Advanced $UsnJrnl Forensics

blueangel

blueangel1275@gmail.com

http://forensic-note.blogspot.kr/

Junghoon Oh

1. $UsnJrnl

2. $UsnJrnl Record Carving

3. NTFS Log Tracker v1.4

4. Conclusion

$UsnJrnl

Journal(Change) Log File of NTFS

- **This file is used to determine whether any change is occurred in a specific file by applications.**

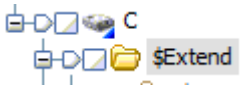
- **From Win7, Journal Function is activated by default**
  - In case of deactivation setting(in Win XP), it is possible to activate through "Fsutil".

    > fsutil usn [createjournal] m=<MaxSize> a=<AllocationDelta> <VolumePath>
  - For more information about "Fsutil" : http://technet.microsoft.com/en-us/library/cc788042.aspx

- **The file is composed of "$Max" attribute and "$J" attribute**
  - $Max : The meta data of change log is stored.
  - $J : The actual change log records are stored.
    - ✓ Each record has USN(Update Sequence Number) information.
    - ✓ The record order is determined with USN.
    - ✓ USN = the offset value of a record within $J attribute
    - ✓ USN information is also stored in then $STANDARD_INFORMATION attribute of a MFT record

Journal(Change) Log File of NTFS(continue…)

- **Location**
  - The file is located under "$Extend" folder.



Name	File Created	Last Written	Entry Modified	Last Accessed	Logical Size
$UsnJrnl:$J					1,246,483,680
$UsnJrnl:$Max					32

- **The size of log data(generally…)**
  - In case of full time use(24 hours/day), the log for 1~2 days are recorded.
  - In case of regular use(8 hours/day), the log for 4~5 days are recorded.

- **Forensic Readiness**
  - changing log size bigger(more than 1 GB??)

- **Digital Forensic Profit**
  - The investigator can confirm every NTFS's events(creation, deletion, modification…) in specific period.

The Structure of $Max attribute

- **The size of $Max attribute**
  - 32 Bytes fixed size

- **The format of $Max attribute**

Offset	Size	Stored Information	Detail
0x00	8	Maximum Size	The maximum size of log data
0x08	8	Allocation Size	The size of allocated area when new log data is saved.
0x10	8	USN ID	The creation time of "$UsnJrnl" file(FILETIME)
0x18	8	Lowest Valid USN	The least value of USN in current records With this value, investigator can approach the start point of first record within "$J" attribute

The Structure of $J attribute

- **The log records of variable size are listed consecutively.**

- **• The zero-filled "Sparse Area" occupies front part of an attribute.**

- The reason for this structure is because the operating system keeps the same size of the log data saved in the $J attribute.

- The record allocation policy of $J attribute
  1. The new log records are added at the end of the attribute.
  2. If the total size of the added records exceeds "Allocation Size", the operation system assures that the size of the entire log data exceeds "Maximum Size".
  3. If the size of the entire log data exceeds "Maximum Size", the front area of attribute is occupied by zero as much as size of "Allocation Size".(Actually, disk area is not filled by zero.)

- Thus, the logical size of $J attribute grow continuously, but the size of area saving actual data is kept constant.
- The general size of log data is 0x200000 ~ 0x23FFFFF

The Structure of $J attribute(continue…)

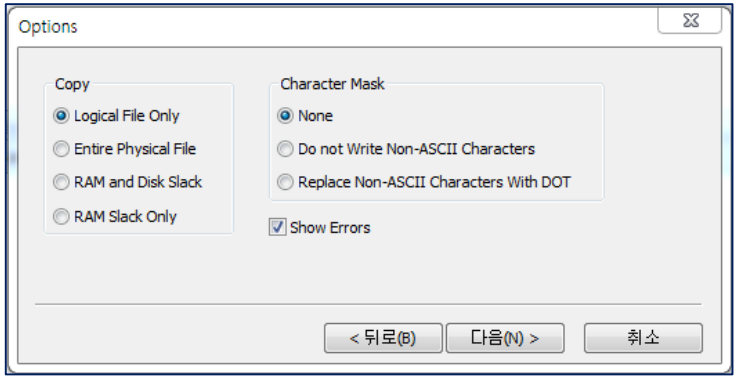
- **After Logical area, there are valid data…**

