

PilotEdit User Manual



| | |
|-----------------|--|
| Author: | support@pilotedit.com |
| Date: | 2018-10-31 |
| Version: | 12.1.0 |
| URL: | http://www.pilotedit.com |

Table of Contents

| | |
|--|----|
| 1. Introduction | 6 |
| 1.1. What is PilotEdit? | 6 |
| 1.2. PilotEdit GUI | 9 |
| 1.2.1. <i>PilotEdit Window</i> | 9 |
| 1.2.2. <i>Edit Huge Files</i> | 14 |
| 1.2.3. <i>Compare Huge Files</i> | 15 |
| 2. Install PilotEdit | 16 |
| 2.1. System Requirements | 16 |
| 2.2. Install PilotEdit | 17 |
| 2.3. Input Serial Number * | 17 |
| 2.4. Uninstall PilotEdit | 17 |
| 2.5. Upgrade PilotEdit | 17 |
| 3. Quick Start | 19 |
| 3.1. How to Edit Huge Files * | 19 |
| 3.2. How to Compare and Merge Two Files * | 20 |
| 3.3. How to Encrypt and Decrypt Multiple Files and Directories * | 21 |
| 3.4. How to Open and Edit Encrypted Files * | 22 |
| 3.5. How to Sort or Find/Remove Duplicate Lines * | 24 |
| 3.6. How to Replace Multiple Strings Automatically * | 25 |
| 4. PilotEdit Functions | 27 |
| 4.1. File | 27 |
| 4.1.1. <i>New File</i> | 27 |
| 4.1.2. <i>Open Files</i> | 27 |
| 4.1.3. <i>Open FTP Files</i> | 27 |
| 4.1.4. <i>Save File</i> | 30 |
| 4.1.5. <i>Save All</i> | 30 |
| 4.1.6. <i>Save As</i> | 30 |
| 4.1.7. <i>Save As With Old Directory Structure</i> | 30 |
| 4.1.8. <i>Save As FTP File</i> | 32 |
| 4.1.9. <i>Print</i> | 32 |
| 4.1.10. <i>Print Preview</i> | 32 |
| 4.1.11. <i>Print Setup</i> | 32 |
| 4.1.12. <i>Print Page Setup</i> | 32 |
| 4.1.13. <i>Close File</i> | 32 |
| 4.1.14. <i>Close All Files</i> | 33 |
| 4.1.15. <i>Clean Recent File List</i> | 33 |
| 4.2. Edit | 34 |
| 4.2.1. <i>Select Line</i> | 34 |

| | |
|---|----|
| 4.2.2. Select All..... | 34 |
| 4.2.3. Select Text Between Start Tag and End Tag *..... | 34 |
| 4.2.4. Undo..... | 34 |
| 4.2.5. Redo..... | 34 |
| 4.2.6. Cut..... | 35 |
| 4.2.7. Copy..... | 35 |
| 4.2.8. Copy Hexadecimal Encoding..... | 35 |
| 4.2.9. Copy File Name..... | 35 |
| 4.2.10. Copy File Path..... | 36 |
| 4.2.11. Paste..... | 36 |
| 4.2.12. Paste Hexadecimal Encoding..... | 36 |
| 4.2.13. Column Mode..... | 37 |
| 4.2.14. Word Wrap..... | 39 |
| 4.2.15. HEX Mode..... | 39 |
| 4.2.16. Tabs and Spaces..... | 39 |
| 4.2.17. Change Case..... | 39 |
| 4.2.18. Indent..... | 40 |
| 4.2.19. Insert..... | 40 |
| 4.2.20. Word..... | 42 |
| 4.2.21. Line..... | 43 |
| 4.2.22. Don't Check File Update..... | 43 |
| 4.2.23. Change File Encoding..... | 43 |
| 4.2.24. Execute Scripts *..... | 45 |
| 4.2.25. Sort and Find/Remove Duplicated Lines *..... | 46 |
| 4.2.26. Extract Strings *..... | 49 |
| 4.2.27. Format Selection *..... | 50 |
| 4.3. Search..... | 52 |
| 4.3.1. Find..... | 52 |
| 4.3.2. Find Previous..... | 54 |
| 4.3.3. Find Next..... | 54 |
| 4.3.4. Replace..... | 54 |
| 4.3.5. Find In Multiple Directories And Files..... | 57 |
| 4.3.6. Replace in Multiple Directories and Files..... | 58 |
| 4.3.7. Go To..... | 59 |
| 4.4. View..... | 60 |
| 4.4.1. Toolbar..... | 60 |
| 4.4.2. Window..... | 60 |
| 4.4.3. Status Bar..... | 60 |
| 4.4.4. Mark as Code Block *..... | 60 |

| | |
|---|-----|
| 4.4.5. Expand All * | 62 |
| 4.4.6. Collapse All * | 62 |
| 4.4.7. Expand/Collapse... * | 62 |
| 4.5. Compare | 63 |
| 4.5.1. Compare and Merge Files... * | 63 |
| 4.5.2. Compare Directories... | 68 |
| 4.6. Encryption * | 69 |
| 4.6.1. Open Encrypted Files... * | 69 |
| 4.6.2. Open Encrypted FTP Files... * | 69 |
| 4.6.3. Save As Encrypted File... * | 70 |
| 4.6.4. Save As Non-Encrypted File... * | 70 |
| 4.6.5. Save As Encrypted FTP File... * | 71 |
| 4.6.6. Save As Non-encrypted FTP File... * | 71 |
| 4.6.7. Encrypt/Decrypt Local Files... * | 71 |
| 4.7. Configure | 73 |
| 4.7.1. FTP Account... | 73 |
| 4.7.2. File Type... | 74 |
| 4.7.3. Set Font... | 79 |
| 4.7.4. Set Print Font... | 80 |
| 4.7.5. Change Language ... | 80 |
| 4.7.6. Edit String Table... | 82 |
| 4.7.7. Edit File Group/Bookmark... | 89 |
| 4.7.8. Theme | 91 |
| 4.7.9. Customize... | 96 |
| 4.7.10. Options... | 96 |
| 5. Regular Expression and PilotEdit Script | 105 |
| 5.1. Create Regular Expression with GUI... | 105 |
| 5.2. Parse Expression with GUI... | 115 |
| 5.3. Search Expression | 116 |
| 5.3.1. Constant String Segment | 116 |
| 5.3.2. Character List Segment | 116 |
| 5.3.3. Multiple Case-sensitive Strings Segment | 118 |
| 5.3.4. Multiple Non-case-sensitive Strings Segment | 119 |
| 5.4. Special Characters and Escape Characters for Regular Expression | 121 |
| 5.5. Search Regular Expression Examples | 123 |
| 5.5.1. Find a Line with Two Words in It | 123 |
| 5.5.2. Find a Line with Two Words in It and One Character between these Two Words | 124 |
| 5.5.3. Find a Line Start with One Word | 124 |
| 5.5.4. Find a Line Start with One Word and End with Another Word | 125 |

| | |
|---|-----|
| 5.5.5. Search Expression for Multi-line Text | 126 |
| 5.5.6. Search Expression for Multiple Case-sensitive Strings | 127 |
| 5.5.7. Find the last word that is not in the double quotation mark | 128 |
| 5.5.8. Find the "int" or "long" Variable Definition in C++ or java..... | 130 |
| 5.5.9. Find non-ASCII Characters in the File | 131 |
| 5.6. Replace Expression | 132 |
| 5.7. Replace Regular Expression Examples | 134 |
| 5.7.1. Change the Position of Two Words | 134 |
| 5.7.2. Replace one Character..... | 135 |
| 5.7.3. Remove a Line together with CR LF | 136 |
| 5.7.4. Change Multiple Case-sensitive Strings into One Word | 137 |
| 5.7.5. Insert a Special Character after the Second Character of the Last Word..... | 138 |
| 5.7.6. Change One Line into two Lines by Adding CR LF..... | 140 |
| 5.7.7. Replace Expression for Multi-line Text | 141 |
| 5.8. PilotEdit Script *..... | 143 |
| 5.8.1. Find regular expression * | 143 |
| 5.8.2. String operations * | 143 |
| 5.8.3. Replace expression * | 144 |
| 5.8.4. PilotEdit Script Examples * | 145 |
| 5.9. Advanced Examples with PilotEdit Regular Expressions | 147 |
| 5.9.1. Sort over Regular Expression * | 147 |
| 5.9.2. Extract Strings in the Double Quotation Marks and Copy Them to the Clipboard * | 148 |
| 5.9.3. Extract Email Addresses and Copy Them to the Clipboard * | 150 |
| 5.9.4. Extract Multiline Strings with Regular Expression * | 151 |
| 5.9.5. Find or Remove Duplicate Multiline Strings * | 152 |
| 6. FAQ | 153 |
| 6.1. Where does PilotEdit save configuration files and temporary files? | 153 |
| 6.2. How to recover the lost data? | 153 |
| 6.3. How to get PilotEdit's support? | 153 |
| 7. Acknowledgments..... | 154 |
| 8. Appendix | 155 |
| 8.1. PilotEdit Revision History | 155 |

1. Introduction

1.1. What is PilotEdit?

PilotEdit is an amazing file editor that can be used to edit and compare huge files.

1. The file size supported by PilotEdit is unlimited

- Edit files larger than 400GB. (The Lite version can edit files larger than 10GB.)
- Download/upload files larger than 10GB through FTP/SFTP. (The Lite version only supports FTP.)
- Encrypt/decrypt files larger than 10GB. *
- Compare and merge files larger than 100GB. *

2. Code collapse *

- Code collapse. *
- Change the source code structure into a tree by adding PilotEdit tags. *

3. Self-defined file types and key words high-lightening

- PilotEdit supports more than 30 kinds of file types.
- Define your own file types.
- Define different TAB and indent for different file types. For example, a user may input TAB for C/C++ and four blanks for Java by presses "TAB".
- Support line-continuation characters.

4. HEX mode

- Input, delete, cut, copy and paste in HEX mode.
- Find/replace in HEX mode.
- Automatically detect file encoding when you change from HEX mode to text mode.
- Endless undo/redo in both HEX mode and text mode.

5. Column mode

- A char input through keyboard will be inserted into all the selected lines in column mode.
- A single-line string pasted will be inserted into all the selected lines in column mode.
- Multiple-line text pasted will be inserted into the selected lines line by line in column mode.

6. Endless undo/redo

- You can undo/redo even after you switch between text mode and HEX mode.

7. Word wrap

- Word wrap is fast even when you are editing a huge file.

8. Edit FTP files

- Open/save as FTP files.
- Delete multiple directories and files through FTP.
- Upload/download multiple directories and files through FTP.

9. Multiple-line find/replace


- Find/replace multiline strings.
- Find/replace multiline strings with regular expression.

10. File comparison and merging *

- Compare two directories.
- Compare and merge two files larger than 100GB. *

- Edit file directly in the comparison window. *
 - Automatically update comparison result when the file content changes. *
 - Find and replace in the comparison window. *
 - Find previous/next different/identical block. *
 - Copy all the different/identical lines to the clipboard. With this feature, you can find lines in the first file but not in the second file. *
 - Compare and merge two huge files in quick mode. *
11. Self-defined string table
- Add a pre-defined string by just one click.
 - Add pre-defined strings before and after the selected text.
 - Run windows command from string table.
 - Find regular expressions defined in the string table.
12. Regular expression
- Find/replace strings defined by regular expression.
 - Create Regular Expression with GUI
 - Find/replace multiline strings with regular expression.
13. Script file *
- You can define the frequently used regular expressions in script files and process the regular expressions by executing the script files. *
 - Remove duplicate lines by running PilotEdit script. *
14. 256-bit AES encryption/decryption *
- Edit an encrypted file transparently with PilotEdit. *
 - Encrypt/decrypt multiple directories and files. *
 - Encrypt/decrypt files more than 4G. *
 - Save the active file as an encrypted/non-encrypted file. *
 - Save the active file as an encrypted/non-encrypted FTP/SFTP file. *
15. File group/bookmark
- File group/bookmark enable you to open multiple files by just one click.
16. Find/replace in multiple directories
- Find a string in multiple directories.
 - Replace a string in multiple directories.
 - Search in zip files. *
 - Find/replace in the specified file types.
 - Find/replace in files except the specified file types. *
17. Sort *
- Sort lines in a file in ascending/descending order. *
 - Sort lines in a file as text or integer number. *
 - Sort lines in a file over one column. *
 - Sort lines in a file over target strings defined by regular expression. *
18. Find/remove duplicate lines *
- Find duplicate lines in a file. *
 - Remove duplicate lines in a file. *
 - Find/remove duplicate multiline strings with regular expression. *
19. Extract Strings *

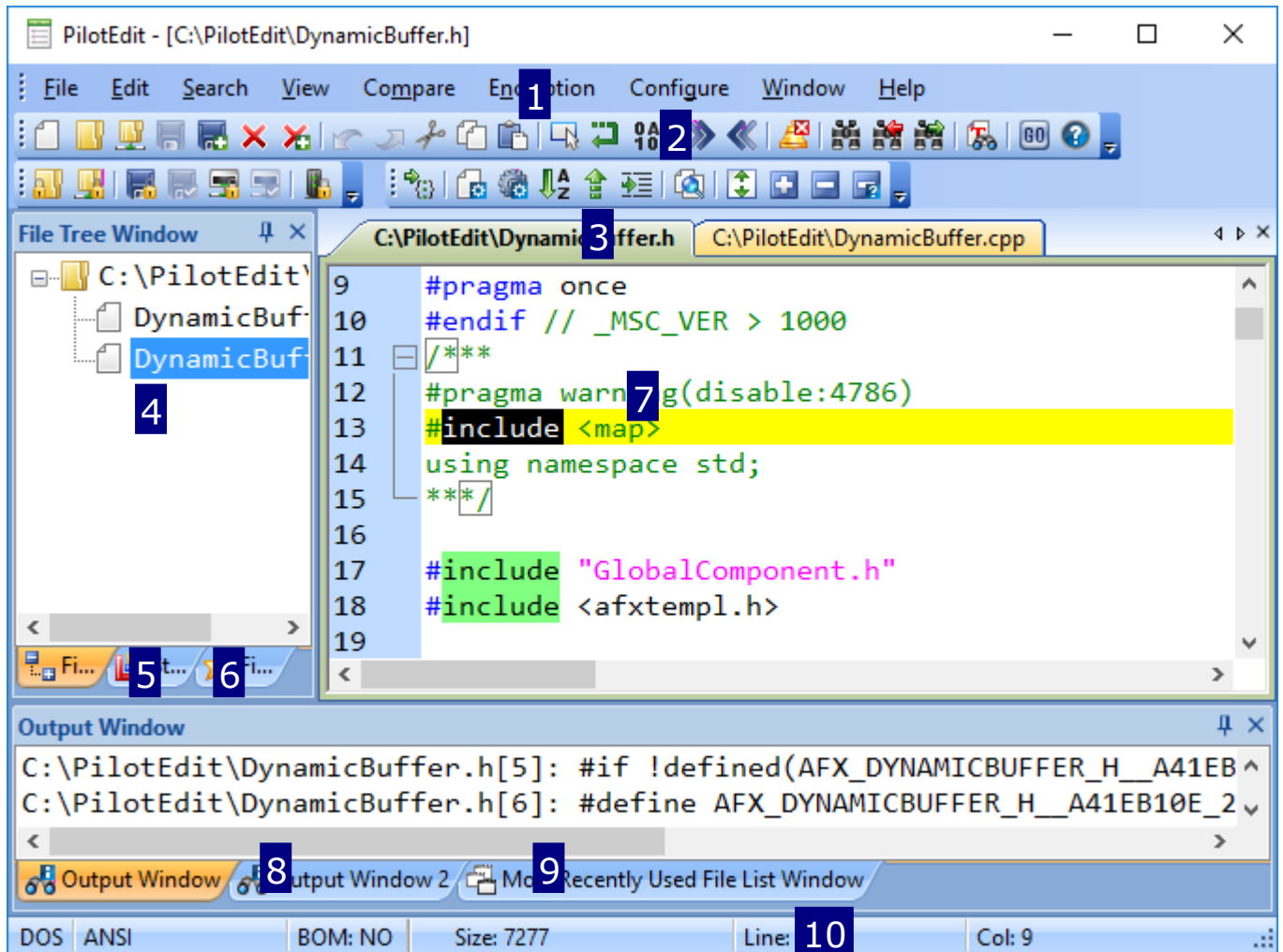
- Copy the target strings matching the regular expression into clipboard. This function is useful when you need to collect some kind of strings from a text file, for example, Email addresses. *
20. Save the selected files as new files with the old directory structure
- Copy the selected files together with the directory structure.
21. Text Formatting *
- Automatically detect start tag and end tag for a text block. *
 - Format selected text. *
22. Edit SFTP files *
- Open/save as SFTP files. *
 - Delete multiple directories and files through SFTP. *
 - Upload/download multiple directories and files through SFTP. *
23. Open very large files in quick mode. *
24. Replace millions of strings in a huge file in quick mode *
25. UNICODE files and DOS/UNIX files are fully supported by PilotEdit
- Automatically detect file encoding.
 - Change the encoding of multiple files by just one click.
 - PilotEdit will adjust text encoding automatically when you copy/paste between files of different encodings.

 Opening a huge file in PilotEdit ASCII mode is four times faster than PilotEdit Lite.

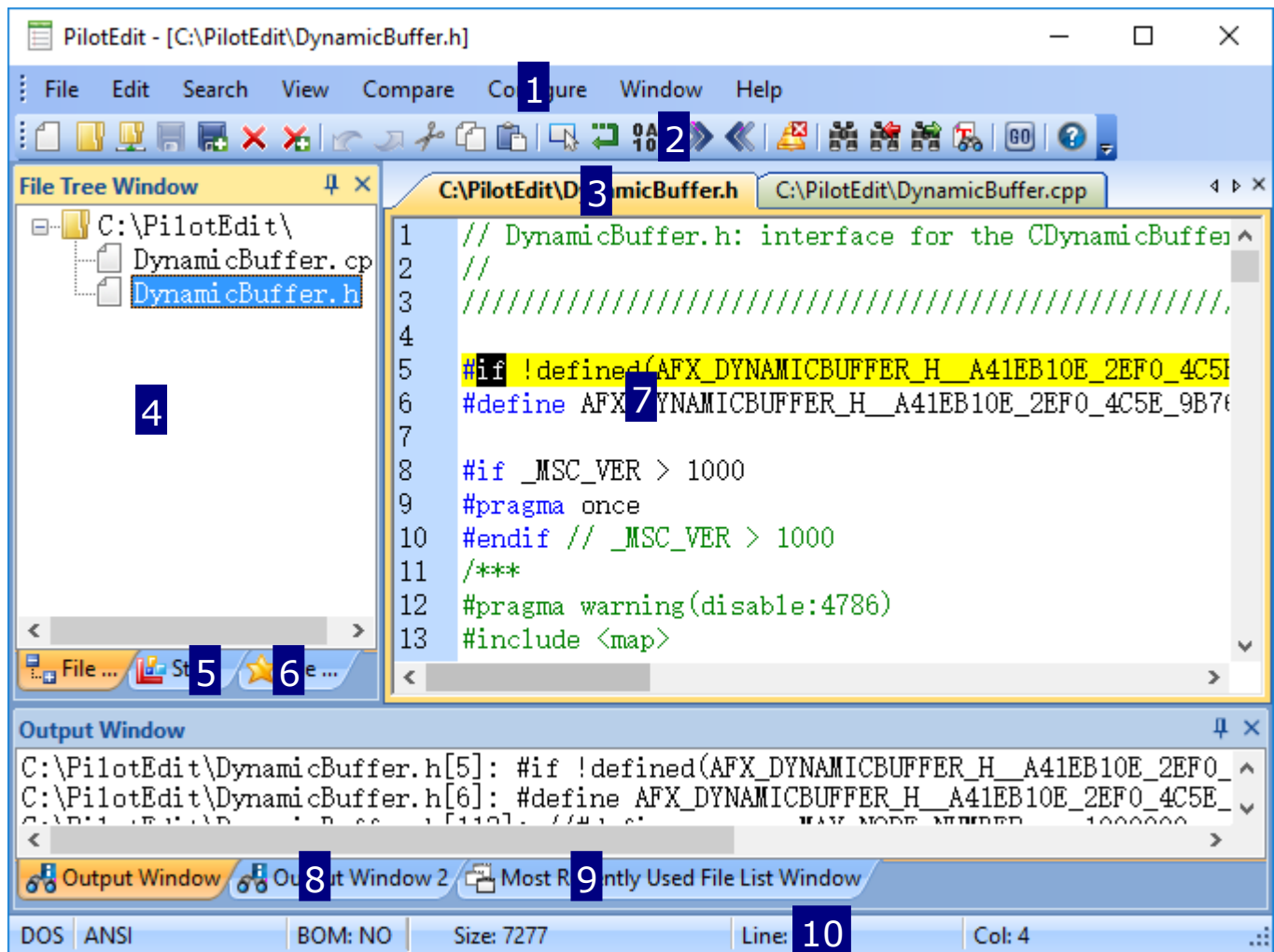
1.2. PilotEdit GUI

1.2.1. PilotEdit Window

PilotEdit Window



PilotEdit Lite Window



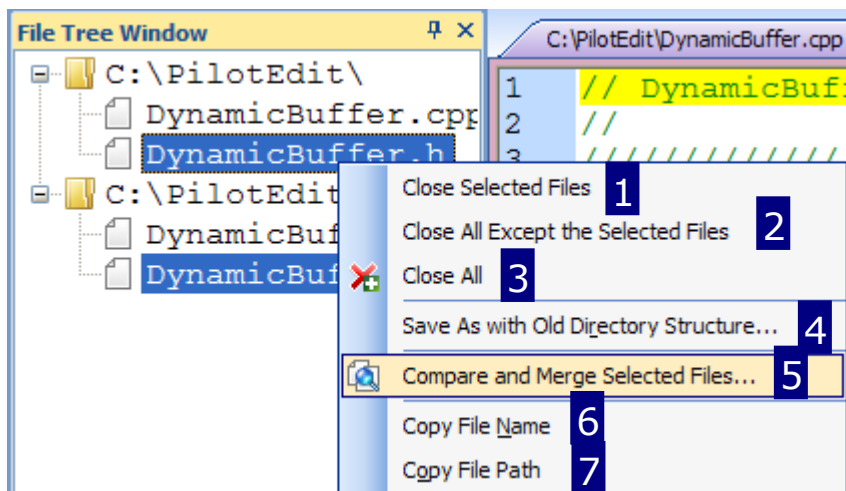
PilotEdit window includes 10 parts:

1. Menu.
2. Toolbar. Toolbar provides some frequently used functions from menu.
3. File Tab Window. File Tab List lists all the opened files.
4. File Tree Window. File Tree List lists all the opened files.
5. String Window. You can input a pre-defined string by just one click.
6. File Group/Bookmark Window. You can open multiple files by just one click.
7. File Edit Window.
8. Output Window. You can find the search and replacement result.
9. Most Recently Used Files List Window. You can find most recently used files here.
10. Status Window. File encoding, size and cursor position will be shown in the status window.

File Tree Window

You can activate a file by double clicking on the file name.

A pop-up menu will show up when you right click on the File Tree List.



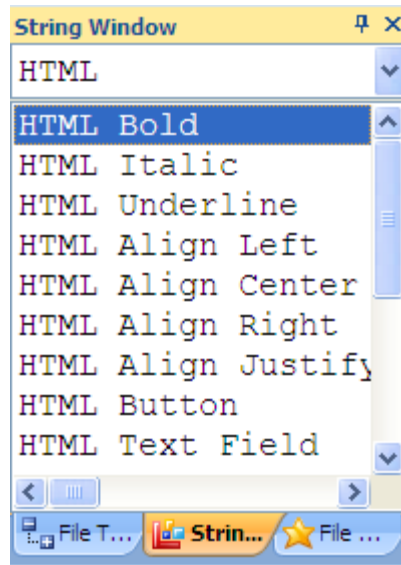
11. Close the selected files.
12. Close all the files except the selected.
13. Close all the opened files.
14. Save the selected files as new files with the old directory structure.
15. Compare the selected files. *
16. Copy the selected file names.
17. Copy the selected file paths.

String Window

Double click on a string in the string list will insert the pre-defined string in the active file.

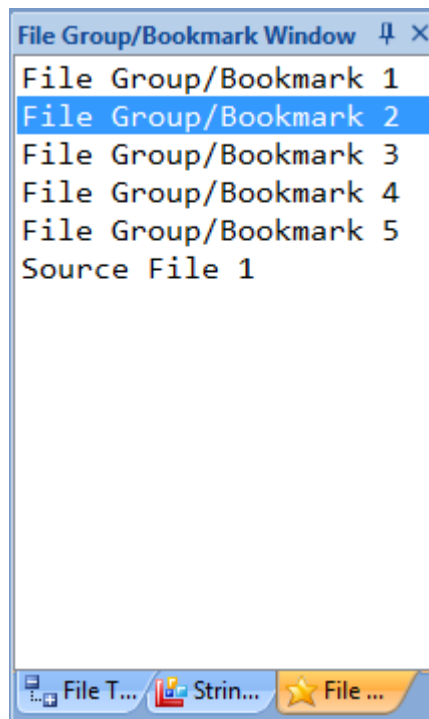
You can also run a windows command or search for a regular expression by right clicking on the string table.

 You can find more information about PilotEdit String Window from the following links:
[Examples about String Table.](#)



File Group/Bookmark Window

Double click on a file group/bookmark will open the files listed in the file group/bookmark.

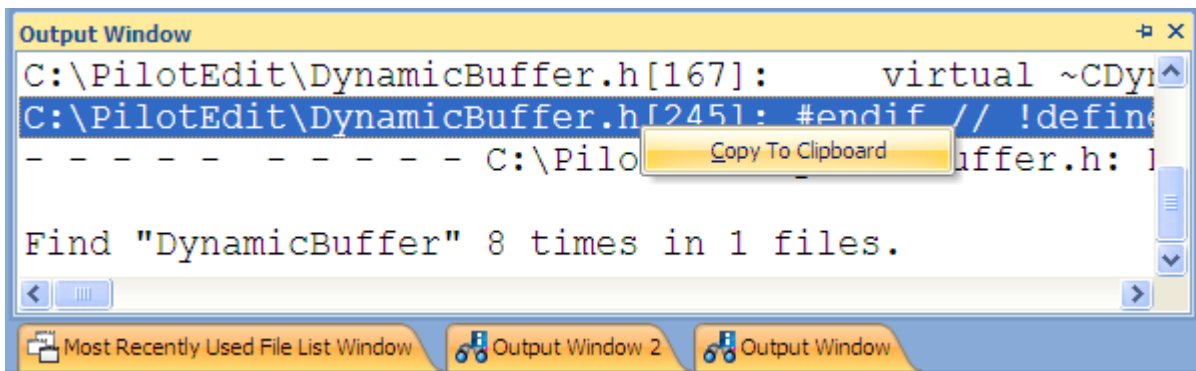


Output Window

You can get the search and replace result in this window. You can go to a found string by double-clicking on it.

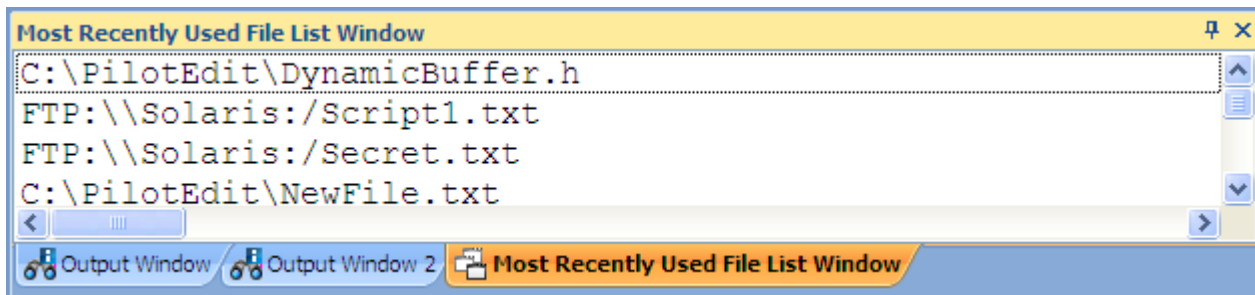
Pilotedit provides two output window. You can change output by clicking on the button >> in the search/replace dialog.

You will be able to copy search/replace result to clipboard by right clicking on the output window.

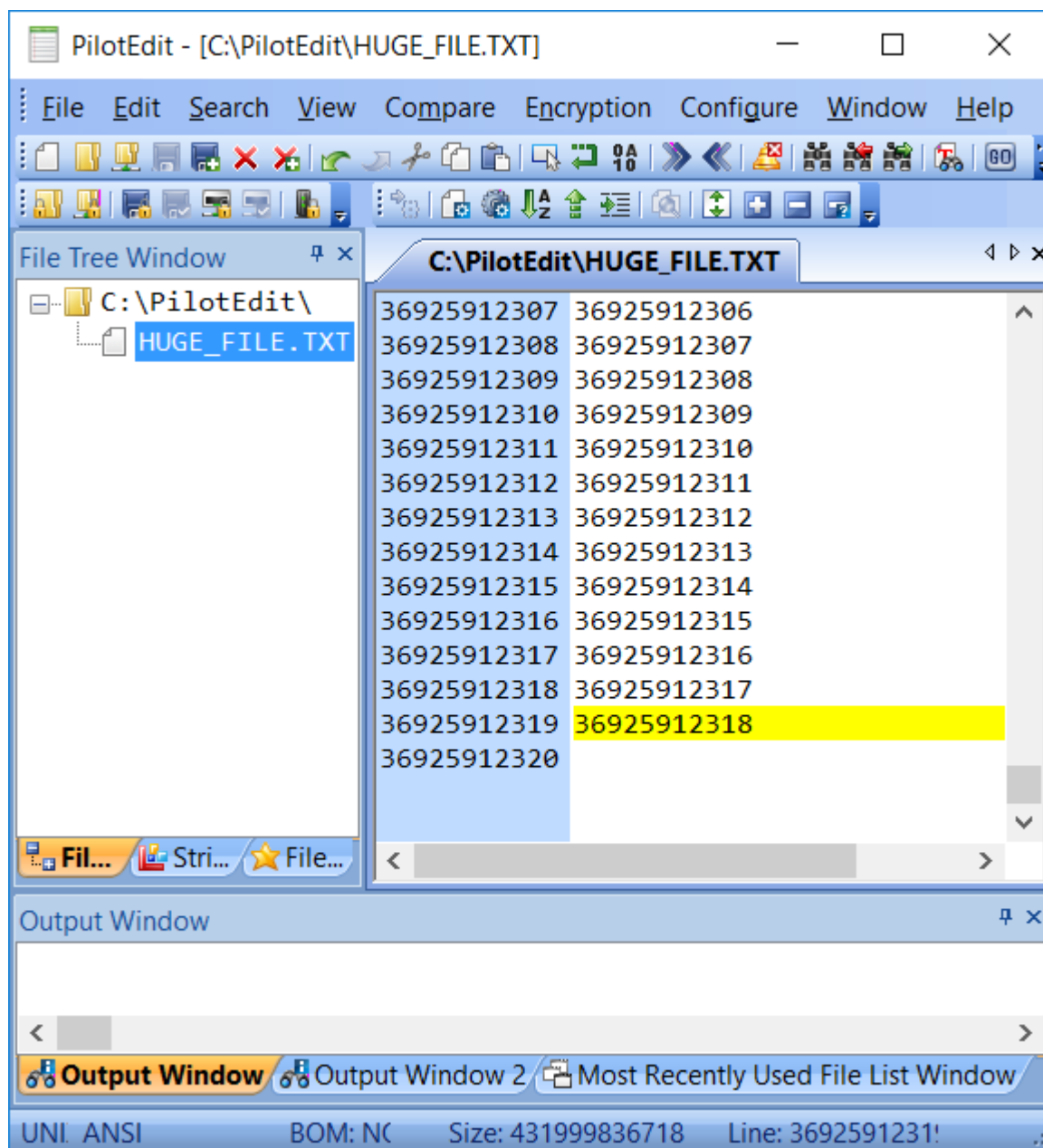


Most Recently Used File List Window

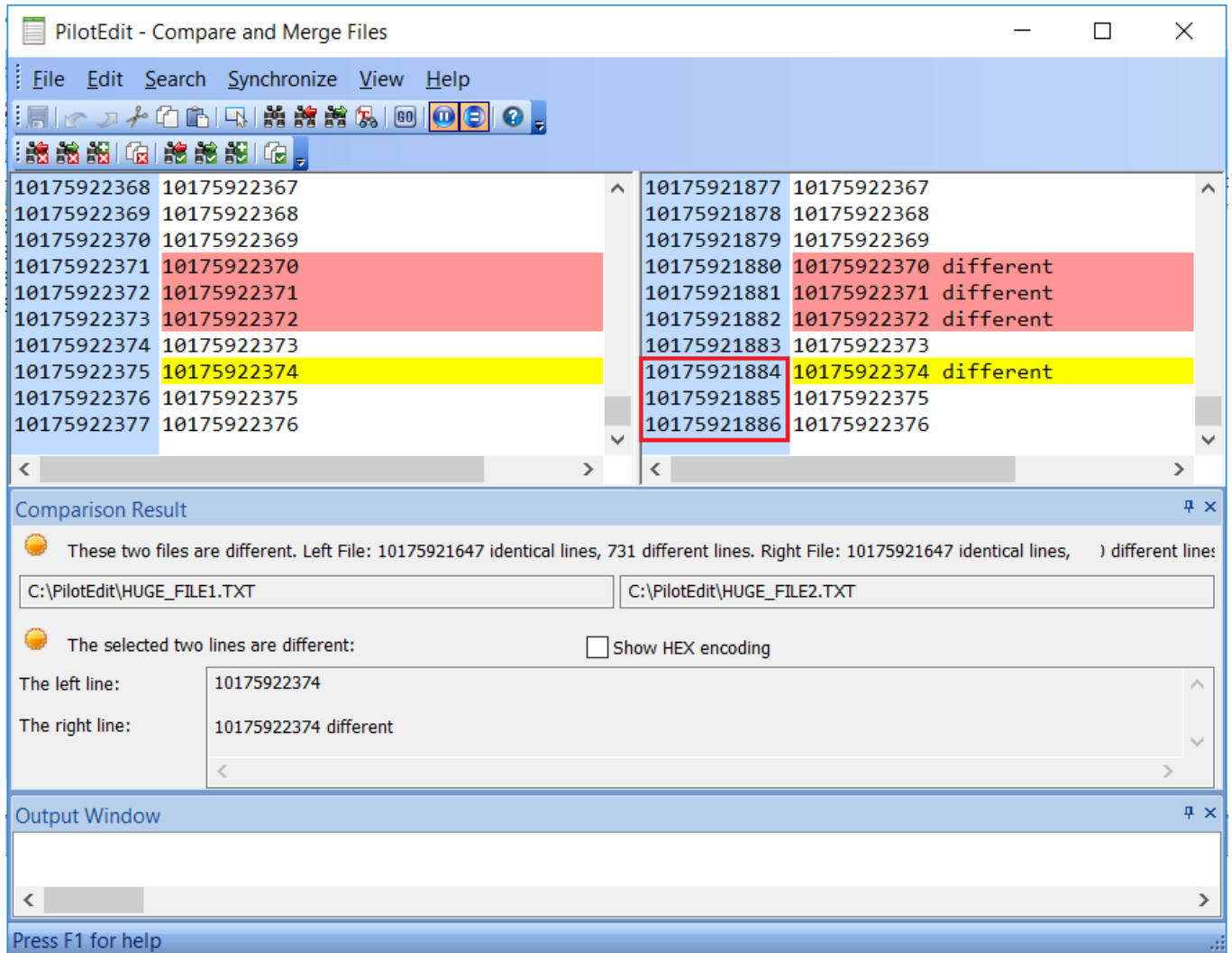
You can activate a recently opened file by double-clicking on its name in the Most Recently Used File List Window.



1.2.2. Edit Huge Files



1.2.3. Compare Huge Files



You can find more information about how to compare files with PilotEdit from the following links:
[Compare and Merge Files](#)

2. Install PilotEdit

2.1. System Requirements

PilotEdit supports the following operating systems.

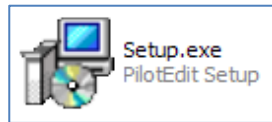
Windows 2000
Windows 2003
Windows XP
Windows Vista
Windows 2008
Windows 7
Windows 8
Windows 10

PilotEdit doesn't support the following operating systems.

Windows 95
Windows 98
Winnt

2.2. Install PilotEdit

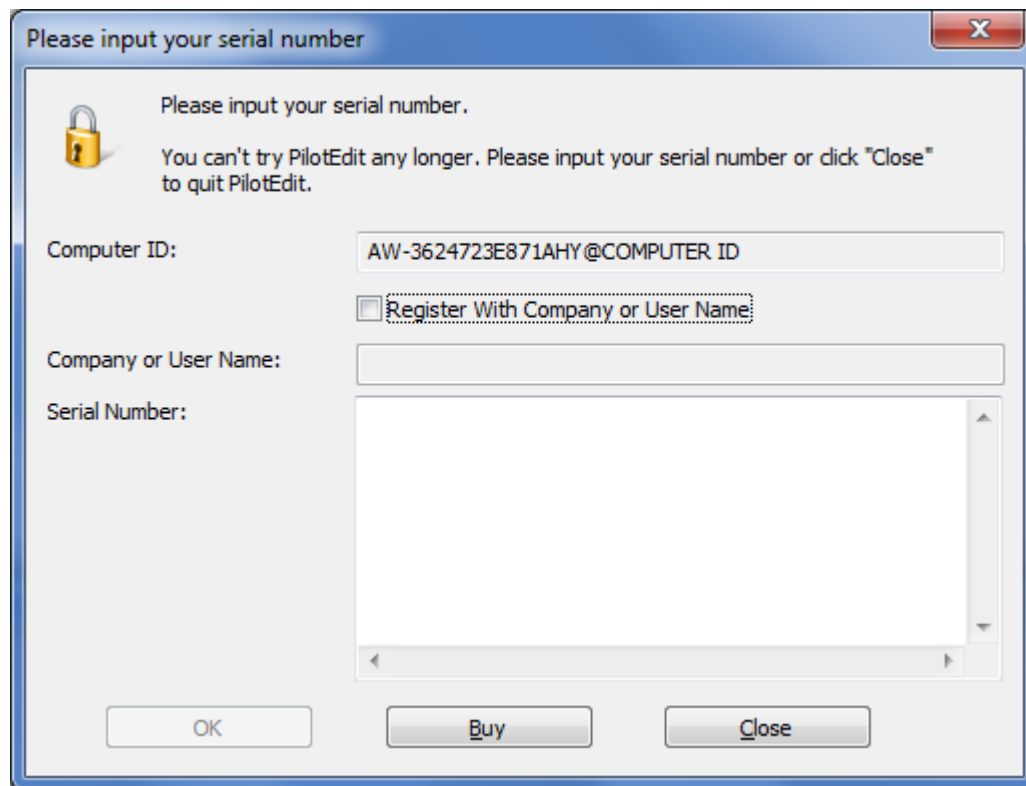
Click on **Setup.exe** and you will be able to install PilotEdit following the installation wizard.



2.3. Input Serial Number *

Run PilotEdit and you will see a window prompting you to input serial number. You will be able to try PilotEdit for 30 days.

Computer ID is generated from the first hard disk ID of your machine. A serial number for a computer ID works for all PilotEdit versions but only works for one machine. If you change your machine or your hard disk, you need to buy a new serial number.



2.4. Uninstall PilotEdit

From the Control Panel, open **Programs and Feature**, and then **Uninstall PilotEdit**.

2.5. Upgrade PilotEdit

1. Close any running PilotEdit.
2. Make a backup of PilotEdit configuration files in <User Application Directory>\PilotEdit.
3. Un-Install the PilotEdit of the old version.
4. Install PilotEdit of the new version.

5. Restore PilotEdit configuration files if you have made some changes to them, for example, String Table, File Groups/Bookmarks, File Types and FTP Addresses.

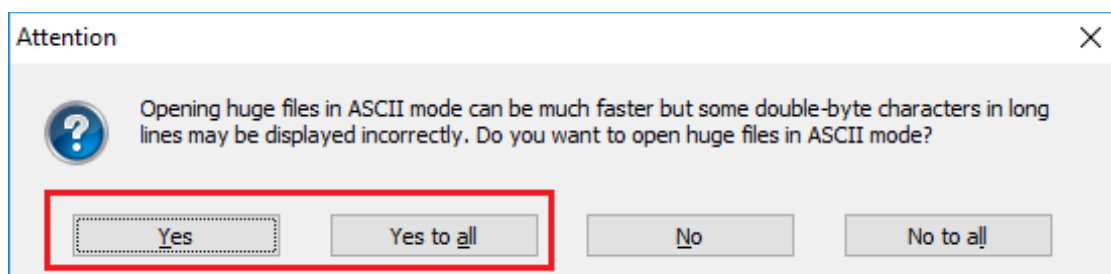
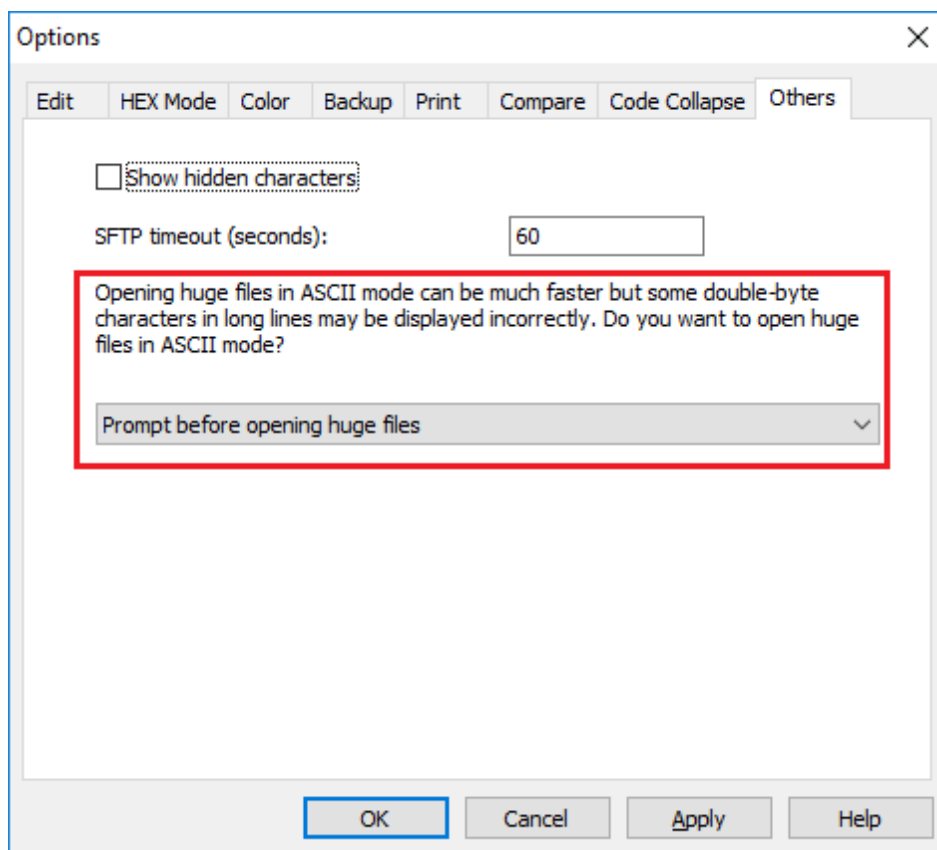
| | |
|--------------------------|---------------------------------------|
| CONFIG.DATA | Configuration file for PilotEdit. |
| STRING_TABLE.DATA | Configuration file for String Table. |
| FILE_GROUP.DATA | Configuration file for File Groups. |
| FILE_TYPE.DATA | Configuration file for File Types. |
| FTP_SERVER.XML | Configuration file for FTP Addresses. |

3. Quick Start

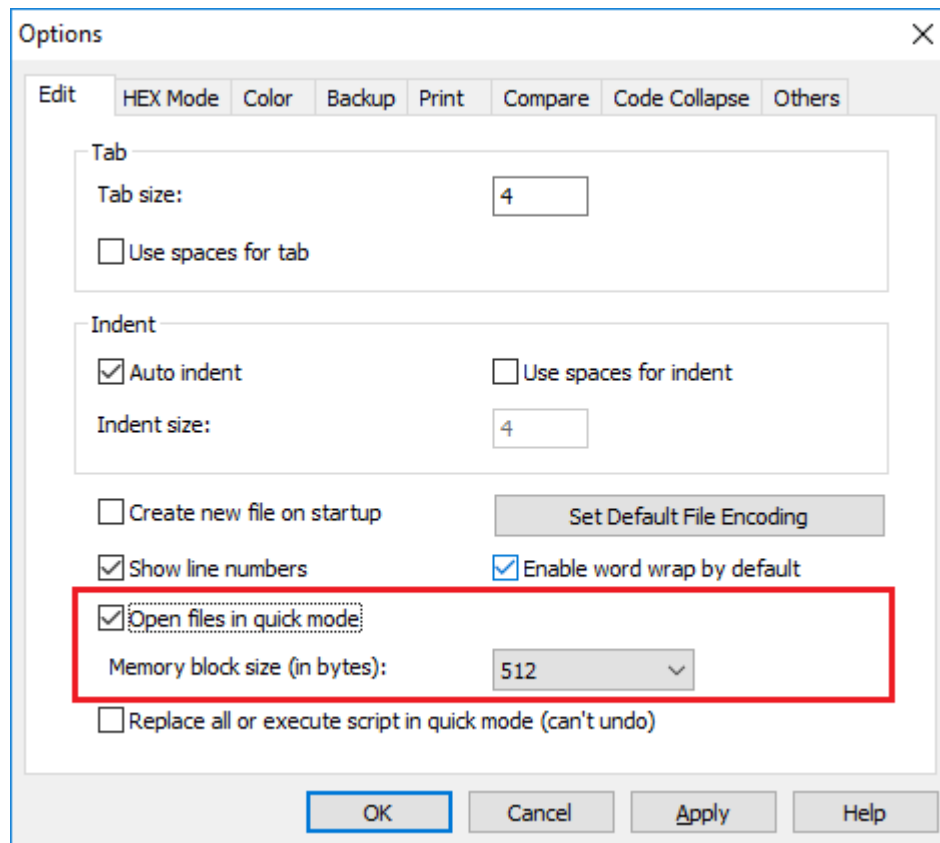
3.1. How to Edit Huge Files *

If you find PilotEdit too slow in opening huge files, you can change file extension to an unknown extension like "*.txt", "*.log" or even remove file extension to enable ASCII mode. ASCII mode is four times faster.

The only disadvantage of ASCII mode is double-byte characters in long lines (longer than 10000 characters) may be displayed incorrectly.

