or more ("non-greedy"), similar to {1,}?
?? zero or one ("non-greedy"), similar to {0,1}?
{n}? exactly n times ("non-greedy")
{n,}? at least n times ("non-greedy")
{n,m}? at least n but not more than m times ("non-greedy")

So, digits in curly brackets of the form {n,m}, specify the minimum number of times to match the item n and the maximum m. The form {n} is equivalent to {n,n} and matches exactly n times. The form {n,} matches n or more times. There is no limit to the size of n or m, but large numbers will chew up more memory and slow down r.e. execution.

If a curly bracket occurs in any other context, it is treated as a regular character.

Examples:
foob.*r matchs strings like 'foobar', 'foobalkjdflkj9r' and 'foobr'
foob.+r matchs strings like 'foobar', 'foobalkjdflkj9r' but not 'foobr'
foob.?r matchs strings like 'foobar', 'foobbr' and 'foobr' but not 'foobalkj9r'
fooba{2}r matchs the string 'foobaar'
fooba{2,}r matchs strings like 'foobaar', 'foobaaar', 'foobaaaar' etc.
fooba{2,3}r matchs strings like 'foobaar', or 'foobaaar' but not 'foobaaaar'

A little explanation about "greediness". "Greedy" takes as many as possible, "non-greedy" takes as few as possible. For example, 'b+' and 'b*' applied to string 'abbbbc' return 'bbbb', 'b+?' returns 'b', 'b*?' returns empty string, 'b{2,3}?' returns 'bb', 'b{2,3}' returns 'bbb'.

You can switch all iterators into "non-greedy" mode (see the modifier /g).

17.9 Meta characters - alternatives

You can specify a series of alternatives for a pattern using "|" to separate them, so that fee|fie|foe will match any of "fee", "fie", or "foe" in the target string (as would f(e|i|o)e). The first alternative includes everything from the last pattern delimiter ("(", "[", or the beginning of the pattern) up to the first "|", and the last alternative contains everything from the last "|" to the next pattern delimiter. For this reason, it's common practice to include alternatives in parentheses, to minimize confusion about where they start and end. Alternatives are tried from left to right, so the first alternative found for which the entire expression matches, is the one that is chosen. This means that alternatives are not necessarily greedy. For example: when matching foo|foot against "barefoot", only the "foo" part will match, as that is the first alternative tried, and it successfully matches the target string. (This might not seem important, but it is important when you are capturing matched text using parentheses.)
Also remember that "|" is interpreted as a literal within square brackets, so if You write [fee|fie|foe] You're really only matching [feio|].

Examples:
foo(bar|foo) matchs strings 'foobar' or 'foofoo'.

17.10 Meta characters - sub expressions

The bracketing construct (...) may also be used for define r.e. sub expressions (after parsing You can find sub expression positions, lengths and actual values in MatchPos, MatchLen and Match properties of Multi Replacer, and substitute it in template strings by Multi Replacer. Substitute).

Sub expressions are numbered based on the left to right order of their opening parenthesis.
First sub expression has number '1' (whole r.e. match has number '0' - You can substitute it in Multi Replacer. Substitute as '$0' or '$&').

Examples:
 (foobar){8,10} matchs strings which contain 8, 9 or 10 instances of the 'foobar'
 foob([0-9]|a+)r matchs 'foob0r', 'foob1r' , 'foobar', 'foobaar', 'foobaar' etc.

17.11 Meta characters - back references

Meta characters \1 through \9 are interpreted as back references. \<n> matches previously matched sub expression #<n>.

Examples:
 (.)\1+   matchs 'aaaa' and 'cc'.
 (.+)\1+   also match 'abab' and '123123'
 (['"]?)(\d+)\1 matchs '"13" (in double quotes), or '4' (in single quotes) or 77 (without quotes) etc

17.12 Modifiers

Modifiers are for changing behaviour of Multi Replacer.

There are many ways to set up modifiers.
Any of these modifiers may be embedded within the regular expression itself using the (?...) construct.
Also, You can assign to appropriate Multi Replacer properties (ModifierX for example to change /x, or ModifierStr to change all modifiers together). The default values for new instances of Multi Replacer object defined in global variables, for example global variable RegExprModifierX defines value of new Multi Replacer instance ModifierX property.

i
 Do case-insensitive pattern matching (using installed in you system locale settings), see also InvertCase.

m
 Treat string as multiple lines. That is, change "^" and "$" from matching at only the very start or end of the string to the start or end of any line anywhere within the string, see also Line separators.
s

Treat string as single line. That is, change "." to match any character whatsoever, even a line separators (see also Line separators), which it normally would not match.

g

Non standard modifier. Switching it Off You'll switch all following operators into non-greedy mode (by default this modifier is On). So, if modifier /g is Off then '+' works as '+?', '*' as '*?' and so on

x

Extend your pattern's legibility by permitting whitespace and comments (see explanation below).

r

Non-standard modifier. If is set then range à-ÿ additional include russian letter '¸', À-ß additional include '¨', and à-ß include all russian symbols.

Sorry for foreign users, but it's set by default. If you want switch if off by default - set false to global variable RegExprModifierR.

The modifier /x itself needs a little more explanation. It tells the Multi Replacer to ignore whitespace that is neither backslashed nor within a character class. You can use this to break up your regular expression into (slightly) more readable parts. The # character is also treated as a meta character introducing a comment, for example:

```
(
(abc) # comment 1
  |  # You can use spaces to format r.e. - Multi Replacer ignores it
(efg) # comment 2
)
```

This also means that if you want real whitespace or # characters in the pattern (outside a character class, where they are unaffected by /x), that you'll either have to escape them or encode them using octal or hex escapes. Taken together, these features go a long way towards making regular expressions text more readable.

17.13 Perl extensions

(?imsxr-imsxr)

You may use it into r.e. for modifying modifiers by the fly. If this construction inlined into sub expression, then it effects only into this sub expression

Examples:

(?i)Saint-Petersburg	matchs 'Saint-petersburg' and 'Saint-Petersburg'
(?i)Saint-(?-i)Petersburg	matchs 'Saint-Petersburg' but not 'Saint-petersburg'
(?i)(Saint-)?Petersburg	matchs 'Saint-petersburg' and 'saint-petersburg'
((?i)Saint-)?Petersburg	matchs 'saint-Petersburg', but not 'saint-petersburg'

(?#text)

A comment, the text is ignored. Note that Multi Replacer closes the comment as soon as it sees a ")", so there is no way to put a literal ")" in the comment.