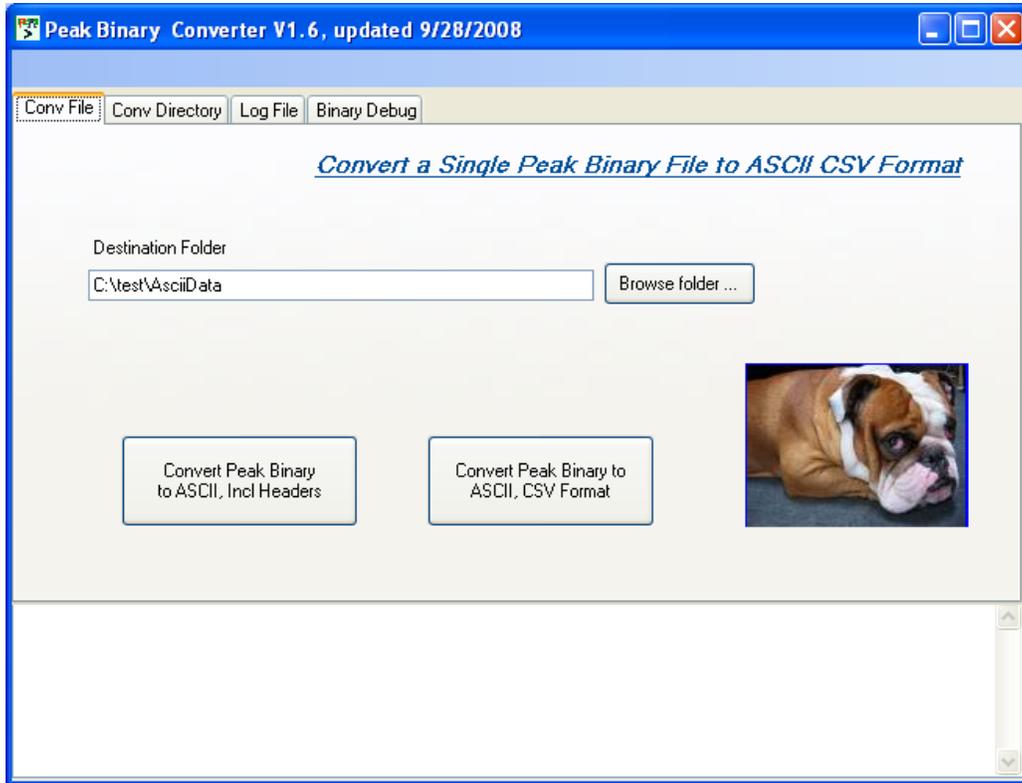


# Peak Binary Converter User Manual : Version 1.6



By

Prof. Take M. Kwon  
Transportation Data Research Laboratory (TDRL)  
University of Minnesota Duluth

# Peak Binary Converter User Manual

By Prof. Taek Kwon

This utility program translates Peak binary files into easy-to-read ASCII files with a Coma Separated Value (CSV) format. The acceptable binary formats are Peak volume, speed, and classification files. However, this utility cannot convert the Peak vehicle record (PVR) files. PVR files can be converted using the BullConverter utility provided to the WIM group.

## 1. Prerequisite

In order to correctly use this utility program, it is important to understand the basic file structure of the Mn/DOT raw data archive. The basic structure is illustrated using the directory structure implemented at the UMD data center. The classification data, which Mn/DOT calls Piezo data and UMD data center calls “Class” data, consists of the Raw and Rawcsv folder, as shown in Figure 1. All peak binary data are stored in a single folder called “Raw” and has the structure shown in Figure 2. Notice that the root folder of the Peak binary data in this case is the “Raw” folder which contains site folders. The site folders have a name that consists of 12 digit numeric numbers and houses all binary data for that site. The first six digits of the site folder name represent the site number, but the actual site number used by Mn/DIT is the three digits used in the middle. Under each site folder, data is further separated into “month\_year” folders. In this documentation, the path to the “Raw” folder is referred to as the Peak binary archive folder.

The Rawcsv folder is the folder structure created as the output of this utility using the binary data in the Raw archive. The basic structure is shown in Figure 3. In this case, the raw csv archive name is “Rawcsv” and consists of site folders. The site folder names are now reduced to six digits. This change was made to avoid the long 12 digit names that are often arduous to type in and easy to make a mistake. Under each site folder, the CSV data are separated into year folders. The year folder contains all CSV files of that year of the site.