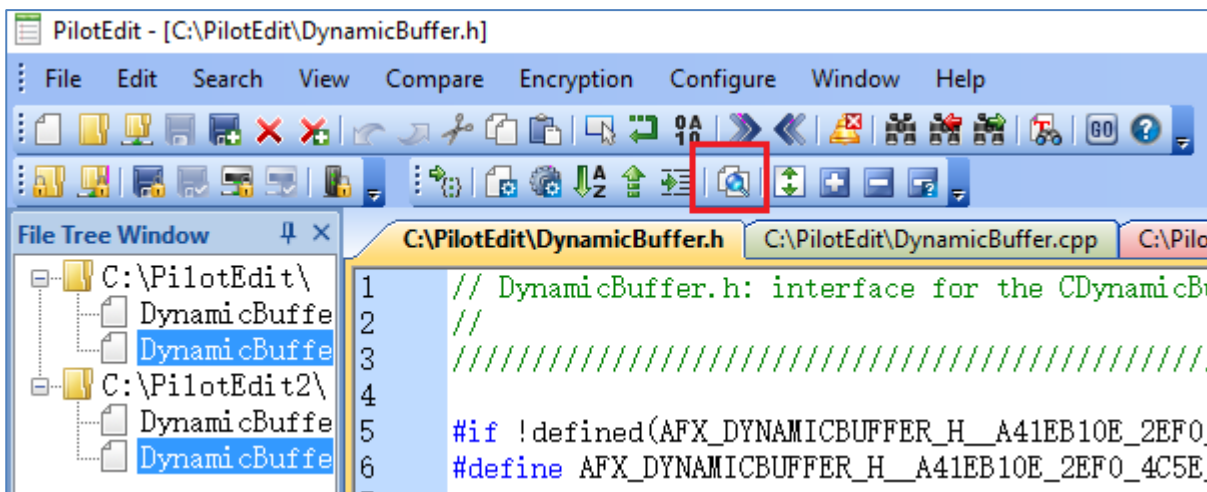


If you get out of memory error, you can change to PilotEdit x64. If you still get out of memory error with PilotEdit x64, you can enable quick mode. Quick mode should be enabled ONLY when you get out of memory error.

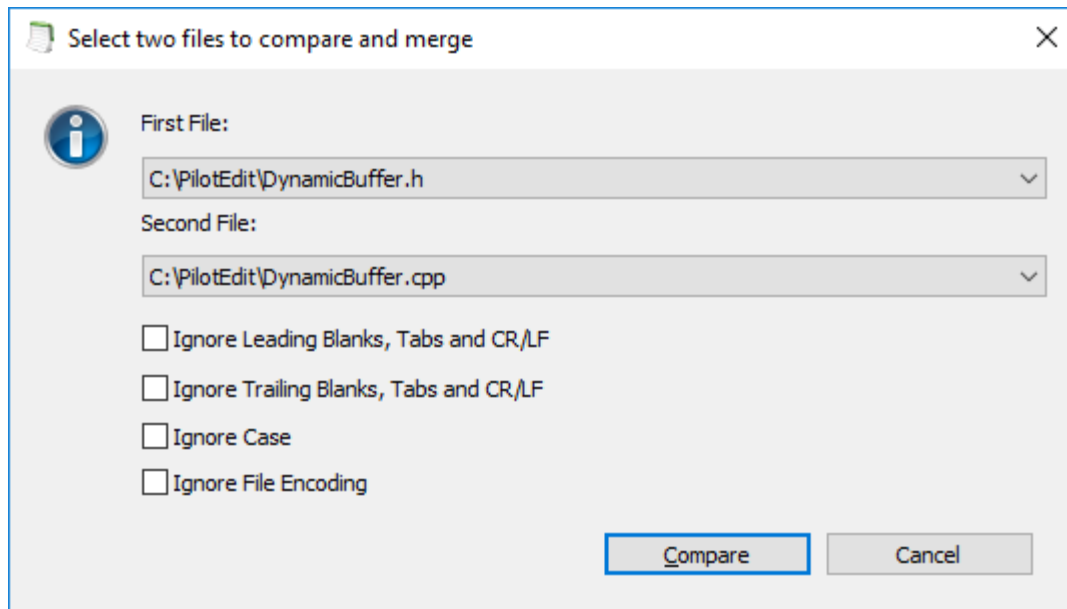


3.2. How to Compare and Merge Two Files *

Open the files and click the button below to compare and merge files.

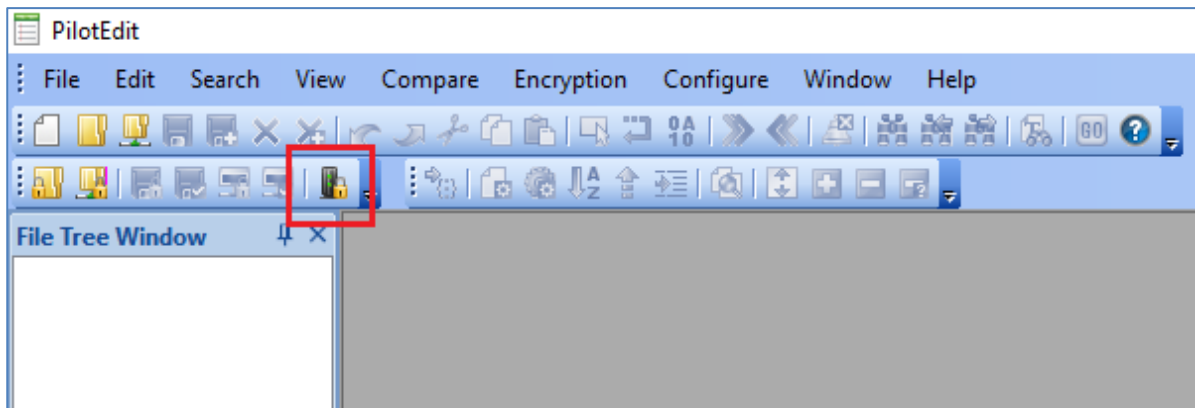


Select two files and click **Compare**.

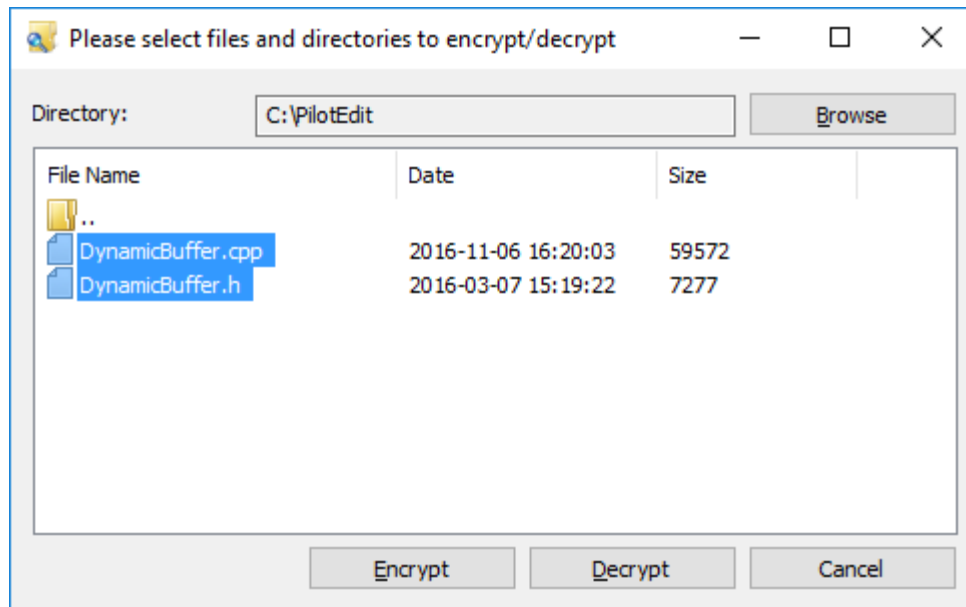


3.3. How to Encrypt and Decrypt Multiple Files and Directories *

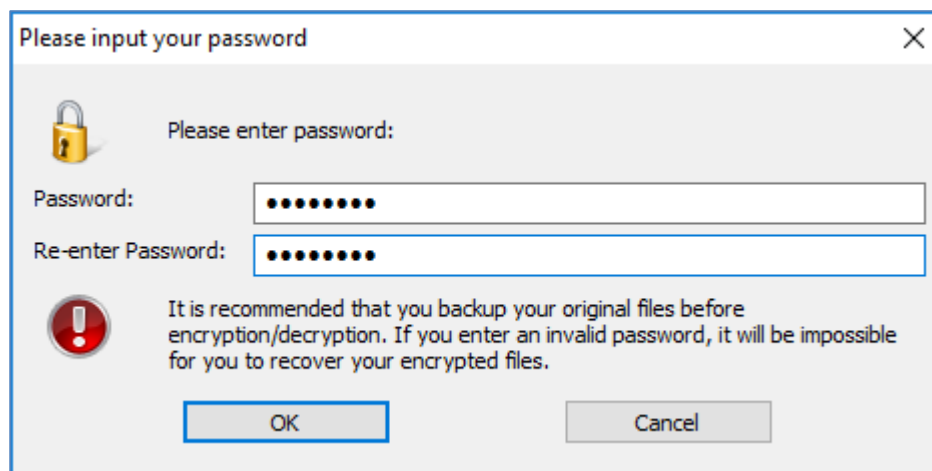
Click the button below.



Select the files and directories to encrypt or decrypt and click **Encrypt** or **Decrypt**.

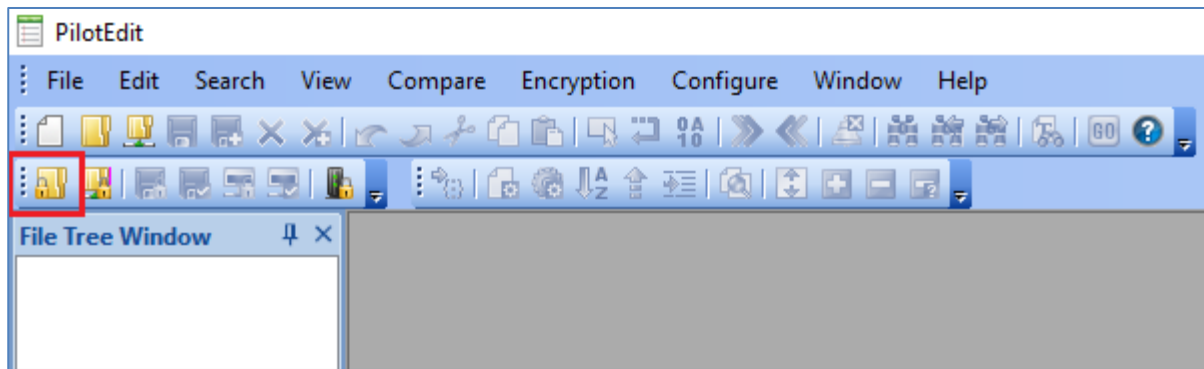


Input password and the selected files will be encrypted or decrypted.

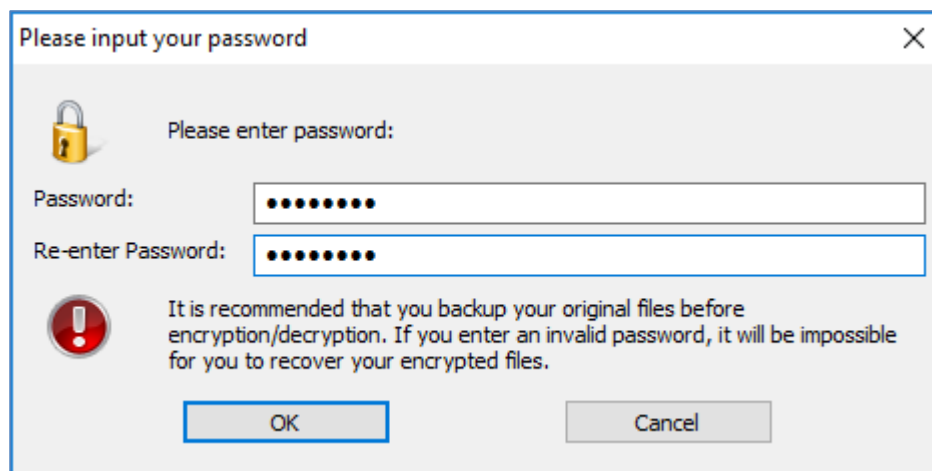


3.4. How to Open and Edit Encrypted Files *

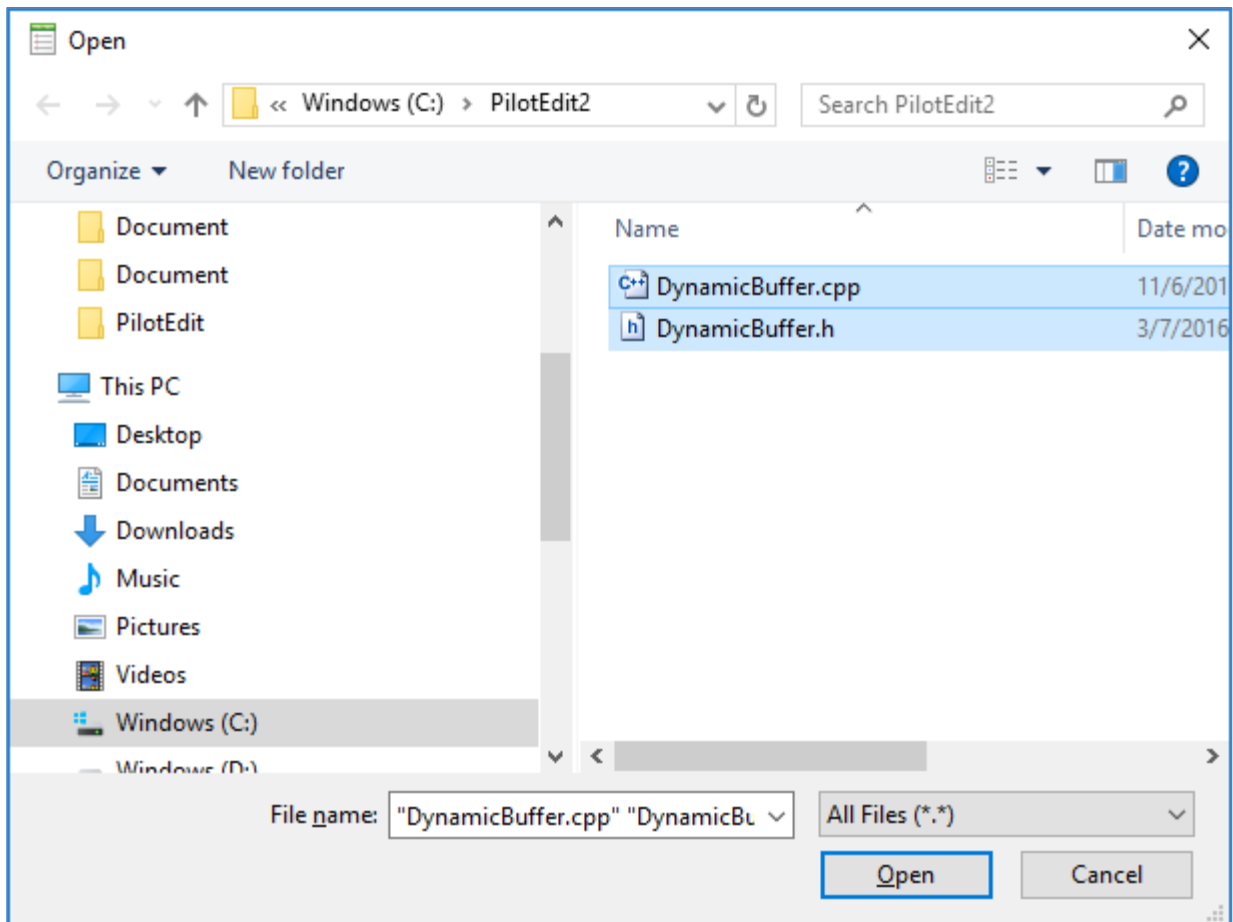
You can open an encrypted file and edit it transparently with PilotEdit. Click the button below.



Input password.

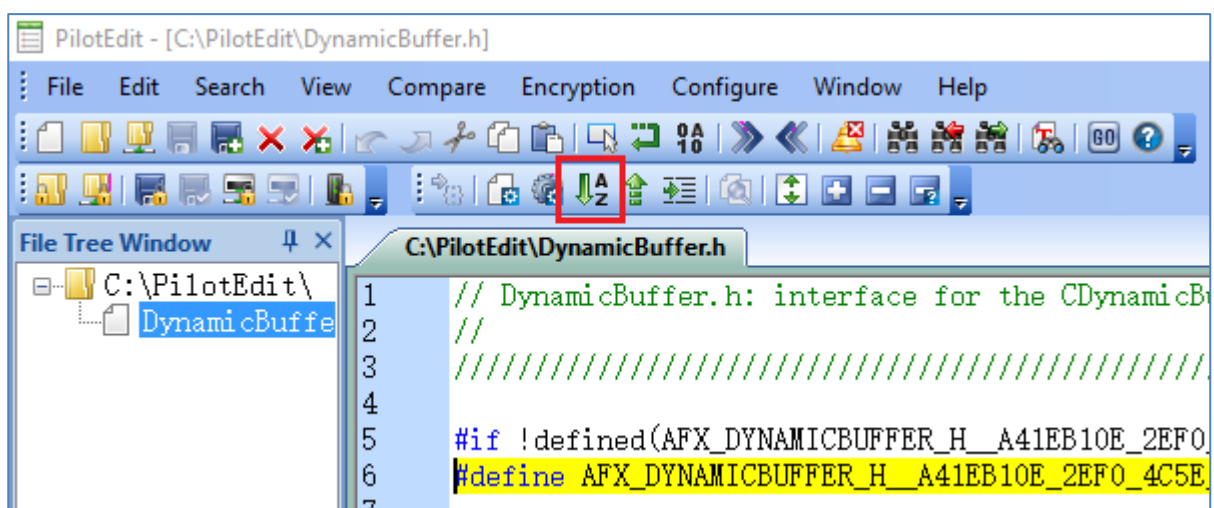


You will be able to open and edit the file transparently.

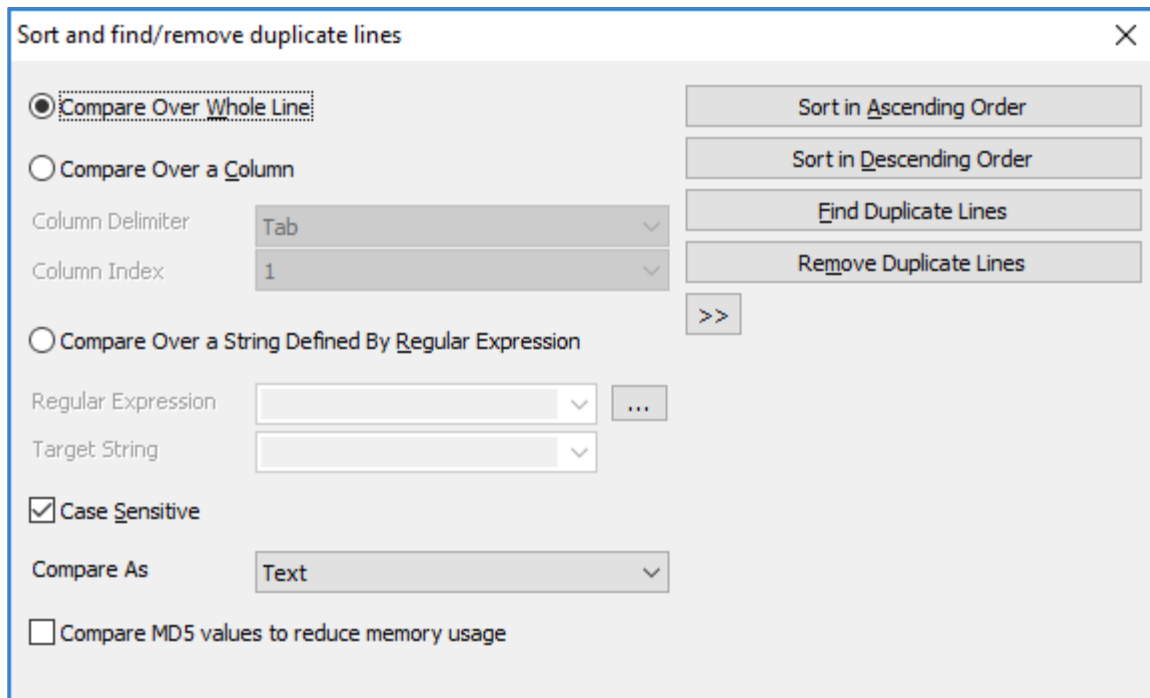


3.5. How to Sort or Find/Remove Duplicate Lines *

Click the button below.



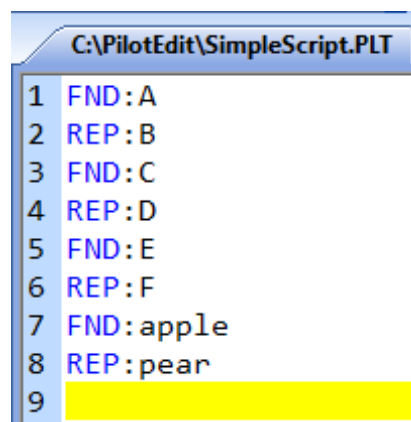
You will be able to sort and find/remove duplicate lines.



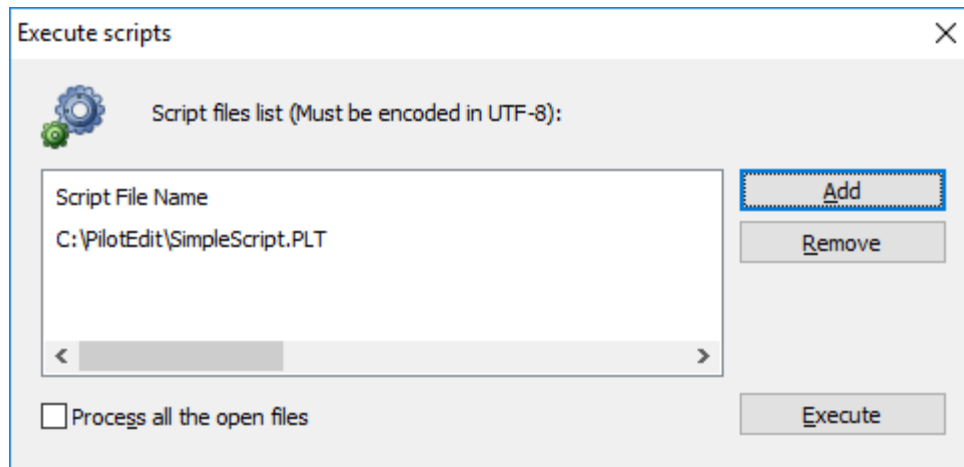
3.6. How to Replace Multiple Strings Automatically *

You can replace multiple strings automatically with PilotEdit script. For example, if you want to replace **A** to **B**, **C** to **D**, **E** to **F**, **apple** to **pear** automatically, you can create and run PilotEdit script as below.

1. Create a simple PilotEdit script to replace **A** to **B**, **C** to **D**, **E** to **F**, **apple** to **pear**.




2. Save this script as an **UTF-8** file. Open the target file, select the script file and execute it.



The text before and after running this script are listed below:

| Text before running this script | Text after running this script |
|--|---|
| Like many fruits, apples , bananas and pears contain important vitamins. All three provide vitamins A and C , with a medium-sized fruit containing between 2 and 5 micrograms of vitamin A and between 8 and 10 milligrams of vitamin C . | Like many fruits, pears, bananas and pears contain important vitamins. Bll three provide vitamins B and D, with a medium-sized fruit containing between 2 and 5 micrograms of vitamin B and between 8 and 10 milligrams of vitamin D. |

 You can find more information about PilotEdit Script from the following links:
[PilotEdit Script](#)

4. PilotEdit Functions

4.1. File

4.1.1. New File

Selecting menu **File**, then **New** or pressing **Ctrl + N** will create a new file in PilotEdit.

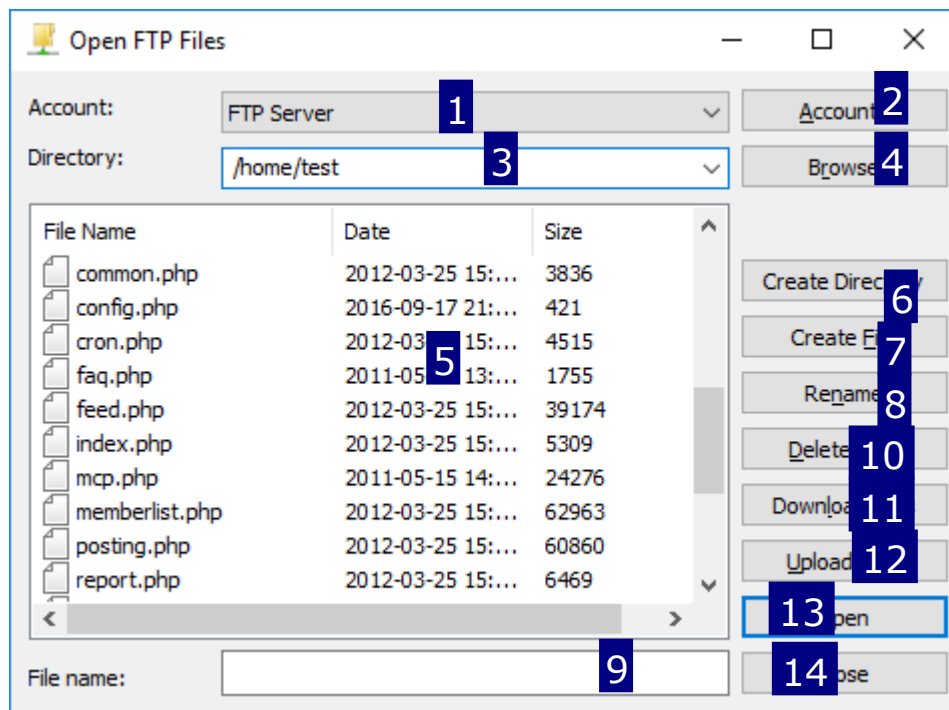
4.1.2. Open Files

You can select menu **File**, then **Open** or press **Ctrl + O** to open files on local host. Files may also be opened from windows explorer in one of the following two ways:

1. Drag and drop them to PilotEdit. This is the recommended method.
2. Right click on the file and select "PilotEdit". This method may be a little slow due to the implementation of PilotEdit.

4.1.3. Open FTP Files

You can select menu **File**, then **Open From FTP...** to open FTP files. You can download/upload FTP files more than 4 GB.



1. Select FTP server.
2. Click **Account** to edit FTP server information.
3. Select or enter FTP directory.
4. Click **Browse** to list the files and directories in the selected directory.
5. File and directory list.
6. Create a new directory in the selected directory.
7. Create a new file in the selected directory.

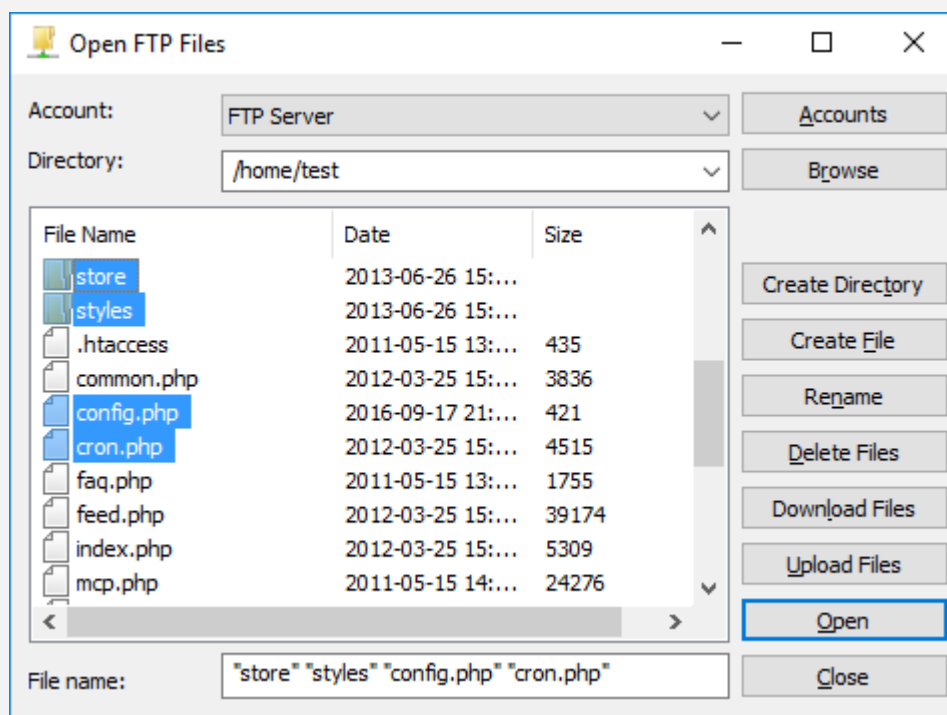
8. Change the name of the selected file.
9. Selected file names.
10. Delete the selected files.
11. Download the selected files and directories to local host.
12. Select local files and directories and upload them to the FTP server.
13. Open the selected files.
14. Close FTP window.

 PilotEdit 1.6 and above support FTP files larger than 4GB.

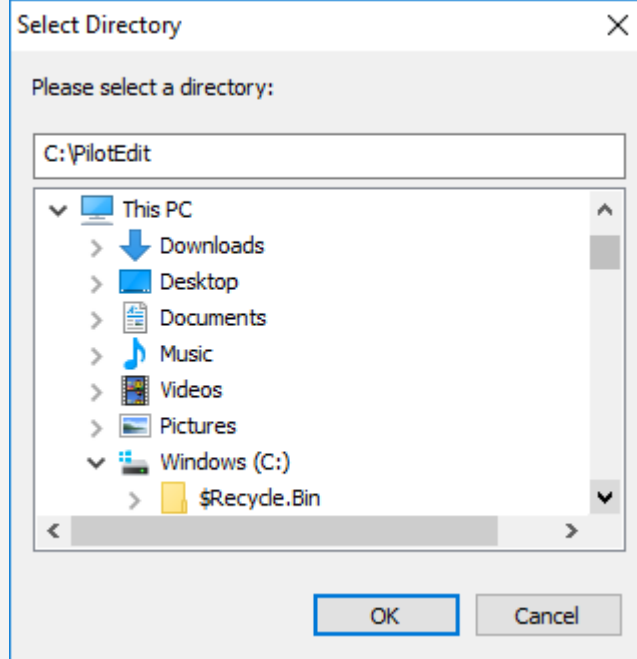
Example 1:

This example demonstrates how to download FTP files and directories through PilotEdit.

1. Select files and directories.



2. Click **Download Files** and select a local directory.

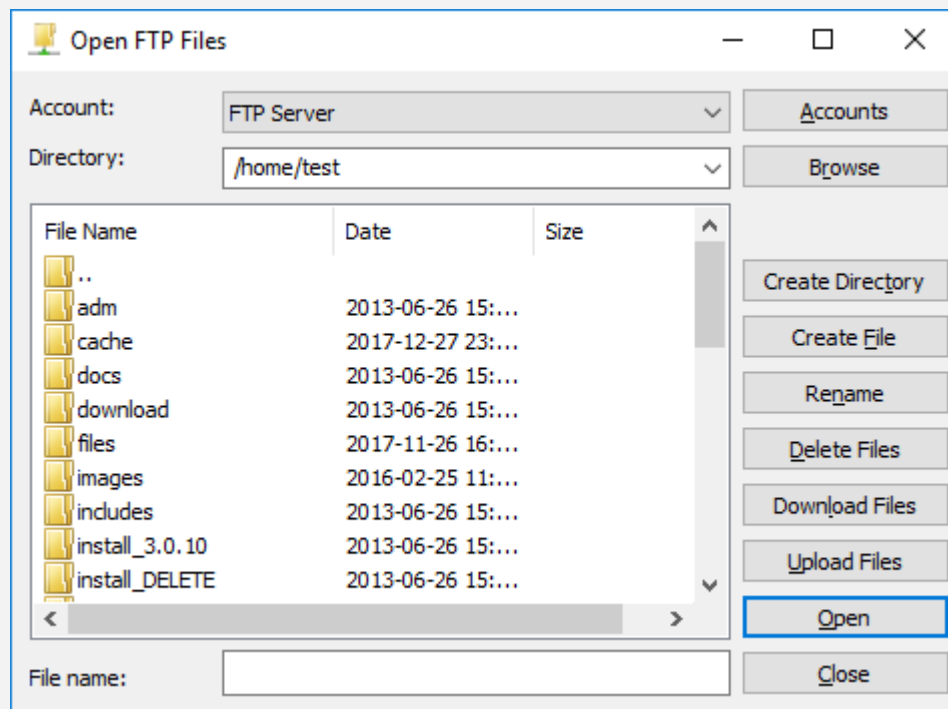


3. Click **OK** and the selected files and directories will be downloaded to local host.

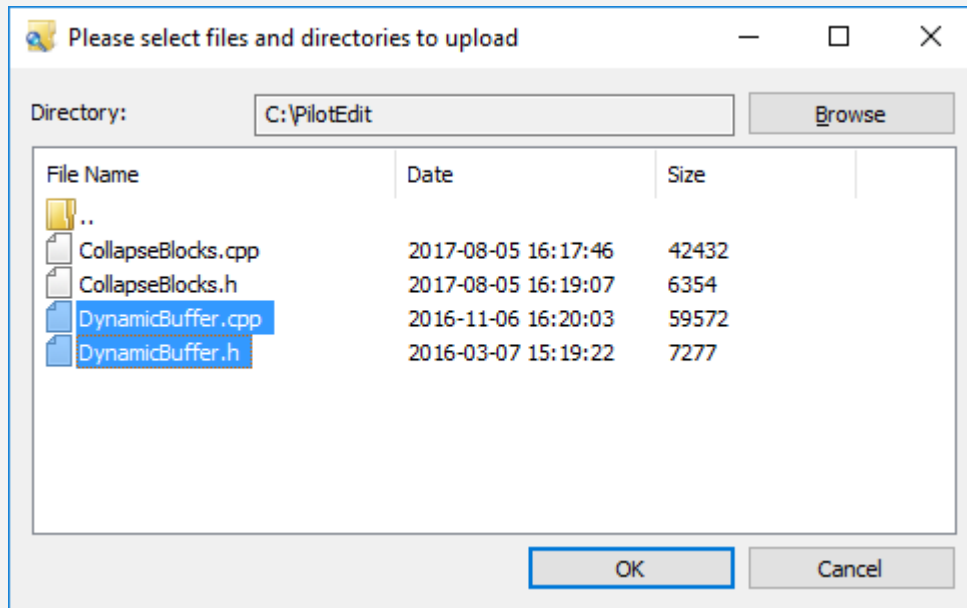
Example 2:

This example will demonstrate how to upload FTP files and directories through PilotEdit.

1. Select FTP directory.




2. Click **Upload Files** and select the files and directories to upload.



3. Click **OK** and the selected files and directories will be uploaded to the FTP server.

4.1.4. Save File

Selecting menu **File**, then **Save** or pressing **Ctrl + S** will save the active file in PilotEdit.

 When you are saving a big file, PilotEdit will first save a copy of the old file, then write the updated content into another temporary file, so that the free space size you need is double of the file size. For example, if you are saving a file of 2GB, you must have at least 4GB free space in your disk.

4.1.5. Save All

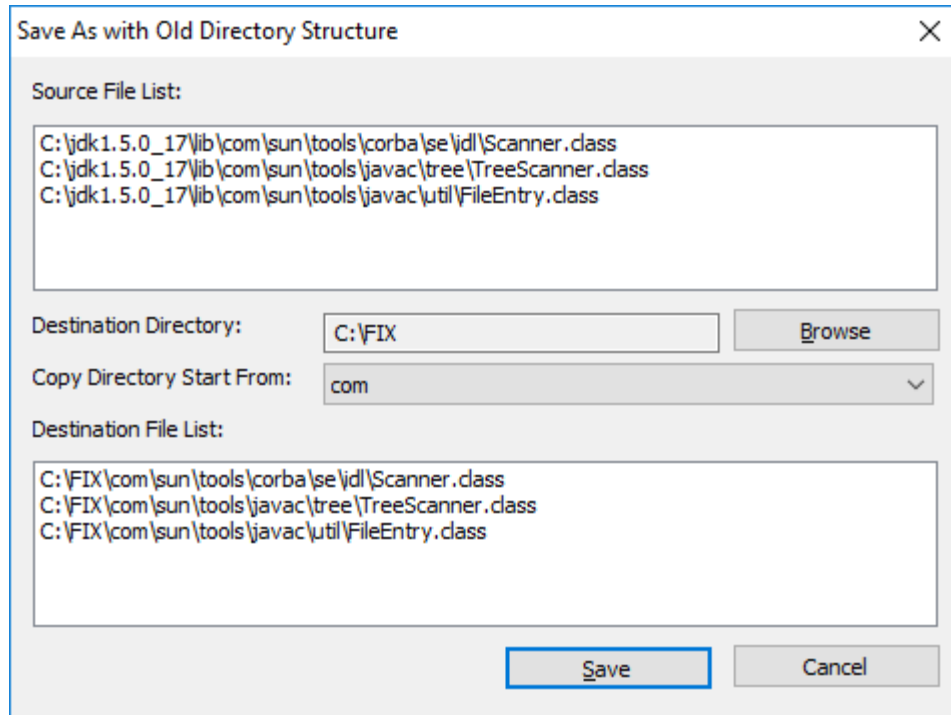
You can save all the opened files by selecting menu item **File**, then **Save All**.

4.1.6. Save As

You can save the active file to another position by selecting menu **File**, then **Save As....**

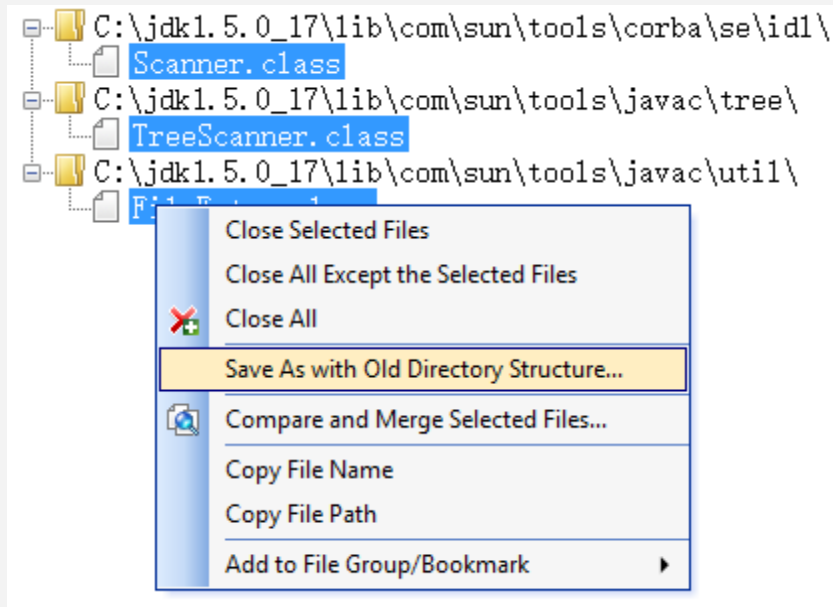
4.1.7. Save As With Old Directory Structure

You can save the selected files as new files with old directory structure by selecting menu **File**, then **Save As With Old Directory Structure....**

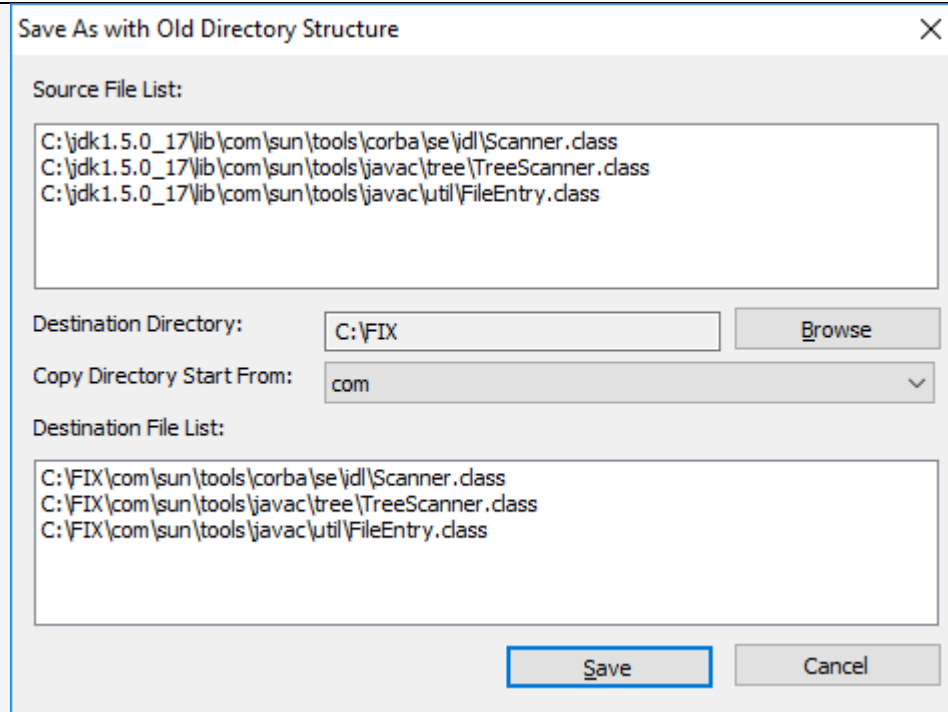
**Example 1:**

This example shows how to copy the selected the files to a new place with old directory structure.

1. Right click on the selected the files and choose menu item **Save As With Old Directory Structure....**



2. Select the destination directory and the source directory to copy from.



3. Click **Save** and the files will be saved to the new directory with the old directory structure **C:\FIX\com\sun\...**

4.1.8. Save As FTP File

You can save the active file to an FTP server by selecting menu **File**, then **Save As FTP File...**
A file opened from one FTP server can also be saved to another FTP server.

4.1.9. Print

You can print the active document by selecting menu **File**, then **Print...**

4.1.10. Print Preview

You can preview the printing result by selecting menu **File**, then **Print Preview...**

4.1.11. Print Setup

You can change the settings of printing by selecting menu **File**, then **Print Setup...**

4.1.12. Print Page Setup

You can change the page settings of printing by selecting menu **File**, then **Print Page Setup...**

4.1.13. Close File

Selecting menu **File**, then **Close** will close the active file in PilotEdit.

4.1.14. Close All Files

Selecting menu **File**, then **Close All** will close all the files opened in PilotEdit.

4.1.15. Clean Recent File List

Selecting menu **File**, then **Clean Recent File List** will clean recently opened file list in PilotEdit menu.

4.2. Edit

4.2.1. Select Line

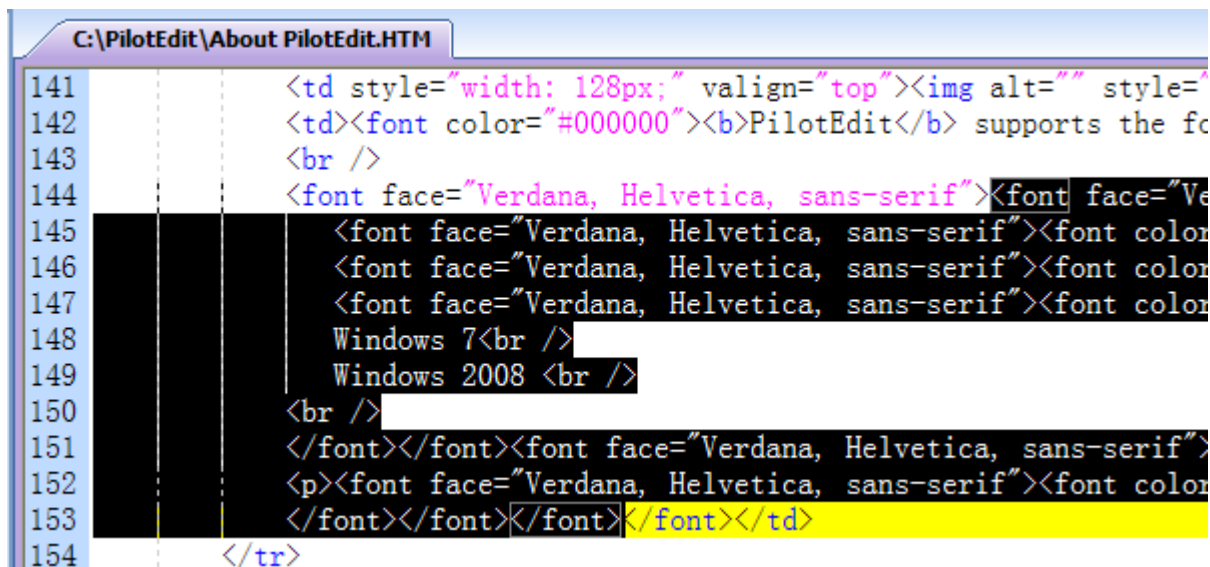
Selecting menu **Edit**, then **Select Line** will select the entire line at cursor's position in current file.

4.2.2. Select All

Selecting menu **Edit**, then **Select All** or pressing **Ctrl + A** will select all the content in current file.

4.2.3. Select Text Between Start Tag and End Tag *

Selecting menu **Edit**, then **Select Text Between Start Tag and End Tag** will select all the text between the start tag and end tag in current file.



⚠ The tags for non-HTML files are defined in menu **Configure**, then **File Type...**, then **Advanced...**, then **Indent Tags:**.

⚠ The tags for HTML are defined in variable **HTML_TAGS** in <User Application Dir>\PilotEdit\CONFIG.DATA.

4.2.4. Undo

You can select menu **Edit**, then **Undo** or press **Ctrl + Z** to undo the last operation in current file. With PilotEdit, you may undo any operations as long as you don't close or re-open the file.

4.2.5. Redo

You can select menu **Edit**, then **Redo** or press **Ctrl + Y** to redo the previously undone action in current file.

With PilotEdit, you may redo any operations as long as you don't close or re-open the file.

4.2.6. Cut

Selecting menu **Edit**, then **Cut** or pressing **Ctrl + X** will cut the selected content in current file. In text mode, the text cut will be translated into ANSI DOS strings before put into clipboard.

4.2.7. Copy

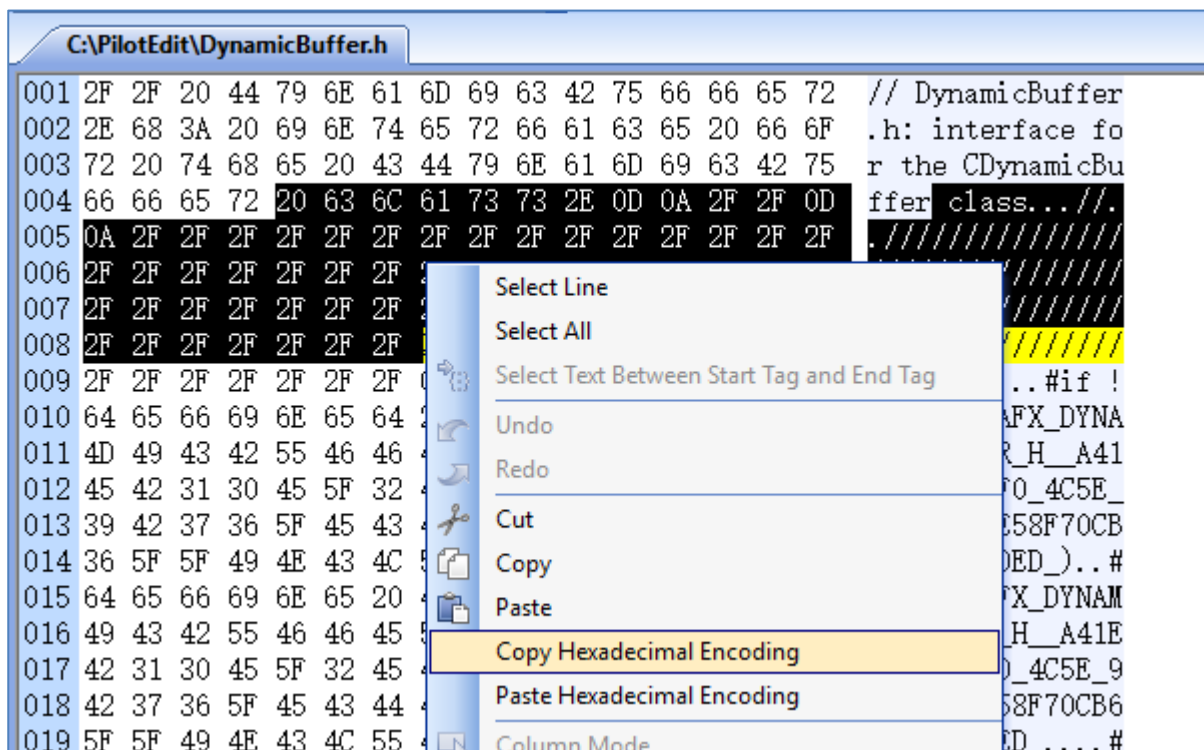
Selecting menu **Edit**, then **Copy** or pressing **Ctrl + C** will copy the selected content in current file. In text mode, the text copied will be translated into ANSI DOS strings before put into clipboard.