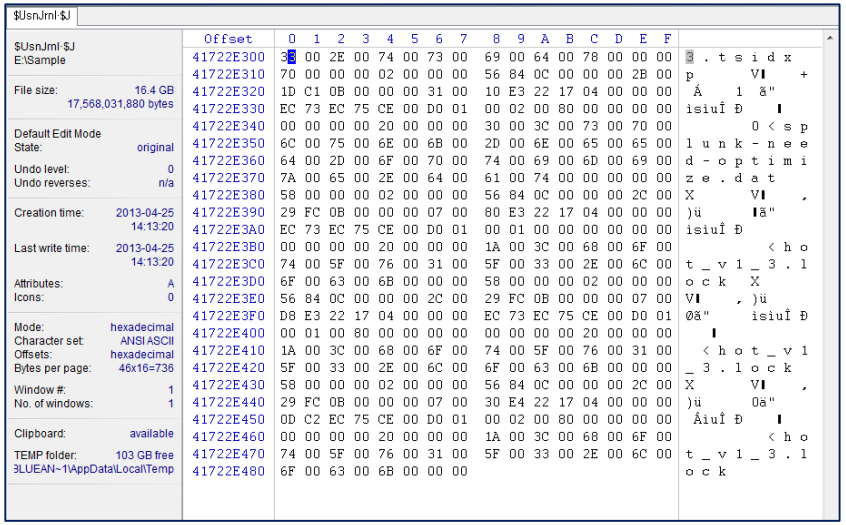
	Name	Initialized Size	Logical Size	Physical Size
1	$ObjId	0	0	0
2	$ObjId:$O	393,216	393,216	393,216
3	$Quota	0	0	0
4	$Quota:$O	88	88	88
5	$Quota:$Q	208	208	208
6	$Reparse	0	0	0
7	$Reparse:$R	4,096	4,096	4,096
8	$RmMetadata	336	336	336
9	$UsnJrnl	0	0	0
10	$UsnJrnl:$J	17,568,031,880	17,568,031,880	17,568,301,056
11	$UsnJrnl:$Max	32	32	32

Text | Hex | Doc | Transcript | Picture | Report | Console | Details | Output

```
17568031728 D8 E3 22 17 04 00 00 00 EC 73 EC 75 CE 00 D0 01   Øã"·····ìsìuÎ·Ð·
17568031744 00 01 00 80 00 00 00 00 00 00 00 00 20 00 00 00   ···|····· ···
17568031760 1A 00 3C 00 68 00 6F 00 74 00 5F 00 76 00 31 00   ··<·h·o·t·_·v·1·
17568031776 5F 00 33 00 2E 00 6C 00 6F 00 63 00 6B 00 00 00   _·3·.·l·o·c·k···
17568031792 58 00 00 00 02 00 00 00 56 84 0C 00 00 00 2C 00   X·····V|····,·
17568031808 29 FC 0B 00 00 00 07 00 30 E4 22 17 04 00 00 00   )ü·····0ä"·····
17568031824 0D C2 EC 75 CE 00 D0 01 00 02 00 80 00 00 00 00    ÂìuÎ·Ð····|····
17568031840 00 00 00 00 20 00 00 00 1A 00 3C 00 68 00 6F 00   ···· ·····<·h·o·
17568031856 74 00 5F 00 76 00 31 00 5F 00 33 00 2E 00 6C 00   t·_·v·1·_·3·.·l·
17568031872 6F 00 63 00 6B 00 00 00 68 00 00 00 02 00 00 00   o·c·k···h·····
17568031888 2A 7E 03 00 00 00 42 00 92 99 01 00 00 00 01 00   *~····B·||·····
17568031904 88 E4 22 17 04 00 00 00 1C A1 7D 77 CE 00 D0 01   |ä"·····¡}wÎ·Ð·
17568031920 03 81 00 00 00 00 00 00 00 00 00 00 20 00 00 00   ·|····· ··
17568031936 28 00 3C 00 4F 00 54 00 65 00 6C 00 65 00 44 00   (·<·O·T·e·l·e·D·
17568031952 61 00 74 00 61 00 5F 00 33 00 34 00 38 00 30 00   a·t·a·_·3·4·8·0·
17568031968 5F 00 37 00 2E 00 65 00 74 00 6C 00 00 00 00 00   _·7·.·e·t·l·····
17568031984 68 00 00 00 02 00 00 00 6F 7E 0B 00 00 00 63 00   h·····o~····c·
17568032000 92 99 01 00 00 00 01 00 F0 E4 22 17 04 00 00 00   ||·····ðä"·····
```

