

Wex: Edit history and comment extractor

Instructions

1.1.0

2019/10/1

RIKEN AIP NLU team

Table of contents

1. Introduction	2
1.1. Operating environment.....	2
1.2. Deployment	2
2. Program parameters.....	3
2.1. Execution.....	3
2.1.1. Parameters	3
2.1.2. Options	3
2.2. Output file	4
2.3. Usage example.....	4
3. Output.....	6
3.1. Standard output	6
3.2. Standard error output	6

1. Introduction

This document explains the program operating procedure for extracting Microsoft Word edit history and comments and converting them to XML/TSV format.

1.1. Operating environment

This program requires Windows and Microsoft Word to operate. See the table below for the guaranteed operating environment.

No.	OS
1.	Windows 10 (64-bit)
2.	Operation confirmed on Word 2010 and Word 2016

Table 1 Guaranteed operating environment list

1.2. Deployment

This program is installed and uninstalled using an installer. The installer can be run by double-clicking WexSetup-1.1.1.msi. After the program has been successfully installed, restart Windows to enable it.

2. Program parameters

2.1. Execution

To run the edit history information extraction program, start a command prompt and execute the following command.

```
$ wex [options] source
```

Parameters and options shall be discussed in future chapters.

2.1.1. Parameters

Specific parameters for the edit history information extraction program are listed in Table 2.

No.	Parameters	Mandatory	Explanation
1.	source...	○	Specify the path to the Word file that is to be extracted or the folder path. Note that you can specify multiple paths. When specifying the folder path, all Word files (all files with the extension .doc, .docx, or .docm) in the folder, including those in subfolders, will be extracted. When specifying the folder, if any type of error occurs in the processing of a file, then that file's processing will be skipped, and the next file will be processed. (The application will not close and processing will continue.)

Table2 Parameter list

2.1.2. Options

The options for the edit history information extraction program are listed in Table 3. None of these items are mandatory.

No.	Options	Explanation
3.	-a	The author attribute is omitted when outputting the comment XML file.
4.	-c	Only the change history XML file is generated.
5.	-f	If a file with the same name already exists when outputting a change history XML file, comment XML file, or comment CSV file, the file is overwritten without warning. If this option is not specified, the existing file will not be overwritten.
6.	-m	Only a comment CSV file is generated.
7.	-r	Only a comment XML file is generated.

8.	-s	All CSV files are generated. In version 1.1.0, only the comment CSV file is generated.
9.	-x	Generates all change history and comment XML files. This option will be enabled if none of the options of -c, -m, -r, -s, and -x are specified.
10.	-v, --version	The following is output to standard output, and the program closes. wex version {version}
11.	-h, --help	Help is output to standard output, and the program closes.

Table 3 Option list

2.2. Output file

When executing the program, files are output in the formats listed in Table 4.

No.	Type	Filename
1.	Change history XML file	{Word filename}-revision.xml
2.	Comment history XML file	{Word filename}-comment.xml
3.	Comment CSV file	{Word filename}-comment.csv

Table 4 Output filename list

2.3. Usage example

Usage examples for the edit history information extraction program are presented in Table 5.

No.	Usage example	Explanation
1.	wex sample.docx	This is an example of extracting the editing change history and comments from sample.docx in the current folder, and then generating a change history XML file and comment XML file in the same folder.
2.	wex .%sample.docx	Same as above.
3.	wex C:%sample.doc	This scenario is the same, except that this example specifies an absolute path for the source.
4.	wex "sample with space.docx"	This scenario is the same, except that this example specifies a file with the same name including spaces for the source.
5.	wex samples	This scenario is the same, except that this example specifies a path to a folder for the source.
6.	wex -a sample.docx	This is an example in which the author attribute in

		the comment XML file is omitted.
7.	-c sample.docx	This is an example of exclusively generating a change history XML file.
8.	wex -f sample.docx	This is an example of forcibly overwriting the files when sample.docx-comment.xml and sample.docx-revision.xml already exist.
9.	wex -r sample.docx	This is an example of exclusively generating a comment XML file.
10.	wex -m sample.docx	This is an example of exclusively generating a comment CSV file.
11.	wex -s sample.docx	This is an example of generating all CSV files. In version 1.1.0, only the comment CSV file is generated.
12.	wex -x sample.docx	This is an example of generating all XML files.
13.	wex -a -r -f sample.docx samples	This is an example of simultaneously specifying 1, 5, 6, 8, and 9.
14.	wex -arf sample.docx samples	Same as above.

Table 5 Usage example list

3. Output

In this chapter, we explain the messages that are displayed while executing the edit history information extraction program.

3.1. Standard output

If the program is operating normally, messages presented in Table 6 will be displayed as standard output.

No.	Message	Detail
1.	Start application initialization processing.	Displayed immediately after program execution.
2.	Application initialization processing has completed.	This is displayed at the stage when the preparation for program execution is finished.
3.	Output of the change history XML file was successful. : {Path}	This is displayed when the {path} change history XML file is output normally.
4.	Output of the comment XML file was successful. : {Path}	This is displayed when the {path} comment XML file is output normally.
5.	Output of the comment CSV file was successful. : {Path}	This is displayed when the {path} comment CSV file is output normally.
6.	Finished.	This is displayed when the program has finished.

Table 6 Standard output message list

3.2. Standard error output

If a problem occurs while executing the program, the messages presented in Table 7 are output to standard error output.

No.	Message	Detail
1.	An unexpected error has occurred.	This is displayed when an unexpected error has occurred.
2.	This usage is invalid. Check the usage of "wex -h."	This is displayed in case of a syntax error.
3.	Initialization of the application failed. Check	This is displayed if Microsoft Word is not

	whether Microsoft Word is installed.	installed.
4.	Unable to open the specified file or folder. : {Path}	This is displayed when the file or folder cannot be opened owing to reasons such as {path} not existing or no access permissions.
5.	The specified file is invalid. Specify a Microsoft Word file. : {Path}	This is specified when {path} is not recognized as a Microsoft Word file.
6.	File overwrite failed. File with the same name already exists. Consider using the -f option. : {Path}	This is displayed when {path} already exists, and file overwrite fails.
7.	File overwrite failed. : {Path}	This is displayed when an existing file cannot be overwritten owing to reasons such as no access permissions to {path}.
8.	Output of comment file failed. Non-supported comments are included. Please resolve the comment crossover. : {Path}:{Line number}	This is displayed when non-supported format comments (particularly, comment crossover) are included in the {path}.

Table 7 Standard error output message list