4.2.8. Copy Hexadecimal Encoding

You can copy the hexadecimal encoding in HEX mode by selecting items in the current file, right click on them, then select **Copy Hexadecimal Encoding**. The text string copied to the clipboard may be pasted in text mode.

The string copied to the clipboard will be something like:

```
20 63 6C 61 73 73 2E 0D 0A 2F 2F 0D 0A 2F 2F 2F
2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F
2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F
2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F 2F
2F 2F 2F
```



4.2.9. Copy File Name

Selecting menu item **Edit**, then **Copy File Name** will copy the active file name on the clipboard.

4.2.10. Copy File Path

Selecting menu item **Edit**, then **Copy File Path** will copy the active file path on the clipboard.

4.2.11. Paste

Selecting menu **Edit**, then **Paste** or pressing **Ctrl + V** will paste data into current file. Data from clipboard will first be translated to the encoding of current text file before pasted. Data copied from PilotEdit HEX mode can only be pasted into PilotEdit HEX mode.

Example 1:

PilotEdit will adjust text encoding automatically when copy/paste in TEXT mode.

1. Open an UTF-8 UNIX file and an UTF-16 DOS file.
2. Copy some text from the UTF-8 UNIX file.
3. The text from UTF-8 file will be changed into ANSI DOS text and put into clipboard.
4. Paste it to the UTF-16 DOS file.
5. The text from clipboard will be changed into UTF-16 DOS text and pasted into the UTF-16 DOS file.

Example 2:

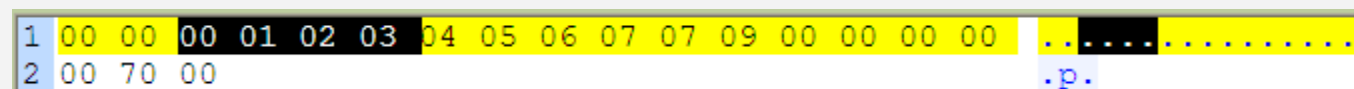
You can make an EXACT copy in HEX mode.

1. Open an UTF-8 UNIX file and an ANSI DOS file.
2. Change the two files into HEX mode.
3. Delete all the data from the ANSI DOS file.
4. Copy all data from the UTF-8 UNIX file and paste it to the ANSI DOS file.

In this way, the two file will be identical. PilotEdit will check the encoding of the two files automatically after you change back to text mode.

Example 3:

In HEX mode, you can copy/paste any special characters even those not supported in text mode like 00 01 02.



```

1 00 00 00 01 02 03 04 05 06 07 07 09 00 00 00 00 .....
2 00 70 00 .p.

```

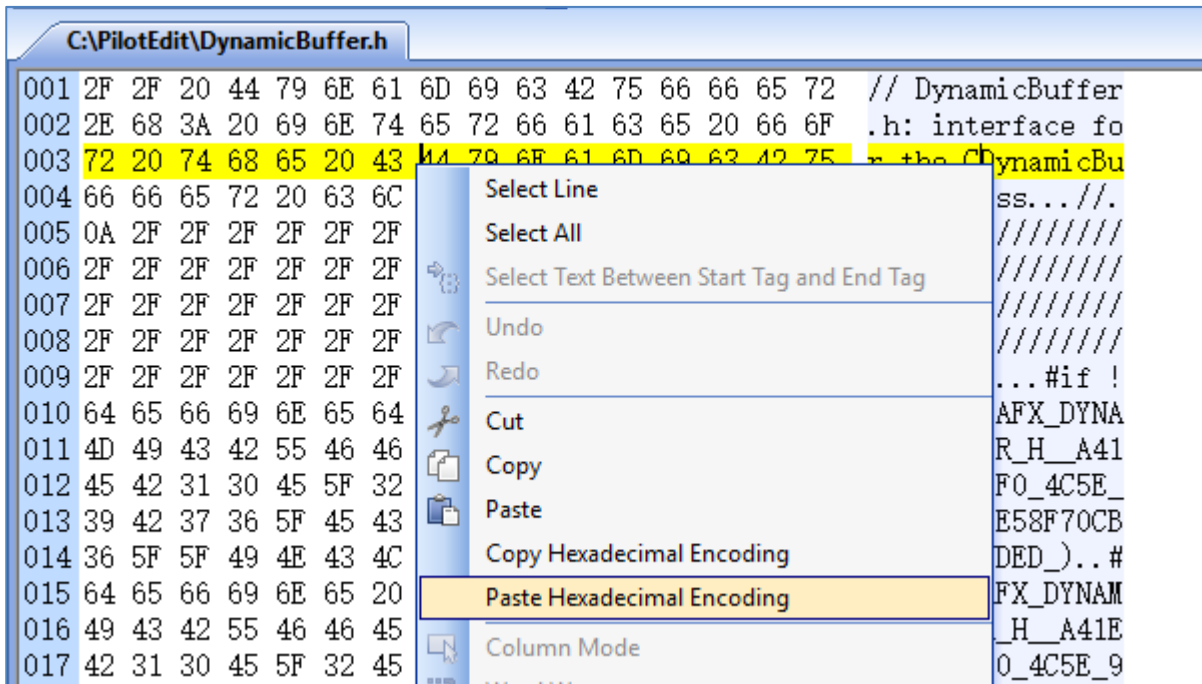
4.2.12. Paste Hexadecimal Encoding

You can paste the hexadecimal encoding in HEX mode by right clicking in the current file, then select **Paste Hexadecimal Encoding**. The strings copied in the text mode may be pasted in HEX mode in this way.

The string pasted from the clipboard should be something like:

```
6E 74 65 72 66 61 63 65 20 66
```

```
6F 72 20 74 68 65 20 43 44 79
6E 61 6D 69 63 42
```



4.2.13. Column Mode

Selecting menu **Edit**, then **Column Mode** or pressing **Ctrl + E** (or **Alt + C**) will change the active file into column mode.

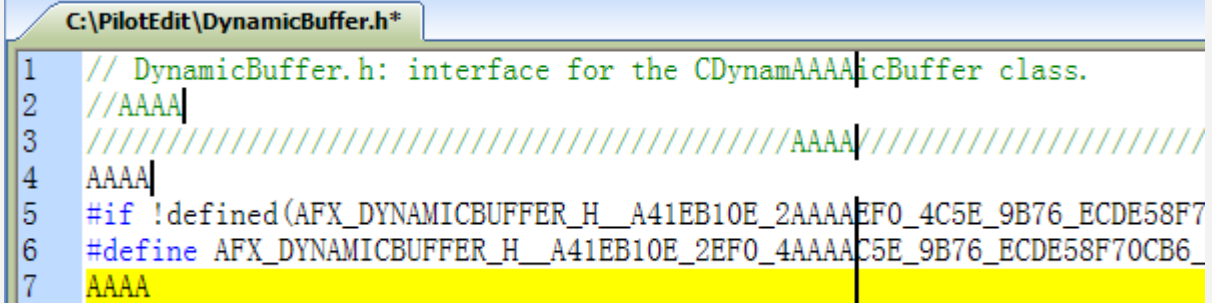
In column mode,

1. If you select multiple lines and input some characters through keyboard, these characters will be inserted into all the selected lines.
2. If you selected multiple lines and paste a single-line string into current file, this string will be inserted into all the selected lines.
3. If you paste multiple-line text into current file, the text will be inserted line by line.

Example 1:

Any character input through keyboard will be inserted into each selected line.

1. Change to column mode in PilotEdit and select several lines.
2. Input some characters through keyboard.
3. These characters will be inserted into each line.



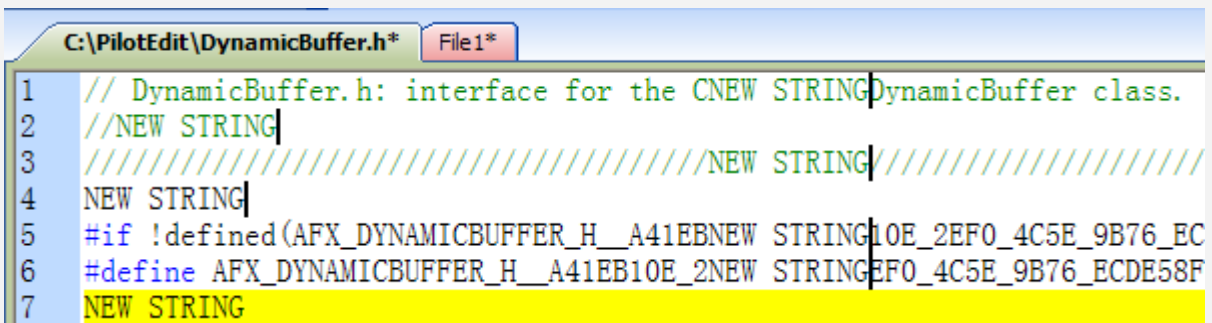
```

C:\PilotEdit\DynamicBuffer.h*
1 // DynamicBuffer.h: interface for the CDynamAAAAicBuffer class.
2 //AAAA|
3 ///////////////////////////////////////////////////AAAA|////////////////////////////////////
4 AAAA|
5 #if !defined(AFX_DYNAMICBUFFER_H__A41EB10E_2AAAAEF0_4C5E_9B76_ECDE58F7
6 #define AFX_DYNAMICBUFFER_H__A41EB10E_2EF0_4AAAAC5E_9B76_ECDE58F70CB6_
7 AAAA
  
```

Example 2:

A one-line string pasted from clipboard will be inserted into each line.

1. Copy one line text into clipboard.
2. Change to column mode in PilotEdit and select several lines.
3. The string from clipboard will be inserted into each line.



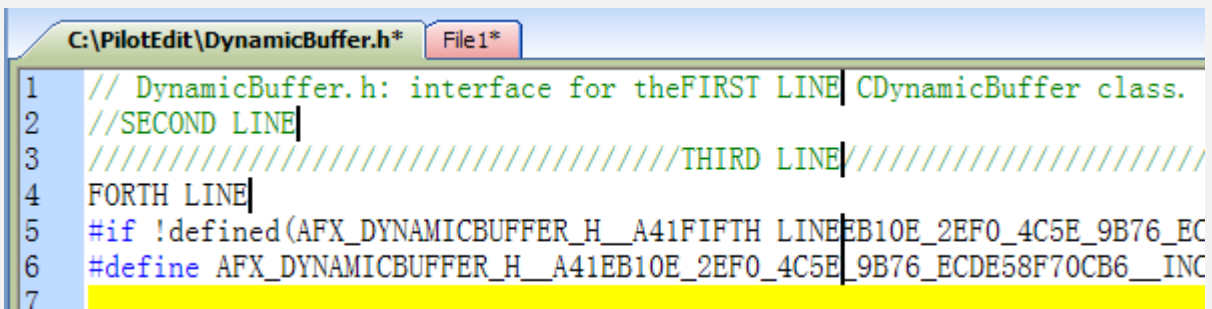
```

C:\PilotEdit\DynamicBuffer.h* File1*
1 // DynamicBuffer.h: interface for the CNEW STRINGDynamicBuffer class.
2 //NEW STRING|
3 ///////////////////////////////////////////////////NEW STRING|////////////////////////////////////
4 NEW STRING|
5 #if !defined(AFX_DYNAMICBUFFER_H__A41EBNEW STRING10E_2EF0_4C5E_9B76_EC
6 #define AFX_DYNAMICBUFFER_H__A41EB10E_2NEW STRINGEF0_4C5E_9B76_ECDE58F
7 NEW STRING
  
```

Example 3:

A multiple-lines string pasted from clipboard will be inserted line-by-line into current file in column mode.

1. Copy multiple-lines text into clipboard.
2. Change to column mode in PilotEdit.
3. The string from clipboard will be inserted line-by-line into current file.



```

C:\PilotEdit\DynamicBuffer.h* File1*
1 // DynamicBuffer.h: interface for theFIRST LINE| CDynamicBuffer class.
2 //SECOND LINE|
3 ///////////////////////////////////////////////////THIRD LINE|////////////////////////////////////
4 FORTH LINE|
5 #if !defined(AFX_DYNAMICBUFFER_H__A41FIFTH LINEEB10E_2EF0_4C5E_9B76_EC
6 #define AFX_DYNAMICBUFFER_H__A41EB10E_2EF0_4C5E|9B76_ECDE58F70CB6__INC
7
  
```


4.2.14. Word Wrap

Selecting menu **Edit**, then **Word Wrap** or pressing **Ctrl + W** will change the current file into word wrap mode.

In the word wrap mode, if a line is too long, it will be displayed in several lines.

4.2.15. HEX Mode

Selecting menu **Edit**, then **HEX Mode** or pressing **Ctrl + H** will change the current file into HEX mode.

4.2.16. Tabs and Spaces

4.2.16.1. Remove Leading Spaces

Selecting menu **Edit, Tabs and Spaces**, then **Remove Leading Spaces** will remove leading spaces from the selected lines.

4.2.16.2. Remove Leading Tabs and Spaces

Selecting menu **Edit, Tabs and Spaces**, then **Remove Leading Tabs and Spaces** will remove leading tabs and spaces from the selected lines.

4.2.16.3. Remove Trailing Spaces

Selecting menu **Edit, Tabs and Spaces**, then **Remove Trailing Spaces** will remove trailing spaces from the selected lines.

4.2.16.4. Remove Trailing Tabs and Spaces

Selecting menu **Edit, Tabs and Spaces**, then **Remove Trailing Tabs and Spaces** will remove trailing tabs and spaces from the selected lines.

4.2.17. Change Case

4.2.17.1. To Upper Case

Selecting menu **Edit, Change Case**, then **To Upper Case** or pressing **Ctrl + U** will change the selected text into upper case.

4.2.17.2. To Lower Case

Selecting menu **Edit, Change Case**, then **To Lower Case** or pressing **Ctrl + L** will change the selected text into lower case.

4.2.17.3. Capitalize

Selecting menu **Edit, Change Case**, then **Capitalize** will capitalize the selected text.

4.2.17.4. Invert Case

Selecting menu **Edit, Change Case**, then **Invert Case** will invert case of the selected text.

4.2.18. Indent

4.2.18.1. Increase Indent

Selecting menu **Edit, Indent**, then **Increase Indent** or pressing **TAB** will increase indent of the selected text.

4.2.18.2. Decrease Indent

Selecting menu **Edit, Indent**, then **Decrease Indent** or pressing **Shift + TAB** will decrease indent of the selected text.

4.2.19. Insert

4.2.19.1. Date

Selecting menu **Edit, Insert**, then **Date** will insert date.

4.2.19.2. Time

Selecting menu **Edit, Insert**, then **Time** will insert time.

4.2.19.3. Date and Time


Selecting menu **Edit, Insert**, then **Date and Time** will insert date and time.

4.2.19.4. Customized Date and Time

Selecting menu **Edit, Insert**, then **Customized Date and Time** will insert customized date and time.

Please refer to the following table and examples about how to customize date and time. You can change the customized date and time format by changing variable

"CUSTOMIZED_DATE_AND_TIME" in "<User Application Directory>\PilotEdit\CONFIG.DATA" and restart PilotEdit.

 Before changing CUSTOMIZED_DATE_AND_TIME, you must close PilotEdit and edit CONFIG.DATA with any other file editors. Otherwise your update will be lost.

Date and Time Tag List:

| Tag | Notes |
|-----|--------------------------|
| %a | Abbreviated weekday name |
| %A | Full weekday name |
| %b | Abbreviated month name |
| %B | Full month name |
| %c | Date and time |

| | |
|---|--|
| | representation appropriate for locale |
| %d | Day of month as decimal number (01 "C 31) |
| %H | Hour in 24-hour format (00 "C 23) |
| %I | Hour in 12-hour format (01 "C 12) |
| %j | Day of year as decimal number (001 "C 366) |
| %m | Month as decimal number (01 "C 12) |
| %M | Minute as decimal number (00 "C 59) |
| %p | Current locale's A.M./P.M. indicator for 12-hour clock |
| %S | Second as decimal number (00 "C 59) |
| %U | Week of year as decimal number, with Sunday as first day of week (00 "C 53) |
| %w | Weekday as decimal number (0 "C 6; Sunday is 0) |
| %W | Week of year as decimal number, with Monday as first day of week (00 "C 53) |
| %x | Date representation for current locale |
| %X | Time representation for current locale |
| %y | Year without century, as decimal number (00 "C 99) |
| %Y | Year with century, as decimal number |
| %Z, %Z | Either the time-zone name or time zone abbreviation, depending on registry settings; no characters if time zone is unknown |
| %% | Percent sign |
| %#a, %#A, %#b, %#B, %#p, %#X, %#z, %#Z, %#% | # flag is ignored. |

| | |
|--|--|
| %#c | Long date and time representation, appropriate for current locale. For example: "Tuesday, March 14, 1995, 12:41:29". |
| %#x | Long date representation, appropriate to current locale. For example: "Tuesday, March 14, 1995". |
| %#d, %#H, %#I, %#j, %#m, %#M, %#S, %#U, %#w, %#W, %#y, %#Y | Remove leading zeros (if any). |

Examples:

| Date and Time Format Examples | Date and Time String Examples |
|---|--------------------------------------|
| CUSTOMIZED_DATE_AND_TIME<<>>%Y-%m-%d %H:%M:%S | 2009-12-24 22:29:06 |
| CUSTOMIZED_DATE_AND_TIME<<>>%#c | Thursday, December 24, 2009 22:33:49 |

4.2.20. Word

4.2.20.1. Jump to the Left Word

Selecting menu **Edit, Word**, then **Jump to the Left Word** will jump to the left word.

4.2.20.2. Jump to the Right Word

Selecting menu **Edit, Word**, then **Jump to the Right Word** will jump to the right word.

4.2.20.3. Select to the Left Word

Selecting menu **Edit, Word**, then **Select to the Left Word** will select to the left word.

4.2.20.4. Select to the Right Word

Selecting menu **Edit, Word**, then **Select to the Right Word** will select to the right word.

4.2.20.5. Remove to the Left Word

Selecting menu **Edit, Word**, then **Remove to the Left Word** will remove to the left word.

4.2.20.6. Remove to the Right Word

Selecting menu **Edit, Word**, then **Remove to the Right Word** will remove to the right word.

4.2.21. Line

4.2.21.1. Remove Blank Lines from the Active File

Selecting menu **Edit, Line**, then **Remove Blank Lines from the Active File** will remove blank lines from the active file.

4.2.21.2. Remove Redundant Blank Lines from the Active File

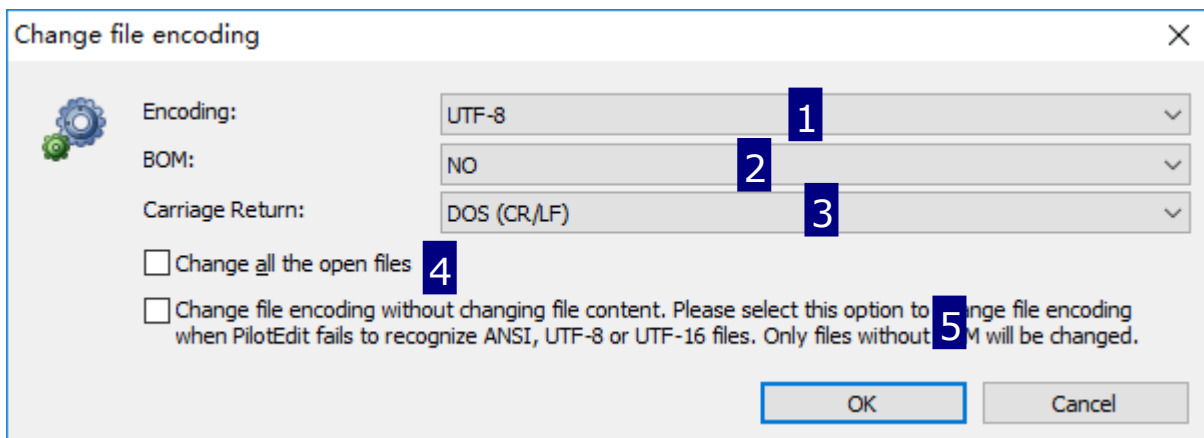
Selecting menu **Edit, Line**, then **Remove Redundant Blank Lines from the Active File** will remove redundant blank lines from the active file.

4.2.22. Don't Check File Update

You can disable file update detection by selecting menu **Edit**, then **Don't Check File Update**.

4.2.23. Change File Encoding

You can change the encoding of multiple files by selecting the menu **Edit**, then **Change File Encoding**. PilotEdit supports ANSI, UTF-8, UTF-16 and UTF-16 big endian files. UNIX, DOS and MAC files can also be processed by PilotEdit.



1. **Encoding.** PilotEdit supports ANSI, UTF-8, UTF-16 and UTF-16 big endian.
2. **BOM.** UTF-8 files may have BOM or not.
3. **Carriage Return.** Windows and UNIX files have different carriage returns.
4. You can select **Change all the open files** to change all the opened files.
5. Select this option to change file encoding when PilotEdit fails to recognize the encoding of ANSI or UTF-8 files. If this option is selected, only the file encoding will be changed and the file content will not be changed. If this option is selected, only files without BOM will be changed. The carriage return will not be changed if a file already includes CR/LF.

Example 1:

Change encoding of all the open files.

1. Open an UTF-8 file, an ANSI file, an UTF-16 file and an UTF-16 big endian file.
2. Select the menu Edit, then Change File Encoding and change all the opened files into UTF-8 UNIX file.

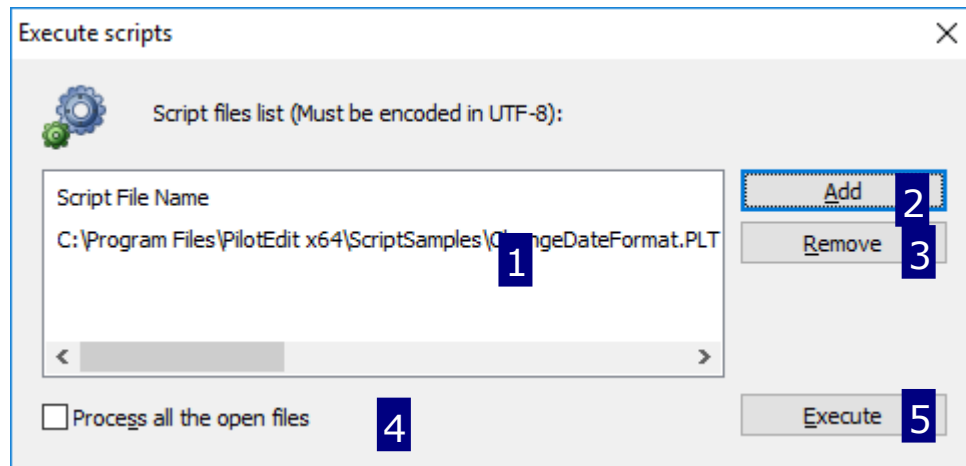
3. You will find that all the files have been changed into UTF-8 UNIX files disregarding their original encodings.

4.2.24. Execute Scripts *


You can define some scripts and execute the scripts over the opened files. In this way, you don't need to repeat your work day after day.

The PilotEdit scripts are normal text files encoded in **UTF-8**.

The script file name must be ***.PLT**.

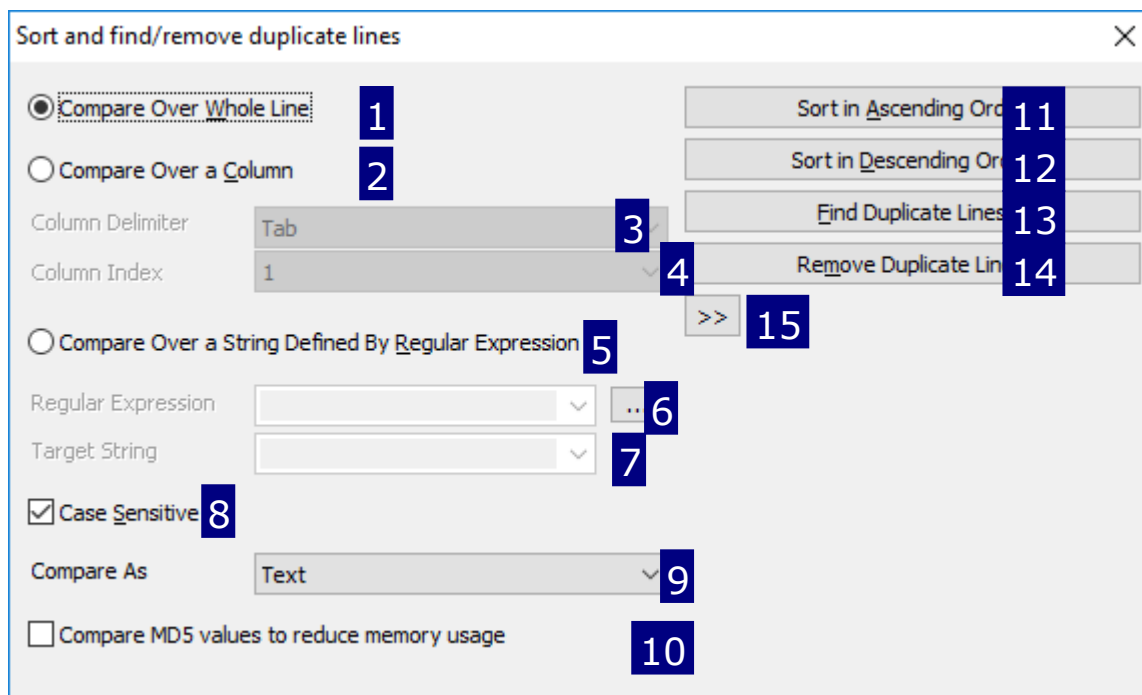


1. Script List. Scripts listed here will be executed.
2. Click **Add** to add some scripts to the list.
3. Click **Remove** to remove the selected scripts from the list.
4. You can choose to process all the open files or only the active file.
5. Click **Execute** to execute the scripts in the list.

 You can find more information about PilotEdit Script from the following links:
[PilotEdit Script](#)

4.2.25. Sort and Find/Remove Duplicated Lines *

You can sort or find/remove duplicate lines by selecting the menu item **Edit**, then **Sort and Find/Remove Duplicate Lines...**



1. Compare over whole lines while sorting finding/removing duplicate lines.
2. Compare over a column while sorting finding/removing duplicate lines.
3. Column delimiter may be Tab, blank or both.
4. Column index.
5. Compare over a target string defined by regular expression while sorting finding/removing duplicate lines.
6. Regular expression.
7. Target string definition.
8. The target string for comparison is case sensitive or not.
9. The target string may be compared as text or integer number.
10. Compare the MD5 values to reduce memory usage. This option only works while removing duplicate lines. It is recommended to select this option only when you get OOME while removing duplicate lines.
11. Sort in ascending order.
12. Sort in descending order.
13. Find duplicate lines. The search result will be sent to the output window.
14. Remove duplicate lines.
15. Change output window.

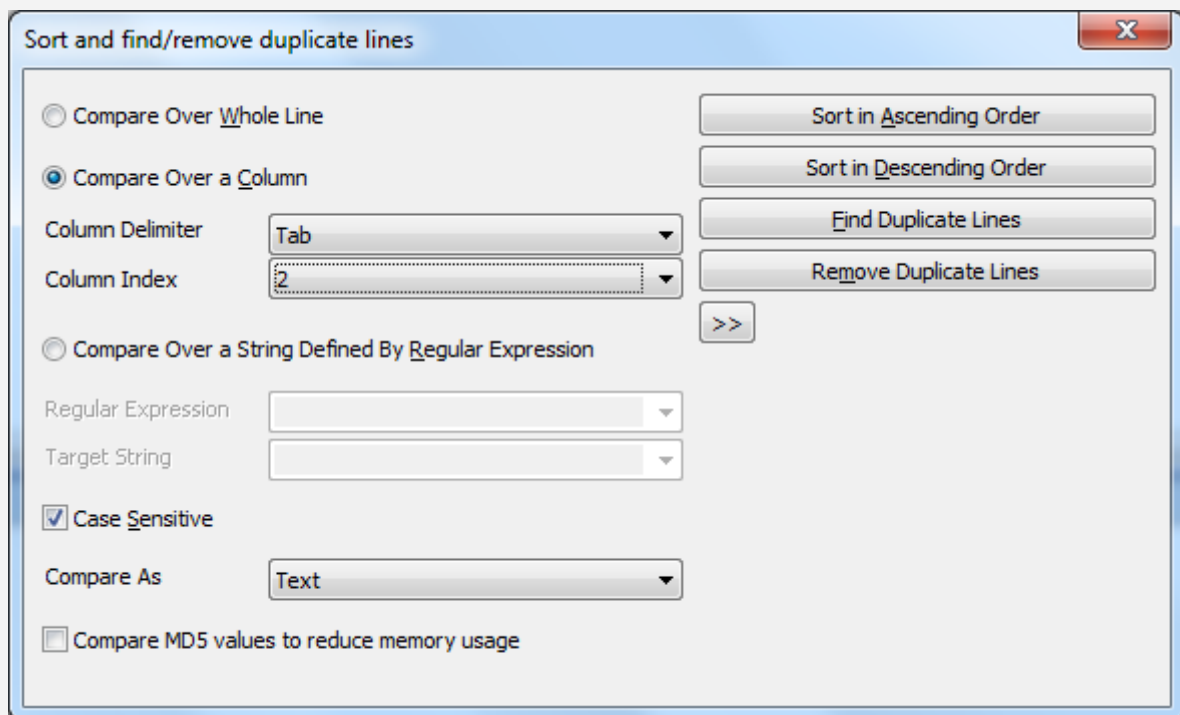
Example 1:

This example demonstrates how to sort over a column with PilotEdit.

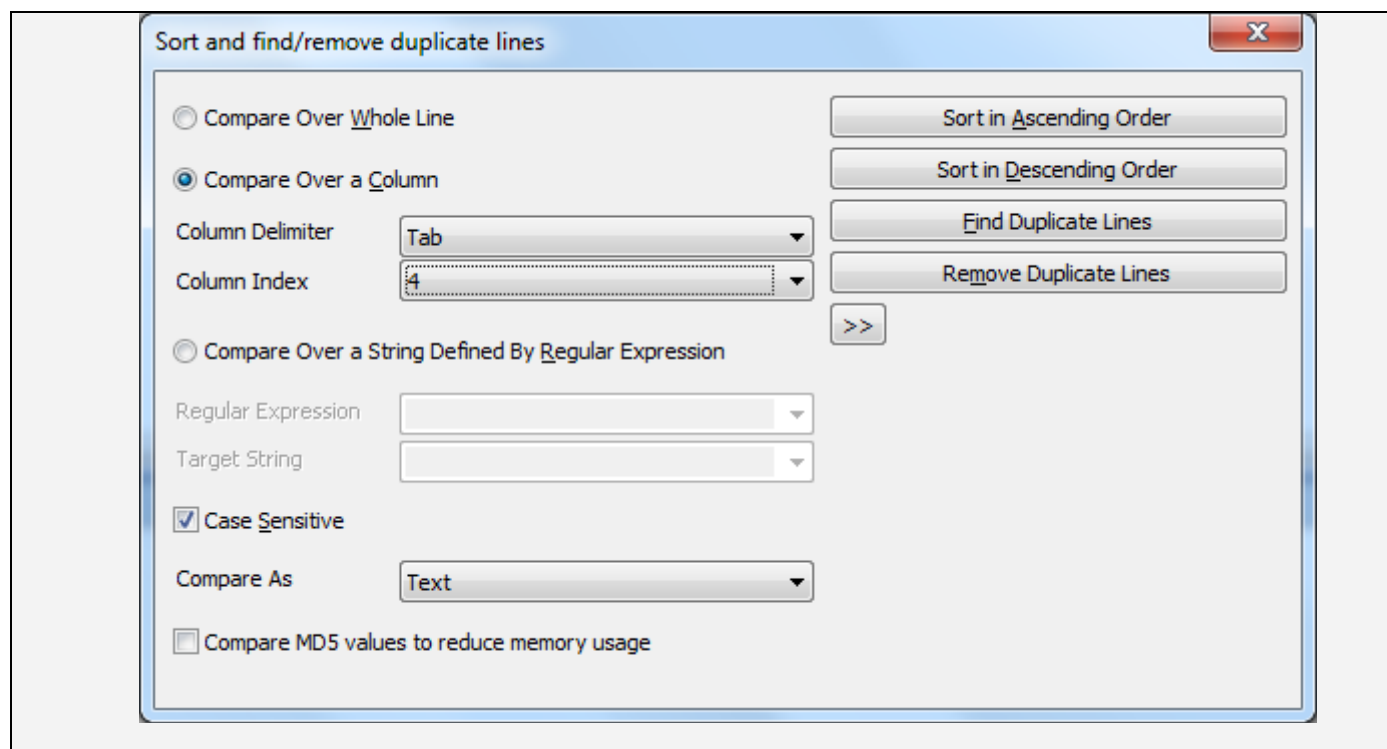
1. This is the file content.

| | | | | |
|---|---|--------|------------------|----|
| 1 | 1 | Bob | Bob@gmail.com | 22 |
| 2 | 2 | John | John@gmail.com | 32 |
| 3 | 3 | Thomas | Thomas@gmail.com | 25 |
| 4 | 4 | Dean | Dean@gmail.com | 24 |
| 5 | 5 | Paul | Paul@gmail.com | 33 |
| 6 | 6 | Jack | Jack@gmail.com | 36 |
| 7 | 7 | Brooke | Brooke@gmail.com | 28 |

2. This picture shows how to sort over user name (the second column).



3. This picture shows how to sort over age (the forth column).

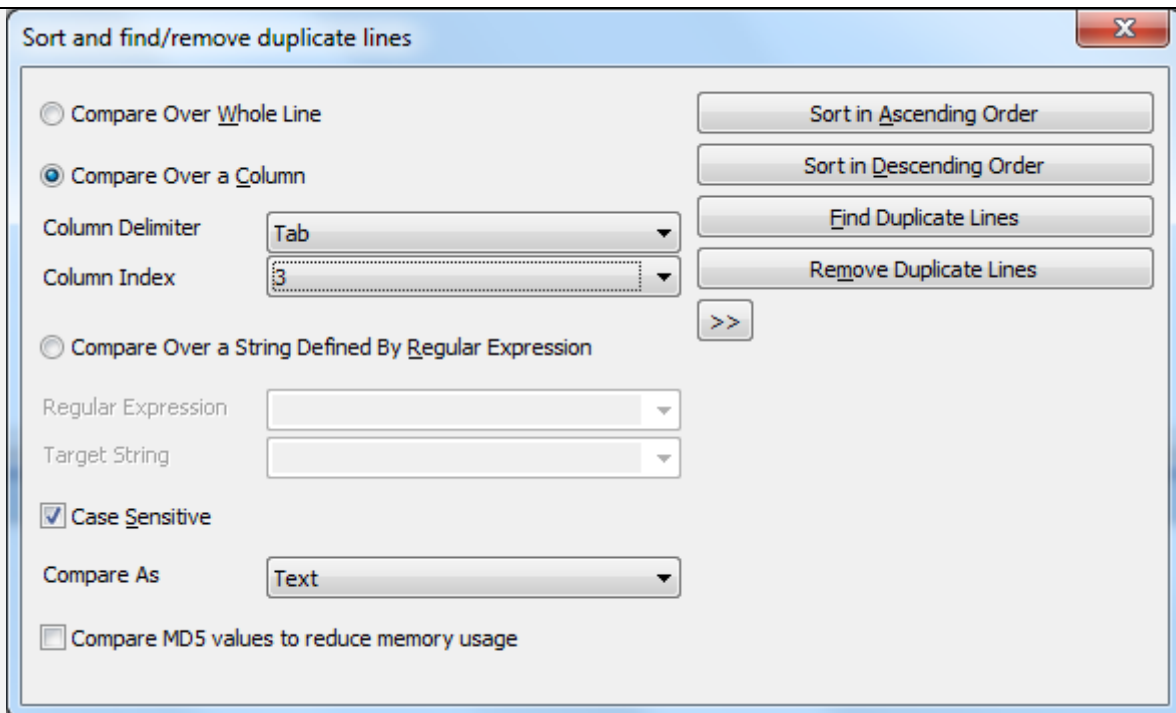
**Example 2:**

This example demonstrates how to remove duplicated lines.

1. This is the file with duplicated Email address in it.

| | | | | |
|---|---|--------|------------------|----|
| 1 | 1 | Bob | Bob@gmail.com | 22 |
| 2 | 2 | John | John@gmail.com | 32 |
| 3 | 3 | Thomas | Thomas@gmail.com | 25 |
| 4 | 4 | Dean | Dean@gmail.com | 24 |
| 5 | 5 | Paul | Bob@gmail.com | 33 |
| 6 | 6 | Jack | Jack@gmail.com | 36 |
| 7 | 7 | Brooke | Brooke@gmail.com | 28 |

2. This picture shows how to remove lines with duplicated Email address.



3. This picture shows the file after removing lines with duplicated Email address

```

1 1   Bob   Bob@gmail.com   22
2 2   John   John@gmail.com   32
3 3   Thomas Thomas@gmail.com   25
4 4   Dean   Dean@gmail.com   24
5 6   Jack   Jack@gmail.com   36
6 7   Brooke Brooke@gmail.com   28

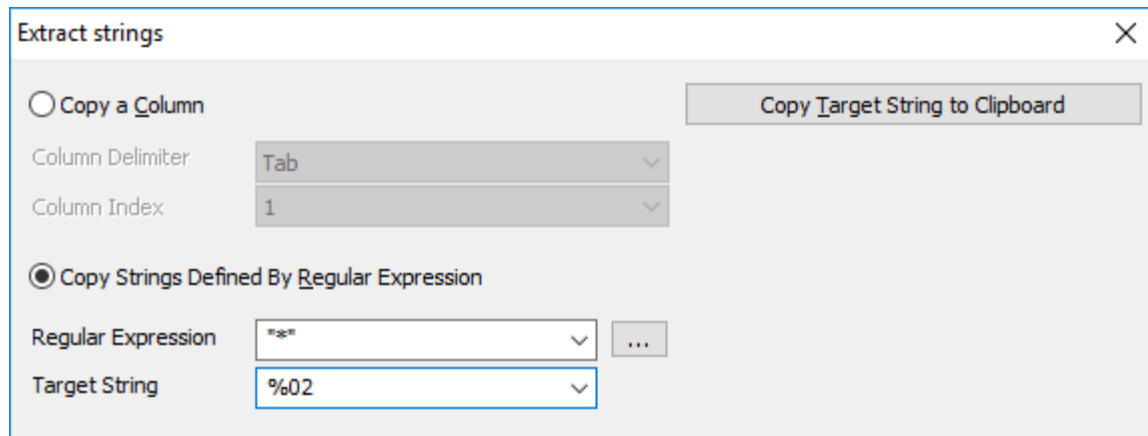
```

You can get examples about how to sort and find/remove duplicate lines with Regular Expression from the following links:

[Sort over regular expression](#)

4.2.26. Extract Strings *

You can extract strings from the active file by selecting the menu item **Edit**, then **Extract Strings....**



You can get examples about how to extract strings with Regular Expression from the following links:
[Extract strings in the double quotation marks and copy them to the clipboard](#)
[Extract Email addresses and copy them to the clipboard](#)

4.2.27. Format Selection *

Selecting menu **Edit**, then **Format Selection** and you will be able to format the selected text.


```

131  BOOL CDynamicBuffer::CreateHeap()
132  {
133      m_sHeap = HeapCreate(HEAP_GENERATE_EXCEPTIONS, 0, 0);
134      m_pBlockArray = (NODE_BLOCK **)malloc((unsigned int)m_nBlockArray
135      |         |         |         return TRUE;
136  }
137  BOOL CDynamicBuffer::DestroyHeap()
138  {
139      if (HeapDestroy(m_sHeap))
140      {
141          m_sHeap = NULL;
142      }
143      free(m_pBlockArray);
144      return TRUE;
145  }

```

```

131  BOOL CDynamicBuffer::CreateHeap()
132  {
133      m_sHeap = HeapCreate(HEAP_GENERATE_EXCEPTIONS, 0, 0);
134      m_pBlockArray = (NODE_BLOCK **)malloc((unsigned int)m_nBlockAr
135      return TRUE;
136  }
137  BOOL CDynamicBuffer::DestroyHeap()
138  {
139      if (HeapDestroy(m_sHeap))
140      {
141          m_sHeap = NULL;
142      }
143      free(m_pBlockArray);
144      return TRUE;
145  }

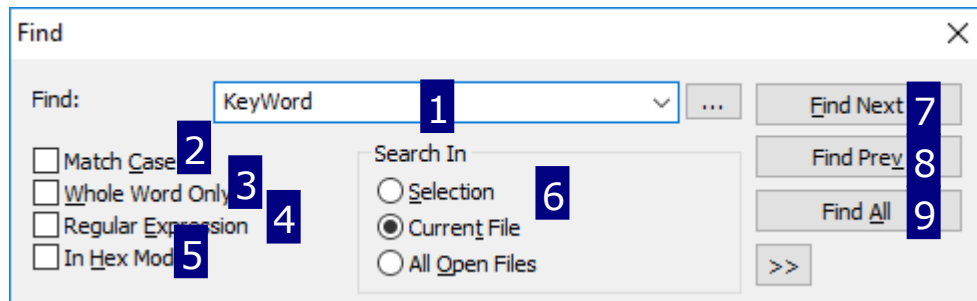
```

 The tags for formation are defined in menu **Configure**, then **File Type...**, then **Advanced...**, then **Format Tags**:

4.3. Search

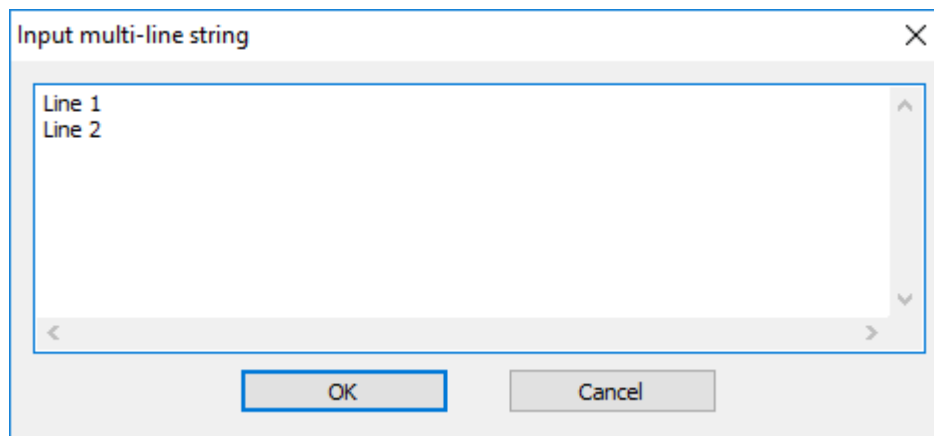
4.3.1. Find...

You can select menu Search, then Find... or press Ctrl + F to find a string in the opened files.



1. The string to search for.
2. **Match Case**. If you choose this option, the match string will be case-sensitive.
3. **Whole Word Only**.
4. **Regular Expression**. If you choose Regular Expression, you can't select the other options.
5. **In HEX Mode**. The search will be performed in HEX Mode. If you choose In HEX mode, you can't select the other options.
6. **Search In**. You can choose to search in selection, in the active file or in all the open files.
7. Click **Find Next** to find the next match string.
8. Click **Find Prev** to find the previous match string.
9. Click **Find All** to find all the match strings. The search result will be send to the output window.

You can input multi-line string by clicking on the ... button.



When you select **Regular Expression**, clicking on the ... button and you will be able to [create the Regular Expression from GUI](#).

Input Regular Expression [X]

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
|--------------------|---------------|

[Add Segment] [Delete] [Move Up] [Move Down]

Regular Expression: []

Target String: []

[Save] [Cancel]

You can also search in HEX mode.

Find [X]

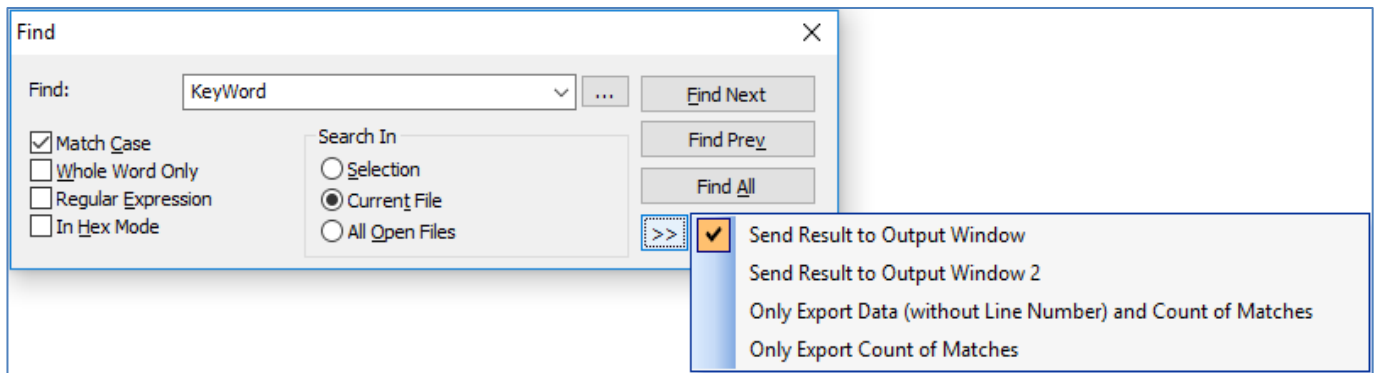
Find: [0D 0A] [v] [...]

☒ Match Case
☐ Whole Word Only
☐ Regular Expression
☒ In Hex Mode

Search In
☐ Selection
☒ Current File
☐ All Open Files

[Find Next] [Find Prev] [Find All] [>>]

You can change the output window or show more options by clicking button >>.



The maximum search string length allowed by PilotEdit is 2000.

⚠ Search/Replace in regular expression may be slower than normal search.

⚠ You can find more information about regular expression from the following links:

[Create Regular Expression with GUI](#)

[Search Expression](#)

[Search Regular Expression Examples](#)

4.3.2. Find Previous

You can select menu **Search**, then **Find Previous** or press **Ctrl + F3** to find the previous match string in the active file.

