seg2_edit: A Program for Editing and Manipulating SEG-2 Files

By Karl J. Ellefsen

Open-File Report 2003-03-141
Version 1.1, December 2024

U.S. Department of the Interior
U.S. Geological Survey

1. OVERVIEW

Program seg2_edit is used to edit and manipulate files that contain seismic (or radar) data stored in the SEG-2 format. The program is written in the standard C++ programming language, including its standard library. The program has been compiled with Microsoft Visual C++ .NET and tested on the Microsoft Windows 98, 2000, and XP operating systems. Nonetheless, because of the usage of the standard language, the source code should readily compile with other compilers, and the program should execute on other operating systems like UNIX or LINUX.

Although users of this program do not need detailed knowledge of the SEG-2 format, they must understand the following aspects of the SEG-2 format: the definition of the file descriptor block, the definition of the trace descriptor block, the keywords and associated values that are part of the file descriptor block, and the keywords and associated values that are part of the trace descriptor block. This information is not included in this report, but it may be obtained from the report establishing the SEG-2 standard (Subcommittee of the SEG Engineering and Groundwater Geophysics Committee, 1990).

This Open-File Report describes

- The contents of the Zip file and installation of the program (section 2).
- The source code, the execution of the program, tests of the installation (section 3).
- Address to which suggestions for improving the program may be sent (section 4).

2. ZIP FILE

2.1 Contents

The name of the Zip file is "seg2_edit.zip," and it contains ofr_03-141.pdf, and four directories. File ofr_03-141.pdf is a copy of this report. The first directory, "source_code," contains the files with the source code for the program, and these files are described in section 3.1. The second directory, "Release," contains the executable program, and this program is described in section 3.2. The third and fourth directories, "test" and "archive," contain files that test the program on the user's computer; this test is described in section 3.4.

The contents of seg2_edit.zip may be extracted using the program Winzip, which may be purchased at http://www.winzip.com.

2.2 Installation

For computers with the Microsoft Windows operating system, the installation has only two steps:

1. Copy the Zip file to a suitable directory on your computer.
2. Extract the contents of the Zip file using WinZip.

For computers with other operating systems like UNIX and LINUX, the installation requires an additional step: The source code must be compiled and linked with a compiler that supports the standard C++ language and the standard library. As every compiler is somewhat different, this step is not described here.

3. PROGRAM seg2_edit

3.1 Source Code

The source code for program seg2_edit is in 15 different files, which are briefly described in Table 1.

Table 1. Files with the source code for program seg2_edit.

File	Short Description
seg2_edit.cpp	Main program.
seg2record.h	Header file defining class seg2record.
seg2record.cpp	Definitions of functions for class seg2record.
seg2_fdb.h	Header file defining class File_Descriptor.
seg2_fdb.cpp	Definitions of functions for class File_Descriptor.
seg2_tb.h	Header file defining class Trace_Block.
seg2_tb.cpp	Definitions of functions for class Trace_Block.
seg2_sb.h	Header file defining class String_Block.
seg2_sb.cpp	Definitions of functions for class String_Block.
seg2_misc.h	Header file defining several data structures and declaring one function.
seg2_misc.cpp	Definition of one function.
str_manip.h	Header file declaring two functions that manipulate strings.
str_manip.cpp	Definitions of two functions that manipulate strings.
Read_String.h	Header file defining a function that reads a string from a stream.
argck.h	Header file that checks command line arguments.

3.2 Executable Program

The executable program, "seg2_edit.exe," has been tested under the Microsoft Windows 98, 2000, and XP operating systems. Although it has not been tested under other Windows operating systems, it is still expected to work.

3.3 Usage of Program

Under the Microsoft Windows operating system, the program is executed using the command line within the console window. Under the UNIX or LINUX operating systems, the program is executed using the command line.

To describe the usage of this program, assume that SEG-2 records are stored in files "n001.dat" and "n002.dat." For both records, the number of traces is "n_traces," and the indices for the traces range from 1 to n_traces.

3.3.1 Delete Traces

Command Line: seg2_edit -delete_traces i j -infile n001.dat -outfile n001a.dat

Explanation : Following argument “-delete_traces” are the indices of the traces to be deleted. Index i corresponds to the first trace, and index j to the last trace. The following conditions apply to i and j: 1 <= i, i <= j, and j <= n_traces. Following argument “-infile” is the name of the input file; following argument “-outfile” is the name of the output file.

Action: The program reads the SEG-2 record in file “n001.dat,” deletes the traces whose indices range from i to j (inclusive), and writes the edited SEG-2 record in file “n001a.dat.”