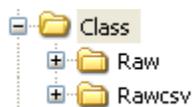


Figure 1: Folder structure of the Peak binary data archive at UMD data center

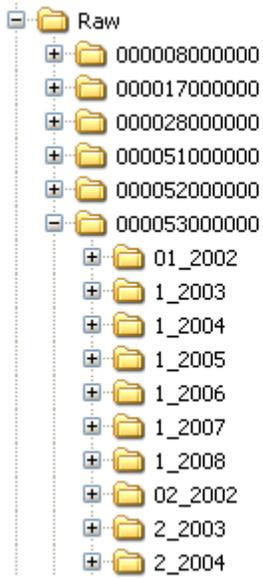


Figure 2: Structure of binary raw data archive

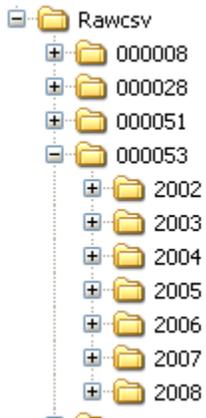


Figure 3: Structure of Rawcsv archive

## 2. CSV Files

All CSV files conform the following filename convention.

**yyyymmdd.000###.ext**

**yyyy:** year

**mm:** month

**dd:** day

**000###:** three digit numeric site ID. Three zeros are for future expansion

**ext:** three extensions used are “vol” for volume, “cls” for classification, and “spd” for speed bin data.

It is important to note that Peak binary files may contain a number of possible combinations of data, for example, some files contain vol and spd data and some files contain vol, spd, and cls data. In the CSV format, each file contains only one type. The data type selection is done automatically by the software, based on the priority of the following type order: (1) cls, (2) spd, and (3) vol. For example, if the binary data contains vol and spd data, only spd data are present in the CSV data. Volume data is still not lost since it can be computed by adding the spd bin data. If all three types of data are present in the Peak binary data, only “cls” data is saved since the “cls” data has the highest priority. Again, volume data can be easily derived from the cls data.

Each CSV file consists of two header lines followed by data. The first header line describes the site and data information. It consists of five fields, siteID, numOfLanes, dataType, date, and lane-by-lane, each separated by a comma. The format is described using an example shown in Figure 4.

```

SiteID=054,numOfLanes=2,dataType=spd,date=20070114,lane-by-lane=False
Time,0,40,45,50,55,60,65,70,75,80,85,100,111
00:00,0,0,0,1,1,0,1,0,0,1,0,0,0
01:00,0,1,2,0,0,1,1,0,0,0,0,0,0
02:00,0,0,0,1,1,1,0,0,0,0,0,0,0
03:00,0,0,0,1,0,0,0,0,0,0,0,0,0
04:00,0,0,0,0,0,0,0,0,0,0,0,0,0
05:00,0,0,0,2,0,1,0,0,0,0,0,0,0
06:00,0,0,0,0,0,0,0,0,0,0,0,0,0
07:00,0,0,1,2,0,0,3,0,0,0,0,0,0
08:00,0,0,0,1,1,3,0,0,0,0,1,0,0
09:00,0,0,1,0,3,1,1,2,0,0,0,0,0
10:00,0,1,1,4,6,5,1,1,0,0,0,0,0
11:00,1,0,3,0,11,4,1,0,0,0,0,0,0
12:00,1,0,1,2,5,5,1,0,0,0,0,0,0
13:00,0,1,0,5,4,2,4,0,0,0,0,0,0
14:00,0,1,3,8,3,2,2,0,0,0,0,0,0
15:00,4,3,8,4,3,1,0,0,0,0,0,0,0
16:00,3,2,4,4,8,0,0,0,0,0,0,0,0
17:00,1,3,1,0,2,0,0,0,0,0,0,0,0
18:00,5,1,5,1,1,0,0,0,0,0,0,0,0
19:00,6,0,1,1,0,0,0,0,0,0,0,0,0
20:00,2,1,1,0,0,0,0,0,0,0,0,0,0
21:00,1,1,1,0,0,0,0,0,0,0,0,0,0
22:00,1,2,0,0,0,0,0,0,0,0,0,0,0
23:00,0,0,0,0,0,0,0,0,0,0,0,0,0

```

Figure 4: CSV data format

The site information heading is in the first line, given by the following:

```
SiteID=054,numOfLanes=2,dataType=spd,date=20070114,lane-by-lane=False
```

This line indicates that the site number is 054, the site has two lanes, data type is speed, the date is Jan 14, 2007, and data is not lane-by-lane data but combined one.