⚠ If you have never searched for any string, PilotEdit will prompt you to input a string for searching.

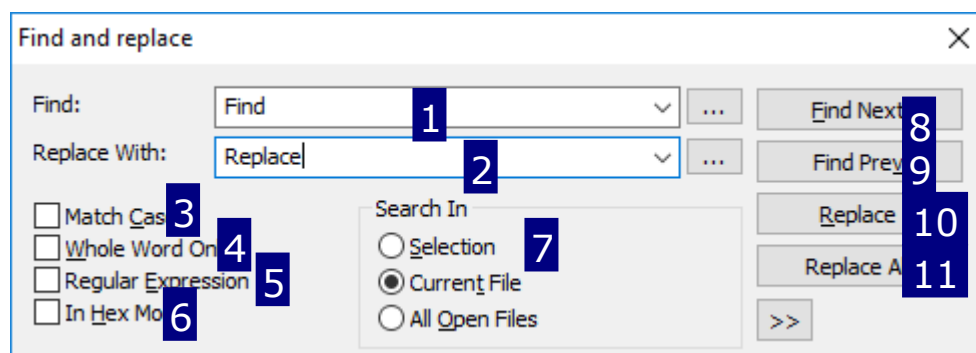
4.3.3. Find Next

You can select menu **Search**, then **Find Next** or press **F3** to find the next match string in the active file.

⚠ If you have never searched for any string, PilotEdit will prompt you to input a string for searching.

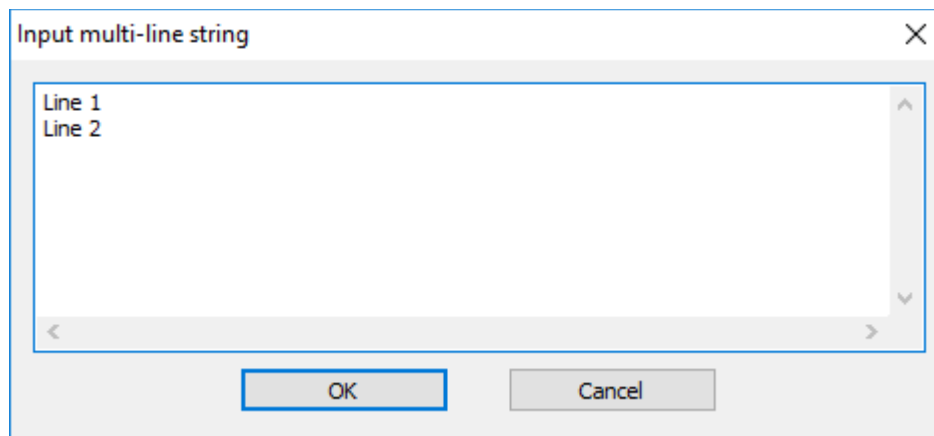
4.3.4. Replace...

You can select menu **Search**, then **Replace...** or press **Ctrl + R** to Replace a string in the opened files.



1. The string to search for.
2. The string to be replaced with.
3. **Match Case.** If you choose this option, the match string will be case-sensitive.
4. **Whole Word Only.**
5. **Regular Expression.** If you choose **Regular Expression**, you can't select the other options.
6. **In HEX Mode.** The search will be performed in HEX Mode. If you choose In HEX mode, you can't select the other options.
7. **Search In.** You can choose to search in selection, in the active file or in all the open files.
8. Click **Find Next** to find the next match string.
9. Click **Find Prev** to find the previous match string.
10. Click **Replace** to replace the next match string.
11. Click **Replace All** to Replace all the match strings. The search result will be send to the output window.

You can input multi-line string by clicking on the ... button.



When you select **Regular Expression**, clicking on the ... button and you will be able to [create the Regular Expression from GUI](#).

Input Regular Expression [X]

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
|--------------------|---------------|

[Add Segment] [Delete] [Move Up] [Move Down]

Regular Expression: []

Target String: []

[Save] [Cancel]

You can also replace in HEX mode.

Find and replace [X]

Find: [0D 0A] [Find Next]

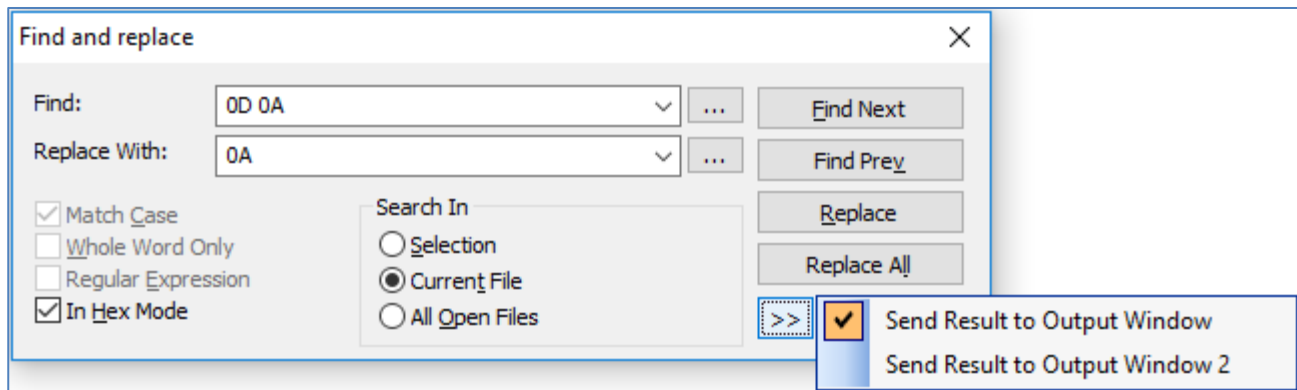
Replace With: [0A] [Find Prey]

☒ Match Case
☐ Whole Word Only
☐ Regular Expression
☒ In Hex Mode

Search In
☐ Selection
☒ Current File
☐ All Open Files

[Replace] [Replace All] [>>]

You can change the output window or show more options by clicking button >>.



The maximum search string length allowed by PilotEdit is 2000.

Search/Replace in regular expression may be slower than normal search.

You can find more information about regular expression from the following links:

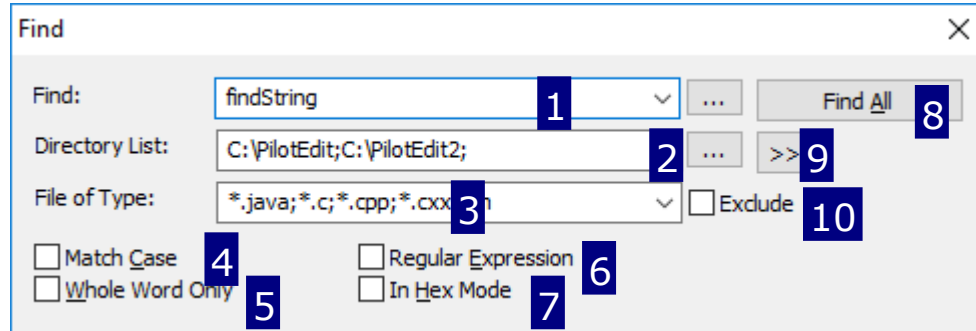
[Create Regular Expression with GUI](#)

[Replace Expression](#)

[Replace Regular Expression Examples](#)

4.3.5. Find In Multiple Directories And Files

You can select menu item **Search**, then **Find In &Multiple Directories And Files...** to find a string in multiple directories and files.



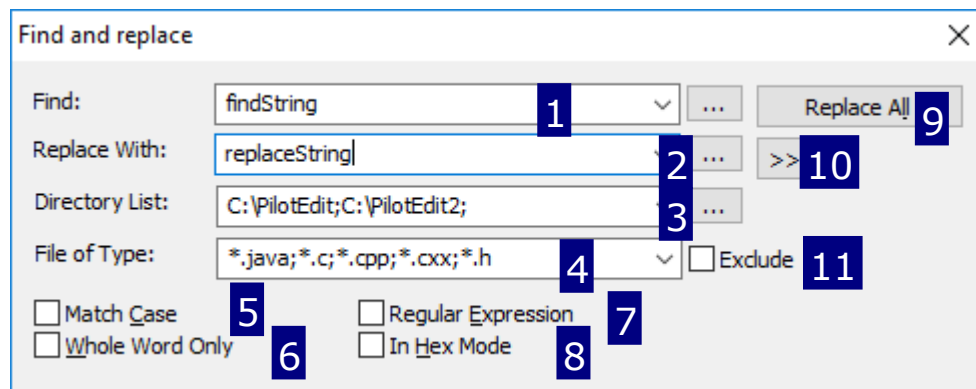
1. The string to search for.
2. You can specify multiple directories and files. The directories and files should be separated by ;.
3. You can specify multiple file types. The file types should be separated by ;. If you want to search in all files, you can use *.*.
4. **Match Case.** If you choose this option, the match string will be case-sensitive.
5. **Whole Word Only.**
6. **Regular Expression.** If you choose **Regular Expression**, you can't select the other options.
7. **In HEX Mode.** The search will be performed in HEX mode. If you choose **In HEX Mode**, you can't select the other options.
8. Click **Find All** to find all the match strings. The search result will be send to the output window.
9. Click **>>** to see more options.
10. You can exclude some file types like "*.zip;*.jar;*.class". *

You can find more information about regular expression from the following links:

[Create Regular Expression with GUI](#)
[Search Expression](#)
[Search Regular Expression Examples](#)

4.3.6. Replace in Multiple Directories and Files

You can select menu item **Search**, then **Replace in Multiple Directories and Files...** to replace a string in multiple directories and files.



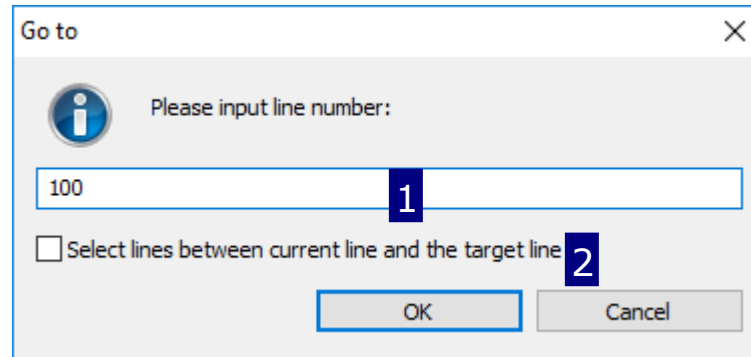
1. The string to search for.
2. The string to be replaced with.
3. You can specify multiple directories and files. The directories and files should be separated by ;.
4. You can specify multiple file types. The file types should be separated by ;. If you want to replace in all files, you can use *.*.
5. **Match Case.** If you choose this option, the match string will be case-sensitive.
6. **Whole Word Only.**
7. **Regular Expression.** If you choose **Regular Expression**, you can't select the other options.
8. **In HEX Mode.** The search will be performed in HEX Mode. If you choose **In HEX Mode**, you can't select the other options.
9. Click **Replace All** to find all the match strings. The replace result will be send to the output window.
10. Click **>>** to change to a different output window.
11. You can exclude some file types like "*.zip;*.jar;*.class". *

 You can find more information about regular expression from the following links:

[Create Regular Expression with GUI](#)
[Replace Expression](#)
[Replace Regular Expression Examples](#)

4.3.7. Go To...

You can select menu **Search**, then **Go To...** or press **Ctrl+G** to go to a line in the active file or select lines by line number.



1. The destination line number to go to.
2. Select lines between current line and the target line.

4.4. View

4.4.1. Toolbar

You can show/hide the following toolbars from the menu item **View**, then **Toolbar**.

| | |
|--------------------------------------|---|
| Standard Toolbar | Some most frequently used functions are listed in it. |
| Encryption/Decryption Toolbar | Encryption/decryption functions are listed in it. |
| Edit Toolbar | Some edit functions are listed in it. |

4.4.2. Window

You can show/hide the following windows from the menu item **View**, then **Window**.

| | |
|--|--|
| File Tree Window | The window for file list tree. |
| String Window | The window for string list. |
| File Group/Bookmark Window | The window for file group/bookmark. |
| Output Window | The window for search/replace result. |
| Output Window 2 | The second window for search/replace result. |
| Most Recently Used File List Window | The window for most recently used 10 files list. |

4.4.3. Status Bar

You can show/hide the status bars from the menu item **View**, then **Status Bar**.

4.4.4. Mark as Code Block *

You can change the source code structure into a tree by adding PilotEdit tags. To add PilotEdit tags please select multiple line and click menu item **View**, then **Mark as Code Block**.

PilotEdit Tag Example

This is the source code before adding PilotEdit Tags.

```

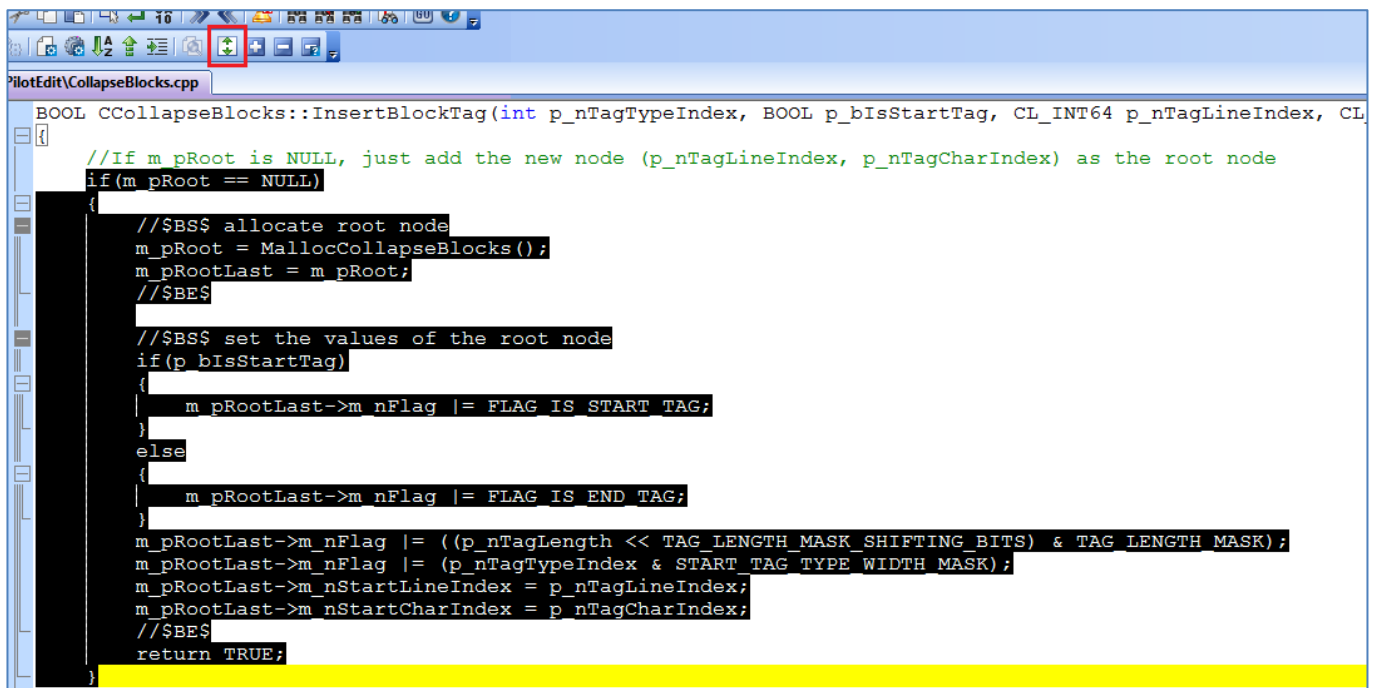
BOOL CCollapseBlocks::InsertBlockTag(int p_nTagTypeIndex, BOOL p_bIsStartTag, CL_INT64 p_nTagLineIndex, CL
{
    //If m_pRoot is NULL, just add the new node (p_nTagLineIndex, p_nTagCharIndex) as the root node
    if(m_pRoot == NULL)
    {
        //allocate root node
        m_pRoot = MallocCollapseBlocks();
        m_pRootLast = m_pRoot;

        //set the values of the root node
        if(p_bIsStartTag)
        {
            m_pRootLast->m_nFlag |= FLAG_IS_START_TAG;
        }
        else
        {
            m_pRootLast->m_nFlag |= FLAG_IS_END_TAG;
        }
        m_pRootLast->m_nFlag |= ((p_nTagLength << TAG_LENGTH_MASK_SHIFTING_BITS) & TAG_LENGTH_MASK);
        m_pRootLast->m_nFlag |= (p_nTagTypeIndex & START_TAG_TYPE_WIDTH_MASK);
        m_pRootLast->m_nStartLineIndex = p_nTagLineIndex;
        m_pRootLast->m_nStartCharIndex = p_nTagCharIndex;
        return TRUE;
    }

    //if the new node (p_nTagLineIndex, p_nTagCharIndex) is before m_pRoot, insert it before m_pRoot
    if (m_pRoot->m_nStartLineIndex > p_nTagLineIndex
        || (m_pRoot->m_nStartLineIndex == p_nTagLineIndex
            && m_pRoot->m_nStartCharIndex > p_nTagCharIndex))
    {
        //Save old root node and create a new root node
        COLLAPSE_BLOCK * pRootOld = m_pRoot;
        m_pRoot = MallocCollapseBlocks();
        m_pRootLast = m_pRoot;
    }
}

```

Mark source code as blocks to insert PilotEdit Tags.



```

PilotEdit\CollapseBlocks.cpp
BOOL CCollapseBlocks::InsertBlockTag(int p_nTagTypeIndex, BOOL p_bIsStartTag, CL_INT64 p_nTagLineIndex, CL
{
    //If m_pRoot is NULL, just add the new node (p_nTagLineIndex, p_nTagCharIndex) as the root node
    if(m_pRoot == NULL)
    {
        //SBS$ allocate root node
        m_pRoot = MallocCollapseBlocks();
        m_pRootLast = m_pRoot;
        //SBE$

        //SBS$ set the values of the root node
        if(p_bIsStartTag)
        {
            m_pRootLast->m_nFlag |= FLAG_IS_START_TAG;
        }
        else
        {
            m_pRootLast->m_nFlag |= FLAG_IS_END_TAG;
        }
        m_pRootLast->m_nFlag |= ((p_nTagLength << TAG_LENGTH_MASK_SHIFTING_BITS) & TAG_LENGTH_MASK);
        m_pRootLast->m_nFlag |= (p_nTagTypeIndex & START_TAG_TYPE_WIDTH_MASK);
        m_pRootLast->m_nStartLineIndex = p_nTagLineIndex;
        m_pRootLast->m_nStartCharIndex = p_nTagCharIndex;
        //SBE$
        return TRUE;
    }
}

```

After adding PilotEdit Tags, you can see the source code includes 6 steps.

```

BOOL CCollapseBlocks::InsertBlockTag(int p_nTagTypeIndex, BOOL p_bIsStartTag, CL_INT64 p_nTagLineIndex, CL
{
    // $BSS$ If m_pRoot is NULL, just add the new node (p_nTagLineIndex, p_nTagCharIndex) as the root node
    // $BSS$ if the new node (p_nTagLineIndex, p_nTagCharIndex) is before m_pRoot, insert it before m_pRoot
    // $BSS$ If the position of the new node (p_nTagLineIndex, p_nTagCharIndex) equals m_pRoot, report error
    // $BSS$ If m_pRootLast is NULL or after the new node, change m_pRootLast to m_pRoot
    // $BSS$ Move m_pRootLast to the node before the new node (p_nTagLineIndex, p_nTagCharIndex)
    // $BSS$ Insert the new node (p_nTagLineIndex, p_nTagCharIndex) after m_pRootLast
    return TRUE;
}

```

The second step includes 3 child steps.

```

BOOL CCollapseBlocks::InsertBlockTag(int p_nTagTypeIndex, BOOL p_bIsStartTag, CL_INT64 p_nTagLineIndex, CL
{
    // $BSS$ If m_pRoot is NULL, just add the new node (p_nTagLineIndex, p_nTagCharIndex) as the root node
    // $BSS$ if the new node (p_nTagLineIndex, p_nTagCharIndex) is before m_pRoot, insert it before m_pRoot
    if (m_pRoot->m_nStartLineIndex > p_nTagLineIndex)
    {
        // $BSS$ Save old root node and create a new root node
        // $BSS$ Set the values of the new root node
        // $BSS$ Append the old root node to the new root node
        return TRUE;
    }
    // $BES$
    // $BSS$ If the position of the new node (p_nTagLineIndex, p_nTagCharIndex) equals m_pRoot, report error
    // $BSS$ If m_pRootLast is NULL or after the new node, change m_pRootLast to m_pRoot
    // $BSS$ Move m_pRootLast to the node before the new node (p_nTagLineIndex, p_nTagCharIndex)
    // $BSS$ Insert the new node (p_nTagLineIndex, p_nTagCharIndex) after m_pRootLast
    return TRUE;
}

```

4.4.5. Expand All *

You can expand all blocks in selection from the menu item **View**, then **Expand All**.

4.4.6. Collapse All *

You can collapse all blocks in selection from the menu item **View**, then **Collapse All**.

4.4.7. Expand/Collapse... *

You can expand/collapse all blocks in selection with more options from the menu item **View**, then **Expand/Collapse...**

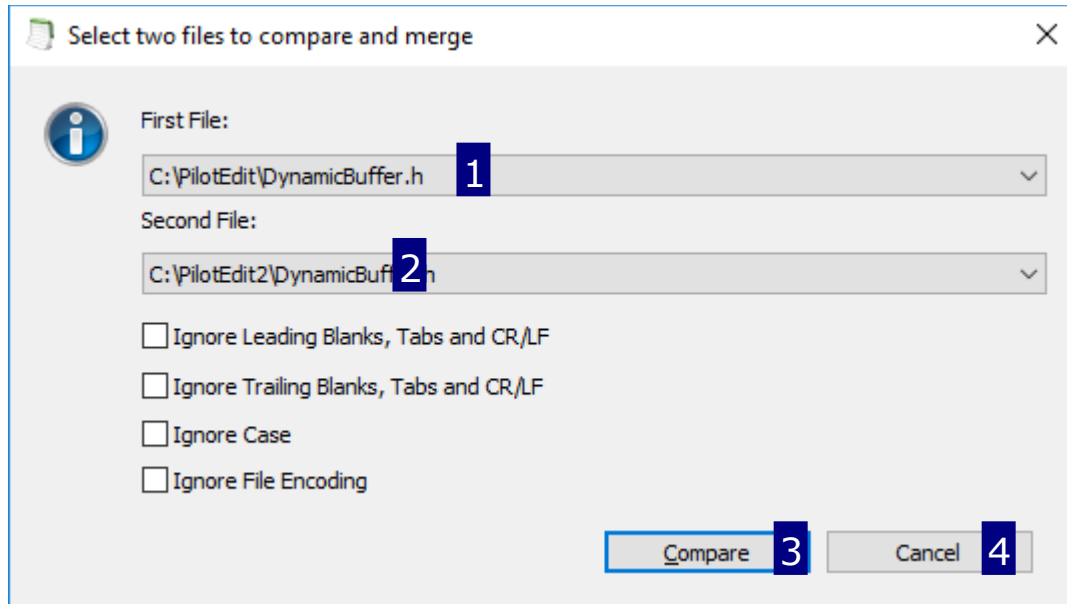
4.5. Compare

4.5.1. Compare and Merge Files... *

You can compare two opened files by selecting the menu item **Compare**, then **Compare and Merge Files....**

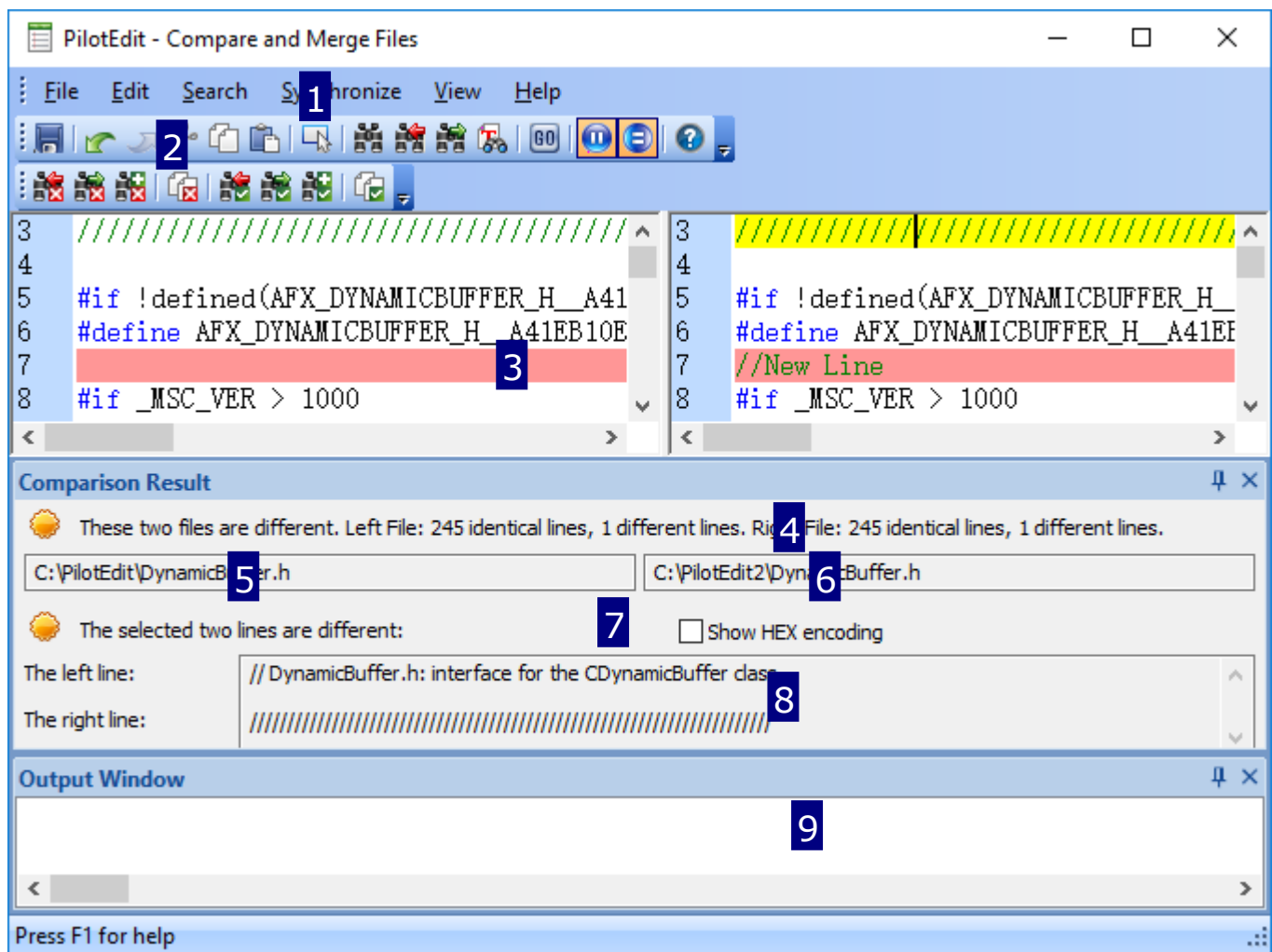
To compare two files, you must open them in PilotEdit first.

PilotEdit will update the comparison result automatically when you make any change to the file content.















1. Select the first file to be compared.
2. Select the second file to be compared.
3. Click **Compare** to compare two files.
4. Click **Cancel** to close this window.

PilotEdit Comparison Window



1. Menu.
2. Toolbar. Toolbar provides some frequently used functions from menu. You can choose to synchronize horizontally or vertically between the two compared files.
3. File content. The lines with red background are different lines and lines with white background are identical lines. The lines with yellow background are selected lines. You can select two lines and compare them in panel 8.
4. File comparison summary.
5. The left file name.
6. The right file name.
7. The selected two lines comparison summary.
8. The text or HEX encoding of the selected two lines. This will help you to find out the difference between two lines.
9. The output window.

Toolbar

| Icons | Notes |
|---|---|
|  | Synchronize vertical scroll bar. If this button is selected, when you move the vertical scroll bar in one window, the vertical scroll bar in the other window will also be changed. |
|  | Synchronize horizontal scroll bar. If this button is selected, when you move the horizontal scroll bar in one window, the horizontal scroll bar in the other window will also be changed. |
|  | Click this button to find the previous different block. |
|  | Click this button to find the next different block. |
|  | Click this button to find all the different blocks.  Only the first line of all the different blocks will be sent to the output window. |
|  | Click this button to copy all the different lines in the active file to the clipboard. This function will help to find out lines in the first file but not in the second file. |
|  | Click this button to find the previous identical block. |
|  | Click this button to find the next identical block. |
|  | Click this button to find all the identical blocks.  Only the first line of all the identical blocks will be sent to the output window. |
|  | Click this button to copy all the identical lines in the active file to the clipboard. This function will help to find out lines in both files. |

Export Comparison Results

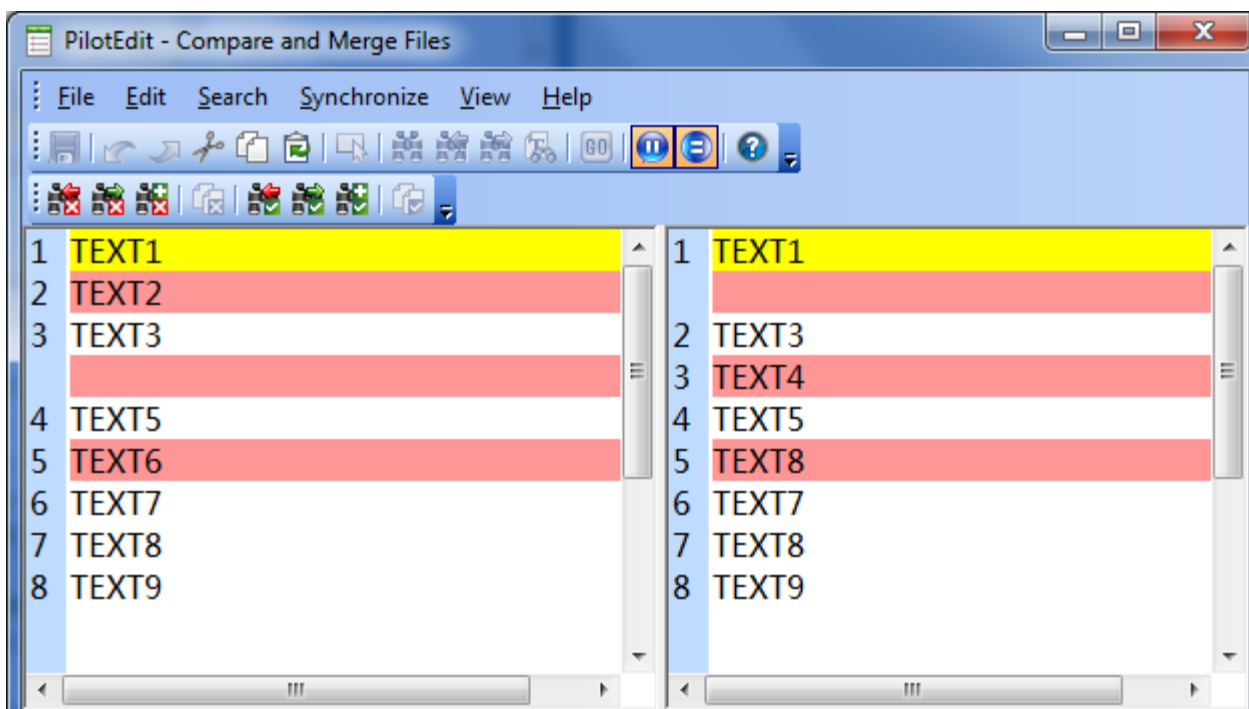
Selecting menu Edit, then Save Comparison Results in a File... will export comparison results as below:

| |
|--|
| [Flag][Line Index of the Left File]Line Content of the Left File<<>>[Line Index of the Right File]Line Content of the Right File |
|--|

Here are the definitions of flags:

| Flag | Notes |
|------|---------------------------------------|
| [Y] | means identical lines |
| [N] | means different lines |
| [L] | means the line only in the left file |
| [R] | means the line only in the right file |

Here is an example of comparison results.



After saving comparison results, the following file will be generated:

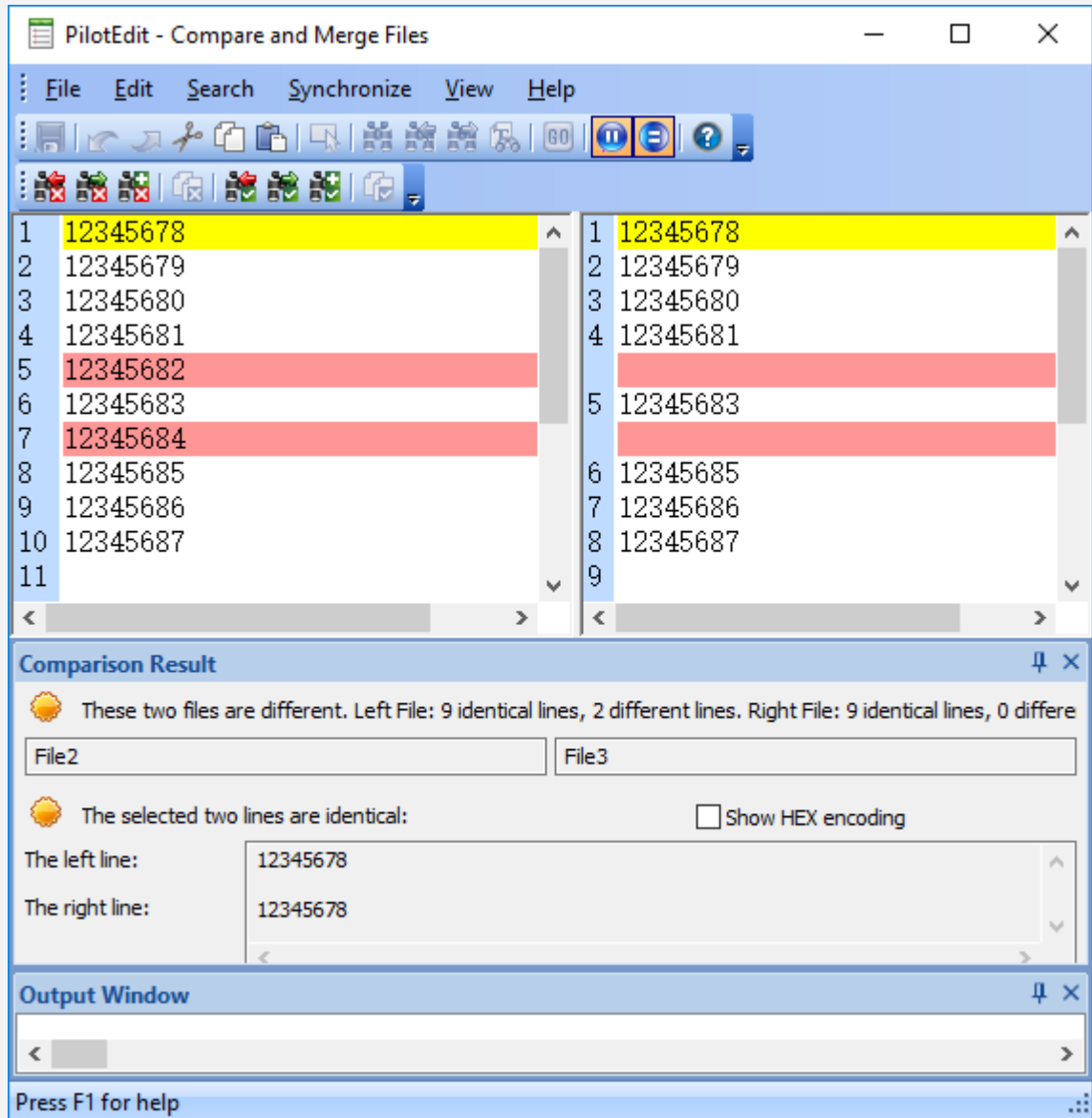
```
[Y][1]TEXT1<<>>[1]TEXT1
[L][2]TEXT2<<>>[]
[Y][3]TEXT3<<>>[2]TEXT3
[R][]<<>>[3]TEXT4
[Y][4]TEXT5<<>>[4]TEXT5
[N][5]TEXT6<<>>[5]TEXT8
[Y][6]TEXT7<<>>[6]TEXT7
[Y][7]TEXT8<<>>[7]TEXT8
[Y][8]TEXT9<<>>[8]TEXT9
```

If you want to change it to another format, you can create a PilotEdit script and run the script.

Example 1:

In this example, you will see how to find lines in the first file but not in the second file.

1. Open the two files and compare them.

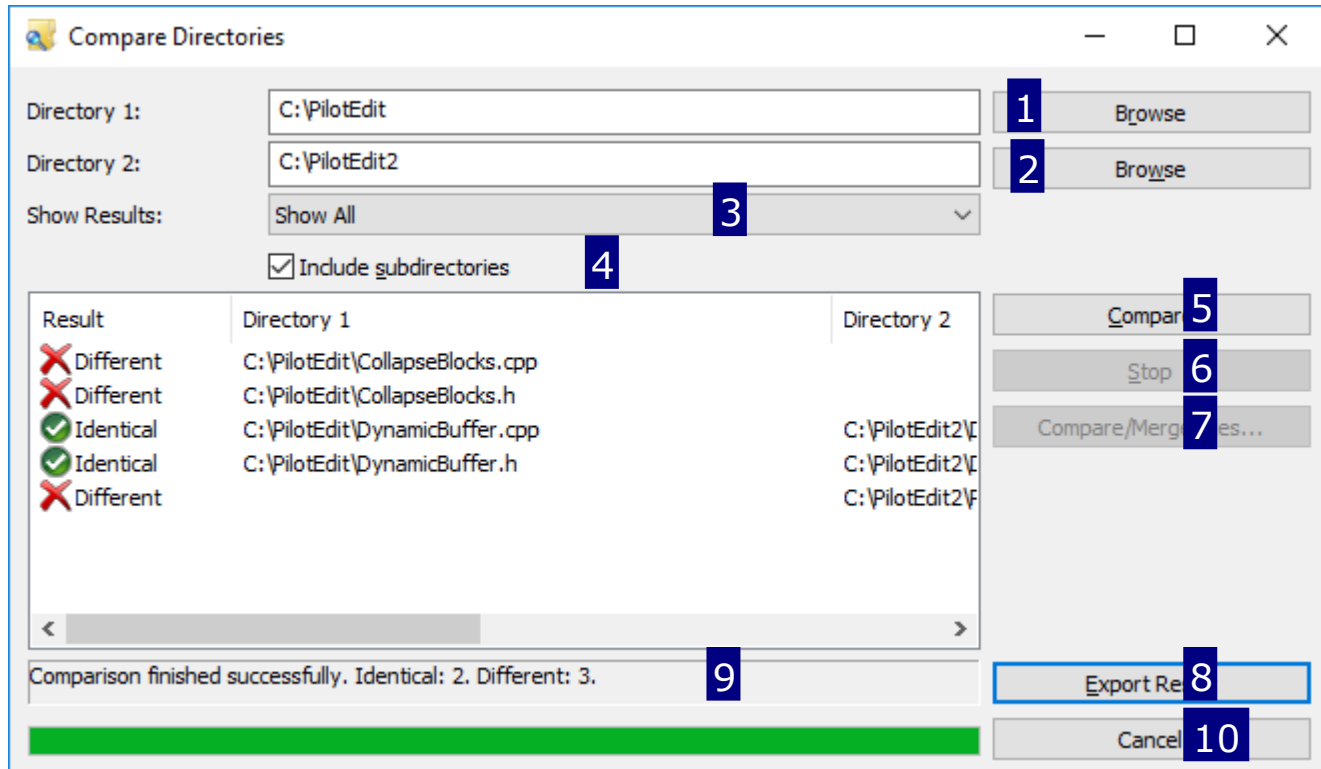


2. Click the first file and click button **Copy All Different Lines** and the lines in the first file but not in the second file will be copied to the clipboard.

```
12345682
12345684
```

4.5.2. Compare Directories...

You can compare the files in two directories by selecting the menu item **Compare**, then **Compare Directories...**




1. Click **Browse** to select the first directory.
2. Click **Browse** to select the second directory.
3. You can choose to show all result or only part of the result.
4. You can choose to compare the files in the child directories or not.
5. Click **Compare** to begin comparison.
6. Click **Stop** to stop comparison.
7. You can click **Compare/Merge Files...** to compare and merge the selected two files. *
8. You can export the comparison result to a text file.
9. Summary about comparison status.
10. Click **Cancel** to close this window.

4.6. Encryption *

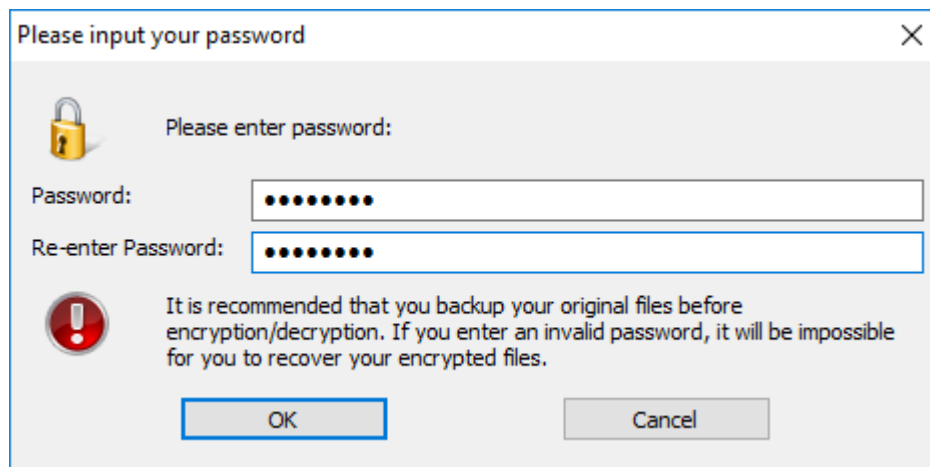
You can encrypt your sensitive files, like password files, with PilotEdit. PilotEdit provides the function to encrypt with 256-bit AES encryption.


AES is the Advanced Encryption Standard, which is the result of a three-year competition sponsored by the U.S. Government's National Institute of Standards (NIST). This encryption method, also known as Rijndael, has been adopted by NIST as a Federal Information Processing Standard.


 It is recommended that you should backup your original files before encryption/decryption. If you enter an invalid password, it will be impossible for you to recover your encrypted files.

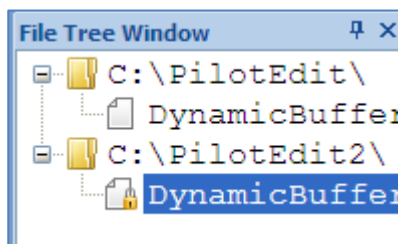
4.6.1. Open Encrypted Files... *

You can open encrypted files by selecting the menu item **Encryption**, then **Open Encrypted Files....** After input your password, you will be able to open encrypted files, edit them and save them transparently.



 If you choose to open a non-encrypted file by selecting menu item **Encryption**, then **Open Encrypted Files....**, this file will be changed into an encrypted file after you save it.

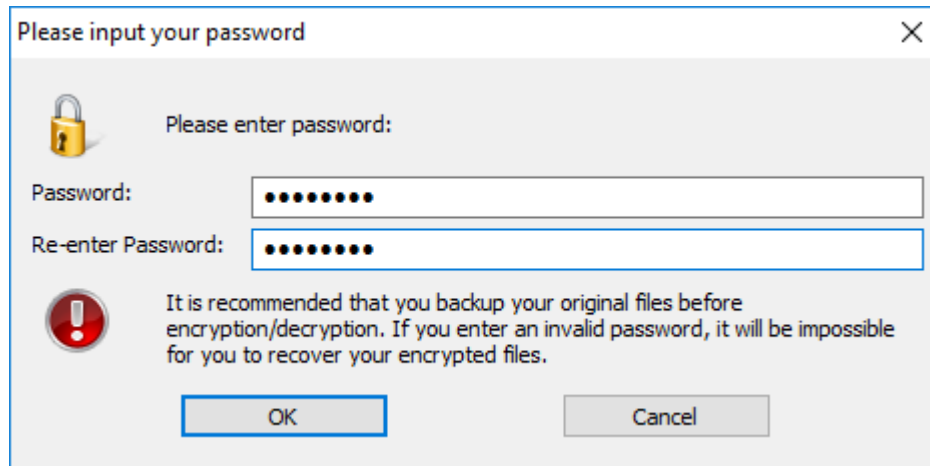
 The icon for an encrypted file is different from a non-encrypted file. For the screenshot below, the second file is an encrypted file.




4.6.2. Open Encrypted FTP Files... *

You can open encrypted FTP files by selecting the menu item **Encryption**, then **Open Encrypted FTP Files....** After input your password, you will be able to open encrypted FTP files.

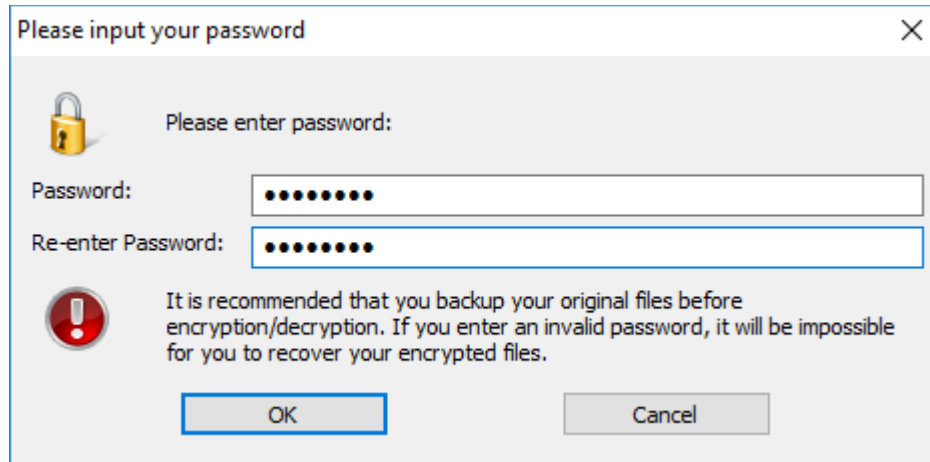
After you open an encrypted FTP file, you will be able to edit it and save it transparently.




 If you choose to open a non-encrypted FTP file by selecting menu item **Encryption**, then **Open Encrypted FTP Files...**, this file will be changed into an encrypted FTP file after you save it.

4.6.3. Save As Encrypted File... *

You can save a non-encrypted file as an encrypted file by selecting the menu item **Encryption**, then **Save As Encrypted File....** After input your password, you will be able to save the active file as an encrypted file.