3.3.2 Zero Traces

Command Line: seg2_edit -zero_traces i j -infile n001.dat -outfile n001a.dat

Explanation: Following argument "-zero_traces" are the indices of the traces to be set to zero. Index i corresponds to the first trace, and index j to the last trace. The following conditions apply to i and j: 1 <= i, i <= j, and j <= n_traces. Following argument “-infile” is the name of the input file; following argument “-outfile” is the name of the output file.

Action: The program reads the SEG-2 record in file “n001.dat,” zeroes the traces whose indices range from i to j (inclusive), and writes the edited SEG-2 record in file “n001a.dat.”

3.3.3 Scale Traces

Command Line: seg2_edit -scale_traces 2.0 -infile n001.dat -outfile n001a.dat

Explanation: Following argument “-scale_traces” is the scale factor, “2.0,” to be applied to all traces in the record. (The value “2.0” is used only for this example. In practice, any value may be used.) Following argument “-infile” is the name of the input file; following argument “-outfile” is the name of the output file.

Action: The program reads the SEG-2 record in file “n001.dat,” scales all traces by multiplying every sample by the scale factor, and writes the edited SEG-2 record in file “n001a.dat.”

3.3.4 Stack Traces

Command Line:
seg2_edit -stack_traces add -infile1 n001.dat -infile2 n002.dat -outfile n001a.dat

Explanation: Following argument “-stack_traces” is the type of stack: either “add” or “subtract.” Following arguments “-infile1” and “-infile2” are the names of the two input

files; following argument “-outfile” is the name of the output file. To stack the traces, both files must be compatible, meaning that both files must have the same number of traces and must use the same unit for the spatial dimension (for example, feet or meters). In addition, the corresponding traces in the two files must have the same number of samples, the same data format, the same time delay, the same descaling factor, and the same sample interval.

Explanation of the Type of Stack: If the type of stack is “add,” then

outfile = infile1 + infile2

where + means adding the corresponding traces. If the type of stack is “subtract,” then

outfile = infile1 - infile2

where - means subtracting the corresponding traces.

Action: The program reads the SEG-2 records in files “n001.dat” and “n002.dat,” checks whether the files are compatible, adds or subtracts the traces in the records, writes the stacked SEG-2 record in file “n001a.dat.”

3.3.5 Print a SEG-2 Record

Command Line: seg2_edit -print -infile n001.dat > n001_hdr.txt

Explanation: Argument “-print” indicates that the entire SEG-2 record is printed, except for the traces themselves. (Instead of the traces, their root mean squares are printed.) Following argument “-infile” is the name of the input file. Following the symbol “>” is the name of the text file in which the record is printed. If this symbol and the name of the text file are omitted, then the record is printed on the console.

Action: The program reads the SEG-2 record in file “n001.dat,” and prints the record in file “n001_hdr.txt.”

3.3.6 Print a Concise Summary of a SEG-2 Record

Command Line: seg2_edit -print -concise -infile n001.dat > n001_info.txt

Explanation: This command is identical to the previous command, except for the argument "-concise", which indicates that a concise summary of the record is printed. The summary includes the number of traces and the unit for the spatial dimension (for example, feet or meters). For each trace, the summary includes the data format, the number of samples, the number of stacks, the time delay, the descaling factor, the sample interval, the receiver location, and the source location.

Action: See the description for the previous command.

3.3.7 Set Keywords

Command Line:

seg2_edit -set_keywords -infile n001.dat -outfile n001a.dat <set_keywords.txt

Explanation: Argument "-set_keywords" indicates that keywords and their associated values are to be set; these keywords may be in either the file descriptor block or the trace descriptor blocks. Following argument "-infile" is the name of the input file; following argument "-outfile" is the name of the output file. Following the symbol "<" is the name of the input text file with the list of the keywords.

Explanation of the File with the Keywords: A typical input file might be:

```
f          CLIENT      U.S. Geological Survey
f          JOB_ID      TD-127a
f          UNITS       METERS
f          NOTE        Seismic line parallel to road CR-3.
t   1   48 LINE_ID     D
t   1   48 RECEIVER    SH_HORIZONTAL_GEOPHONE
t   1   48 SOURCE      HAMMER, 20 lb.
```

In the first column is either "f" or "t," which indicates whether the keyword is in the file descriptor block or the trace descriptor block. If "f," then the second and the third columns are blank. If "t," then the second and the third columns contain the beginning and ending traces, respectively, for which the keywords are set. For this example, keywords in traces 1 to 48 (inclusive) are set. In the fourth column is the keyword. If this keyword does not match the keywords in the SEG-2 standard, a warning will be printed, and the keyword and its value are set anyway. [Because the keywords in the SEG-2 standard are capitalized, the keywords in the fourth column must be capitalized. In other words, non-capitalized keywords will appear to be non-standard.] In the fifth column is the value associated with the keyword.