The column heading is in the second line, i.e.,

```
Time,0,40,45,50,55,60,65,70,75,80,85,100,111
```

In this case, it is a speed bin data and consists of time, and speed bin column headings. The meaning of each column is summarized in Table 1.

Table 1: Column headings of speed CSV file

| Column number | Column name | Description  |
|---------------|-------------|--|
| 1             | Time        | hh:mm It is the beginning time of the time slot (one hour). 2:00 means data between 2:00-3:00 am. The 24 hour military system is used. |
| 2             | 0           | 0-40 mph bin   |
| 3             | 40          | 40-45 mph bin  |
| 4             | 45          | 45-50 mph bin  |

|    |     |                   |
|----|-----|-------------------|
| 5  | 50  | 50-55 mph bin     |
| 6  | 55  | 55-60 mph bin     |
| 7  | 60  | 60-65 mph bin     |
| 8  | 65  | 65-70 mph bin     |
| 9  | 70  | 70-75 mph bin     |
| 10 | 75  | 75-80 mph bin     |
| 11 | 80  | 80-85 mph bin     |
| 12 | 85  | 85-100 mph bin    |
| 13 | 100 | 100-111 mph bin   |
| 14 | 111 | 111-above mph bin |

If “lane-by-lane=True” is present in the site information heading, the data is recorded by lane-by-lane. This means that the number of lines for each time period is equal to the number lanes. It is illustrated using a file generated from site 54, shown in Figure 5. In this case, the data type is classification and the data bins are arranged from types 1 though 15 from left to right. The second column is the lane number in the order present in the binary data. Since, this site has two lanes, each time period consists of two lines, each for the corresponding lane. Since the first line contains date, if data and time information is needed, the date information in the first line along with the time in the first column should be combined in the data processing. Again, the time is the beginning time of each one hour period.

```

20080808.000054.cts - Notepad
File Edit Format View Help
SiteID=054,numOfLanes=2,dataType=c1s,date=20080808,lane-by-lane=True
Time,Lane#,Type1,Type2,Type3,Type4,Type5,Type6,Type7,Type8,Type9,Type10,Type11,Type12,Type13,Type14,Type15
00:00,1,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0
00:00,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
01:00,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
01:00,2,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0
02:00,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
02:00,2,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
03:00,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0
03:00,2,0,0,0,0,0,0,0,0,3,0,0,0,0,0,0
04:00,1,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0
04:00,2,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0
05:00,1,1,2,1,0,0,1,0,0,1,0,0,0,0,0,0
05:00,2,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0
06:00,1,0,7,1,0,0,0,0,0,2,0,0,0,0,0,0
06:00,2,1,15,1,1,0,0,0,0,1,0,0,0,0,0,0
07:00,1,0,3,2,0,1,0,0,0,1,0,0,0,0,0,0
07:00,2,0,7,3,0,0,1,0,0,1,0,0,0,0,0,0
08:00,1,1,5,3,1,0,1,0,0,4,0,0,0,0,0,0
08:00,2,0,1,1,2,1,0,0,1,1,0,0,0,0,0,0
09:00,1,4,4,4,0,0,1,0,0,1,0,0,0,0,0,0
09:00,2,0,4,3,0,0,0,1,0,0,0,0,0,0,0,0
10:00,1,0,3,6,0,0,0,0,0,1,0,0,0,0,0,0
10:00,2,0,5,2,0,2,0,0,1,0,0,0,0,0,0,0
11:00,1,0,2,2,0,0,0,0,0,1,0,0,0,0,0,0
11:00,2,0,3,4,1,0,0,0,0,1,1,0,0,0,0,0
12:00,1,2,3,3,0,0,0,0,0,3,0,0,0,0,0,0
12:00,2,0,6,2,1,1,0,0,1,1,0,0,0,0,0,0

```

Figure 5: Classification file example with lane-by-lane=True format

### 3. Peak converter taps and buttons

The functions are implemented using four taps as shown in Figure 6. The functions are described based on each tap.