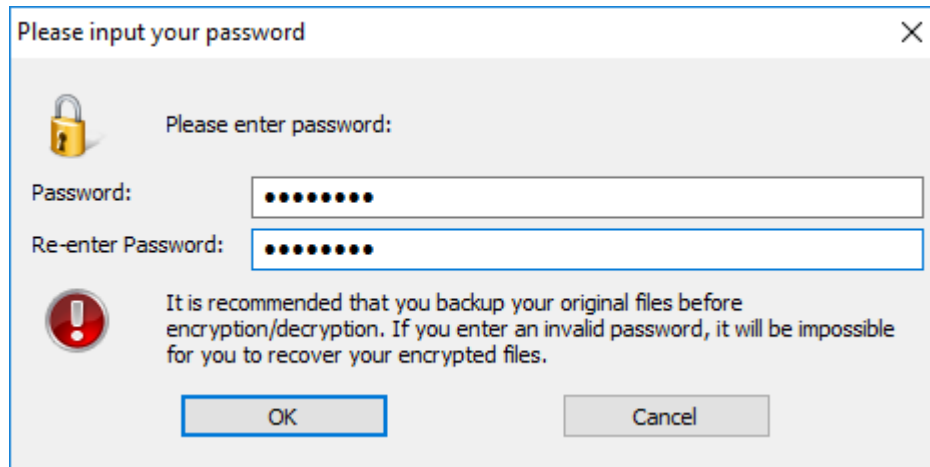
 You can change the password of an encrypted file by selecting the menu item **Encryption**, then **Save As Encrypted File....**


4.6.4. Save As Non-Encrypted File... *

You can save an encrypted file as a non-encrypted file by selecting the menu item **Encryption**, then **Save As Non-Encrypted File....**

4.6.5. Save As Encrypted FTP File... *

You can save a normal file as an encrypted FTP file by selecting the menu item **Encryption**, then **Save As Encrypted FTP File....** After input your password, you will be able to save the active file as an encrypted FTP file.



 You can change the password of an encrypted file by selecting the menu item **Encryption**, then **Save As Encrypted FTP File....**

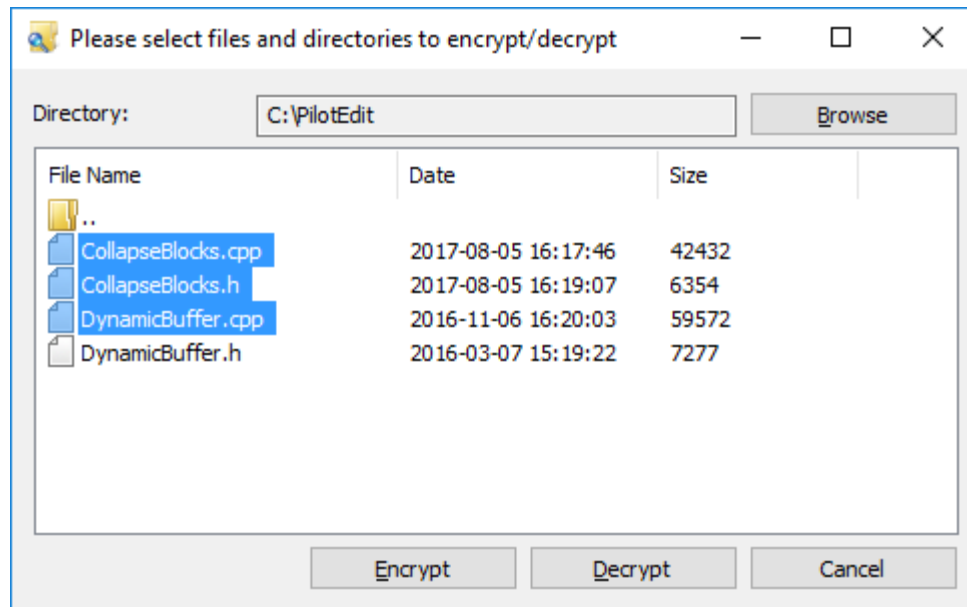
4.6.6. Save As Non-encrypted FTP File... *

You can save an encrypted file as a non-encrypted FTP file by selecting the menu item **Encryption**, then **Save As Non-encrypted FTP File....**

4.6.7. Encrypt/Decrypt Local Files... *

You can encrypt/decrypt multiple files and directories by selecting the menu item **Encryption**, then **Encrypt/Decrypt Local Files....**

You can encrypt/decrypt files larger than 4GB.



⚠ If you choose to encrypt an already encrypted file or decrypt a non-encrypted file, the file will be skipped.

4.7. Configure

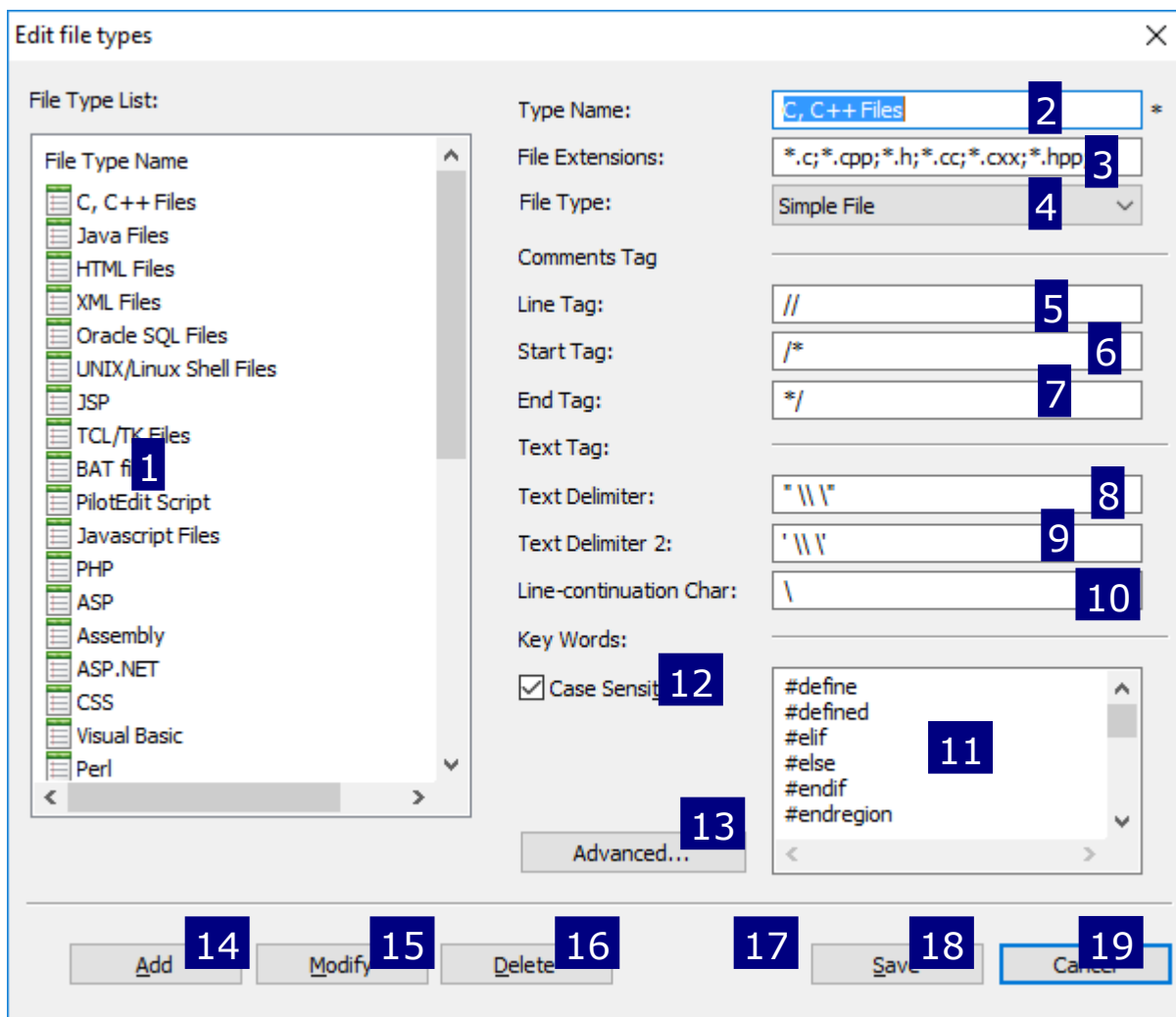
4.7.1. FTP Account...

You can open/edit FTP files with PilotEdit. To edit FTP server information, you may select the menu item **Configure**, then **FTP Account....**

1. **FTP server list.**
2. **FTP server name.** Server name is the unique ID of a server. When you make any change to a FTP server, you can't change its server name.
3. **FTP server IP address.**
4. **FTP server user name.**
5. **FTP server password.**
6. **FTP port.** By default, the FTP port is 21 and the SFTP port is 22.
7. **FTP protocol.** FTP or SFTP.
8. **FTP Server Directory List.** The directory list will be updated automatically when you access an FTP server.
9. You can select some FTP servers and move them up in the server list.
10. You can select some FTP servers and move them down in the server list.
11. Click **Add** to add the server to the FTP server list.
12. Click **Modify** to change the selected FTP server account.
13. Click **Delete** to delete the selected FTP server accounts.
14. When you make any change to the FTP server accounts, you **MUST** click Save to save the changes. Otherwise, your change will be lost after you close this window.
15. Click Cancel to close this window.

4.7.2. File Type...

You can edit file types by selecting the menu item **Configure**, then **File Type....**



1. **File type list**
2. **File type name.** Type name is the unique ID of a file type. When you make change to a file type, you can't change its type name.
3. **File Type.** File type may be one of the following values.

| File Type | Notes |
|----------------------|---|
| Simple File | File type that is not web server pages or XML file, like C/C++ files, java files etc. |
| Web Server Page File | <p>Web Server Page File Web server pages like HTML, ASP, JSP, ASP .NET or PHP etc.</p> <p>⚠ For Web Server Page File, the comment tag, text tag and key words should be specified for the inner programming language.</p> |

4. **File extensions.** You can define multiple file extensions for a file type. The file extensions should be separated by ;.
5. **Line comment tag.** For example, `//` in C++. If a file type doesn't have line comment, you may leave this item empty.
6. **Start comment delimiter.** For example, `/*` in C++.
7. **End comment delimiter.** For example, `*/` in C++.
8. **Text delimiter and special characters.** For example, `" \" \\` in C++;
9. **Another text delimiter and special characters.** For example, `' \' \\` in C++;
10. **Line-continuation character.** The line-continuation character indicates the next line is a continuation of current line.
11. **Key words list** for this file type.
12. Are these key words case sensitive? For example, key words of C++ are case-sensitive while those for Oracle script are not.
13. Click **Advanced...** button to show the advanced file type dialog.
14. Click **Add** to add the file type to the file type list.
15. Click **Modify** to change the selected file type.
16. Click **Delete** to delete the selected file types.
17. Click **Set Default** to set file types to the default setting.
18. **When you make any change to the file type, you MUST click Save to save the changes. Otherwise, you change will be lost after you close this window.**
19. Click **Cancel** to close this window.

Advanced File Types

When you click **Advanced...** button, you will be able to see the advanced file type configuration dialog.

1. **Web Page Code Start Tag.** This tag is for web server page type. For example, `<%` for ASP.
2. **Web Page Code End Tag.** This tag is for web server page type. For example, `%>` for ASP.
3. **Tag Type.** You can choose to collapse by File Tag or PilotEdit Tag.
4. **File Tags.** Tags for a file type.
5. **Code Block Start Tag.** Start Tag of PilotEdit Tag.
6. **Code Block End Tag.** End Tag of PilotEdit Tag.
7. **Tab Type.** You can define tab for a specific file type. For example, if you want to use TAB for C/C++ files and blanks for Java file, you may define different Tab Type here.

| Tab Type | Notes |
|-------------------------|---|
| User Default Definition | Use the default definition defined in Menu Item Configure , then Options... |
| Use Tab | Use Tab |
| Use Blanks | User Blanks |

8. **Tab Size.** If you choose to use blanks for Tab, you need to fill in blank number.
9. **Indent Type.** You can define Indent type for a specific file type.

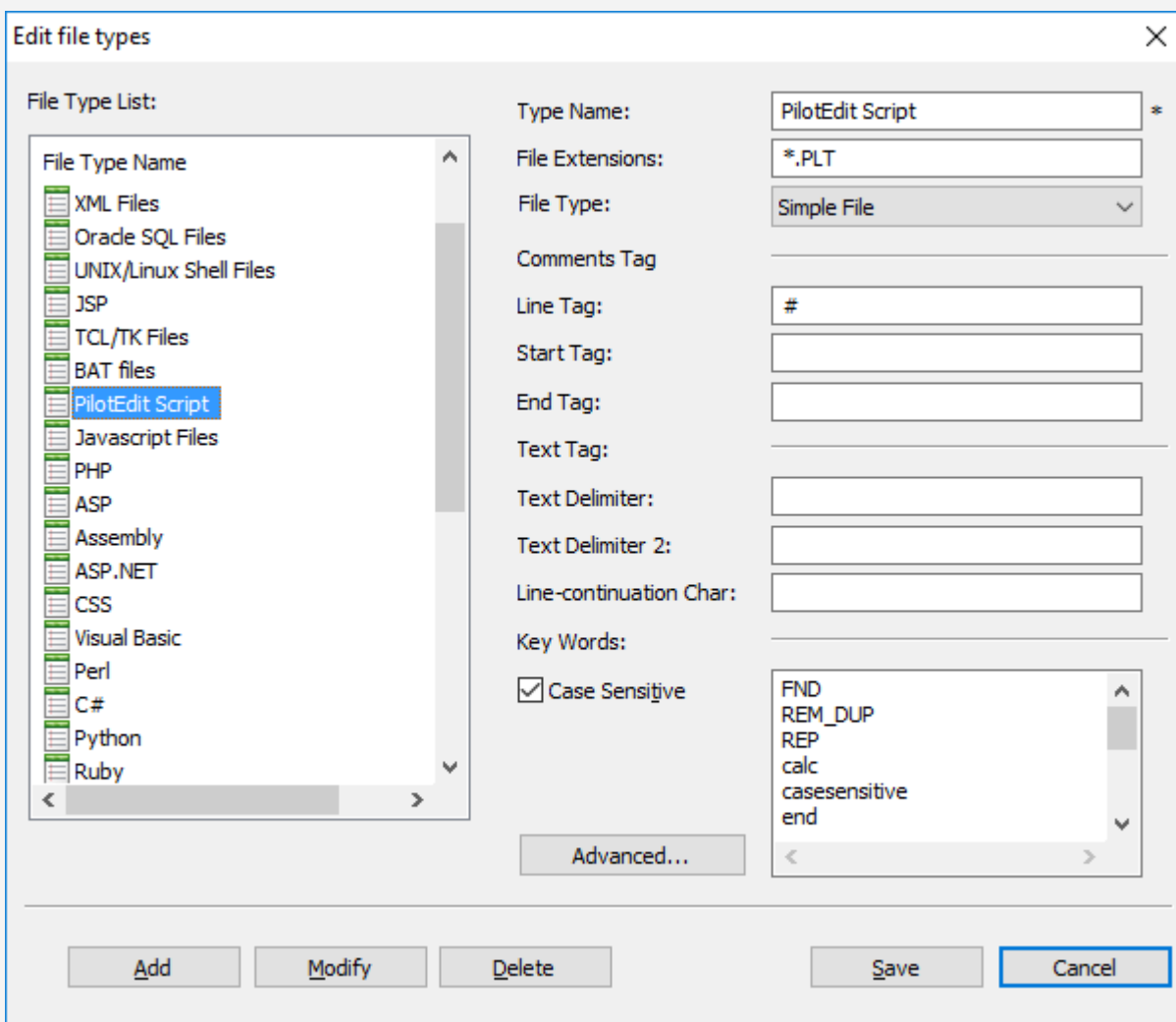
| Tab Type | Notes |
|-------------------------|---|
| User Default Definition | Use the default definition defined in Menu Item Configure , then Options... |
| Use Tab | Use tab |
| Use Blanks | User blanks |

| | |
|-----------|-----------|
| No Indent | No indent |
|-----------|-----------|

10. **Indent Size.** If you choose to use blanks for indent, you need to fill in blank number.

Example 1:

In this example, you will add a file type for PilotEdit script.



After re-start PilotEdit, if you edit a ***.PLT** file, the text color will be changed as below:

```

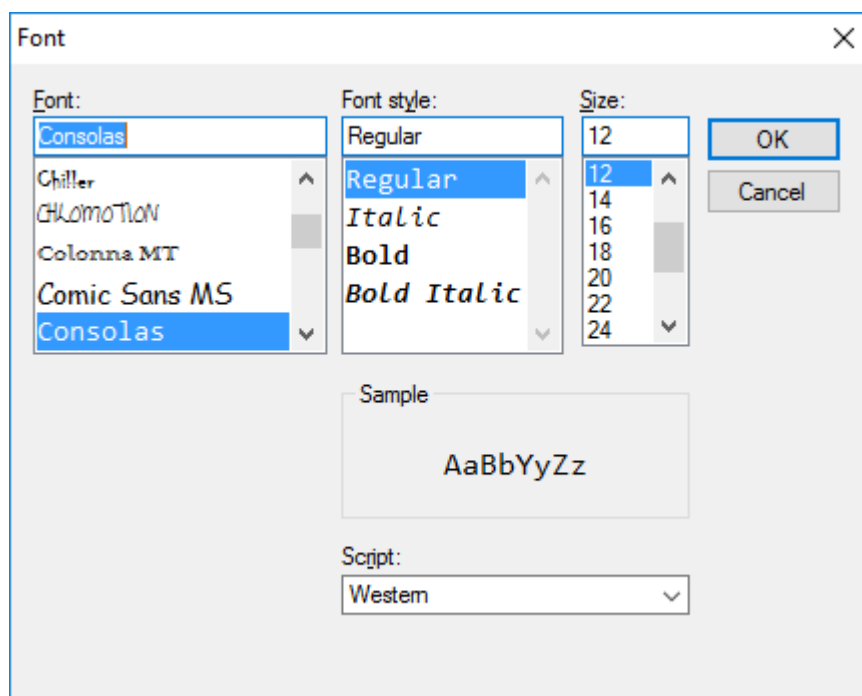
C:\Progr...\ChangeDateFormat.PLT
1 #Execute this scrip to change date from YYYY-MM-DD to MM/DD/YYYY
2 #Steps:
3 #1. Open the target files including date like "2008-05-31"
4 #2. Choose menu "Execute Scripts..." and execute this script "Cha
5 #3. The date will be changed into something like "05/31/2008"
6 FND: [0-9] [0-9] [0-9] [0-9] - [0-9] [0-9] - [0-9] [0-9]
7 REP: %06%07/%09%10/%01%02%03%04
  
```

4.7.3. Set Font...

You can change the font of PilotEdit by selecting the menu item **Configure**, then **Set Font...**

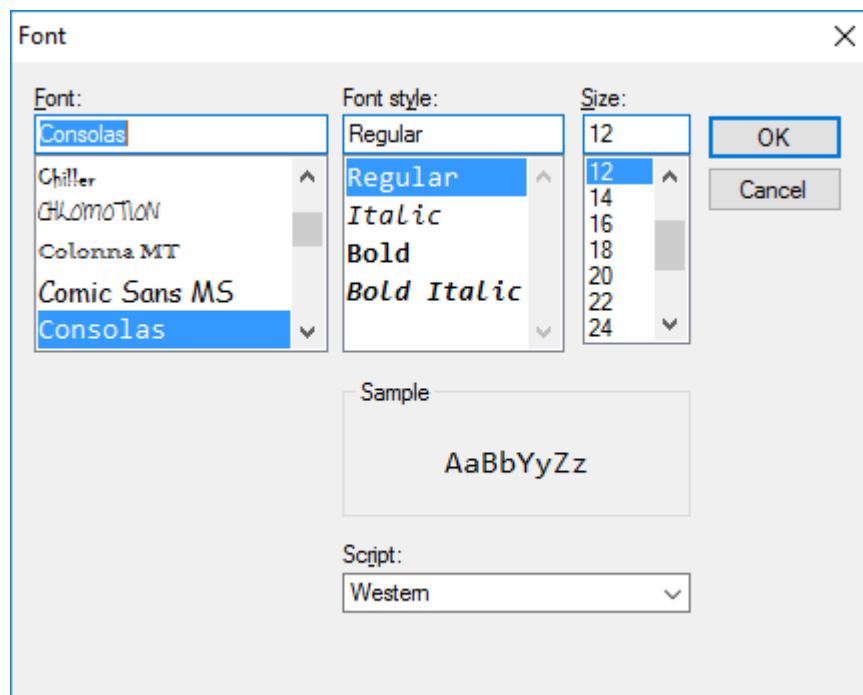
The font in the file tab list, file tree list and output window will also be changed.

To change the font of printing and printing preview, please select menu item **Configure**, then **Set Print Font...**



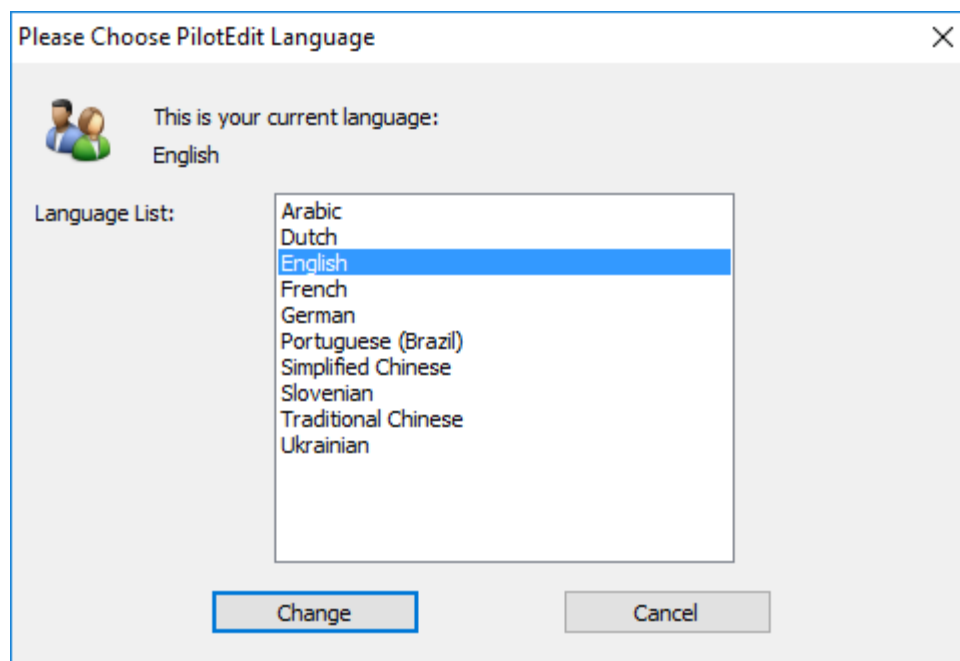
4.7.4. Set Print Font...

You can change the font of printing and printing preview by selecting menu item **Configure**, then **Set Print Font...**



4.7.5. Change Language ...

You can change the language of PilotEdit by selecting menu item **Configure**, then **Change Language...**

**Example 1:**

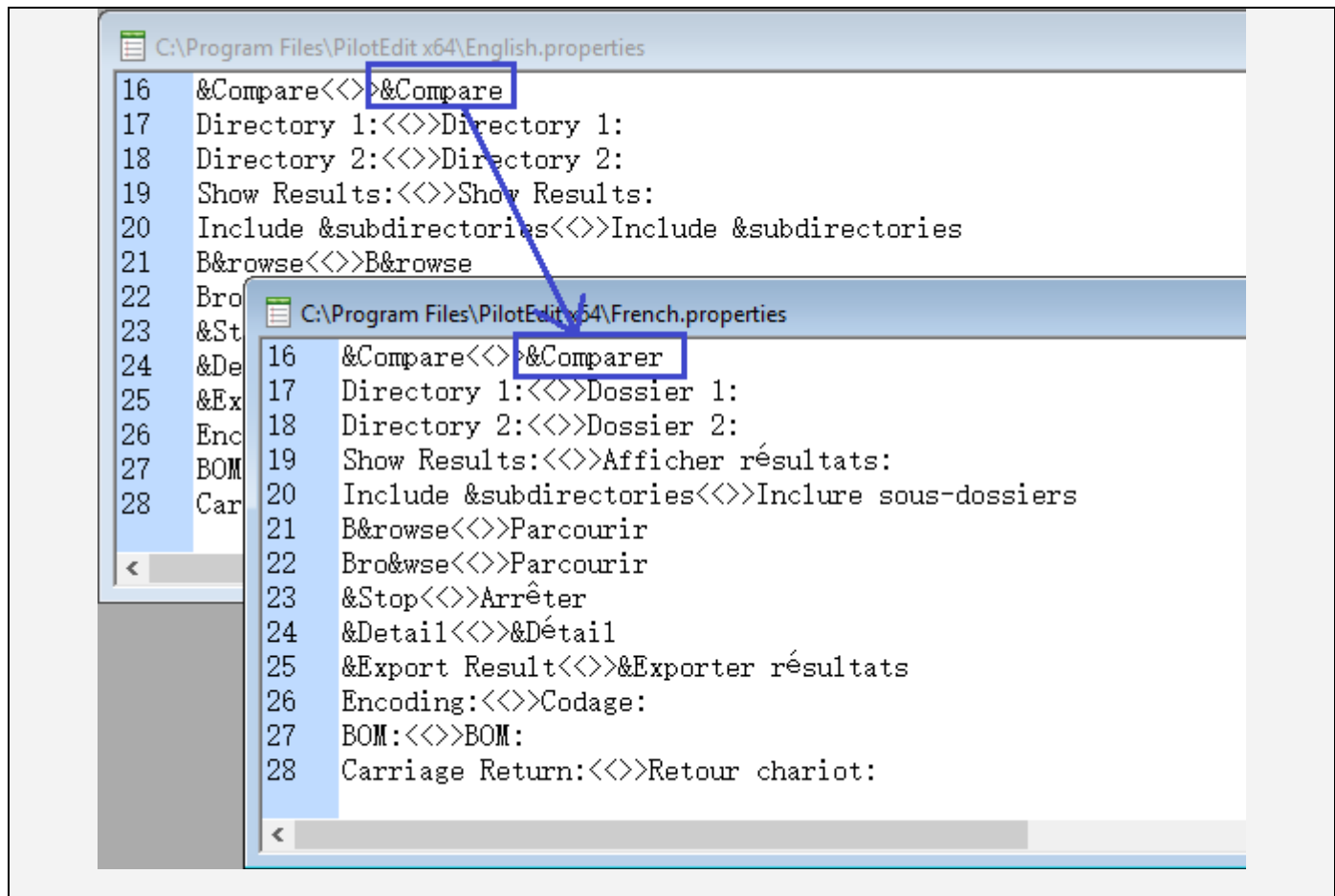
You can change PilotEdit into Simplified Chinese in this way:

1. Select menu item **Configure**, then **Change Language...**
2. Select the language **Simplified Chinese**.
3. The PilotEdit GUI will be changed into Simplified Chinese after re-start it.

Example 2:

How to create a property file for a new language:

1. Open **English.properties** and save it as another *.properties file.
2. Translated the string at the right of <<>> into localized string as shown below.
3. Save this properties file in "Program Files\PilotEdit".



4.7.6. Edit String Table...


You can define your own string table by selecting menu item **Configure**, then **Edit String Table...**

The screenshot shows the 'Edit String Table' dialog box. It has a title bar with a close button (X). The dialog is divided into several sections:

- Category List:** A list box on the left containing various categories like HTML, ASCII Characters, HTML Special Characters, etc. It has 'Move Up' (13), 'Move Down' (14), and 'Delete' (15) buttons below it.
- String List:** A list box in the middle showing strings for the selected category (HTML). It has 'Move Up' (9), 'Move Down' (10), and 'Delete' (11) buttons below it.
- Category Name:** A text field at the top right containing 'HTML' (1).
- String Name:** A text field below the category name containing 'HTML Bold' (2).
- Delete Selected Text:** A checkbox (3) with the label 'Delete Selected Text'.
- String before Selection:** A text area (4) containing ''.
- String after Selection:** A text area (5) containing ''.
- Buttons:** 'Add' (6) and 'Modify' (7) buttons are located below the text areas. 'Save' (16) and 'Cancel' (17) buttons are at the bottom right.

1. Category name
2. String name
3. Select this checkbox if you want the selected text in the active text be removed. If you don't select this checkbox, the string in the edit 4 will be inserted before the selected text and the string in the edit 5 will be inserted after the selected text. (**Note:** This field is not necessary if this string is used as windows command or regular expression)
4. String value. If you choose **NOT to Delete Selected Text**, the string in the edit 4 will be inserted before the selected text.
5. If you choose **NOT to Delete Selected Text**, the string in the edit 5 will be inserted after the selected text. (**Note:** This field is not necessary if this string is used as windows command or regular expression)
6. Click **Add** to add string into string table.
7. Click **Modify** to modify the selected string value. You can't change the category name and string name.
8. String list of the selected category.
9. Click **Move Up** to move the selected strings up.
10. Click **Move Down** to move the selected strings down.
11. Click **Delete** to delete the selected strings.
12. Category list.
13. Click **Move Up** to move the selected categories up.
14. Click **Move Down** to move the selected categories down.
15. Click **Delete** to delete the selected categories.

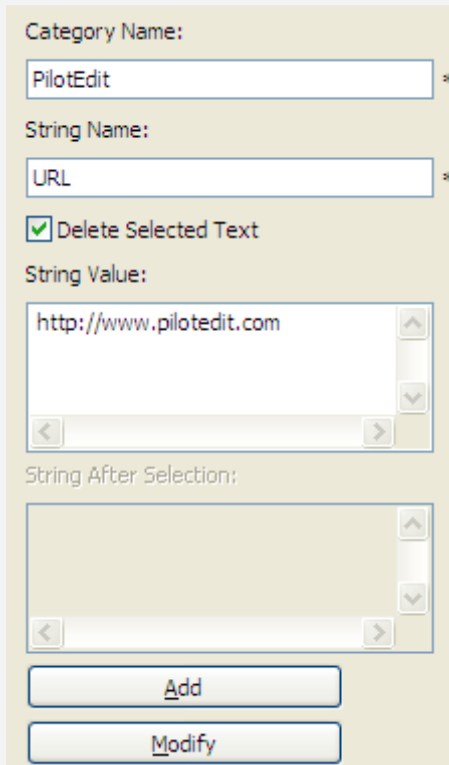
16. When you make any change to the string table, you **MUST** click **Save** to save the changes. Otherwise, your change will be lost after you close this window.
17. Click **Cancel** to close this window.

 String table doesn't work in HEX mode. If you double-click on a string in the string table in HEX mode, PilotEdit will change back to text mode and insert the string.

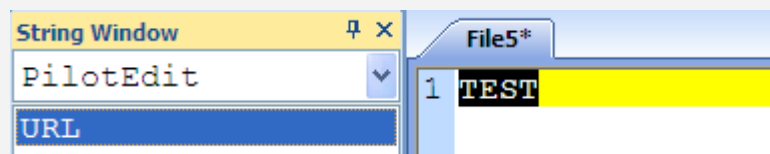
Example 1:

In this example you will define a string and delete the selected text in the active file:

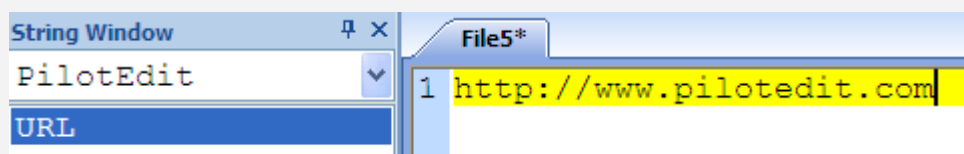
1. Add a string and select **Delete Selected Text**.



2. Select some text in the active file and double click the added string in the string table.



3. The selected text in the active file will be removed and the string from the string table will be inserted into the active file.



Example 2:

In this example you will define a string and preserve the selected text in the active file:

1. Add a string and **don't** select **Delete Selected Text**.

Category Name:
HTML *

String Name:
HTML Bold *

☐ Delete Selected Text

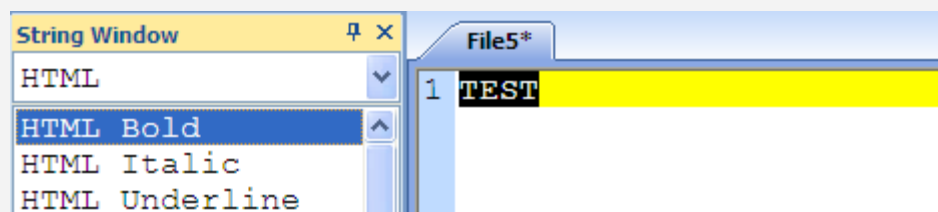
String Before Selection:

String After Selection:

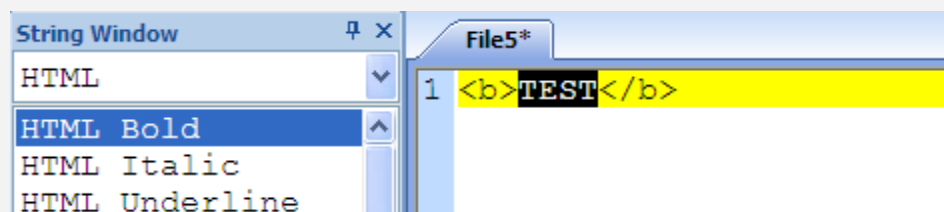
Add

Modify

2. Select some text in the active file and double click the added string in the string table.



3. The first string will be inserted before the selected text and the second string will be inserted after the selected text.



Example 3:

In this example you will define a windows command and run it in string table:

1. Add a windows command **copy /Y "%cp%\%cff%" "%cp%\%cff%.bak"**.

Category Name:
Command Examples *

String Name:
Make a backup of the active file *

☐ Delete Selected Text

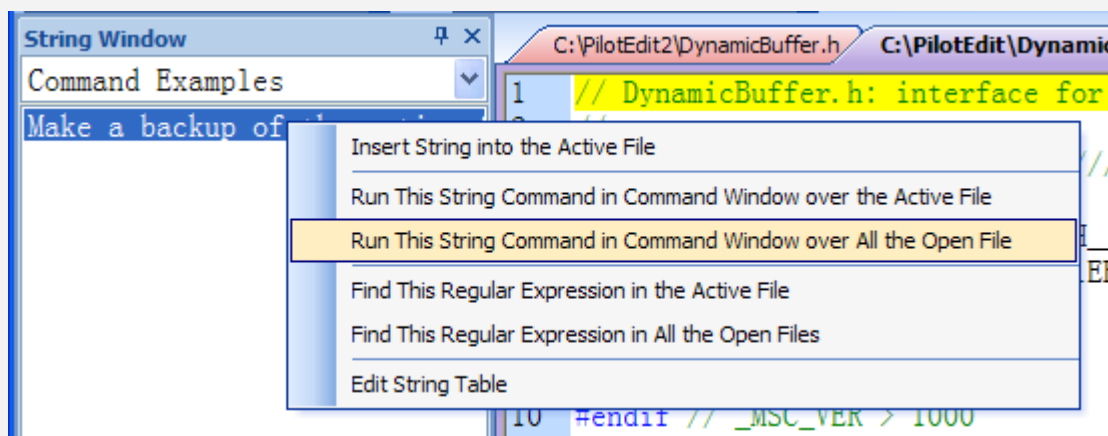
String before Selection:
copy /Y "%cp%\%cff%" "%cp%\%cff%.bak"

String after Selection:

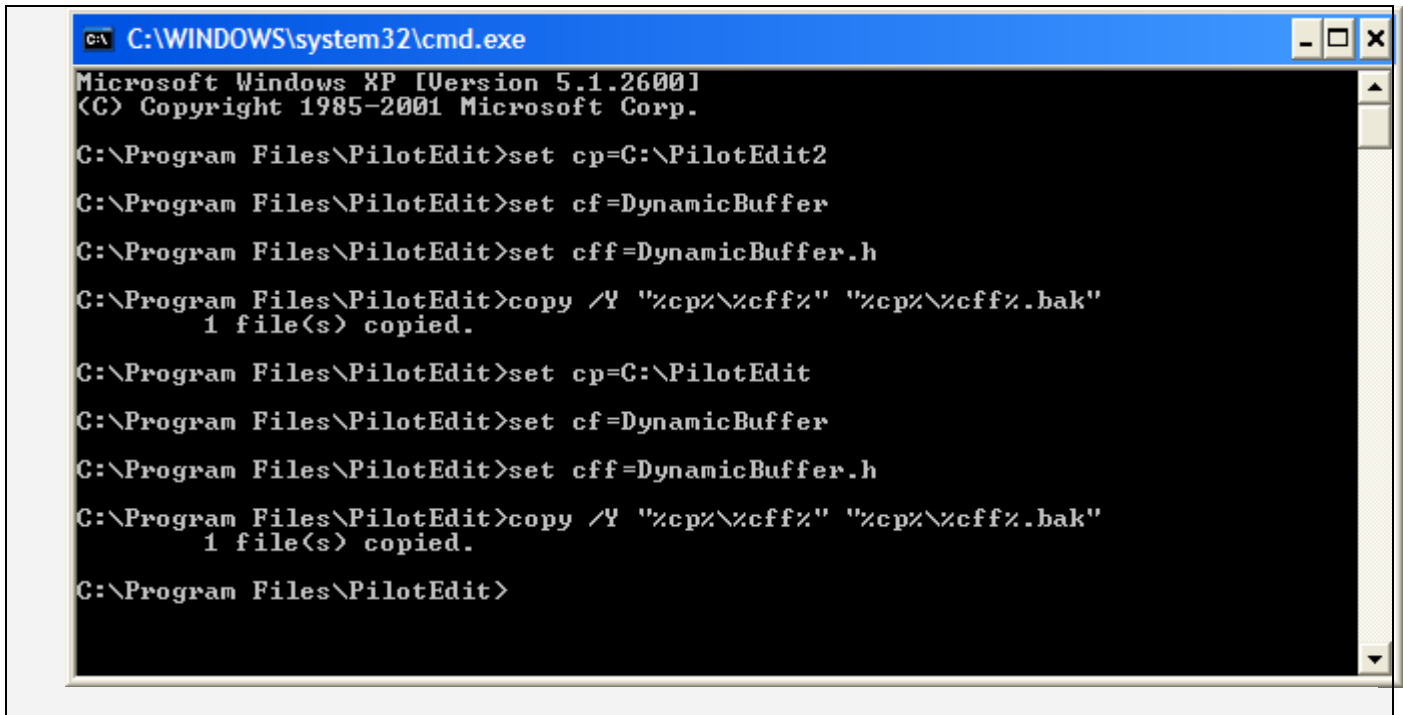
Add

Modify

2. Right click this string in string table to run this command over all the opened files.



3. This operation will create a backup file for each of the opened files.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\PilotEdit>set cp=C:\PilotEdit2
C:\Program Files\PilotEdit>set cf=DynamicBuffer
C:\Program Files\PilotEdit>set cff=DynamicBuffer.h
C:\Program Files\PilotEdit>copy /Y "%cp%\%cff%" "%cp%\%cff%.bak"
1 file(s) copied.
C:\Program Files\PilotEdit>set cp=C:\PilotEdit
C:\Program Files\PilotEdit>set cf=DynamicBuffer
C:\Program Files\PilotEdit>set cff=DynamicBuffer.h
C:\Program Files\PilotEdit>copy /Y "%cp%\%cff%" "%cp%\%cff%.bak"
1 file(s) copied.
C:\Program Files\PilotEdit>
```

Example 4:

In this example you will define a regular expression in the string table to search for multi-line comments in the C/C++/java files.

1. Add a string `/*[*]**/`.

Category Name:
 *

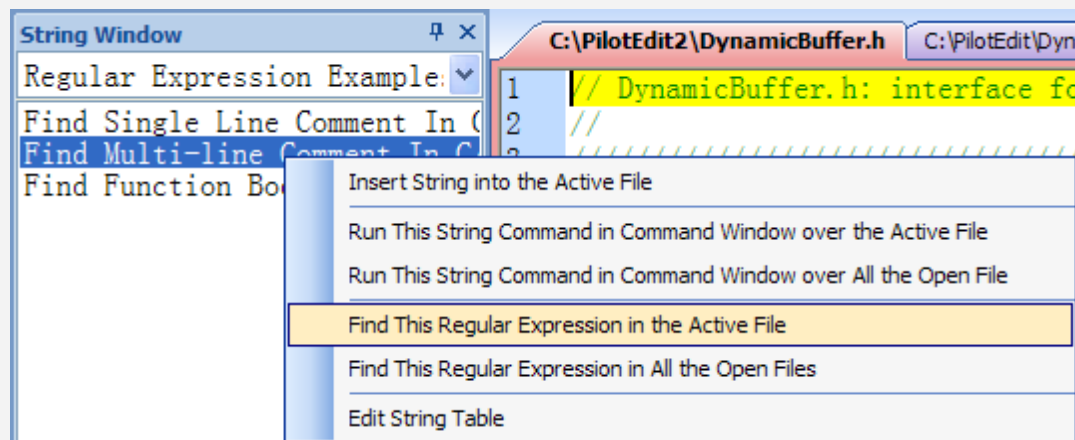
String Name:
 *

☐ Delete Selected Text

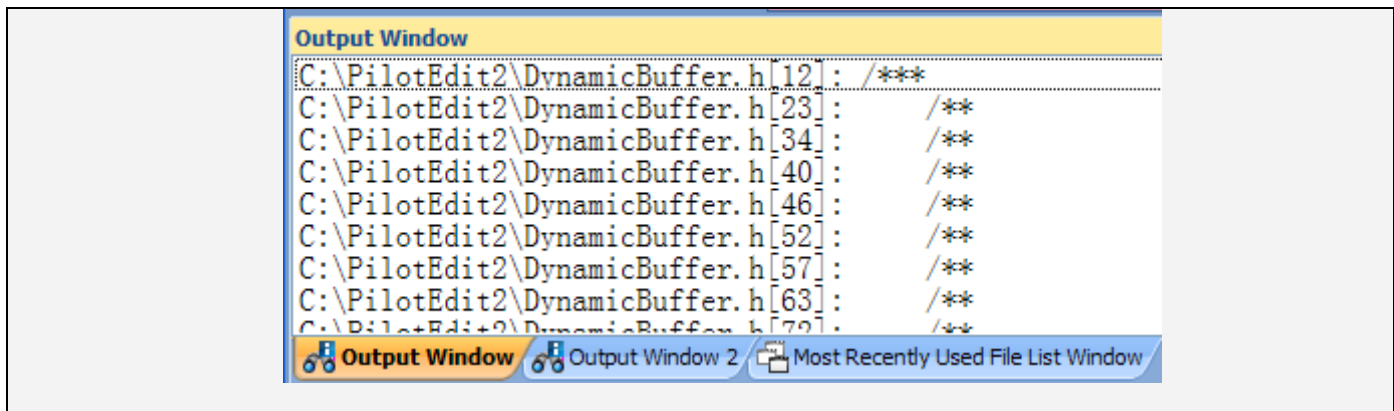
String before Selection:

String after Selection:

- Right click on the string in the string table to search for this regular expression in the active file.

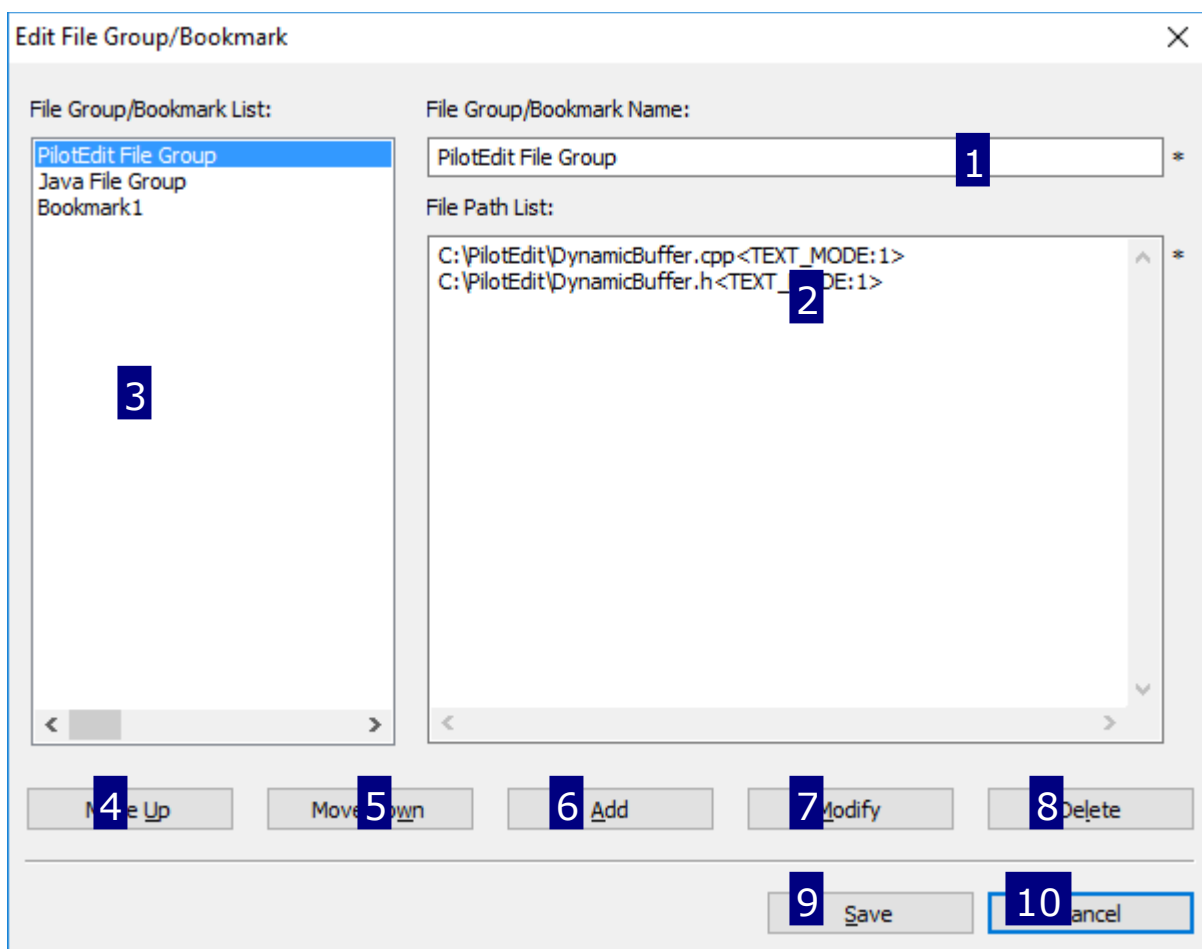


- The multi-line comments matching this regular expression in the active file will be listed in the output window.



4.7.7. Edit File Group/Bookmark...

You can define your own file group/bookmark by selecting menu item **Configure**, then **Edit File Group/Bookmark...**



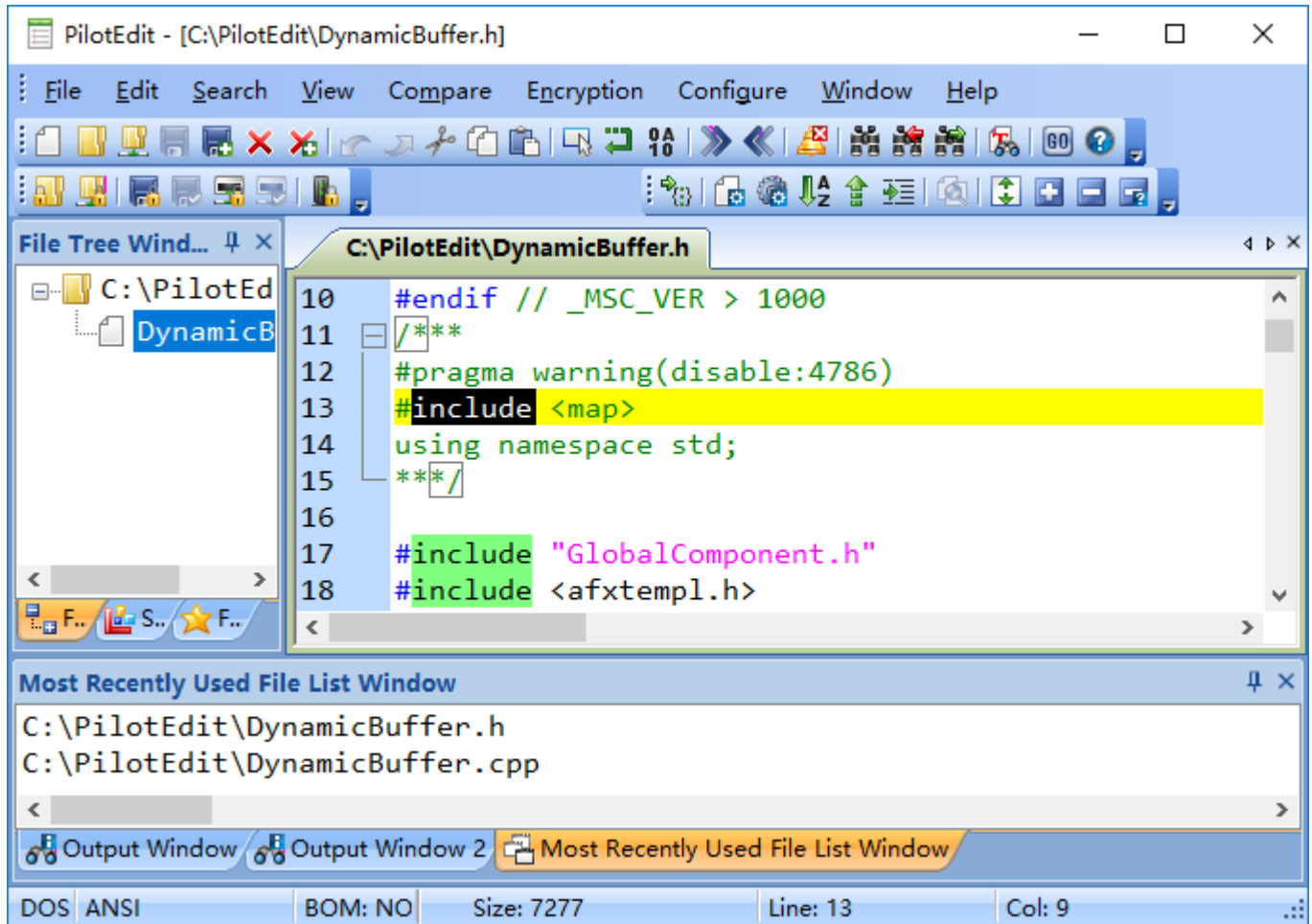
1. File Group/Bookmark Name.
2. File Path List. The files listed here will be opened when you choose to open the file group/bookmark.
3. File Group List.

4. Click **Move Up** to move the selected strings up.
5. Click **Move Down** to move the selected strings down.
6. Click **Add** to add a file group/bookmark into the file group table.
7. Click **Modify** to modify the selected file group/bookmark.
8. Click **Delete** to delete the selected file groups/bookmarks.
9. **When you make any change to the string table, you MUST click Save to save the changes. Otherwise, you change will be lost after you close this window.**
10. Click **Cancel** to close this window.

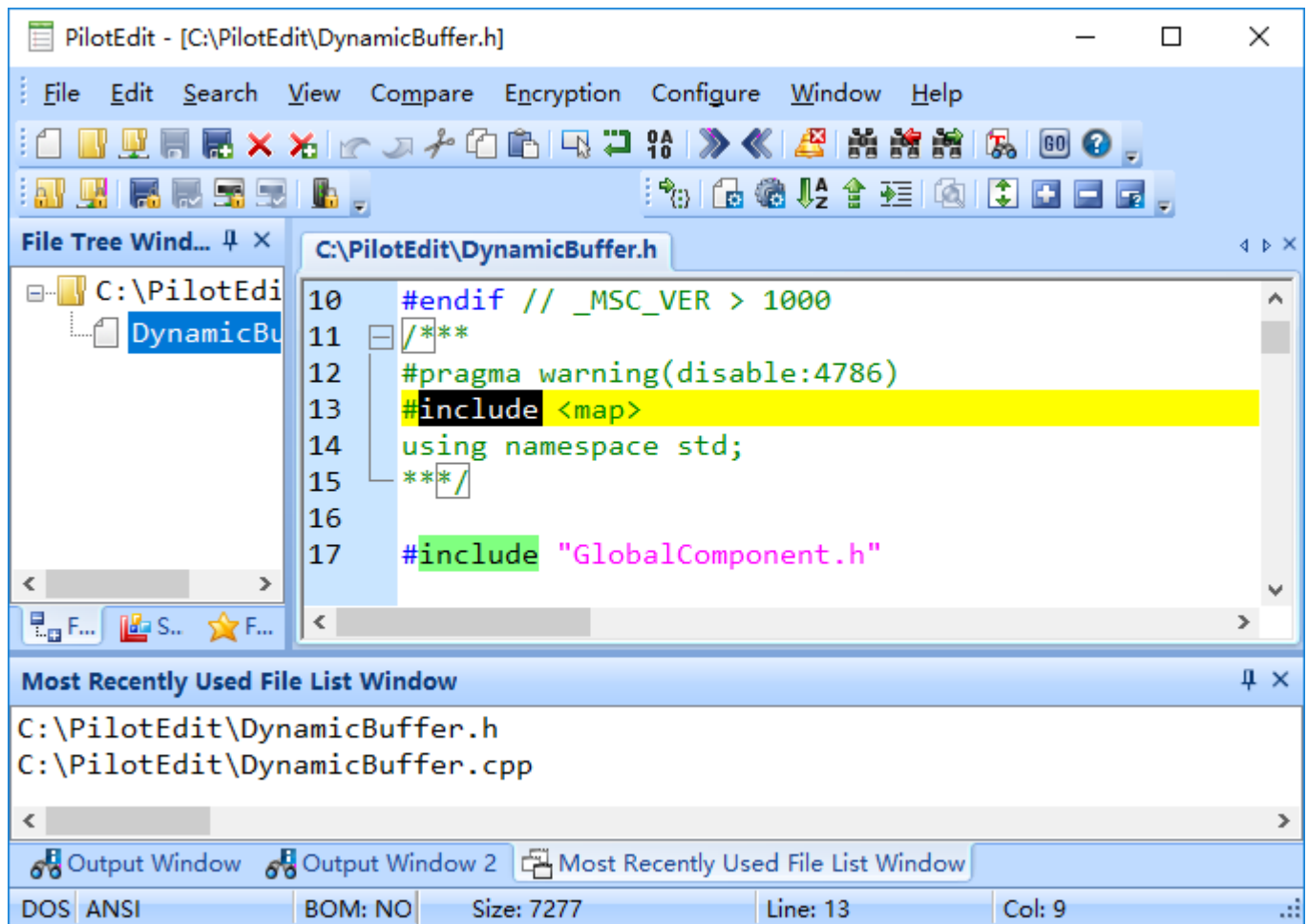
4.7.8. Theme

You can change the theme of PilotEdit by selecting the menu item **Configure**, then **Theme**.

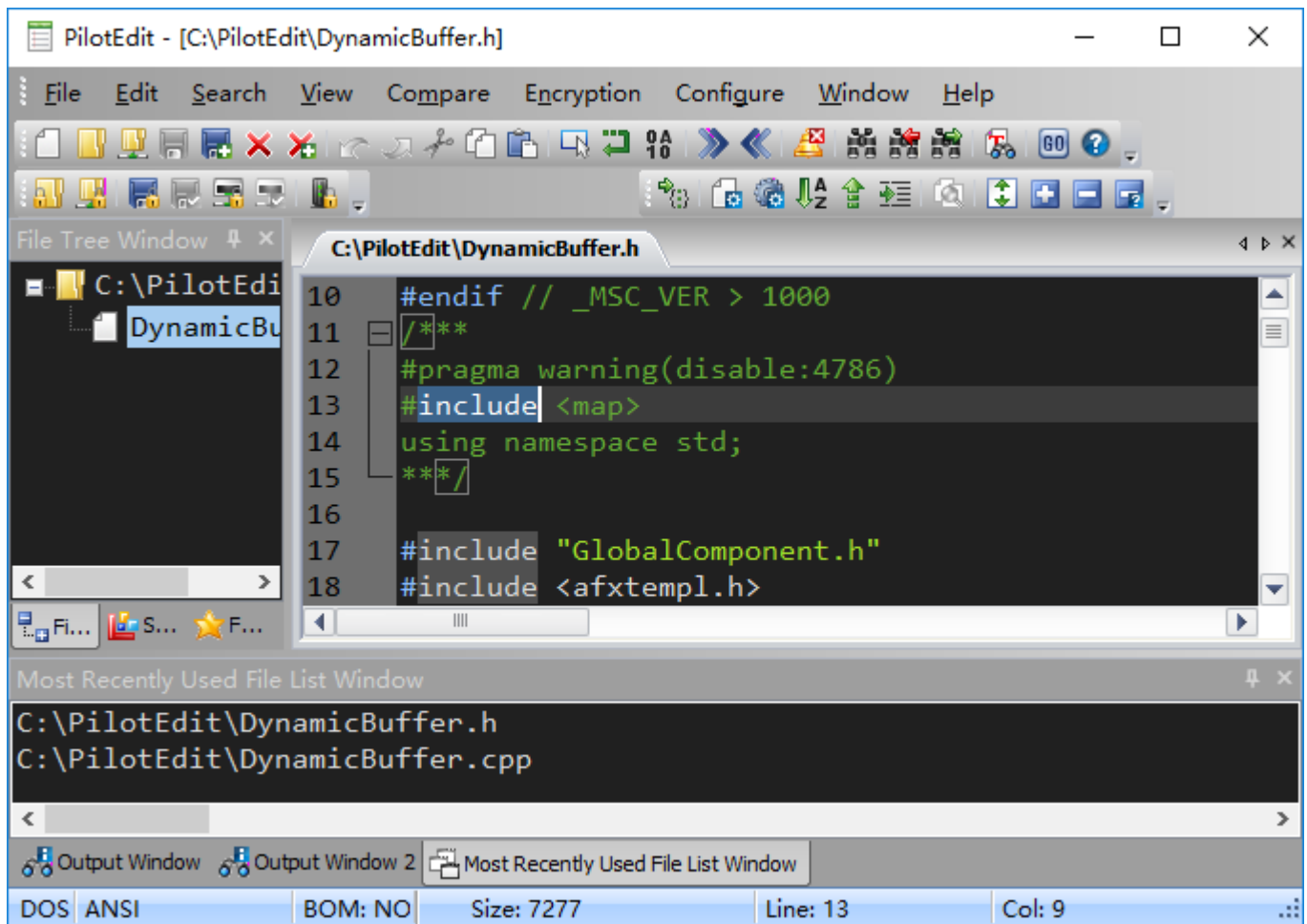
1. Office 2003



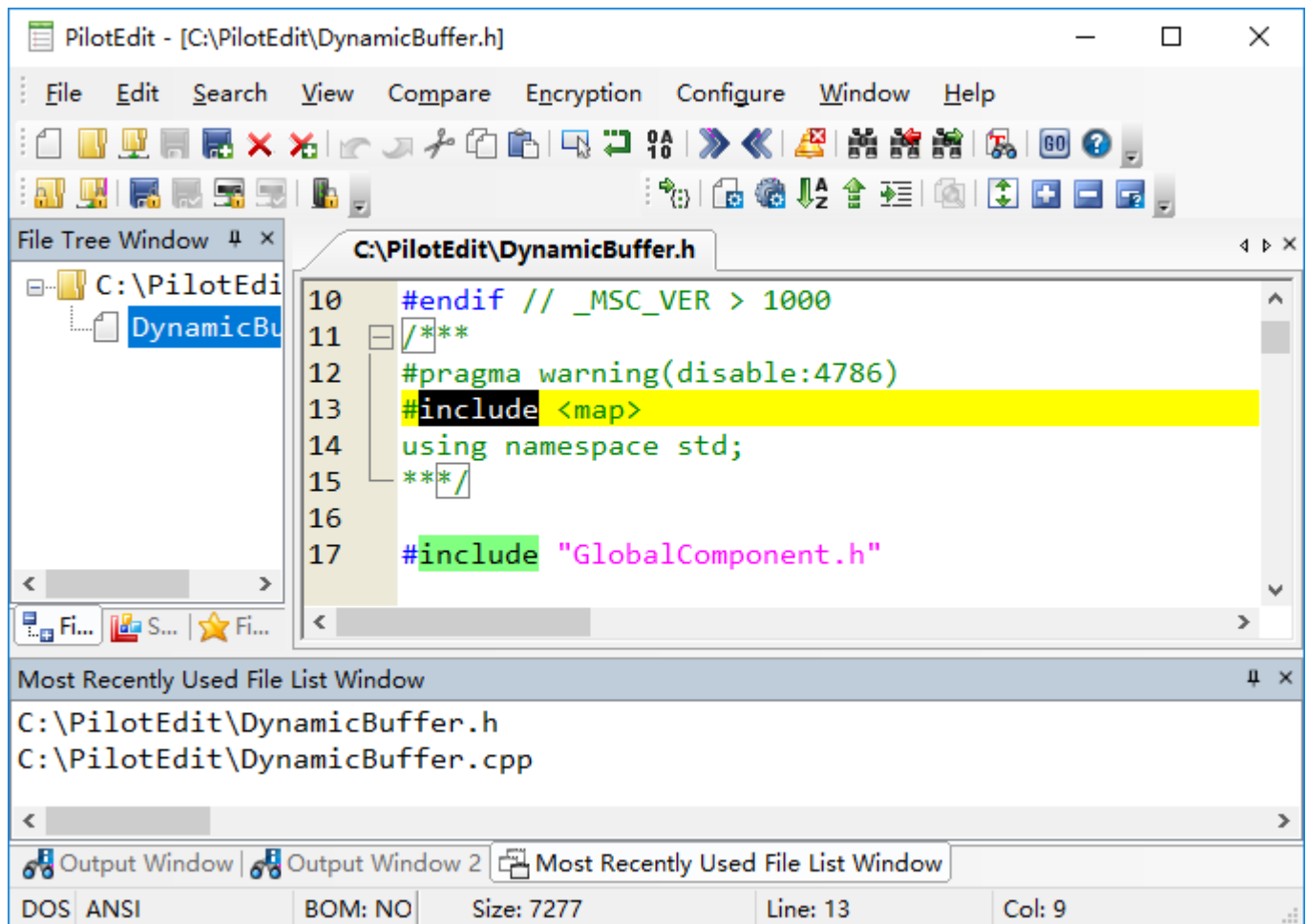
2. Office 2007



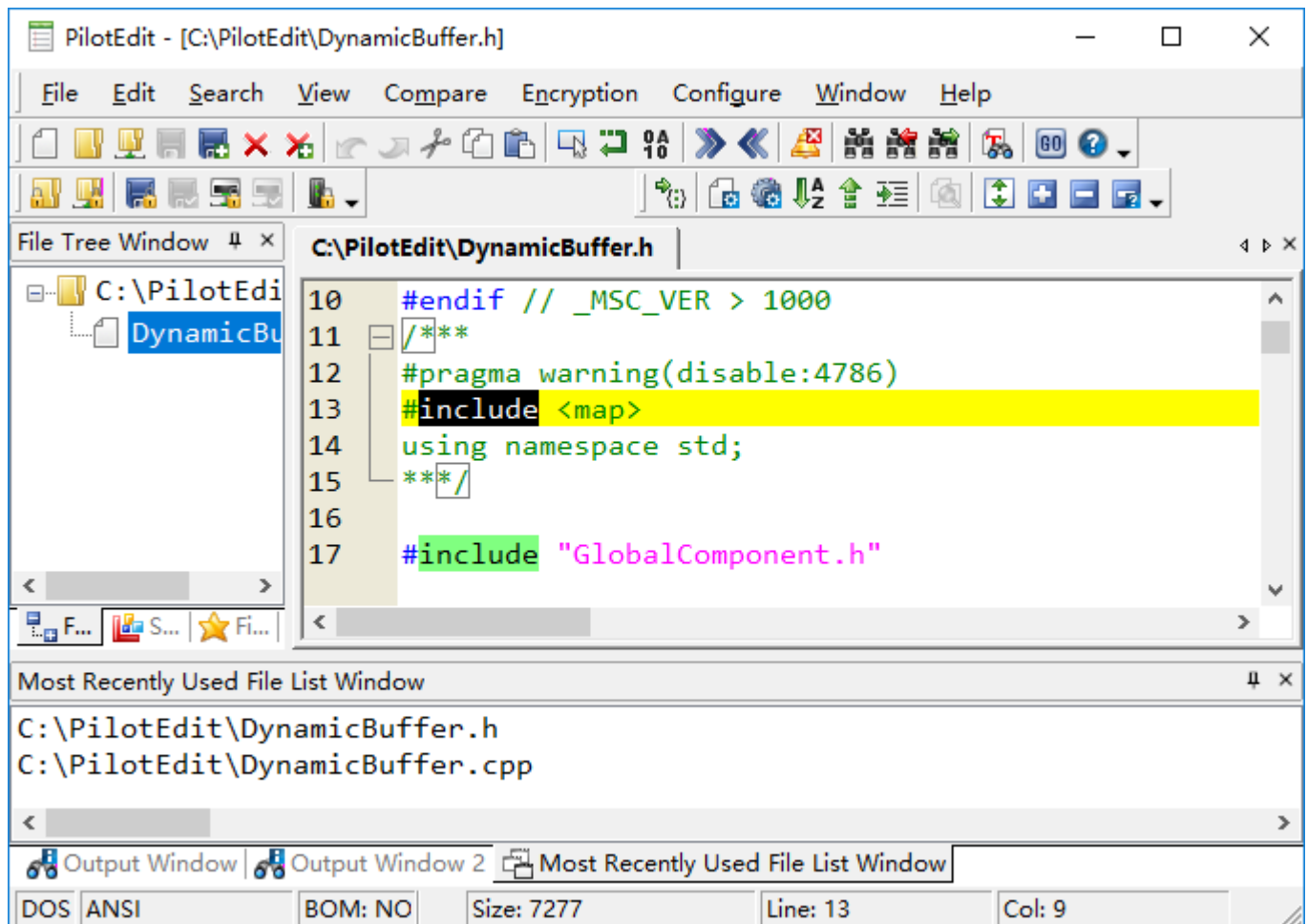
3. Office 2010 Black



4. Visual Studio

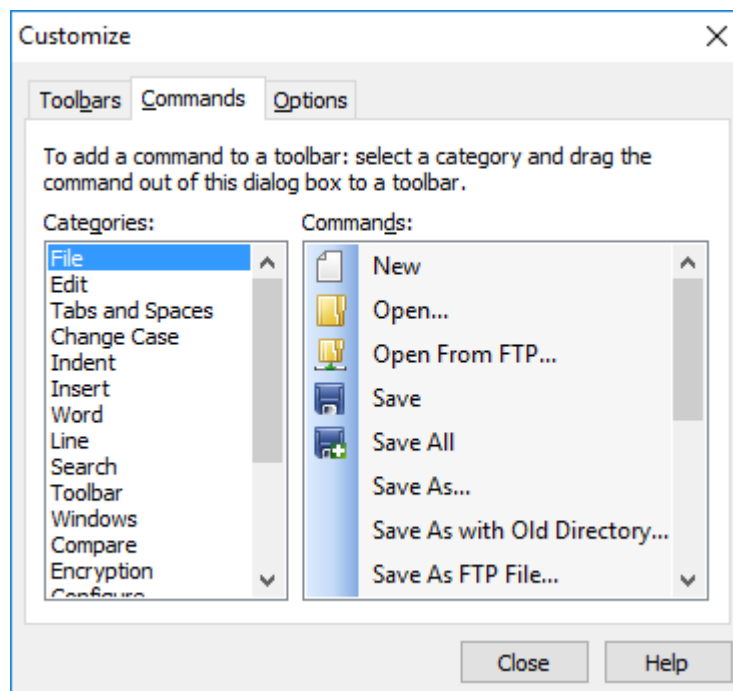


5. Windows Classic



4.7.9. Customize...

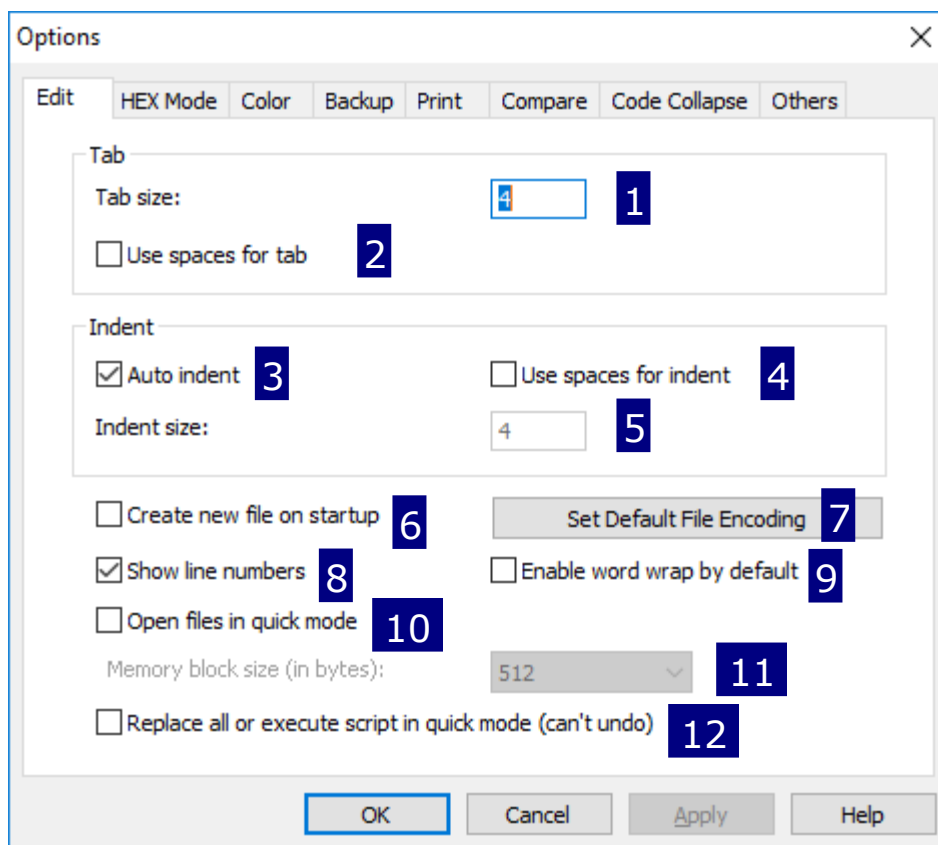
You can customize the toolbar and menu of PilotEdit by selecting the menu item **Configure**, then **Customize...**



4.7.10. Options...

You can change the options of PilotEdit by selecting the menu item **Configure**, then **Options...**


Edit



1. Change the width of a TAB key displayed in PilotEdit editor.
2. If this is selected, a user will be able to input blanks by clicking on TAB.
3. Automatically indent.
4. If this is selected, PilotEdit will use blanks for indent.
5. The number of blanks for indent.
6. If this is selected, a new file will be created on startup.
7. Set the default encoding of the new files or files without BOM.
8. Show or hide line number.
9. Enable word wrap by default.
10. Open files in quick mode. *
11. Memory block size (in bytes). It is recommended to select quick mode ONLY when you run out of memory. Selecting quick mode and a big memory block size will reduce the time and memory spent in opening a huge file but slow down some other operations. Please change to a small memory block size if you find some operations too slow. *
12. You can choose Quick Mode to replace millions of strings in a huge file. If you choose Quick Mode, you can't undo the changes with Ctrl + Z after replacing all or executing PilotEdit Script. *

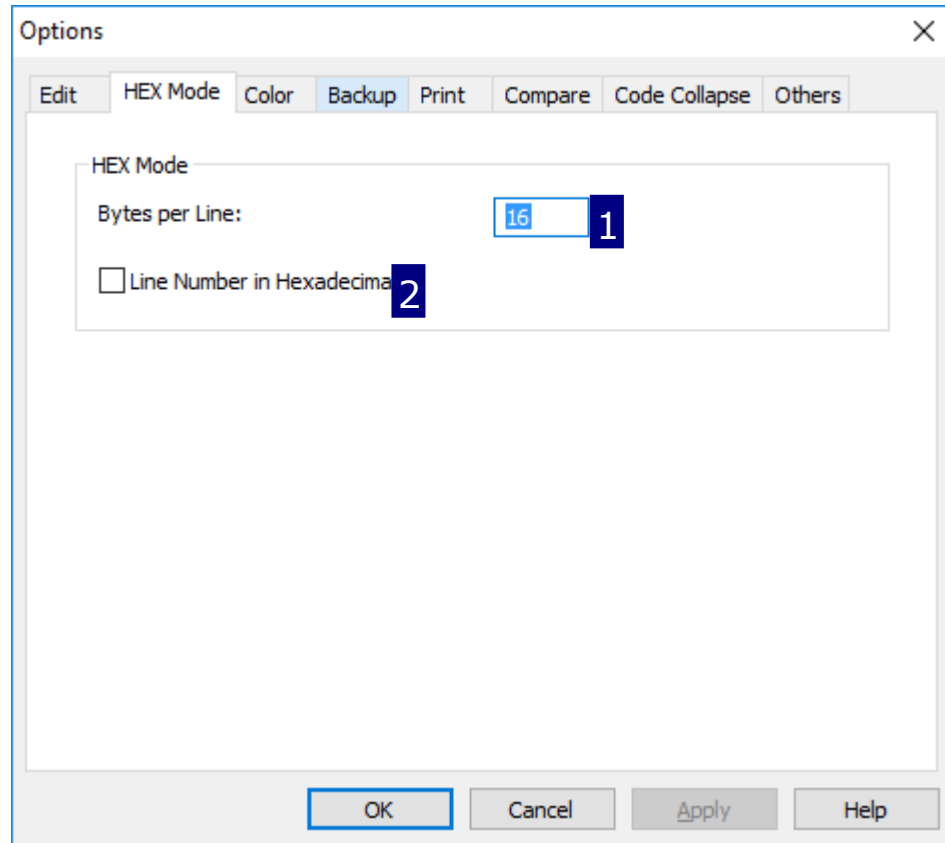
| File Size | Recommended Quick Mode Configuration |
|--------------------|---|
| File Size < 50 GB | Use non-quick mode |
| File Size < 500 GB | Use quick mode and select memory block size 512, 1024 or 2048 bytes |
| File Size > 500 GB | Use quick mode and select memory block size 1024 or bigger bytes |

12. You can choose Quick Mode to replace millions of strings in a huge file. If you choose Quick Mode, you can't undo the changes with Ctrl + Z after replacing all or executing PilotEdit Script. *

 You can define TAB and Indent for a specified file type by selecting the menu item **Configure**, then **File Type...**

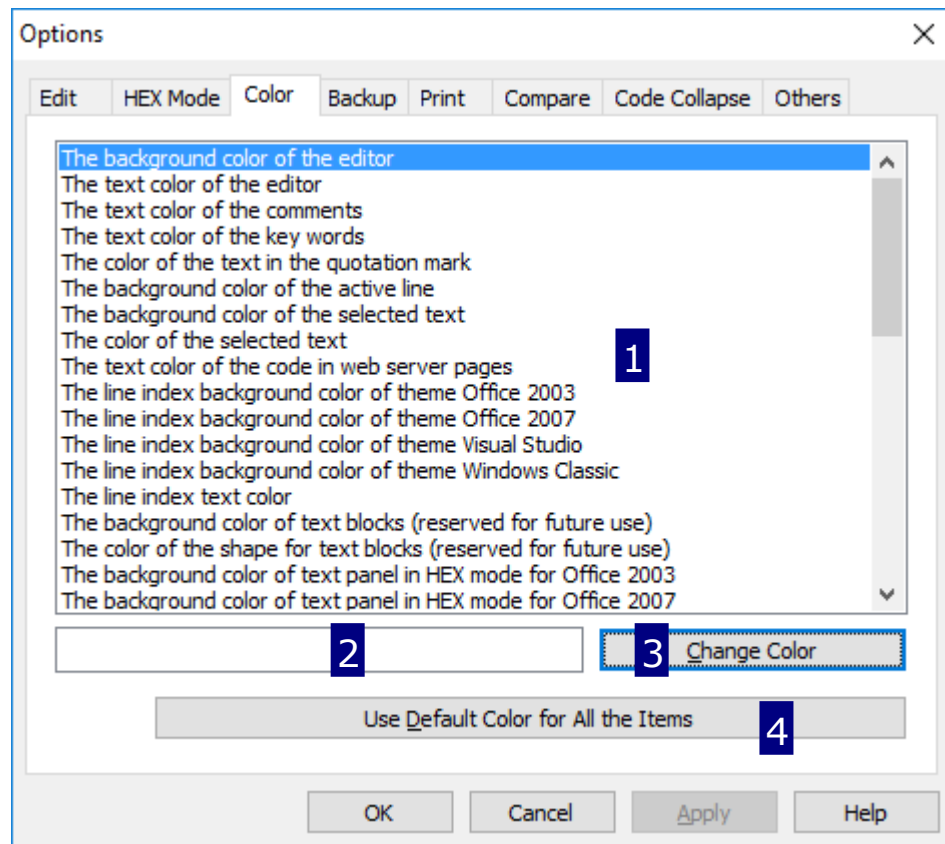
TAB and Indent specified in menu item **Configure**, then **File Type...** has the highest priority.

HEX Mode



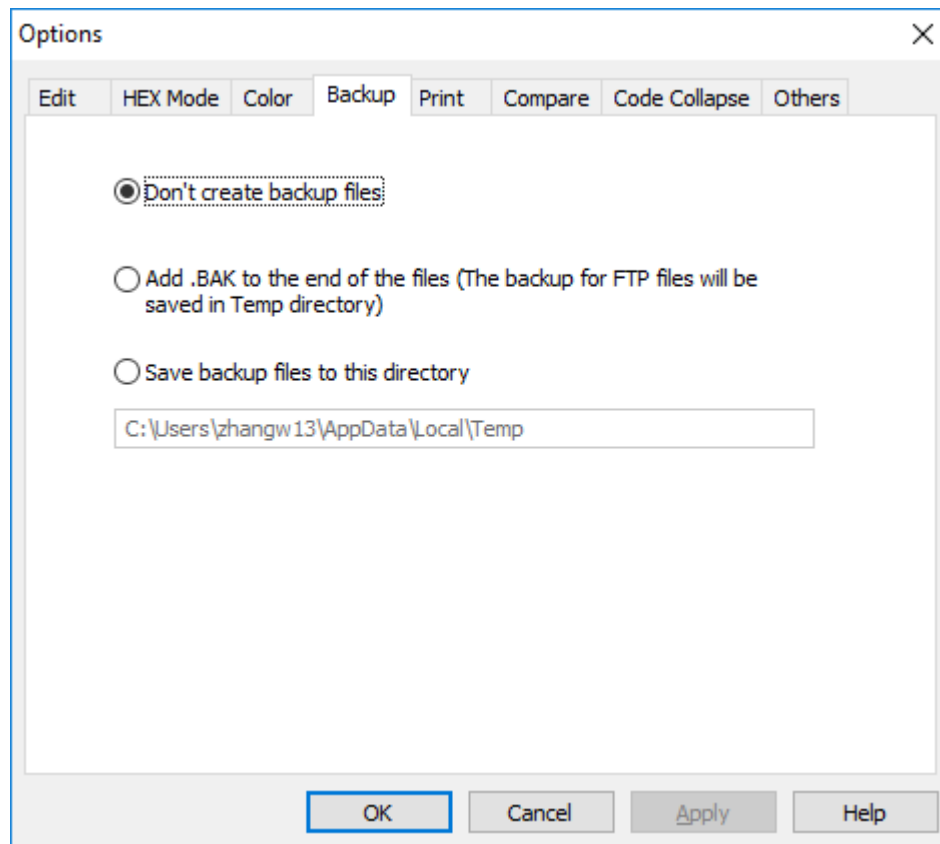
1. The number of bytes displayed in each line in HEX mode.
2. If this is selected, the line number in HEX mode will be displayed as a hex-decimal string.

Color



1. **Color List.**
2. Color of the selected item in **Color List.**
3. Click this button to change the color of the selected item in **Color List.**
4. Click this button to change to the default color settings of PilotEdit.

Backup



Print

The screenshot shows the 'Options' dialog box with the 'Print' tab selected. The dialog has a title bar with a close button (X) and a tab bar with the following tabs: Edit, HEX Mode, Color, Backup, Print (selected), Compare, Code Collapse, and Others. The main content area is divided into two sections: 'Margins (inches)' and 'Header and Footer'. In the 'Margins' section, there are four text input fields: 'Left' (0.30), 'Right' (0.30), 'Top' (0.30), and 'Bottom' (0.30). In the 'Header and Footer' section, there are two checked checkboxes: 'Print header' and 'Print footer'. Below 'Print header' are two dropdown menus: 'Left top' (File name) and 'Right top' (empty). Below 'Print footer' are two dropdown menus: 'Left bottom' (empty) and 'Right bottom' (Page number). At the bottom of the dialog are four buttons: OK, Cancel, Apply, and Help.

Options

Edit HEX Mode Color Backup **Print** Compare Code Collapse Others

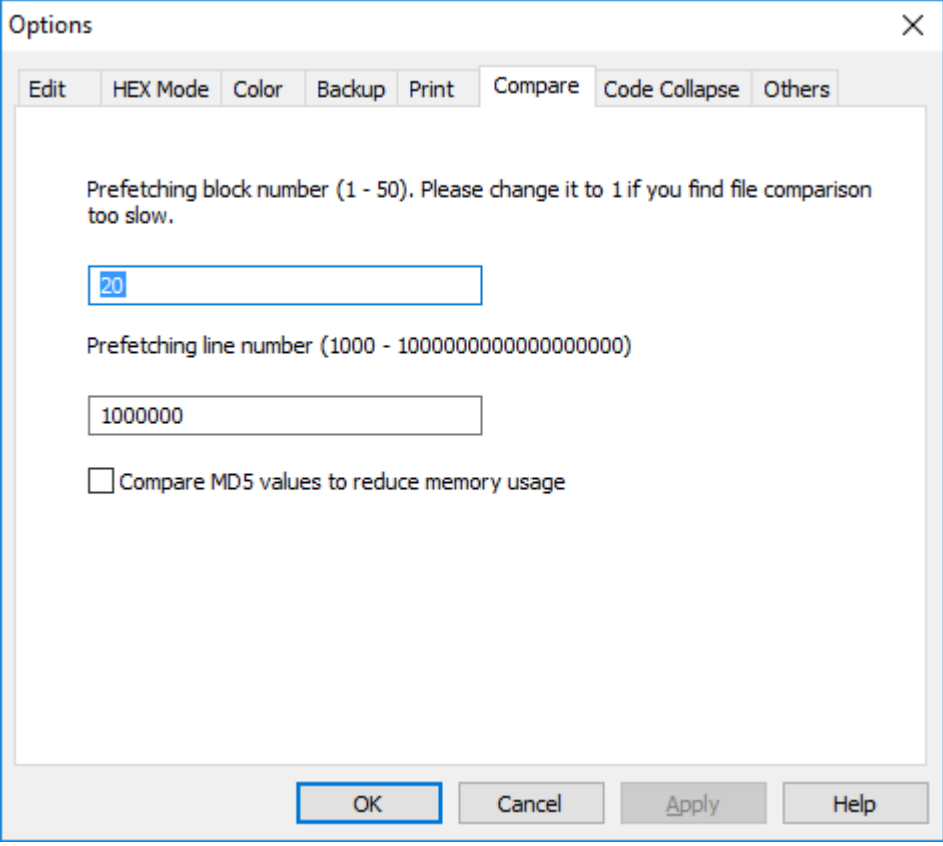
Margins (inches)

Left: 0.30 Top: 0.30
Right: 0.30 Bottom: 0.30

Header and Footer

☒ Print header
Left top: File name Right top:
☒ Print footer
Left bottom: Right bottom: Page number

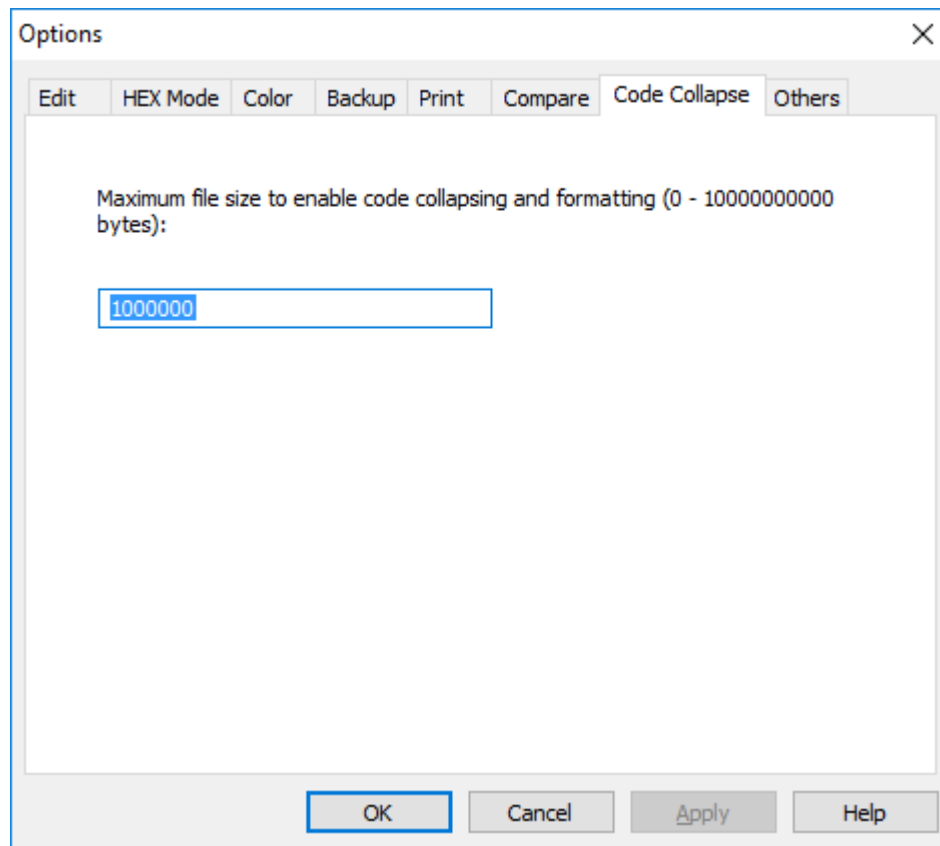
OK Cancel Apply Help

Compare *

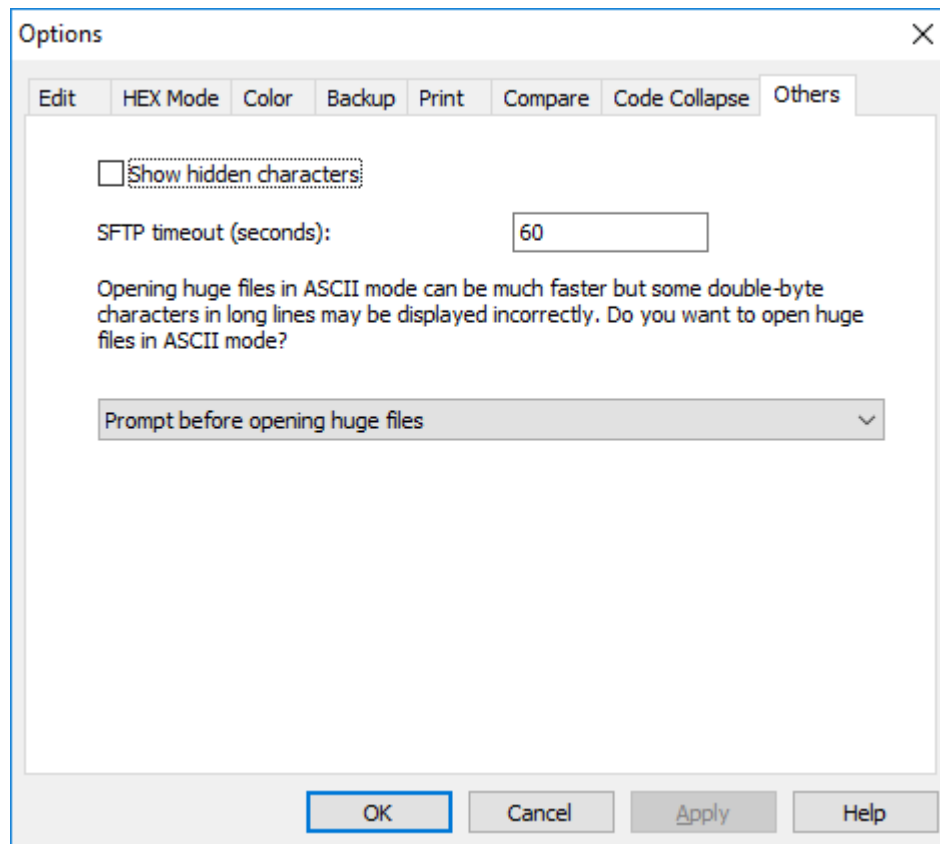
The screenshot shows the 'Options' dialog box with the 'Compare' tab selected. The dialog has a title bar with a close button (X). Below the title bar is a tabbed interface with tabs for 'Edit', 'HEX Mode', 'Color', 'Backup', 'Print', 'Compare', 'Code Collapse', and 'Others'. The 'Compare' tab is active and contains the following settings:

- A text label: "Prefetching block number (1 - 50). Please change it to 1 if you find file comparison too slow."
- A text input field containing the value "20".
- A text label: "Prefetching line number (1000 - 10000000000000000000)"
- A text input field containing the value "1000000".
- A checkbox labeled "Compare MD5 values to reduce memory usage", which is currently unchecked.

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

Code Collapse *

Others



5. Regular Expression and PilotEdit Script

5.1. Create Regular Expression with GUI

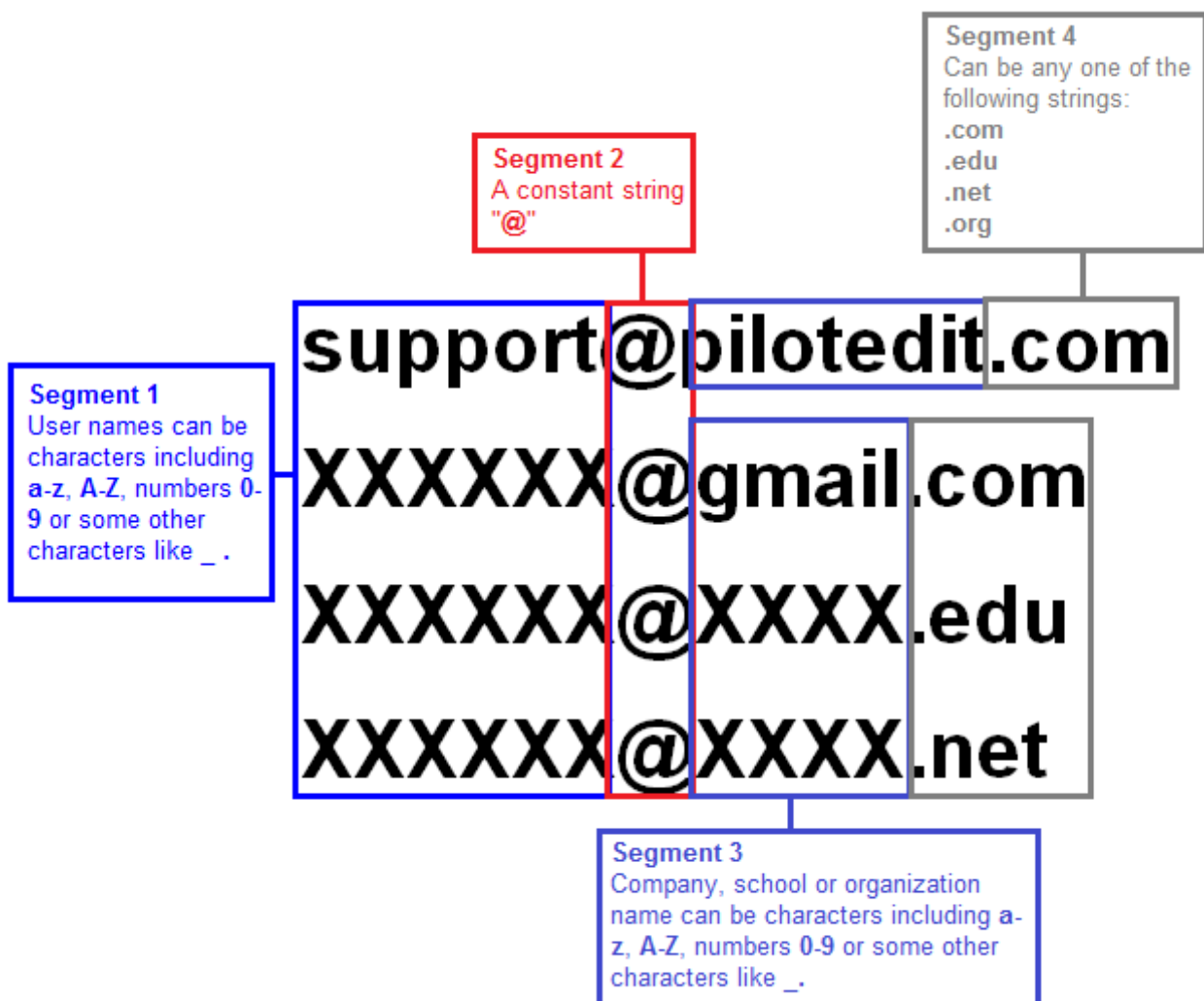
A PilotEdit regular expression includes several segments.

The match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.

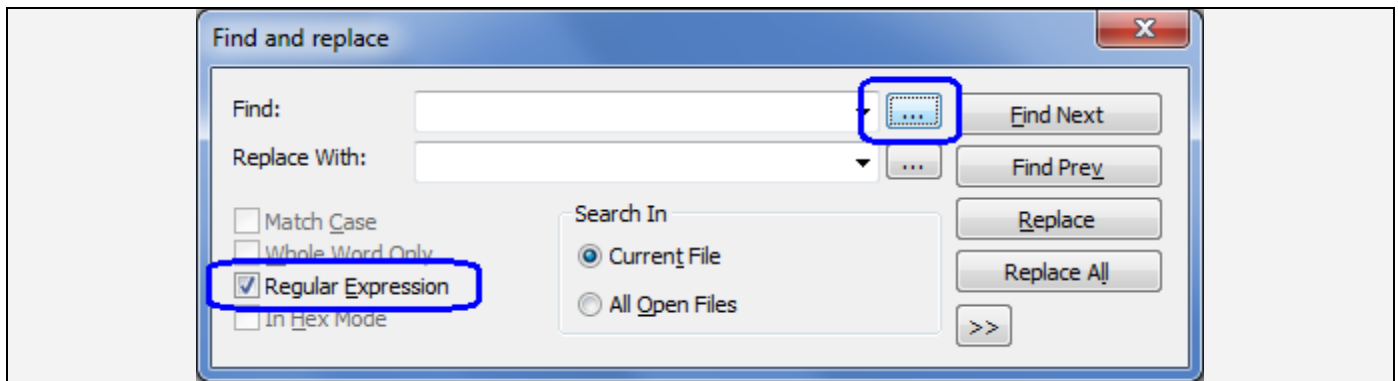
In the following example, you will use PilotEdit regular expression to replace all email addresses to "XXXX@gmail.com".

Example 1:

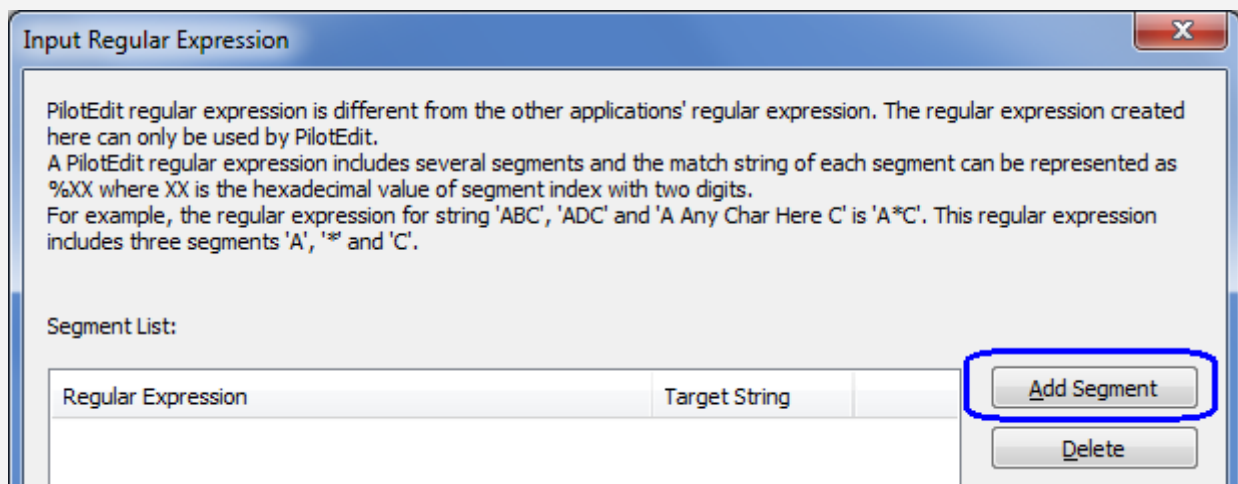
An email address includes four segments as shown below.



Select "**Regular Expression**" and click button "... " to create regular expression with GUI.



Click button "**Add Segment**" to add segments.



Select "**Match Characters**", select "**Character Type:**" and click button "**Include**" to include "**a-z**", "**A-Z**" and "**0-9**".

Add New Segment

☐ Match a Constant String

String:

☒ Match Characters

Character Type:

Character List:

Character Regular Expression:

☒ Match One of Multiple Strings

String:

String Regular Expression:

☐ Case Sensitive

Occurrences:

☐ Match Line Begin ☐ Match Line End

Regular Expression:

Select **"Character List"**, fill in **"_."** and click button **"Include"** to include **"_."**.
Select **"Match One or More Occurrences"** and click **"Add"** to add segment 1.

Add New Segment

☐ Match a Constant String

String:

☒ Match Characters

Character Type:

Character List:

Character Regular Expression:

☐ Match One of Multiple Strings

String:

String Regular Expression:

☐ Case Sensitive

Occurrences:

☐ Match Line Begin ☐ Match Line End

Regular Expression:

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|---|---------------|
| <input type="text" value="[a-z A-Z 0-9 _]+"/> | %01 |

Select "**Match a Constant String**", fill in "@" and click "**Add**" to add segment 2.

Add New Segment

☒ Match a Constant String

String: @

☐ Match Characters

Character Type: Any Characters Except CR/LF

Character List: Include Exclude

Character Regular Expression: Clean

☐ Match One of Multiple Strings

String: Add

String Regular Expression: Clean

☐ Case Sensitive

Occurrences: Match Zero or More Occurrences

☐ Match Line Begin ☐ Match Line End

Regular Expression: \\@

Add Cancel

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.

A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.

For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| [a-z A-Z 0-9 _ .]+ | %01 |
| \@ | %02 |

Add Segment

Delete

Move Up

Move Down

Regular Expression:

[a-z|A-Z|0-9|_|.]+\@

Target String:

%01%02

Save

Cancel

Add segment 3 for company, school or organization name.

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| [a-z A-Z 0-9 _]+ | %01 |
| \@ | %02 |
| [a-z A-Z 0-9 _]+ | %03 |

Regular Expression: [a-z|A-Z|0-9|_]+\@[a-z|A-Z|0-9|_]+

Target String: %01%02%03

Select "**Match One of Multiple Strings**" and add string ".com", ".edu" and ".org".
 Select "**Match Only One Occurrence**" and click "**Add**" to add segment 4.

Add New Segment

☐ Match a Constant String

String:

☐ Match Characters

Character Type:

Character List:

Character Regular Expression:

☒ Match One of Multiple Strings

String:

String Regular Expression:

☐ Case Sensitive

Occurrences:

☐ Match Line Begin ☐ Match Line End

Regular Expression:

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.

A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.

For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| [a-z A-Z 0-9 _ .]+ | %01 |
| \@ | %02 |
| [a-z A-Z 0-9 _ .]+ | %03 |
| {.com .edu .org}? | %04 |

Add Segment

Delete

Move Up

Move Down

Regular Expression:

[a-z|A-Z|0-9|_|.]+\@[a-z|A-Z|0-9|_|.]+{.com|.edu|.org}?

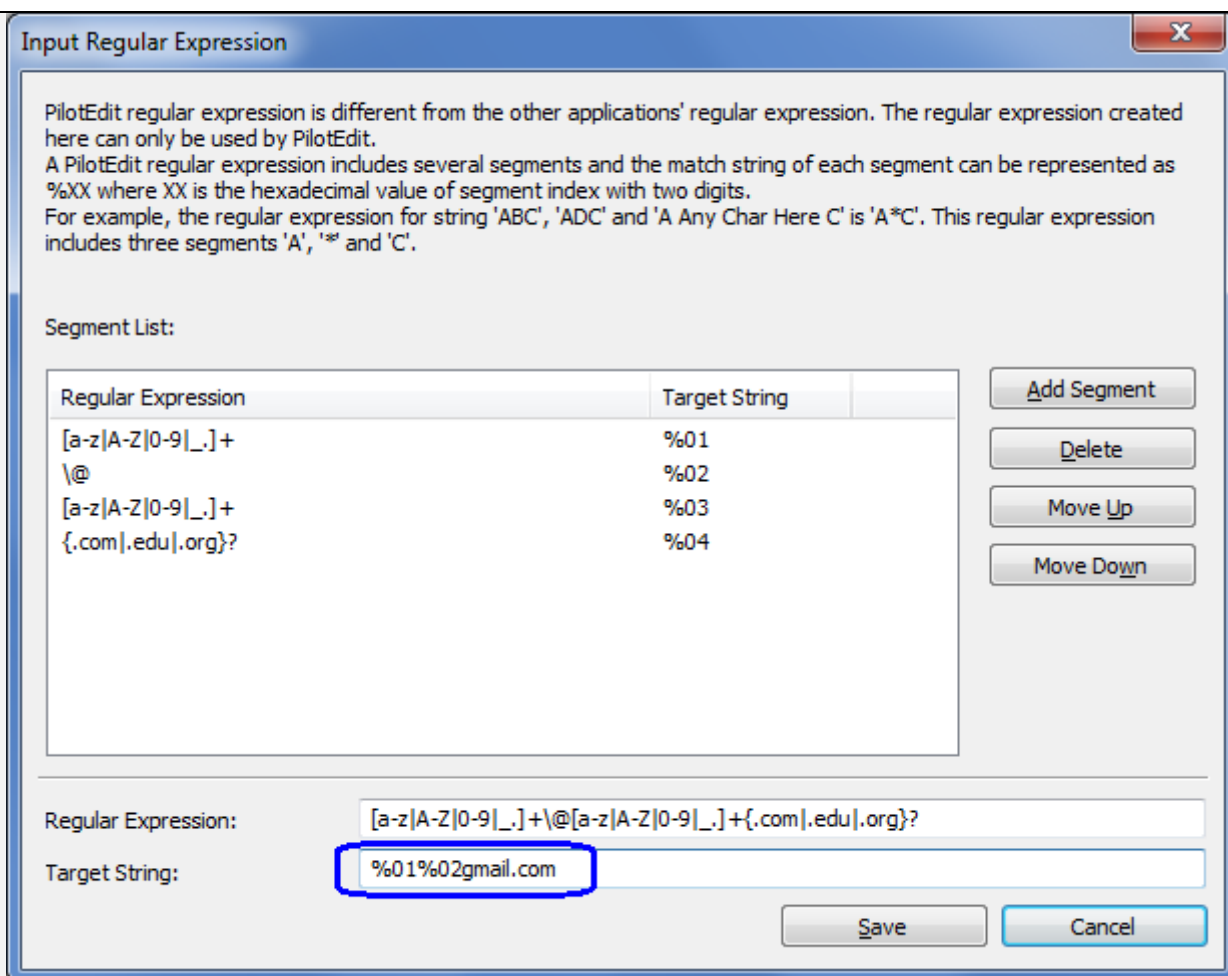
Target String:

%01%02%03%04

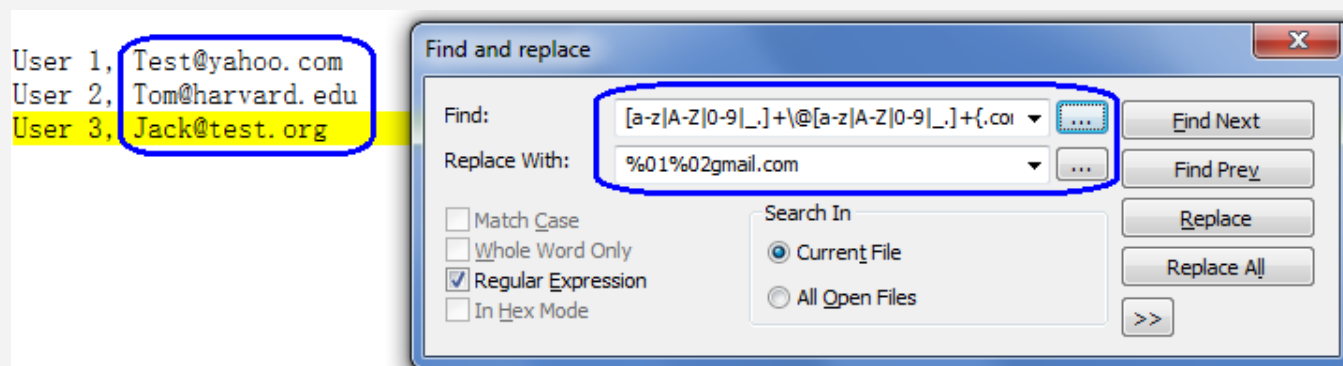
Save

Cancel

Change "%03%04" to "gmail.com" to replace all email addresses to "XXXX@gmail.com".



Change "**Replace All**" to replace all email addresses to "**XXXX@gmail.com**".



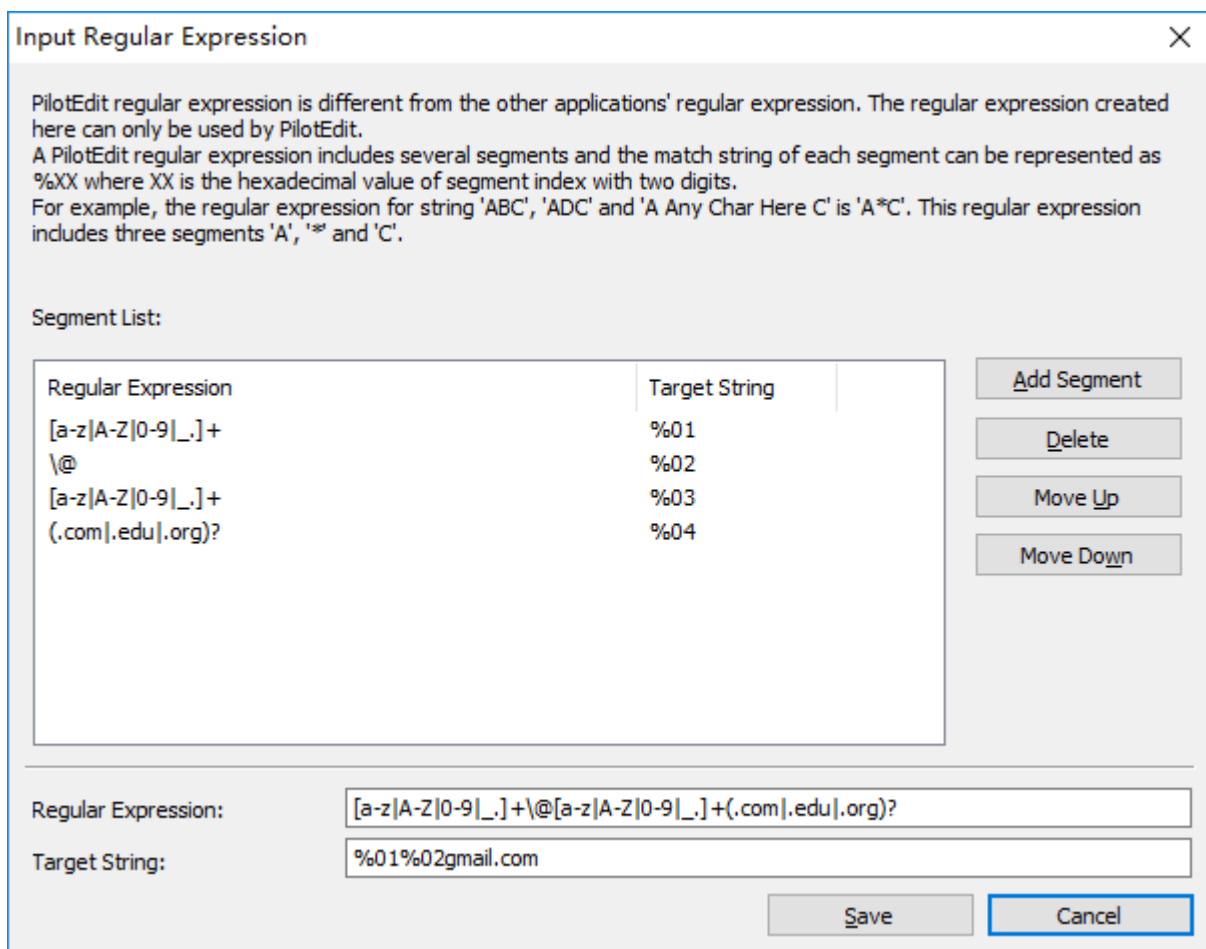
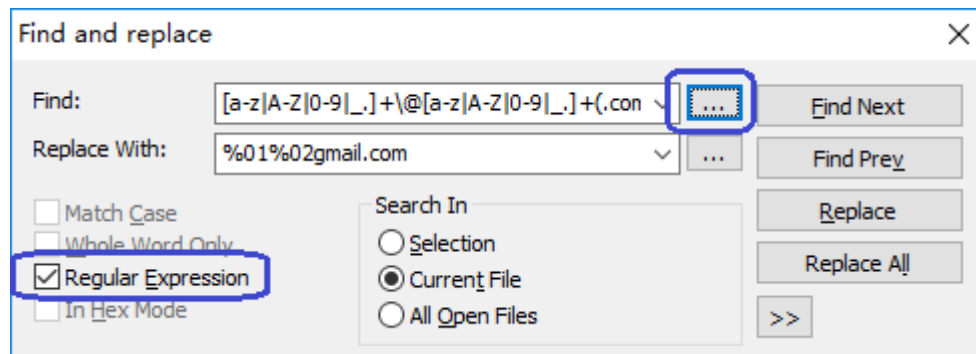
This is the result after replacing all email addresses to "**XXXX@gmail.com**".