Assume, for a moment, that the file descriptor block of "n001.dat" already includes the keyword "CLIENT." In this case, the associated value is replaced. On the other hand, assume the file descriptor block does not include the keyword "JOB_ID." In this case, the keyword "JOB_ID" and its associated value are created. These two rules also apply to the trace descriptor block.

Action: The program reads the SEG-2 record in file "n001.dat," reads the file with the keywords to be set, sets the keywords and their values in n001.dat, and writes the edited SEG-2 record in file "n001a.dat."

3.3.8 Delete Keywords

Command Line:
seg2_edit -delete_keywords -infile n001.dat -outfile n001a.dat <delete_keywords.txt

Explanation:

Argument “-delete_keywords” indicates that keywords and their associated values are to be deleted; these keywords may be in either the file descriptor block or the trace descriptor blocks. Following argument “-infile” is the name of the input file; following argument “-outfile” is the name of the output file. Following the symbol “<” is the name of the input text file with the list of the keywords that will be deleted.

Explanation of File with Keywords: A typical input file might be:

```
f          CLIENT
f          NOTE
f          GENERAL_CONSTANT
t   1   48  NOTCH_FREQUENCY
t   1   48  SOURCE_LOCATION
t   1   48  SOURCE_STATION_NUMBER
```

In the first column is either “f” or “t,” which indicates whether the keyword is in the file descriptor block or the trace descriptor block. If "f", then the second and the third columns are blank. If “t,” then the second and the third columns contain the beginning and ending traces, respectively, for which the keywords are deleted. In this example, keywords in traces 1 to 48 (inclusive) are deleted. In the fourth column is the keyword. If this keyword does not match any keywords in the SEG-2 record, then a warning is printed. To match two keywords, their cases (that is, upper case or lower case) must be identical.

Action: The program reads the SEG-2 record in file “n001.dat”, reads the file with the keywords, deletes the keywords in n001.dat, and writes the edited SEG-2 record in file “n001a.dat”.

3.3.9 Extract Traces

Command Line:
seg2_edit -extract_traces -infile n001.dat -tracefile t001 >extract_traces.txt

Explanation: Argument “-extract_traces” indicates that the traces are to be extracted from the SEG-2 file. Following argument “-infile” is the name of the input file; following argument “-tracefile” is the name of the binary file in which the traces are written. Following the symbol “>” is the name of the output text file with a list of information about the traces. The information includes the number of traces and the unit for the spatial dimension (for example, feet or meters), the number of samples, the sample interval, and the time delay.

Action: The program reads the SEG-2 record in file “n001.dat”; extracts information about the traces; prints that information in the text file; checks whether the traces meet several criteria, which are described later. If so, then the program extracts the traces and writes them (as a binary file) in “t001.” There are four criteria: (1) All traces must have the same number of samples.(2) All traces must have the same sample interval. (3) The

samples for all traces must be stored in the same format (that is, 16-bit fixed point, 32-bit fixed point, 32-bit floating point, or 64-bit floating point). (4) The samples for all traces must not be stored as 20-bit floating point (Subcommittee of the SEG Engineering and Groundwater Geophysics Committee, 1990).

Format of Binary File: Let n_samp be the number of samples in each trace. Each sample is written as a floating-point number, which consists of 4 bytes. The first n_samp numbers correspond to the first trace, the second n_samp numbers correspond to the second trace, and so on. The number of bytes in the file equals 4 × n_samp × n_traces.

3.3.10 Create a SEG-2 Record

Command Line: seg2_edit -create_seg2 -tracefile t001 -outfile n001a.dat <create_seg2.txt

Explanation: Argument "-create_seg2" indicates that a SEG-2 record is to be created. Following argument "-tracefile" is the name of the input file with the traces in binary format. (The format of this file is identical to that described in the section "Extract Traces"). Following the argument "-outfile" is the name of the output (SEG-2) file. Following the symbol "<" is the name of the input text file with information about the traces, which is needed to create the SEG-2 file.