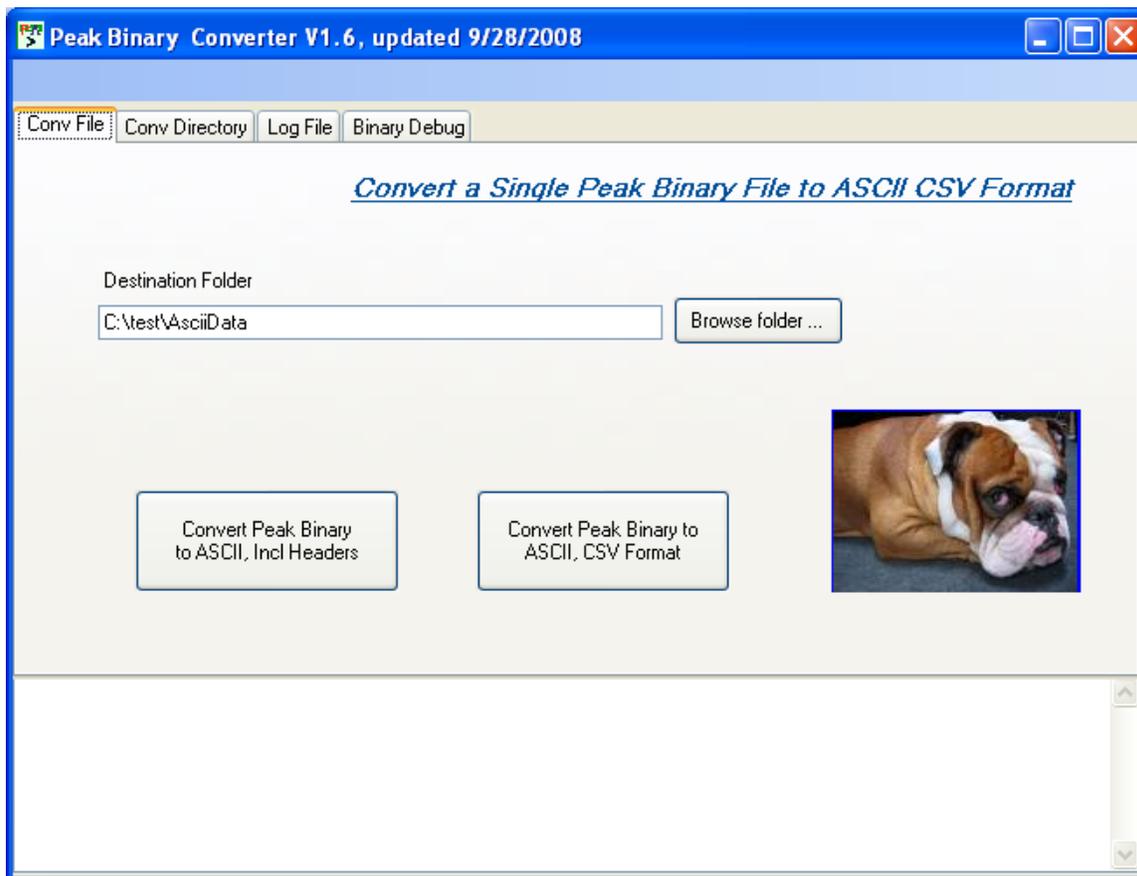


Figure 6. A screen shot of Peak Binary Converter

#### **Conv File Tap**

This tap is designed for examining individual binary files and contains two processing buttons. First, the user must specify the Destination Folder before using the two functions in the tap. Please do not use the CSV archive folder, since running individual files is likely a simple test. The files produced by each button are stored in the Destination Folder.

**“Convert Peak Binary to ASCII, Incl Headers” button.** Pressing this button triggers a file dialog from which user selects a peak binary file. Selecting a binary file and clicking the Open button produces a file that contains all header information of the original followed by the binary data. The output file is also opened on a notepad for immediate viewing. This function is useful to examine the details of a binary file for a debugging purpose.

**“Convert Peak Binary to ASCII, CSV Format” button.** Pressing this function produces a CSV output file, i.e. it removes the detailed header information and organizes the data into the CSV data defined (Section 2).