```
User 1, Test@gmail.com
User 2, Tom@gmail.com
User 3, Jack@gmail.com
```

5.2. Parse Expression with GUI

You can parse PilotEdit regular expression with GUI. This is helpful when you need to troubleshoot a regular expression that is not working as expected.

Input the regular expression, select **Regular Expression**, click button ... and you will see the segments of the parsed regular expression.



5.3. Search Expression

5.3.1. Constant String Segment

Constant String Segment in regular expression is **case-sensitive**.

The screenshot shows the 'Add New Segment' dialog box. The 'Match a Constant String' option is selected and highlighted with a blue rectangle. The 'String' field contains the text 'Test'. Below this, there are options for 'Match Characters' and 'Match One of Multiple Strings', which are not selected. The 'Match One of Multiple Strings' section has fields for 'String' and 'String Regular Expression', with 'Add' and 'Clean' buttons. At the bottom, there is a 'Regular Expression' field containing 'Test' and 'Add' and 'Cancel' buttons.

Examples:

| Constant String | Notes |
|------------------|---|
| Test | Find the case-sensitive string Test . For example, test is NOT a match string. |
| *\?\+\\" | Find the string *?+\\" |

5.3.2. Character List Segment

You can define Character List Segment with **[]**.

- | will be used to separate several character lists.
- ! will be used to exclude the characters in current list.
- can be used to define a range of characters.

| Character List | Notes |
|---------------------------|---|
| [LIST1 LIST2 ...]* | Match zero or more characters defined in the list. |
| [LIST1 LIST2 ...]+ | Match one or more characters defined in the list. |
| [LIST1 LIST2 ...]? | Match one character defined in the list. |
| [LIST1 LIST2 ...] | Equals to [LIST1 LIST2 ...]? Match one character defined in the list. |

Add New Segment

☐ Match a Constant String

String:

☒ Match Characters

Character Type:

Character List:

Character Regular Expression:

☐ Match One of Multiple Strings

String:

String Regular Expression:

☐ Case Sensitive

Occurrences:

☐ Match Line Begin ☐ Match Line End

Regular Expression:

Only Characters segment in regular expression can match CR/LF. If you want to match multiple-lines, you must use Characters segments.
But if you add **\$** at the end of the segment, this segment will not match CR/LF. Please refer to the second example **[]*\$**.

Examples:

| Character List | Notes |
|----------------|--|
| * | Match any number of characters except CR/LF. |
| ? | Match any one character except CR/LF. |
| + | Match one or more characters except CR/LF. |
| []* | Match any characters including CR/LF. |

| | |
|----------------------------|---|
| []*\$ | Equals to *\$. Match any characters to the end of the line. The CR/LF will not be included. |
| [ABC DEF] | Equals to [ABC DEF]? Match one character if this character is in ABCDEF . |
| [A-Z]? | Equals to [A-Z] Match one character if this character is an upper case letter. |
| [A-Z 0-9]* | Match any characters if these characters are upper case letters or numbers. |
| ^[A-G !D-E] | Equals to ^[A-G !D-E]? Match one character if this character is in ABCFG . (In A-G but not in D-E) This character must be in the beginning of the line. |
| [!ABC]* | Match any characters if these characters are not in ABC . |
| [\U516c] | Match UNICODE character \u516c . (The UNICODE expression is not case sensitive, equals to [\U516C]) |
| [\U516c-\u5170] | Match UNICODE character from \u516c to \u5170 |
| [\u0-\uffff]*\$ | Equals to []*\$ or *\$. (The UNICODE expression is not case sensitive, equals to [\U0-\UFFFF]*\$) Match any characters to the end of the line. The CR/LF will not be included. |
| [\0-\65535]*\$ | Equals to []*\$ or *\$ or [\U0-\UFFFF]*\$. Match any characters to the end of the line. The CR/LF will not be included. |
| [\65-\90] | Equals to [A-Z]? . Match one character if this character is an upper case letter. |
| [ABC !DEF]+ | Match one or more character if these characters are in ABC!DEF . Here \ will serve as a translator and ! will be treated as normal character ! . |
| [ABC !CDE !BCD DEF] | Equals to [ABC !CDE !BCD DEF]? . Match one character if this character is in ADEF . When there are any conflictions among several lists, the last definition will take effect. In this example, the first list includes C , but the second and third list removes C . So that C is not in the final list. |

5.3.3. Multiple Case-sensitive Strings Segment

You can define Multiple Case-sensitive Strings Segment with **()**.

| will be used to separate several strings.

| Multiple case-sensitive strings | Notes |
|---------------------------------|---|
| (String1 String 2 ...)* | Match one string in the list zero or more time. The match strings are case-sensitive. |
| (String1 String 2 ...)+ | Match one string in the list one or more time. The match strings are case-sensitive. |
| (String1 String 2 ...)? | Match one string in the list. The match string is case-sensitive. |
| (String1 String 2 ...) | Equals to (String1 String 2 ...)? . Match one string in the list. The match string is case-sensitive. |

Examples:

| Multiple case-sensitive strings | Notes |
|---------------------------------|---|
| (Apple Orange Banana)+ | Find one or more case-sensitive string Apple , Orange or Banana . For example, apple , ORANGE and baNaNa are not match strings. |
| (*\ \\+ \\\$\\^)? | Find one case-sensitive string * + or \$ ^ . |

⚠ For the above example, (Apple|Orange|Banana)+ means one of the string in the list being repeated one or more time. So that AppleApple is a match string while AppleOrange is NOT a match string.

5.3.4. Multiple Non-case-sensitive Strings Segment

You can define Multiple Non-case-sensitive Strings Segment with **{ }**.

| will be used to separate several strings.

| Multiple non-case-sensitive strings | Notes |
|-------------------------------------|---|
| {String1 String 2 ...}* | Match one string in the list zero or more time. The match strings are |

| | |
|--------------------------------|---|
| | NOT case-sensitive. |
| {String1 String 2 ...}+ | Match one string in the list one or more time. The match strings are NOT case-sensitive. |
| {String1 String 2 ...}? | Match one string in the list. The match string is NOT case-sensitive. |
| {String1 String 2 ...} | Equals to {String1 String 2 ...}? . Match one string in the list. The match string is NOT case-sensitive. |

Add New Segment

☐ Match a Constant String

String:

☐ Match Characters

Character Type:

Character List:

Character Regular Expression:

☒ Match One of Multiple Strings

String:

String Regular Expression:

☒ Case Sensitive

Occurrences:

☐ Match Line Begin ☐ Match Line End

Regular Expression:

Examples:

| Multiple case-sensitive strings | Notes |
|---------------------------------|---|
| {Apple Orange Banana}+ | Find one or more non-case-sensitive string Apple , Orange or Banana . For example, apple , ORANGE and baNaNa are also match strings. |
| {*\\ \\+ \\\$\\^}? | Find one non-case-sensitive string * , + or \$. |

⚠ For the above example, {Apple|Orange|Banana}+ means one of the string in the list being repeated one or more time. So that AppleApple is a match string while AppleOrange is NOT a match string.

5.4. Special Characters and Escape Characters for Regular Expression

There are some special characters for PilotEdit regular expression as listed below.

If you want to use the special characters as normal character, you must add a transform character `\` before them.

For example, if you want to find string `*?+\`, you must define the search regular expression as `*\\?\\+\\`.

Special Characters

| Special Characters | Notes |
|---------------------|--|
| <code>^</code> | Match line begin. |
| <code>\$</code> | Match line end. |
| <code>*+?</code> | Wildcard characters. |
| <code>()</code> | You use them to define multiple case-sensitive strings in regular expression. |
| <code>{ }</code> | You use them to define multiple non-case-sensitive strings in regular expression. <code> </code> will be used to separate several strings. |
| <code>[!-]</code> | You use them to define characters in regular expression. |
| <code>%</code> | In the replace expression, you can use <code>%XX</code> to get the value of the XXth segment. This will be described in detail later. |
| <code>\</code> | Transform character. If you want to use the special characters as normal character, you must add a prefix <code>\</code> before them. For example, if you want to find string <code>*?+\</code> , you must define the search regular expression as <code>*\\?\\+\\</code> . Note: If you add <code>\</code> before a normal character, these two characters will be translated into the second character. For example, <code>\A</code> equals to <code>A</code> . |
| <code>@&</code> | Reserved for future use. Note: If you want your regular expression or script compatible with a future version of PilotEdit, please use <code>\@</code> and <code>\&</code> instead of <code>@</code> and <code>&</code> . |

Escape Characters

The following escape characters can be used in regular expressions and replace expressions.

| Special Characters | Escape Characters |
|---------------------|-----------------------------|
| <code>^</code> | <code>\^</code> |
| <code>\$</code> | <code>\\$</code> |
| <code>*+?</code> | <code>* \+ \?</code> |
| <code>()</code> | <code>\(\ \)</code> |
| <code>{ }</code> | <code>\{ \ \}</code> |
| <code>[!-]</code> | <code>\[\ \! \- \]</code> |
| <code>%</code> | <code>\%</code> |
| <code>\</code> | <code>\\</code> |
| <code>@&</code> | <code>\@ \&</code> |

| | |
|-------------------------------|-----------|
| TAB | \t |
| Carriage Return (0X0D) | \r |
| Line Feed (0X0A) | \n |

5.5. Search Regular Expression Examples

5.5.1. Find a Line with Two Words in It

Find a line with **This** and **test** in it and **This** should be in front of **test**.

The search expression can be defined as below:

| | |
|-------------------|------------------|
| Search Expression | This*test |
|-------------------|------------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| This | %01 |
| * | %02 |
| test | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression:

Target String:

Buttons: Save, Cancel

Examples:

| Source String | Notes |
|------------------------------------|---|
| This is a test. | It is a match string. |
| That test and this test. | Not match. Because This and test are case-sensitive |
| This is an apple. Just test it. | Not match. Because This and test are in two lines. |

5.5.2. Find a Line with Two Words in It and One Character between these Two Words

Find a line with **This** and **test** in it and there should be only one character between **This** and **test**.
The search expression can be defined as below:

| | |
|-------------------|------------------|
| Search Expression | This?test |
|-------------------|------------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| This | %01 |
| ? | %02 |
| test | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression: This?test
Target String: %01%02%03

Buttons: Save, Cancel

Examples:

| Source String | Notes |
|-------------------------|---|
| This test is OK. | It is a match string. |
| This is a test. | Not match. Because there are more than one characters between This and test . |

5.5.3. Find a Line Start with One Word

Find a line starts with **This**.

The search expression can be defined as below:

| | |
|-------------------|--------------|
| Search Expression | ^This |
|-------------------|--------------|

Examples:

| Source String | Notes |
|---------------------------------|--|
| This test is OK. | It is a match string. |
| That is a test. This is a test. | Not match. Because the first word of the line is not This . |

5.5.4. Find a Line Start with One Word and End with Another Word

Find a line start with **This** and end with **test**.

The search expression can be defined as below:

| | |
|-------------------|---------------------|
| Search Expression | ^This*test\$ |
|-------------------|---------------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| ^This | %01 |
| * | %02 |
| test\$ | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression: ^This*test\$

Target String: %01%02%03

Buttons: Save, Cancel

Examples:

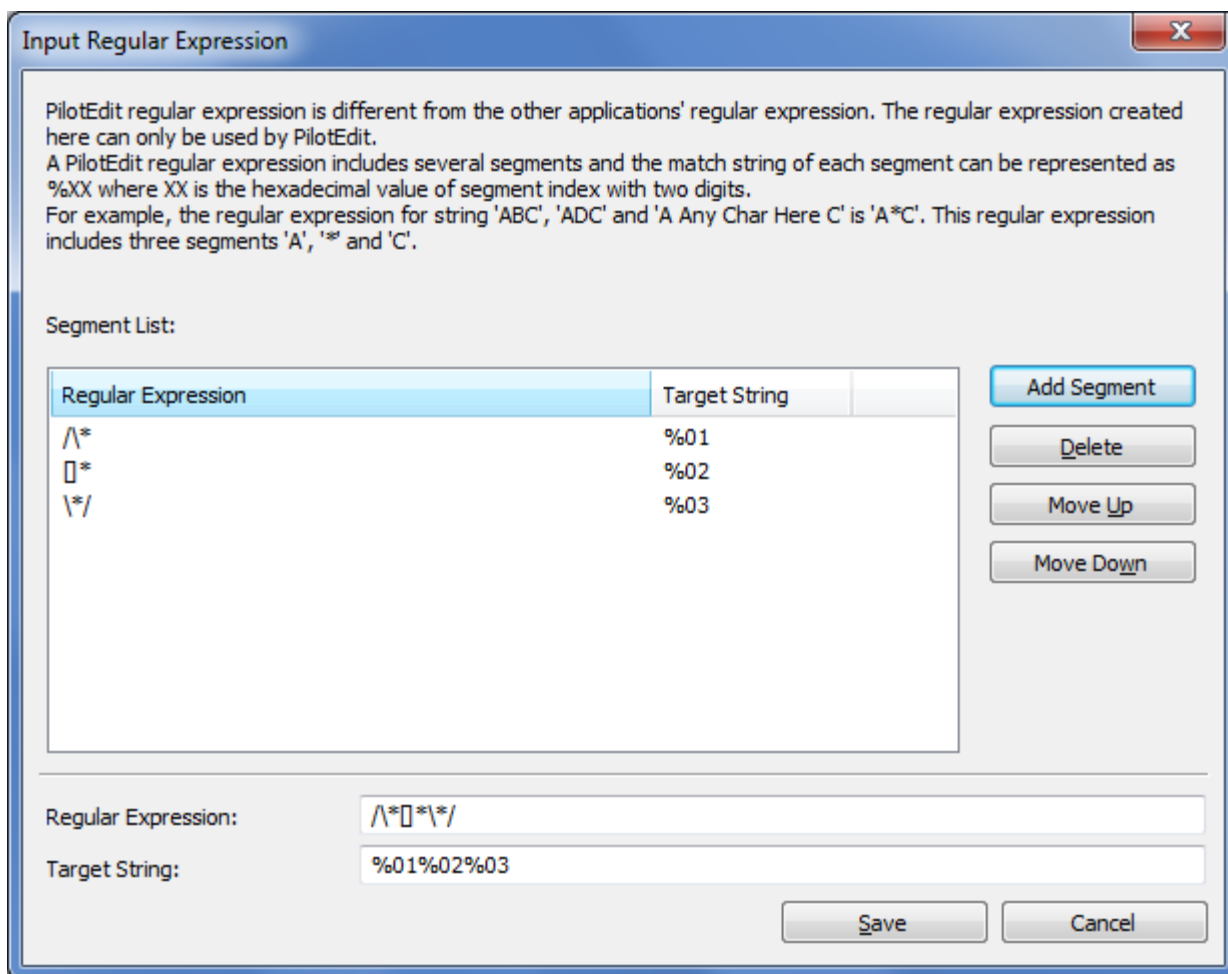
| Source String | Notes |
|--------------------------------|---|
| This is a test | It is a match string. |
| This test is OK | Not match. Because the line end word doesn't match. |

5.5.5. Search Expression for Multi-line Text

Find multi-line comments in C/C++. These lines start with `/*` and end with `*/`.

The search expression can be defined as below (**If you want to match CR/LF, you must use [...]**):

| | |
|-------------------|----------------------------|
| Search Expression | <code>/*[*\r\n]*/</code> |
|-------------------|----------------------------|



This regular expression will be parsed into three segments by PilotEdit.

| | | |
|----|-----|----|
| /* | []* | */ |
|----|-----|----|

Examples:

| Source String | Notes |
|--|-----------------------|
| /* int nTest = 1; //define a variable */ | It is a match string. |
| /* int nTest = 1;*/ | It is a match string. |

This line will also be treated as multi-line comments.

```
CString sTest = _T("/*test*/");
```

5.5.6. Search Expression for Multiple Case-sensitive Strings

Find a string starts with **This** or **That** and ends with **apple** or **banana**. The match string should be case-sensitive.

The search expression can be defined as below:

| | |
|-------------------|--|
| Search Expression | <code>^(This That)?*(apple banana)?\$</code> |
|-------------------|--|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| (This That)? | %01 |
| * | %02 |
| (apple banana)?\$ | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression:

Target String:

Buttons: Save, Cancel

This regular expression will be parsed into three segments by PilotEdit.

| | | |
|----------------------------|----------------|--------------------------------|
| <code>^(This That)?</code> | <code>*</code> | <code>(apple banana)?\$</code> |
|----------------------------|----------------|--------------------------------|

Examples:

| Source String | Notes |
|------------------------------|--------------------------------|
| This is an apple | It is a match string. |
| That is a banana | It is a match string. |
| This is a banana. I like it. | Not match. Line end not match. |
| This is a BANANA | Not match. Not match case. |

5.5.7. Find the last word that is not in the double quotation mark

Find the last word (there is a blank before this word) if it is not in the double quotation mark.

The search expression can be defined as below (there is a blank before [and]):

| | |
|-------------------|----------|
| Search Expression | [!"]]+\$ |
|-------------------|----------|

This regular expression will be parsed into two segments by PilotEdit (The first one is a blank).

| | |
|--|----------|
| | [!"]]+\$ |
|--|----------|

Examples:


| Source String | Notes |
|------------------------------|--|
| This is an apple | It is a match string (Including the blank before apple). |
| That is a "banana" | Not Match. (The last word can't include ") |
| This is a banana. I like it. | It is a match string (Including the blank before it.). |

5.5.8. Find the "int" or "long" Variable Definition in C++ or java

Find the "int" or "long" variable definition in C++ or java.

The search expression can be defined as below (there is a blank in **[]**):

| | |
|-------------------|---|
| Search Expression | ^[\t]*(int long)?[\t]+[[!\\r\\n!\\(\\)]+\$ |
|-------------------|---|

 This regular expression will be parsed into four segments by PilotEdit (there is a blank in **[\t]**).

| | | | |
|-----------------|--------------------|----------------|-----------------------------|
| ^[\t]* | (int long)? | [\t]+ | [[!\\r\\n!\\(\\)]+\$ |
|-----------------|--------------------|----------------|-----------------------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|----------------------|---------------|
| ^[\t]* | %01 |
| (int long)? | %02 |
| [\t]+ | %03 |
| [[!\\r\\n!\\(\\)]+\$ | %04 |

Add Segment
Delete
Move Up
Move Down

Regular Expression:

Target String:

Save
Cancel

Examples:

| Source String | Notes |
|--|--|
| //this is an int value int nIntValue; //this is a long value long nLongValue; | It is a match string (Including the blanks and tabs in the beginning of the line). |
| //this is a int function int GetIntValue(); | Not match. You have excluded (in the forth segment so |

| | |
|--|---|
| //this is a long function long GetIntValue(); | that function definition will not be treated as a match string. |
|--|---|

5.5.9. Find non-ASCII Characters in the File

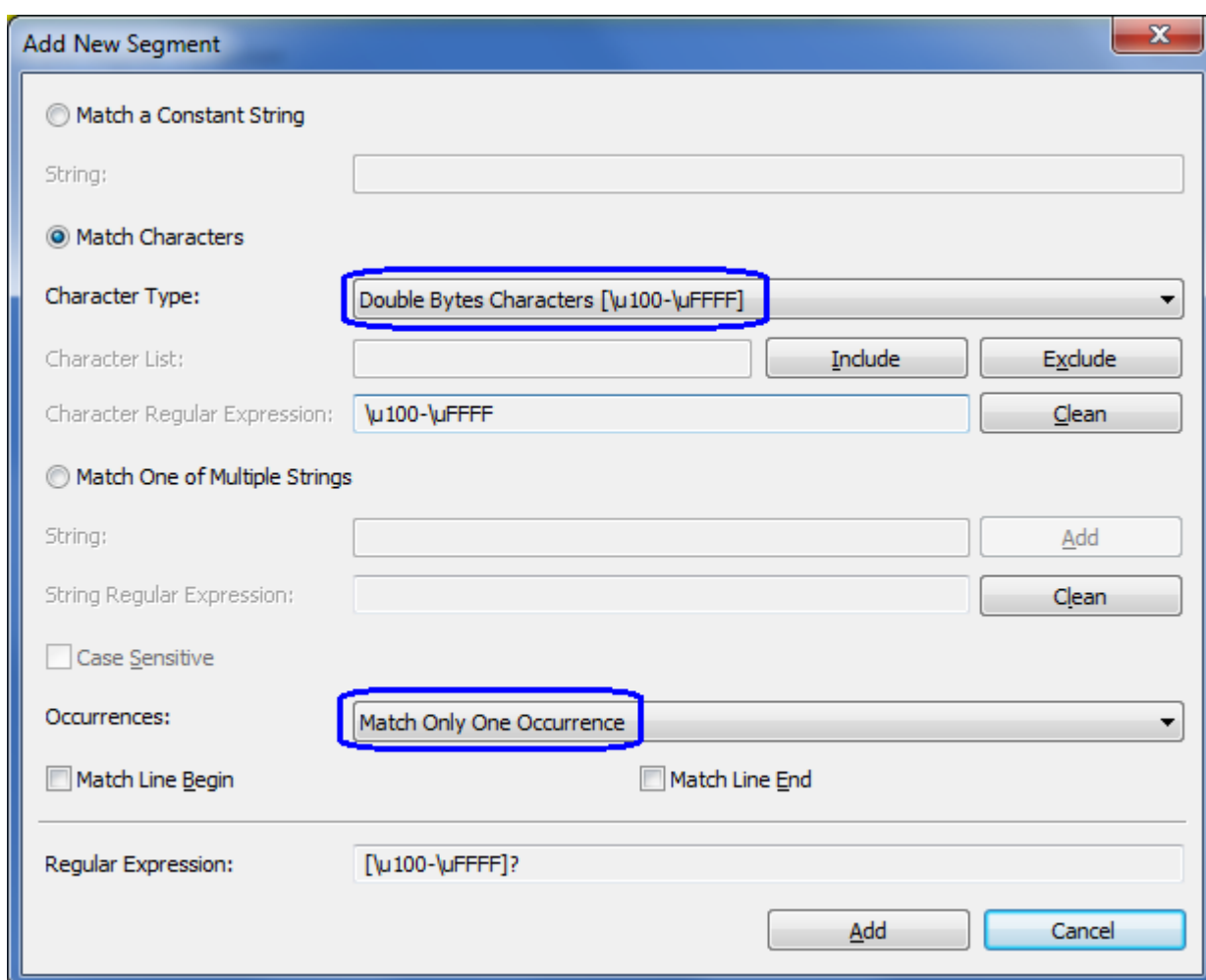
Find non-ASCII characters in the file.

The search expression can be defined as below:

| | |
|-------------------|--------------------------------|
| Search Expression | <code>[\\u100-\\uFFFF]?</code> |
|-------------------|--------------------------------|

 This regular expression will be parsed into only one segment by PilotEdit.

| |
|--------------------------------|
| <code>[\\u100-\\uFFFF]?</code> |
|--------------------------------|




Examples:

| Source String | Notes |
|---------------|--|
| "TEST" | It is a match string (There are two non-ASCII characters in it "T" and "E"). |

5.6. Replace Expression

A search expression will be parsed into several segments by PilotEdit. PilotEdit will find a match string for each segment.

In replace expression, you can use **%XX** to get the value of the **XX**th segment. Here **XX** is a two digits decimal number start from 01.

 There must be **TWO decimal digits** after %. If you use the segment value as **%1**, you may get unexpected error.

Example 1:

In this example, you will change the time format from **HH:MM:SS** into **HH:MM** in the following file.

```
2007-09-01 12:33:02 TEST
TEST 2007-09-01 12:33:04
2007-09-01 12:34:04 Apple
Apple 2007-09-01 13:34:04
```

You will define the search and replace regular expression as:

| | |
|--------------------|-------------------------------|
| Search Expression | 2007\ -09\ -01 ??:?:?? |
| Replace Expression | %01%02%03%04%05%06 |

The search expression will be parsed into 9 segments (The first segment includes a blank):

| | | | | | | | | |
|-----------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| 2007\ -09\ -01 | ? | ? | : | ? | ? | : | ? | ? |
|-----------------------|----------|----------|----------|----------|----------|----------|----------|----------|

When PilotEdit finds a match string, for example 2007-09-01 12:33:02, the match string will also be divided into 9 segments in accordance with the search expression (The first segment includes a blank):

| | | | | | | | | |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|
| 2007-09-01 | 1 | 2 | : | 3 | 3 | : | 0 | 2 |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|

You want to remove the 7th, 8th and 9th segment, only leave the first six segments, so you can define the replace regular expression as **%01%02%03%04%05%06**.

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| 2007\-\09\-\01 | %01 |
| ? | %02 |
| ? | %03 |
| : | %04 |
| ? | %05 |
| ? | %06 |
| : | %07 |
| ? | %08 |
| ? | %09 |

Regular Expression: 2007\-\09\-\01 ??:??:??

Target String: %01%02%03%04%05%06

Buttons: Add Segment, Delete, Move Up, Move Down, Save, Cancel

Find and replace

Find: 2007\-\09\-\01 ??:??:??

Replace With: %01%02%03%04%05%06

Options:

- ☐ Match Case
- ☐ Whole Word Only
- ☒ Regular Expression
- ☐ In Hex Mode

Search In:

- ☐ Selection
- ☒ Current File
- ☐ All Open Files

Buttons: Find Next, Find Prev, Replace, Replace All, >>

After replace all, the above text will be changed into:

```

2007-09-01 12:33 TEST
TEST 2007-09-01 12:33
2007-09-01 12:34 Apple
Apple 2007-09-01 13:34
  
```

5.7. Replace Regular Expression Examples

5.7.1. Change the Position of Two Words

Find a line with **This** and **test** in it and **This** should be in front of **test**.

Change the position of **This** and **test**.

The search and replace expression can be defined as below:

| | |
|--------------------|------------------|
| Search Expression | This*test |
| Replace Expression | %03%02%01 |

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| This | %01 |
| * | %02 |
| test | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression:

Target String:

Buttons: Save, Cancel

This regular expression will be parsed into three segments by PilotEdit.

| | | |
|-------------|----------|-------------|
| This | * | test |
|-------------|----------|-------------|

Examples:

| Source String | Notes |
|------------------------|---|
| This is a test. | It is a match string. This line will be changed into: |

| | |
|------------------------------------|--|
| | test is a This. |
| That test and this test. | Not match. Because This and test are case-sensitive. |
| This is an apple. Just test it. | Not match. Because This and test are in two lines. |

5.7.2. Replace one Character

Find a line with **This** and **test** in it and there should be only one character between **This** and **test**. Change the character between This and test into **A**.

The search and replace expression can be defined as below:

| | |
|--------------------|------------------|
| Search Expression | This?test |
| Replace Expression | %01A%03 |

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.


Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| This | %01 |
| ? | %02 |
| test | %03 |

Add Segment
Delete
Move Up
Move Down

Regular Expression: This?test
Target String: %01A%03

Save Cancel

 This regular expression will be parsed into three segments by PilotEdit.

| | | |
|-------------|----------|-------------|
| This | ? | test |
|-------------|----------|-------------|

Examples:

| Source String | Notes |
|------------------|--|
| This test is OK. | It is a match string. This line will be changed into: ThisAtest is OK. |
| This is a test. | Not match. Because there are more than one characters between This and test. |


5.7.3. Remove a Line together with CR LF

Find a line equals to **This is a test.**

Remove this line. The **CR/LF** in this line should also be removed.

The search and replace expression can be defined as below (**If you want to match CR/LF, you must use [...] :**

| | |
|--------------------|-----------------------------------|
| Search Expression | ^This is a test.[\r]*[\n]? |
| Replace Expression | |

 This regular expression will be parsed into three segments by PilotEdit.

| | | |
|-------------------------|--------------|--------------|
| ^This is a test. | [\r]* | [\n]? |
|-------------------------|--------------|--------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| ^This is a test. | %01 |
| [v]* | %02 |
| [n]? | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression: ^This is a test.[v]*[n]?

Target String:

Buttons: Save, Cancel

Examples:

| Source String | Notes |
|---|--|
| This is a test. This is a test. This is a test. | It is a match string. The last line will not match since there is no CR/LF after it. This text will be changed into: This is a test. |
| That is a test. This is a test. | Not match. |

5.7.4. Change Multiple Case-sensitive Strings into One Word

Find a line starts with **This** or **That** and ends with **apple** or **banana**. The match string should be case-sensitive.

Change **apple** or **banana** into **pear**.

The search and replace expression can be defined as below:

| | |
|--------------------|---------------------------------|
| Search Expression | ^(This That)?*(apple banana)?\$ |
| Replace Expression | %01%02pear |

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
 A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
 For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| ^(This That)? | %01 |
| * | %02 |
| (apple banana)?\$ | %03 |

Buttons: Add Segment, Delete, Move Up, Move Down

Regular Expression:

Target String:

Buttons: Save, Cancel

 This regular expression will be parsed into three segments by PilotEdit.

| | | |
|---------------|---|-------------------|
| ^(This That)? | * | (apple banana)?\$ |
|---------------|---|-------------------|

Examples:

| Source String | Notes |
|------------------------------|--|
| This is an apple | It is a match string. This line will be changed into: This is an pear |
| This is a banana. I like it. | Not match. Line end word not match. |
| This is a BANANA | Not match. Not match case. |

5.7.5. Insert a Special Character after the Second Character of the Last Word

Find the last word (there is a blank before this word) if it is not in the double quotation mark and has at least three characters.

Add # after the second character.

The search and replace expression can be defined as below (there is a blank before the first [and all the]):

| | |
|--------------------|--|
| Search Expression | <code>[!\\r\\n !"]]?[!\\r\\n !"]]?[!\\r\\n !"]]+\$</code> |
| Replace Expression | <code>%01%02%03#%04</code> |

⚠ This regular expression will be parsed into four segments by PilotEdit (The first one is a blank).

| | | | |
|--|------------------------------|------------------------------|--------------------------------|
| | <code>[!\\r\\n !"]]?</code> | <code>[!\\r\\n !"]]?</code> | <code>[!\\r\\n !"]]+\$</code> |
|--|------------------------------|------------------------------|--------------------------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------------------|---------------|
| | %01 |
| <code>[!\\r\\n !"]]?</code> | %02 |
| <code>[!\\r\\n !"]]?</code> | %03 |
| <code>[!\\r\\n !"]]+\$</code> | %04 |

Add Segment
Delete
Move Up
Move Down

Regular Expression:
Target String:

Save
Cancel

Examples:

| Source String | Notes |
|------------------------------|--|
| This is an apple | It is a match string. This line will be changed into: This is an ap#ple |
| That is a "banana" | Not Match. (The last word can't include ") |
| This is a banana. I like it. | It is a match string. This line will be changed into: This is a banana. I like it#. |

5.7.6. Change One Line into two Lines by Adding CR LF

Find a line start with **This**.

Add CR/LF before the last word (there is a blank before this word) to change this line into two lines.

The search and replace expressions can be defined as below (there is a blank before [and]). In replace expression, you can use **\r** and **\n** for **CR/LF**.

| | |
|--------------------|---------------------------------|
| Search Expression | ^This* [!\\r\\n!\\t]+\$ |
| Replace Expression | %01%02%03\r\n%04 |

 This regular expression will be parsed into four segments by PilotEdit (The third one is a blank).

| | | | |
|--------------|----------|--|--------------------------|
| ^This | * | | [!\\r\\n!\\t]+\$ |
|--------------|----------|--|--------------------------|

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.

Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| ^This | %01 |
| * | %02 |
| | %03 |
| [!\\r\\n!\\t]+\$ | %04 |

Add Segment
Delete
Move Up
Move Down

Regular Expression: ^This* [!\\r\\n!\\t]+\$
Target String: %01%02%03\r\n%04

Save Cancel

Examples:

| Source String | Notes |
|------------------|--|
| This is an apple | It is a match string. This line will be changed into: This is an apple |

| | |
|------------------------------|---|
| That is a banana | Not Match. (The first word doesn't match) |
| This is a banana. I like it. | It is a match string. This line will be changed into: This is a banana. I like it. |

5.7.7. Replace Expression for Multi-line Text

Replace expression for multi-line text. Find strings start with Bob and end with **Tom**. The target strings may include multiple lines.

Remove the content between **Bob** and **Tom**.

The search and replace expression can be defined as below (**If you want to match CR/LF, you must use [...] :**):

| | |
|--------------------|------------------|
| Search Expression | Bob[]*Tom |
| Replace Expression | %01%03 |

Input Regular Expression

PilotEdit regular expression is different from the other applications' regular expression. The regular expression created here can only be used by PilotEdit.
A PilotEdit regular expression includes several segments and the match string of each segment can be represented as %XX where XX is the hexadecimal value of segment index with two digits.
For example, the regular expression for string 'ABC', 'ADC' and 'A Any Char Here C' is 'A*C'. This regular expression includes three segments 'A', '*' and 'C'.


Segment List:

| Regular Expression | Target String |
|--------------------|---------------|
| Bob | %01 |
| []* | %02 |
| Tom | %03 |

Add Segment
Delete
Move Up
Move Down

Regular Expression: Bob[]*Tom
Target String: %01%03

Save Cancel

 This regular expression will be parsed into three segments by PilotEdit.

| | | |
|------------|------------|------------|
| Bob | []* | Tom |
|------------|------------|------------|

Examples:

| Source String | Notes |
|-------------------------------------|--|
| Bob wrote a mail to Jack and Tom | It is a match string. The text will be changed into: BobTom |

5.8. PilotEdit Script *

If you need to replace strings with some regular expressions frequently, you can define these regular expressions in a script and run this script.

PilotEdit Script is a good choice when you need to replace strings with some regular expressions frequently.

PilotEdit script must be encoded in UTF-8.

A PilotEdit script includes three parts, **Find Regular Expression**, **String Operations** and **Replace Expression**.

5.8.1. Find regular expression *

The find regular expression will be defined in the first line after **FND:**. Don't add extra blank or TAB after **FND:**.

5.8.2. String operations *

String operations are defined between the find expression and the replace expression. String operations are **optional**.

| String operations | Notes |
|---|---|
| set <Variable> = <Value> | <p>Define or change the value of Variable. For example</p> <p>set valueTest = Hello %{03} and %{valueExist}</p> <p>If value of 03 is "Tom" and value of "valueExist" is "Jerry", the value of valueTest is "Hello Tom and Jerry".</p> <p>The value of each segment from the regular expression will be automatically defined as a variable. The name of the variables is the two digits decimal index, like 01, 02, 03...</p> |
| uppercase <Variable> | <p>Change Variable into upper case.</p> <p>For example:</p> <p>uppercase 01</p> |
| lowercase <Variable> | <p>Change Variable into lower case.</p> <p>For example:</p> <p>lowercase name</p> |
| replace <Variable> <String1> <String2> | <p>Replace String1 into String2 in the Variable value.</p> <p>String1 and String2 may be defined in "" or not. Variable is the variable name.</p> <p>Example 1, Replace A into AB in the first segment: replace 01 A AB</p> <p>Example 2, replace " into ' in the second segment: replace 02 "\" ""</p> <p>Example 3, replace \ into \" in variable value of name:</p> |

| | |
|---|---|
| | <p>replace name "\\\" "\\\""</p> <p>Example 4, replace P into test it! in the first segment: replace 01 "P" "test it!"</p> |
| <p>ifequal <Value1> <Value2> ... end</p> | <p>If the Value1 equals to Value2, run the commands in it.</p> <p>For example, switch the second segment value and the forth segment value if the third segment value equals to the forth segment value. ifequal %{03} %{04} set temp = %{02} set 02 = %{04} set 04 = %{temp} end</p> |
| <p>ifnequal <Value1> <Value2> ... end</p> | <p>If the Value1 doesn't equal to Value2, run the commands in it.</p> <p>For example, change the second segment value to uppercase if the third segment value doesn't equal to hello. ifnequal %{03} "hello" uppercase 02 end</p> |
| <p>lfill <Variable> <Length> <Character></p> | <p>Change the length of Variable to Length by adding extra Character to the left of the string.</p> <p>Example 1, Change Variable index into 5 characters by add 0 to the left of the string: lfill index 5 0</p> |
| <p>rfill <Variable> <Length> <Character></p> | <p>Change the length of Variable to Length by adding extra Character to the right of the string.</p> <p>Example 1, Change Variable index into 5 characters by add 0 to the right of the string: rfill index 5 0</p> |
| <p>calc <Variable> = <String1> <+-%/*> <String2></p> | <p>Calculate String1 +-%/* String2 and save the result into Variable.</p> <p>Example 1, calc index = %{index} + 1</p> |


5.8.3. Replace expression *

The replace expression will be defined in the last line after **REP:**. Don't add extra blank or TAB after **REP:**.

You can get the variable value by using "**\${variable name}**". (For a regular expression segment value, you can use "**%XX**" or "**%{XX}**" where **XX** is the two digit decimal index.).

 A line started with **#** is a comment line.

 You can find some sample scripts in **<PilotEdit Install Directory>\ScriptSamples**.

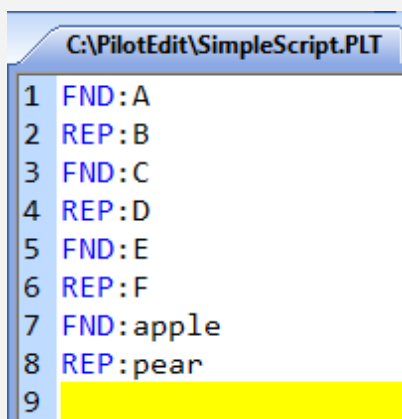
 Please refer to **<PilotEdit Install Directory>\ScriptSamples\RemoveDuplicateLines.PLT** about how to remove duplicate lines by running PilotEdit script.

5.8.4. PilotEdit Script Examples *

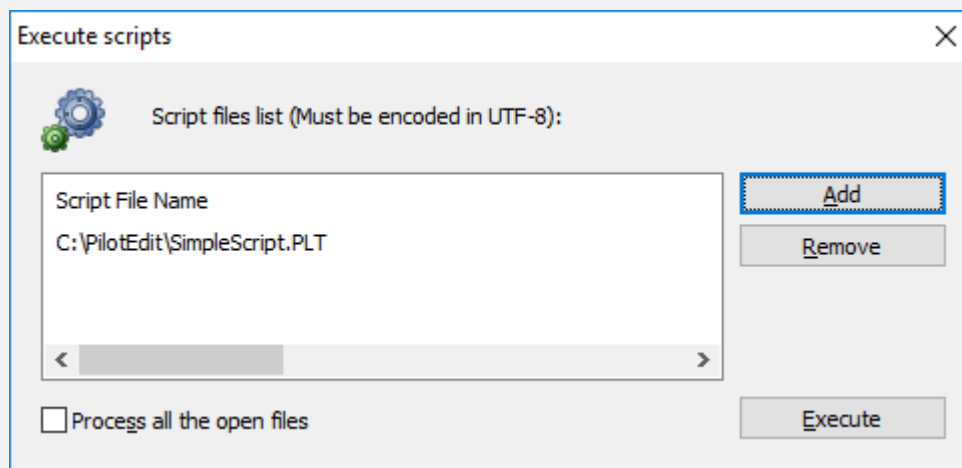
Example 1:

You can create a simple PilotEdit script to replace **A** to **B**, **C** to **D**, **E** to **F**, **apple** to **pear**. (In this example, String Operations are empty.)

3. Create a simple PilotEdit script to replace **A** to **B**, **C** to **D**, **E** to **F**, **apple** to **pear**.



4. Save this script as an **UTF-8** file. Open the target file, select the script file and execute it.



The text before and after running this script are listed below:

| Text before running this script | Text after running this script |
|--|---|
| Like many fruits, apples , bananas and pears contain important vitamins. All three provide vitamins A and C , with a medium-sized fruit containing between 2 and 5 micrograms of vitamin A and between 8 and 10 milligrams of vitamin C . | Like many fruits, pears, bananas and pears contain important vitamins. Bll three provide vitamins B and D, with a medium-sized fruit containing between 2 and 5 micrograms of vitamin B and between 8 and 10 milligrams of vitamin D. |

Example 2:

In this example, you will find the string with **Product** and **=** in it, change the Product into upper case and change the string between Product and = to lower case.

In the first line, the expression after **FND:** is the **find regular expression**.

The lines between the first line and the last line will define the **string operations**.

The first segment will be changed into upper case and the second segment will be changed into lower case.

In the last line, the expression after **REP:** is the **replace expression**.

1. Create PilotEdit script below.

```

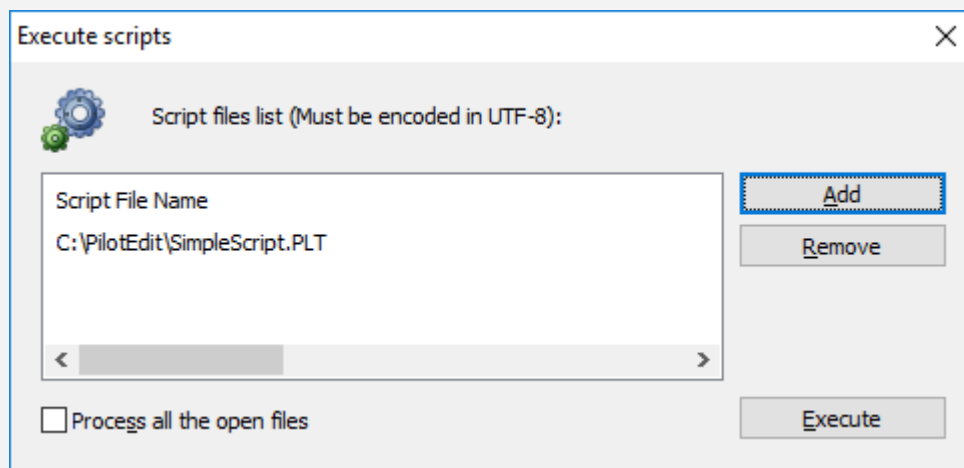
1 FND:Product*=
2 uppercase %01
3 lowercase %02
4 REP:%01%02%03
  
```

← Find regular expression

← String operations

← Replace expression

2. Save this script as an **UTF-8** file. Open the target file, select the script file and execute it.



The text before and after running this script are listed below:

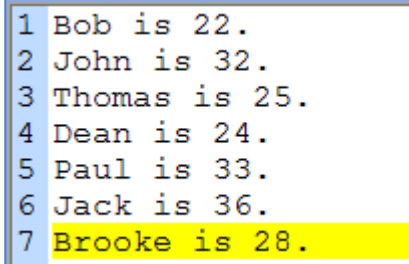
| Text before running this script | Text after running this script |
|---|---|
| ProductName=PilotEdit ProductVersion=1.0 | PRODUCTName=PilotEdit PRODUCTVersion=1.0 |

5.9. Advanced Examples with PilotEdit Regular Expressions

5.9.1. Sort over Regular Expression *

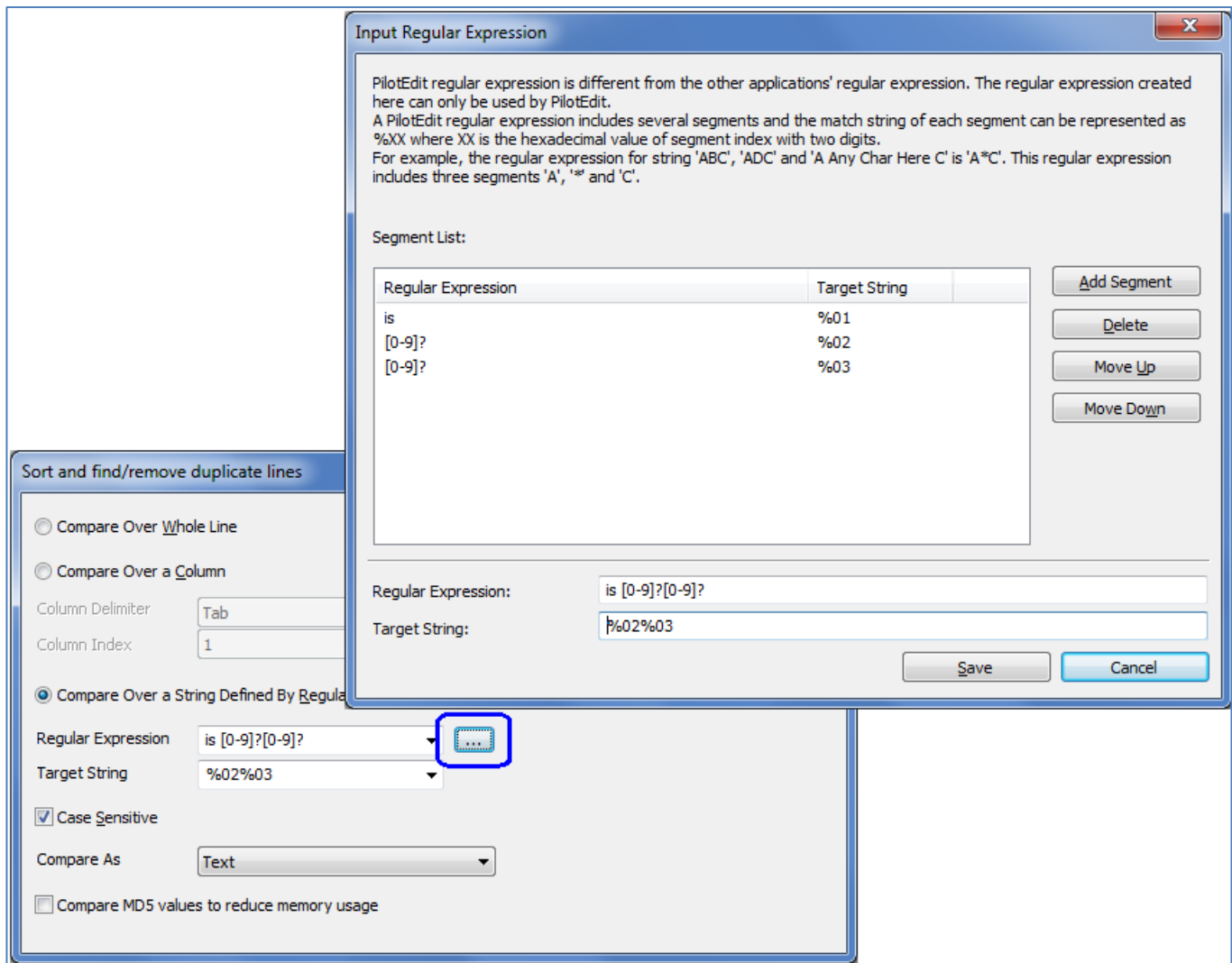
This example demonstrates how to sort over regular expression.

This is the file content.



```
1 Bob is 22.  
2 John is 32.  
3 Thomas is 25.  
4 Dean is 24.  
5 Paul is 33.  
6 Jack is 36.  
7 Brooke is 28.
```

Select menu item **Edit**, then **Sort and Find/Remove Duplicate Lines...**. This picture shows how to sort over age with regular expression.



5.9.2. Extract Strings in the Double Quotation Marks and Copy Them to the Clipboard *

This example demonstrates how to extract strings in the double quotation marks and copy them to the clipboard with regular expression.

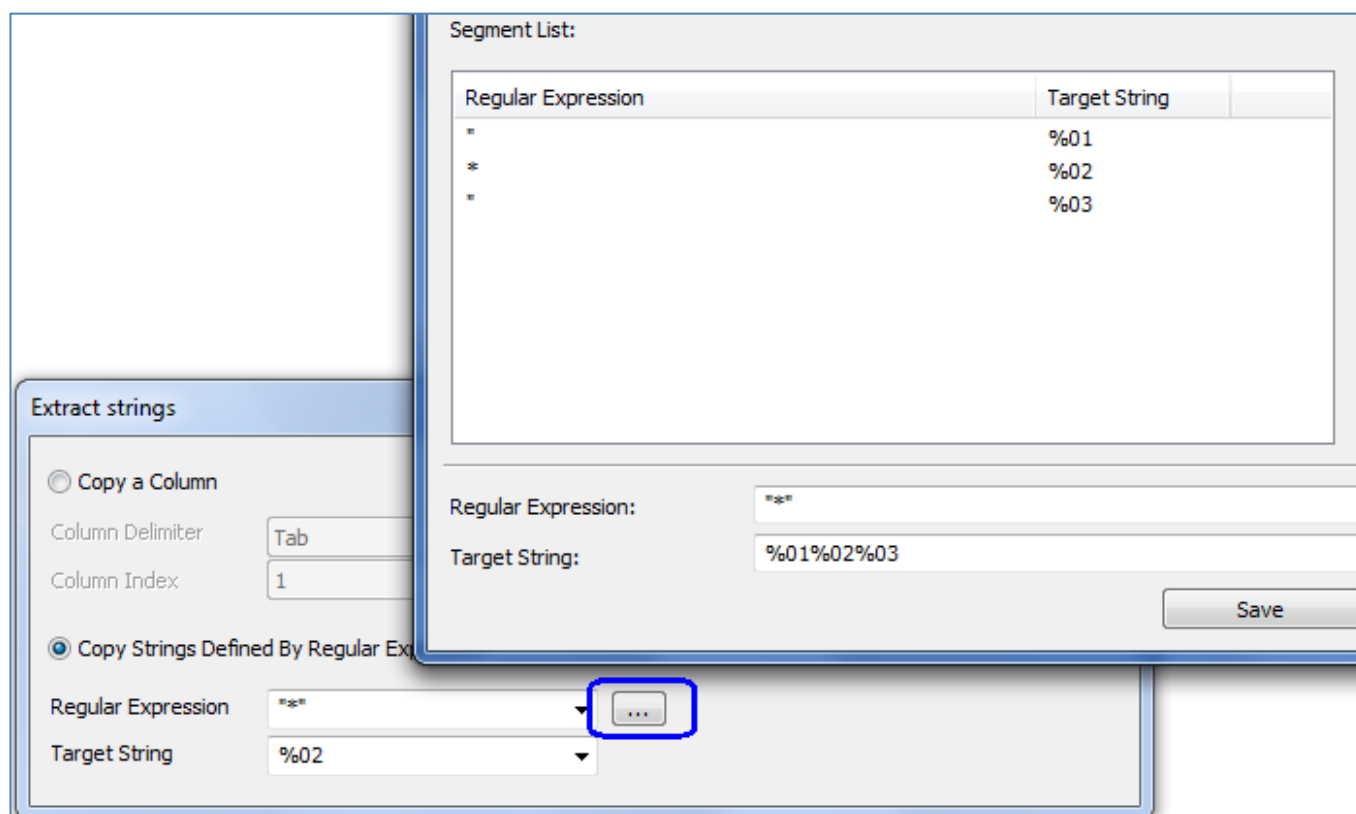
This is the file content.

```

1 XTP_IDR_RIBBONCUSTOMIZEMENUMORE MENU
2 BEGIN
3     POPUP "Quick Access"
4     BEGIN
5         MENUITEM "&Customize Quick Access Toolbar", XTP_ID_F
6         MENUITEM "[Commands]", XTP_ID_F
7         MENUITEM SEPARATOR
8         MENUITEM "&More Commands...", XTP_ID_F
9         MENUITEM "&Show Below the Ribbon", XTP_ID_F
10        MENUITEM "&Show Above the Ribbon", XTP_ID_F
11        MENUITEM SEPARATOR
12        MENUITEM "Mi&nimize the Ribbon", XTP_ID_F
13    END
14 END

```

Select menu item **Edit**, then **Extract Strings....** This picture shows how to copy text in double quotation mark to clipboard with regular expression.



This is the text copied to the clipboard.

```

Quick Access
&Customize Quick Access Toolbar
[Commands]
&More Commands...
&Show Below the Ribbon

```

&Show Above the Ribbon
Mi&nimize the Ribbon

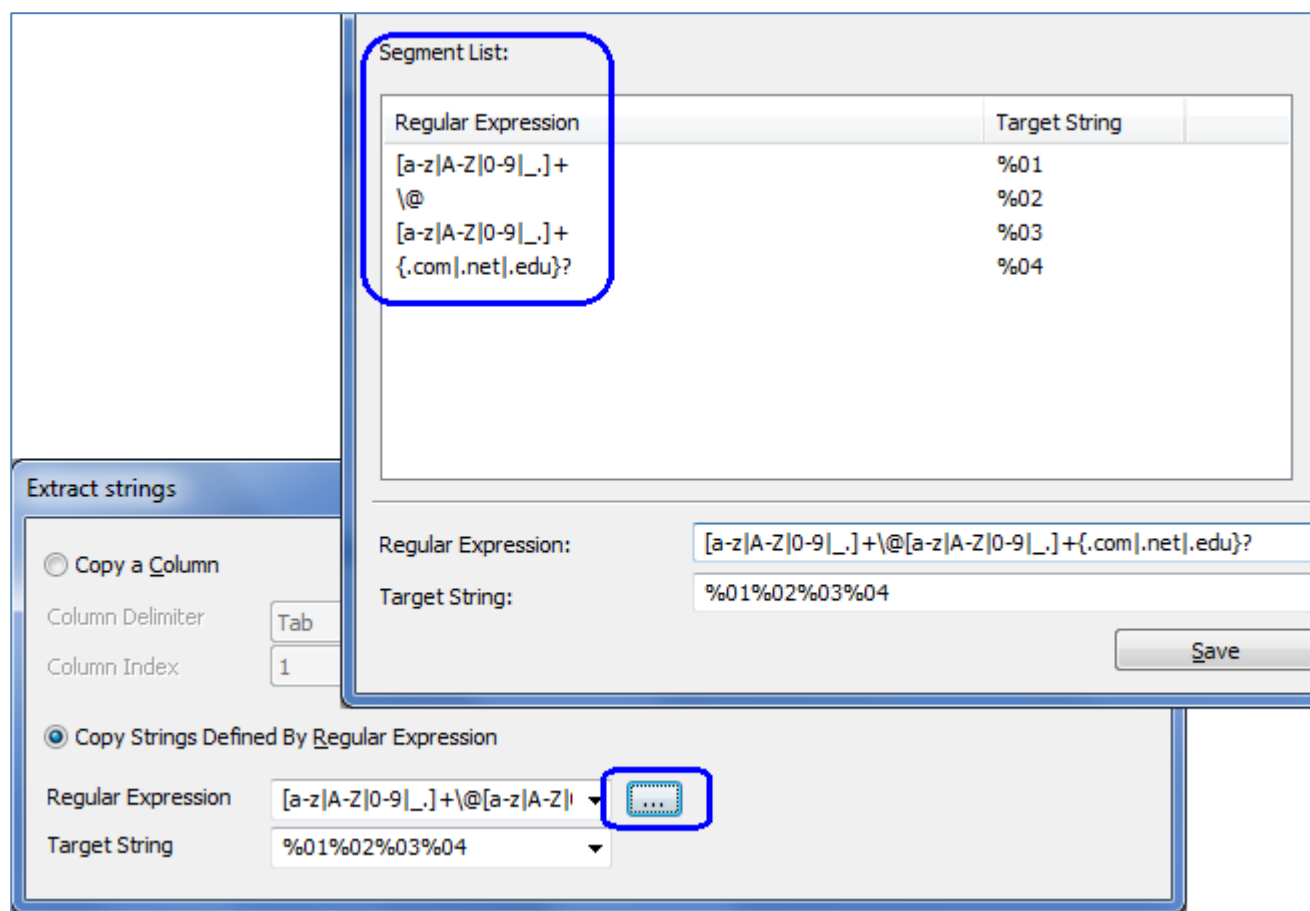
5.9.3. Extract Email Addresses and Copy Them to the Clipboard *

This example demonstrates how to extract Email addresses and copy them to the clipboard with regular expression.

This is the file content.

| | | | | |
|---|---|--------|------------------|----|
| 1 | 1 | Bob | Bob@gmail.com | 22 |
| 2 | 2 | John | John@gmail.com | 32 |
| 3 | 3 | Thomas | Thomas@gmail.com | 25 |
| 4 | 4 | Dean | Dean@gmail.com | 24 |
| 5 | 5 | Paul | Paul@gmail.com | 33 |
| 6 | 6 | Jack | Jack@gmail.com | 36 |
| 7 | 7 | Brooke | Brooke@gmail.com | 28 |

Select menu item **Edit**, then **Extract Strings....** This picture shows how to copy Email addresses to the clipboard with regular expression.



This is the regular expression.

```
[a-z|A-Z|0-9|_|.]+\@[a-z|A-Z|0-9|_|.]+\{.com|.net|.edu}?%01%02%03%04
```

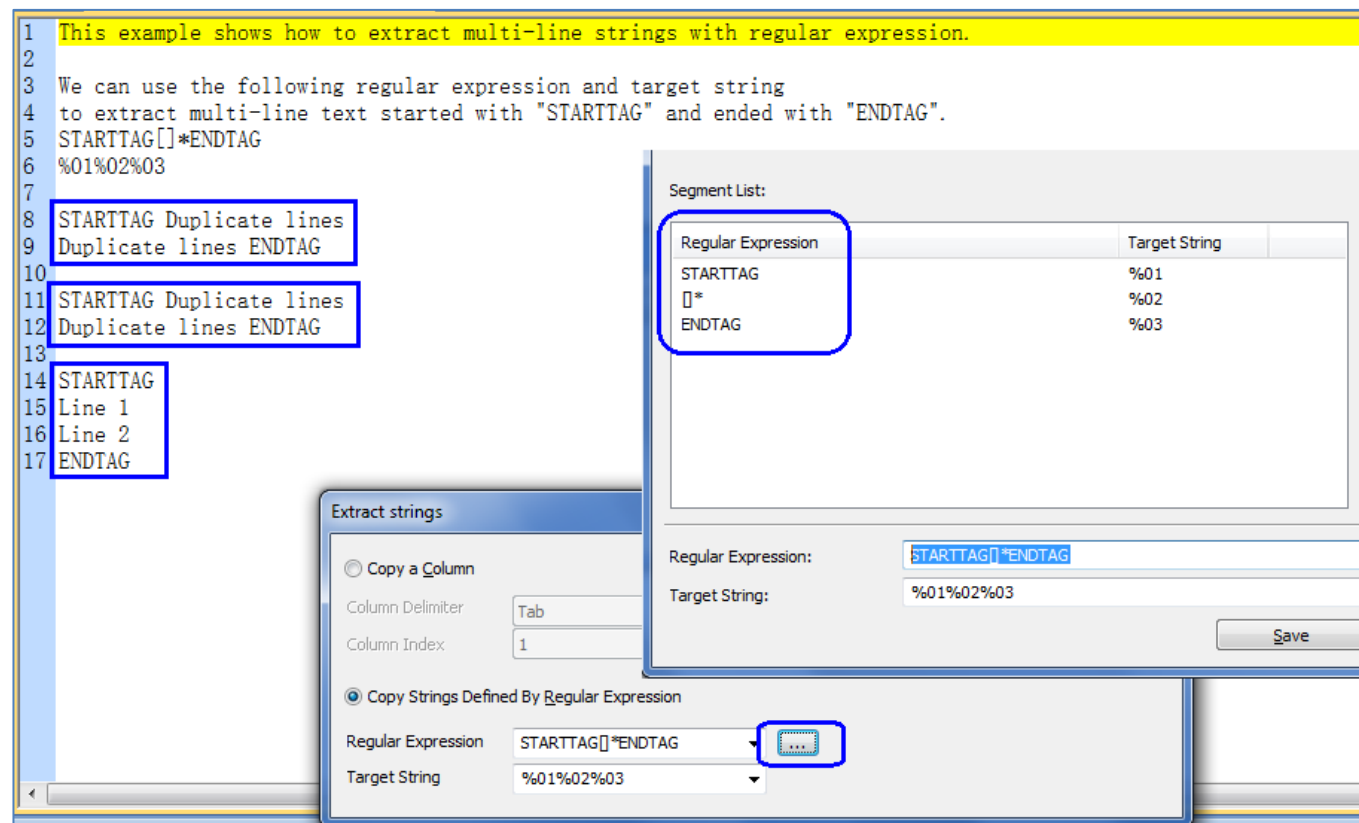
This is the text copied to the clipboard.

```
Bob@gmail.com
John@gmail.com
Thomas@gmail.com
Dean@gmail.com
Paul@gmail.com
Jack@gmail.com
Brooke@gmail.com
```

5.9.4. Extract Multiline Strings with Regular Expression *

This example demonstrates how to extract multiline strings with regular expression.

Select menu item **Edit**, then **Extract Strings....** This picture shows how to extract multiline strings with regular expression.



These are the regular expression and the target string.

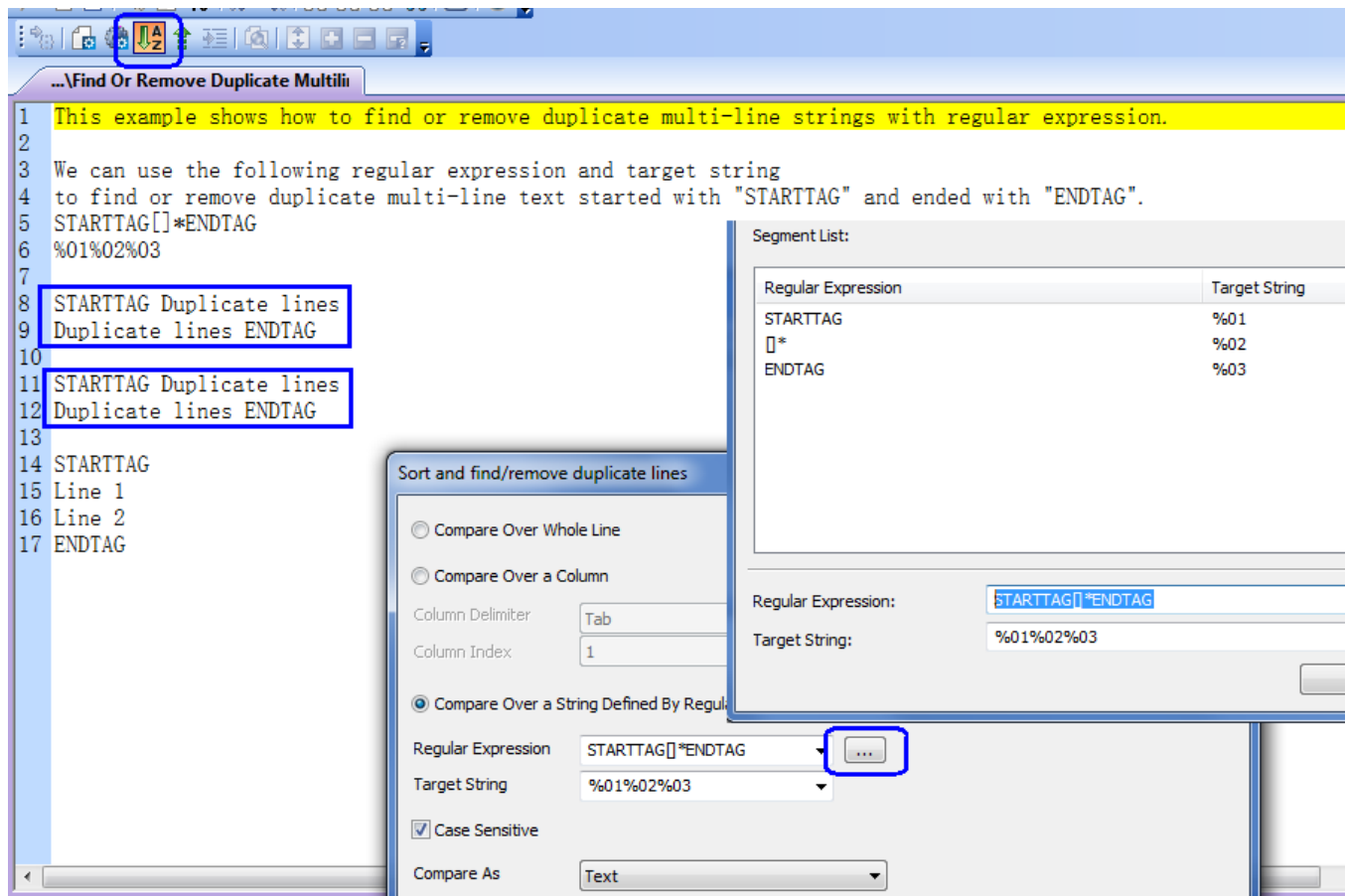
| |
|--------------------------|
| STARTTAG[]*ENDTAG |
|--------------------------|

| |
|------------------|
| %01%02%03 |
|------------------|

5.9.5. Find or Remove Duplicate Multiline Strings *

This example shows how to find/remove duplicate multiline strings with regular expression.

Select menu item **Edit**, then **Sort and Find/Remove Duplicate Lines...** This picture shows how to find/remove duplicate multiline strings with regular expression.



These are the regular expression and the target string.

| |
|--------------------------|
| STARTTAG[]*ENDTAG |
|--------------------------|

| |
|------------------|
| %01%02%03 |
|------------------|

6. FAQ

6.1. Where does PilotEdit save configuration files and temporary files?

1. PilotEdit configuration files will be saved in:

<User Application Dir>\PilotEdit

If you want to clean old PilotEdit setting, you may remove the files from it and re-install PilotEdit.

2. PilotEdit temporary files will be saved as:

<Temp Dir>\PE*.tmp

PilotEdit will remove the temporary files by default. But if a FTP file is not saved successfully to the FTP server, the temporary file for it will not be removed.

6.2. How to recover the lost data?

The problem is how to recover the lost data if the file was not saved successfully due to some errors such as unexpected power off.

1. For a local file, such as "C:\PilotEdit\test.txt", PilotEdit will first write the updated content into a temporary file "C:\PilotEdit\test.txt\$PE.TMP". Then this file will be moved to replace the old file. Even if PilotEdit is stopped unexpectedly before moving file, the old file will not be changed. In this way, "C:\PilotEdit\test.txt" will not be damaged.
2. For a FTP file, PilotEdit will save a copy "C:\Documents and Settings\<User>\Local Settings\Temp\PE**.tmp" first. This temporary file will not be removed if it was not uploaded successfully to the FTP server. So that you can find the lost data in "C:\Documents and Settings\<User>\Local Settings\Temp\PE**.tmp".

6.3. How to get PilotEdit's support?

You can go to PilotEdit's home page:

<http://www.pilotedit.com>

Or send Email to:

support@pilotedit.com

7. Acknowledgments

PilotEdit uses AES encryption and decryption routines written by Dr. Brian Gladman. The source code for these routines is available to any interested party from Dr. Gladman's web site, <http://fp.gladman.plus.com>. We would like to express our appreciations to Dr. Gladman for making this code available.

PilotEdit is translated into French by nickW. We would like to express our appreciations to nickW for it.

8. Appendix

8.1. PilotEdit Revision History

Here is PilotEdit revision history.

| Ver. No. | Ver. Date | Summary of Change |
|----------|------------|--|
| 1.0.0 | 2007-10-31 | PilotEdit 1.0 |
| 1.1.0 | 2007-12-09 | Download/upload FTP files and directories. |
| 1.2.0 | 2008-01-12 | String table added. |
| 1.3.0 | 2008-02-07 | Tested on German OS. |
| 1.4.0 | 2008-03-18 | Fixed some defects. |
| 1.5.0 | 2008-05-09 | Encryption/decryption added. * |
| 1.6.0 | 2008-05-31 | PilotEdit 1.6 supports FTP files more than 4GB. |
| 1.7.0 | 2008-06-20 | PilotEdit 1.7 has made improvement over GUI. |
| 1.8.0 | 2008-07-08 | Fixed some defects. |
| 1.9.0 | 2008-07-31 | PilotEdit 1.9 has made improvement over search and replace. |
| 2.0.0 | 2008-10-06 | File group was added in PilotEdit 2.0. |
| 2.1.0 | 2008-12-06 | A dialog for options was added in PilotEdit 2.1. |
| 2.2.0 | 2009-02-07 | PilotEdit 2.2 supports more than 20 file types. |
| 2.3.0 | 2009-03-18 | PilotEdit 2.3 enables sorting, finding/replacing duplicated lines. |
| 2.4.0 | 2009-04-16 | You can save selected files as new files with old directory structure with PilotEdit 2.4. |
| 2.5.0 | 2009-05-08 | PilotEdit 2.5 provides the ability to merge text differences between compared files. |
| 2.6.0 | 2009-05-31 | PilotEdit 2.6 provides the ability to find and replace in the comparison window. |
| 2.7.0 | 2009-07-18 | PilotEdit 2.7 changes help document from PDF to CHM. |
| 2.8.0 | 2009-08-18 | PilotEdit 2.8 provides the ability to change window and text color. |
| 2.9.0 | 2009-10-01 | PilotEdit 2.9 provides the ability to register with user or company name. * |
| 3.0.0 | 2009-10-31 | PilotEdit 3.0 provides the ability to find/remove duplicate multiline strings with regular expression. |
| 3.1.0 | 2010-01-01 | PilotEdit 3.1.0 begins to support Windows 7 and Windows 2008. |
| 3.2.0 | 2010-05-06 | PilotEdit 3.2.0 has made improvement over the sorting algorithm. * |
| 3.3.0 | 2010-05-31 | PilotEdit 3.3.0 can compare and merge two files more than 100MB. * |
| 3.4.0 | 2010-06-30 | PilotEdit 3.4.0 includes theme for Windows Classic. |
| 3.5.0 | 2010-07-31 | PilotEdit 3.5.0 supports line-continuation character. |
| 3.6.0 | 2010-08-31 | PilotEdit 3.6.0 has made improvement over PilotEdit script. * |
| 3.7.0 | 2010-10-01 | PilotEdit 3.7.0 provides the ability to select text between start tag and end tag. |

| Ver. No. | Ver. Date | Summary of Change |
|----------|------------|--|
| 3.8.0 | 2010-11-30 | PilotEdit 3.8.0 provides the ability to format code. * |
| 3.9.0 | 2010-12-31 | PilotEdit 3.9.0 includes PilotEdit x64. * |
| 4.0.0 | 2011-02-03 | PilotEdit 4.0.0 is a major release. |
| 4.1.0 | 2011-02-16 | PilotEdit 4.1.0 provides the option to create backup files. |
| 4.2.0 | 2011-03-16 | PilotEdit 4.2.0 supports SFTP. * |
| 4.3.0 | 2011-04-06 | PilotEdit 4.3.0 highlights selected word. * |
| 4.4.0 | 2011-04-30 | PilotEdit 4.4.0 provides the ability to print. |
| 4.5.0 | 2011-05-31 | PilotEdit 4.5.0 provides the ability to remove leading and trailing spaces. |
| 4.6.0 | 2011-07-01 | PilotEdit 4.6.0 has made improvement over the performance of comparing and sorting files. * |
| 4.7.0 | 2011-07-31 | PilotEdit 4.7.0 provides word operations. |
| 4.8.0 | 2011-08-31 | PilotEdit 4.8.0 provides quick mode in which millions occurrences of strings can be replaced in a huge file. * |
| 4.9.0 | 2011-09-30 | PilotEdit 4.9.0 provides the ability to compare and merge two huge files in quick mode. * |
| 5.0.0 | 2011-10-31 | PilotEdit 5.0.0 will show a progress window while opening, sorting and comparing huge files. |
| 5.1.0 | 2012-01-16 | PilotEdit 5.1.0 has made improvement over FTP client to work in slow network. |
| 5.2.0 | 2012-02-16 | PilotEdit 5.2.0 provides the ability to customize menu and toolbars in non-English locales. |
| 5.3.0 | 2012-02-29 | PilotEdit 5.3.0 provides the ability to remove blank lines. |
| 5.4.0 | 2012-03-31 | PilotEdit 5.4.0 has made improvement over file comparison. * |
| 5.5.0 | 2012-04-30 | PilotEdit 5.5.0 provides the ability to change the encoding of huge files. * |
| 5.6.0 | 2012-05-31 | PilotEdit 5.6.0 has made improvement over column mode. |
| 5.7.0 | 2012-08-31 | PilotEdit 5.7.0 is translated into French. |
| 5.8.0 | 2012-09-30 | PilotEdit 5.8.0 provides the ability to open very large files in quick mode. * |
| 5.9.0 | 2012-10-31 | PilotEdit 5.9.0 can edit files larger than 400GB in quick mode. * |
| 6.0.0 | 2012-11-30 | PilotEdit 6.0.0 is a major release. |
| 6.1.0 | 2012-12-31 | PilotEdit 6.1.0 has made improvement over the performance of file comparison. * |
| 6.2.0 | 2013-01-31 | PilotEdit 6.2.0 is capable of comparing and merging two huge files of 100GB. * |
| 6.3.0 | 2013-07-31 | PilotEdit 6.3.0 supports code collapsing. * |
| 6.4.0 | 2013-09-30 | PilotEdit 6.4.0 has made improvement over file comparison. * |
| 6.5.0 | 2013-10-31 | PilotEdit 6.5.0 has made improvement over file tree window. |
| 6.6.0 | 2013-11-30 | PilotEdit 6.6.0 provides the ability to search in zip files. * |

| Ver. No. | Ver. Date | Summary of Change |
|----------|------------|--|
| 6.7.0 | 2013-12-31 | PilotEdit 6.7.0 provides the ability to show hidden characters. |
| 6.8.0 | 2014-01-31 | PilotEdit 6.8.0 provides the ability to only export the count of match strings. |
| 6.9.0 | 2014-02-28 | PilotEdit 6.9.0 provides the ability to change SFTP timeout value. * |
| 7.0.0 | 2014-03-31 | PilotEdit 7.0.0 is a major release. |
| 7.1.0 | 2014-04-30 | PilotEdit 7.1.0 includes a fix for searching/replacing non-ASCII characters. |
| 7.2.0 | 2014-05-31 | PilotEdit 7.2.0 provides the ability to find/replace in selection. |
| 7.3.0 | 2014-10-31 | PilotEdit 7.3.0 has made improvement over the performance of replacing all. * |
| 7.4.0 | 2014-11-30 | PilotEdit 7.4.0 has made improvement over the performance of saving files. |
| 7.5.0 | 2014-12-31 | PilotEdit 7.5.0 provides GUI to generate regular expression. |
| 7.6.0 | 2015-01-31 | PilotEdit 7.6.0 supports CSV file type. |
| 7.7.0 | 2015-02-28 | PilotEdit 7.7.0 supports Apache Pig Latin Script. |
| 7.8.0 | 2015-03-31 | PilotEdit 7.8.0 provides the ability to exclude file types while searching/replacing in directories. * |
| 7.9.0 | 2015-04-30 | PilotEdit 7.9.0 provides the ability to remove duplicate lines by running PilotEdit script. * |
| 8.0.0 | 2015-05-31 | PilotEdit 8.0.0 is a major release. |
| 8.1.0 | 2015-06-30 | PilotEdit 8.1.0 includes improvement over column mode. |
| 8.2.0 | 2015-07-31 | PilotEdit 8.2.0 supports groovy and rexx. |
| 8.3.0 | 2015-08-31 | PilotEdit 8.3.0 provides the ability to select a large range of lines by entering line number. |
| 8.4.0 | 2015-09-30 | PilotEdit 8.4.0 include some fixes for word wrap. |
| 8.5.0 | 2015-10-31 | PilotEdit 8.5.0 includes improvements over file group/bookmark. |
| 8.6.0 | 2015-11-30 | PilotEdit 8.6.0 includes some fixes for UTF-8 and UTF-16. |
| 8.7.0 | 2015-12-31 | PilotEdit 8.7.0 improves the performance of finding and replacing. |
| 8.8.0 | 2016-01-31 | PilotEdit 8.8.0 improves the performance of finding previous occurrence. |
| 8.9.0 | 2016-02-29 | PilotEdit 8.9.0 improves the performance of opening huge files. * |
| 9.0.0 | 2016-03-31 | PilotEdit 9.0.0 is a major release. |
| 9.1.0 | 2016-04-30 | PilotEdit 9.1.0 includes improvement over file comparing algorithm. * |
| 9.2.0 | 2016-05-31 | PilotEdit 9.2.0 provides the ability to disable file update detection. |
| 9.3.0 | 2016-06-30 | PilotEdit 9.3.0 has upgraded SFTP library to support new SFTP versions. * |
| 9.4.0 | 2016-07-31 | PilotEdit 9.4.0 provides the ability to save comparison results to a file. |
| 9.5.0 | 2016-08-31 | PilotEdit 9.5.0 is four times faster than PilotEdit 9.4.0 when opening |

| Ver. No. | Ver. Date | Summary of Change |
|----------|------------|---|
| | | huge files in ASCII mode. * |
| 9.6.0 | 2016-09-30 | PilotEdit 9.6.0 provides the ability to register with computer ID. * |
| 9.7.0 | 2016-10-31 | PilotEdit 9.7.0 includes a fix for serial number verification. * |
| 9.8.0 | 2016-11-30 | PilotEdit 9.8.0 supports German. |
| 9.9.0 | 2016-12-31 | PilotEdit 9.9.0 supports file type json and gradle. |
| 10.0.0 | 2017-01-31 | PilotEdit 10.0.0 is a major release. |
| 10.1.0 | 2017-02-28 | PilotEdit 10.1.0 improves the performance of opening UTF16 text files. * |
| 10.2.0 | 2017-03-31 | PilotEdit 10.2.0 includes a fix for serial number verification. * |
| 10.3.0 | 2017-04-30 | PilotEdit 10.3.0 supports file type MicroBasic Pro. |
| 10.4.0 | 2017-05-31 | PilotEdit 10.4.0 supports language Portuguese (Brazil). |
| 10.5.0 | 2017-06-30 | PilotEdit 10.5.0 includes improvements over serial number verification. * |
| 10.6.0 | 2017-07-31 | PilotEdit 10.6.0 includes improvements over PilotEdit configuration file. |
| 10.7.0 | 2017-08-31 | PilotEdit 10.7.0 supports CDATA in XML and XHTML files. |
| 10.8.0 | 2017-09-30 | PilotEdit 10.8.0 includes a fix for start tag and end tag. * |
| 10.9.0 | 2017-10-31 | PilotEdit 10.9.0 increases bookmark menu item number to 20. |
| 11.0.0 | 2017-11-30 | PilotEdit 11.0.0 includes a fix for comparing 100GB files. * |
| 11.1.0 | 2017-12-31 | PilotEdit 11.1.0 supports dark theme. |
| 11.2.0 | 2018-01-31 | PilotEdit 11.2.0 includes improvements over the dark theme. |
| 11.3.0 | 2018-02-28 | PilotEdit 11.3.0 includes improvements over the icons. |
| 11.4.0 | 2018-03-31 | PilotEdit 11.4.0 supports high-DPI. |
| 11.5.0 | 2018-04-30 | PilotEdit 11.5.0 includes End User License Agreement (EULA). |
| 11.6.0 | 2018-05-31 | PilotEdit 11.6.0 provides the ability to parse Regex with GUI. * |
| 11.7.0 | 2018-06-30 | PilotEdit 11.7.0 includes MSI install package. |
| 11.8.0 | 2018-07-31 | PilotEdit 11.8.0 includes improvements over PilotEdit script. * |
| 11.9.0 | 2018-08-31 | PilotEdit 11.9.0 includes Hindi language. |
| 12.0.0 | 2018-09-30 | PilotEdit 12.0.0 is a major release. |
| 12.1.0 | 2018-10-31 | PilotEdit 12.1.0 provides the ability to set default encoding. |