Explanation of File with Trace Information: The minimum information that is recommended to be in all SEG-2 records (Subcommittee of the SEG Engineering and Groundwater Geophysics Committee, 1990) includes two keywords for the file descriptor block and six keywords for each trace descriptor block. These required keywords are specified in the input text file, and a typical example of such a file might be:

```
TRACE_SORT   AS_ACQUIRED
UNITS        METERS

                    Time    Sample
Trace   No. of      Delay   Interval    Receiver Location
No.     Samples     (s)     (s)         X       Y       Z
  1     2001        0.0     0.0005      107.87  0.11    0.12
  2     2001        0.0     0.0005      106.37  0.11    0.10
  3     2001        0.0     0.0005      104.87  0.11    0.08
...
47      2001        0.0     0.0005       38.87  -0.09   -0.22
48      2001        0.0     0.0005       37.38  -0.10   -0.22
```

On the first two lines are the two required keywords and their associated values that must be set in the file descriptor block. For keyword "TRACE_SORT", suitable values can be "AS_ACQUIRED", "CDP_GATHER", "CDP_STACK", "COMMON_OFFSET", "COMMON_RECEIVER", OR "COMMON_SOURCE". For keyword "UNITS", suitable values can be "FEET", "METERS", "INCHES", "CENTIMETERS", or "NONE". On the fourth and subsequent lines is a table with seven columns that are used

to set the six required keywords and their associated values in each trace descriptor block. Because of the column headings, the meaning of each column should be clear. [Program seg2_edit does not use the trace number to create the SEG-2 file: The trace number helps the user keep track of the entries in the other columns.] The order of the entries in this file must match the order of the traces in the binary file. If the user wishes to add more information to the SEG-2 record than may be specified with the input text file, then that information can be added using the command line argument "-set_keywords" (section 3.3.7).

Action: The program reads both the file with the trace information and the file with the traces, creates the SEG-2 record, and writes the record file "n001a.dat."

3.3.11 Merge SEG-2 Files

Command Line: seg2_edit -merge -outfile n001a.dat <files2merge.txt

Explanation: Argument "-merge" indicates that traces from SEG-2 files are to be merged together to create a new SEG-2 file. Following the argument "-outfile" is the name of the new SEG-2 file. Following the symbol "<" is the name of the text file with the list of SEG-2 files that will be merged together.

Explanation of File with the List of SEG-2 Files: A typical input file might be:

Filename	Trace No.
d001.dat	3
d002.dat	3
d003.dat	3
d003.dat	4
d003.dat	7
d004.dat	
...	
d020.dat	3

In the first column is the name of the file from which a trace is extracted; in the second column is the number of the trace that is extracted. Note that three traces are extracted from file "d003.dat" and there is a separate entry for each trace. The order of this list is the order of the traces in the new SEG-2 file.

Action: The program reads the first entry in the list, "d001.dat", and it deletes all traces except trace 3. Thus, what remains is the file descriptor block and trace 3. This file descriptor block is used in the new SEG-2 file, and this trace 3 becomes the first trace in the new SEG-2 file. Next, trace 3 from "d002.dat" is appended to the new SEG-2 file; in other words, it becomes the second trace in the new SEG-2 file. This procedure is repeated until trace 3 from file "d020.dat" has been appended. Finally, the new SEG-2 record is written to file "n001a.dat".

3.4 Test of Installation

On a computer with the Microsoft Windows operating system, the installation of program seg2_edit may be tested using the files in directories “test” and “archive.” To this end, open a console window and make the current directory “test.” On the command line, type “test” to execute a batch file named “test.bat.” This batch file executes program seg2_edit for each of the options listed in section 3.3; for each option one or more output files are created in directory “test.” Now type “compare” to execute a batch file named “compare.bat.” This batch file compares the newly created files to the corresponding files in directory “archive.” If the installation is correct, then after each comparison the phrase “FC: no differences encountered” is printed to the console window. Finally, type “cleanup” to execute a batch file named “cleanup.bat.” This batch file deletes all of these newly created files.

4. SUGGESTIONS FOR IMPROVING THE PROGRAM

If you have any suggestions for improving the program, please contact:

Karl J. Ellefsen
U. S. Geological Survey
MS 964, Box 25046
Denver, CO 80225-0046

Email: ellefsen@usgs.gov

5. DISCLAIMERS

This report has not been reviewed for conformity with U.S. Geological Survey editorial standards.

6. ACKNOWLEDGEMENTS

This work was supported by the Toxics Substances Hydrology Program and Mineral Resources Program of the U.S. Geological Survey.

7. REFERENCES

Subcommittee of the SEG Engineering and Groundwater Geophysics Committee, Pullan, S. E., Chairman, 1990, Recommended standard for seismic (/radar) data files in the personal computer environment: Geophysics, vol. 55, no. 9, p. 1260-1271.