### Conv Directory tap

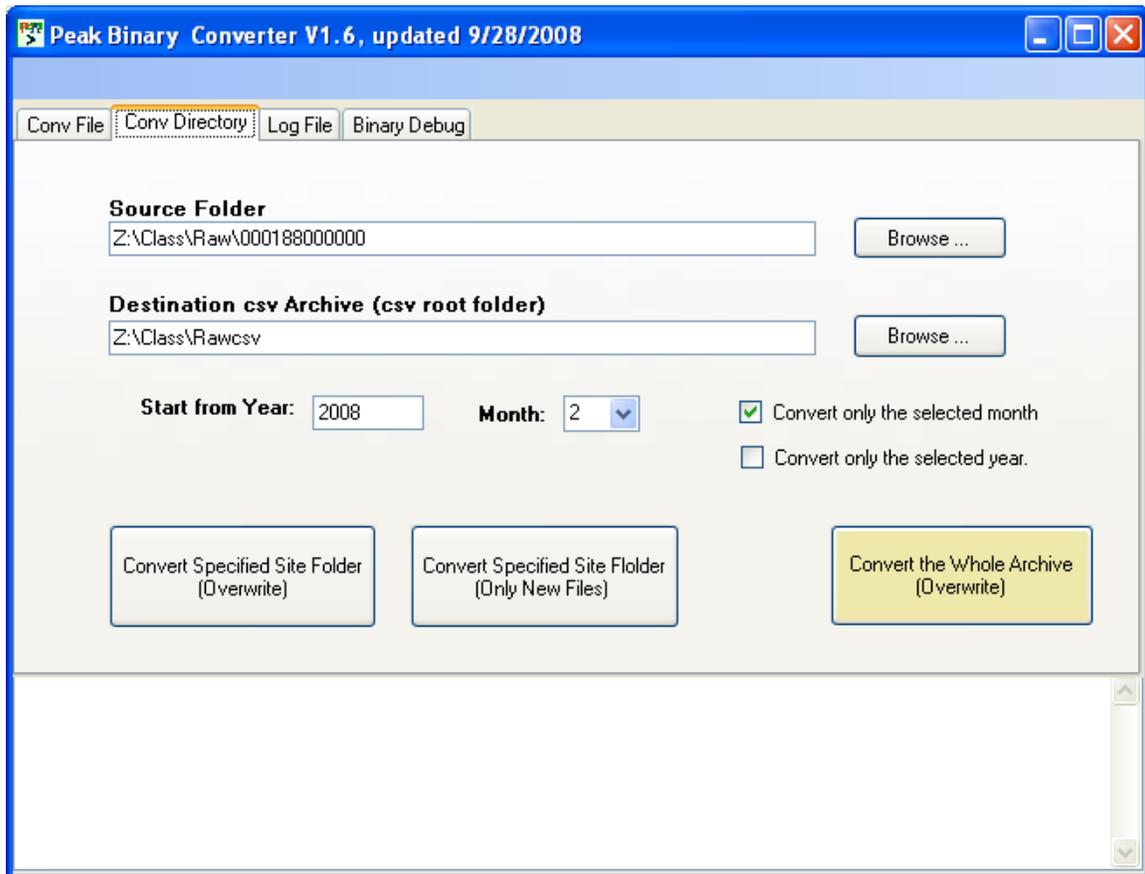


Figure 7: “Conv Directory” tap

This tap contains the conversion utilities that can convert a bulk of files. There are two ways that you can use these utilities. The first approach is converting the data for each site using the left side two buttons, i.e.,



The difference between these two buttons is whether to overwrite the existing converted files or not. The “**Convert Specified Site Folder (Overwrite)**” button overwrites existing converted files while the “**Convert Specified Site Folder (Only New Files)**” button does not overwrite if a converted file already exists. For using these functions, it is important to set the Source Folder to a site directory as shown in Figure 7.

The screenshot shows two input fields with 'Browse...' buttons. The first field is labeled 'Source Folder' and contains the text 'Z:\Class\Raw\000188000000'. The second field is labeled 'Destination csv Archive (csv root folder)' and contains the text 'Z:\Class\Rawcsv'.

Note that the destination folder should always be the CSV archive folder. The software automatically creates the site folders if they do not exist.

Which month or which year to process depends on the selection and year entry, i.e.,

The screenshot shows a 'Start from Year' text box with '2008' and a 'Month' dropdown menu with '2' selected. There are two checkboxes: 'Convert only the selected month' (checked) and 'Convert only the selected year.' (unchecked).

The “Start from Year” text box specifies the year the user want to start processing. If both check marks are unchecked, it will process all of the files starting from the year specified. In the example above, it will only process February 2008 of site 188. The following examples illustrate more choices the user can make.

The screenshot shows a 'Start from Year' text box with '2008' and a 'Month' dropdown menu with '2' selected. There are two checkboxes: 'Convert only the selected month.' (unchecked) and 'Convert only the selected year.' (checked).

All of the 2008 data is processed.

The screenshot shows a 'Start from Year' text box with '2008' and a 'Month' dropdown menu with '8' selected. There are two checkboxes: 'Convert only the selected month' (checked) and 'Convert only the selected year.' (checked).

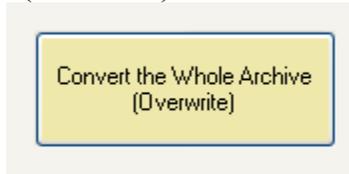
Only 2008 August data is processed. Note that “Convert only the selected month” automatically includes the year restriction.

## Conversion of the whole archive

In order to convert the whole archive, first the source folder must be changed to the Peak binary archive folder, i.e.

The screenshot shows a web interface for converting an archive. It has two text input fields for folders, each with a 'Browse ...' button. The 'Source Folder' field contains 'Z:\Class\Raw' and the 'Destination csv Archive (csv root folder)' field contains 'Z:\Class\Rawcsv'. Below these are two more input fields: 'Start from Year' with '2008' and 'Month' with '8' and a dropdown arrow. To the right are two checkboxes: the first is checked and labeled 'Convert only the selected month', the second is unchecked and labeled 'Convert only the selected year'.

In the above example, “Z:\Class\Raw” is the Peak binary archive folder and “Z:\Class\Rawcsv” is the CSV archive. Next, simply click the “Convert the Whole Archive (Overwrite)” button.



In the above example, the binary data in every site for year 2008 Aug files are converted and saved to the Rawcsv folder. Again, the CSV site folders are automatically created if they do not exist, so there is no need to manually create CSV site folders.

**Caution:** Converting the whole archive can take a long time, since the archive size is very large. It presently contains about 160,000 binary files in 5,355 folders. Therefore, it is strongly recommended that “Convert only the selected year” is always checked.

## 4. Log Files

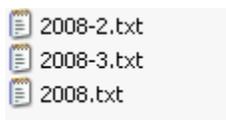
There are two functions for log files, and it is important to read the log file after each processing.



It is important to clean up the log file often, since its file size can grow substantially and can reduce the processing speed. The “Cleanup Log File” button does not actually delete the log file but it copies to a different file name. The log file is plain text files and located in

“C:\Program Files\UMD\Peak\_Conv\Log”

Here is an example of files in the log directory.



In this case, 2008.txt is the current log file and the rest are backup log files that are created when you click on the “Cleanup Log File” button. If these files are not needed, the user should delete the backup log files. You can also delete the current log file. In that case, when you process, the software will recreate the current log file.

The “View Log File” opens the current log file on a notepad. Inside the log file, the line starting with “Error” contains important information. In the example shown in Figure 7, site 460, 11/3/2007 had an error. There are two possibilities. First, the binary data is corrupted, which would be the most case. Second, the decoder was not able to recognize the format. In either case, the error message is present in the csv file. If the data line does not start with time, it is a sign of the error and should not use the CSV file. Some cases, the data is good up to the line indicating Error, in that case, the good data can be still used for further processing.

```
>> site; 000460000000
>>> Converting Dir: Z:\Class\Raw\000460000000\10_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\11_2007
Error: Data may be corrupted. Counter did not change. 11/3/2007
>>> Converting Dir: Z:\Class\Raw\000460000000\12_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\1_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\2_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\3_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\6_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\7_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\8_2007
>>> Converting Dir: Z:\Class\Raw\000460000000\9_2007
---- 305 files converted
```

Figure 7: Log file example