- **Extracting $J attribute by Encase 6(default "Logical File Only")**
  - There is no valid data which is located after logical area.





Collection of $UsnJrnl

- **Encase**
  - Extract $J attribute after selecting "Entire Physical File" option

- **Winhex**
  - Default, this tool extracts file by Physical Size

- **ExtractUsnJrnl (https://github.com/jschicht/ExtractUsnJrnl)**
  - This tool can extract only valid data except sparse area.

이름	수정한 날짜	유형	크기
$UsnJrnl_$J.bin	2014-11-05 오전...	BIN 파일	34,598KB

The format of record (http://msdn.microsoft.com/en-us/library/aa365722.aspx)

Offset	Size	Stored Information	Detail
0x00	4	Size of Record	
0x04	2	Major Version	2(Change Journal Software's major version)
0x06	2	Minor Version	2(Change Journal Software's major version)
0x08	8	MFT Reference Number	"MFT Reference Number" of file or directory that effected by currently change event.
0x10	8	Parent MFT Reference Number	"MFT Reference Number" of parent directory of file and directory that effected by currently change event. The full path information can be obtained with this information and $MFT.
0x18	8	USN	Update Sequence Number
0x20	8	TimeStamp(FILETIME)	Event Time(UTC +0)
0x28	4	Reason Flag	The flag of change event
0x2C	4	Source Information	The subject that triggers change of event
0x30	4	Security ID	
0x34	4	File Attributes	The attribute information of the object effected by current event. Generally, it is used for classifying the object into a file or directory.
0x38	2	Size of Filename	The size of object name effected by current event
0x3A	2	Offset to Filename	The offset of object name within record
0x3C	N	Filename	The object(file or directory) name effected by current event

- The reason for using "Parent MFT Reference Number" instead of "MFT Reference Number"
  - ✓ If "MFT Reference Number" is used, full path information may not be obtained when relevant file is deleted.

Reason Flag (http://msdn.microsoft.com/en-us/library/aa365722.aspx)

Flag	Description
0x01	The file was overwritten.
0x02	The file or directory was added to
0x04	The file or directory was truncated.
0x10	The named data streams for a file is overwritten.
0x20	A named data streams for the file were added.
0x40	A named data streams for the file was truncated
0x100	The file or directory was created for the first time.
0x200	The file or directory was deleted.
0x400	The file's or directory's extended attributes were changed.
0x800	The access rights to the file or directory was changed.
0x1000	The file or directory was renamed.(previous name)
0x2000	The file or directory was renamed.(new name)
0x4000	A user changed the FILE_ATTRIBUTE_NOT_CONTENT_INDEXED attribute.
0x8000	A user has either changed one or more file or directory attributes or one or more time stamps.
0x10000	A hard link was added to or removed from the file or directory
0x20000	The compression state of the file or directory was changed from or to compressed.
0x40000	The file or directory was encrypted or decrypted.
0x80000	The object identifier of the file or directory was changed.
0x100000	The reparse point contained in the file or directory was changed, or a reparse point was added to or deleted from the file or directory.
0x200000	A named stream has been added to or removed from the file, or a named stream has been renamed.
0x80000000	The file or directory was closed.

Source Information (http://msdn.microsoft.com/en-us/library/aa365722.aspx)

Flag	Description
0x00	Normal Event
0x01	The operation provides information about a change to the file or directory made by the operating system
0x02	The operation adds a private data stream to a file or directory.
0x04	The operation creates or updates the contents of a replicated file.

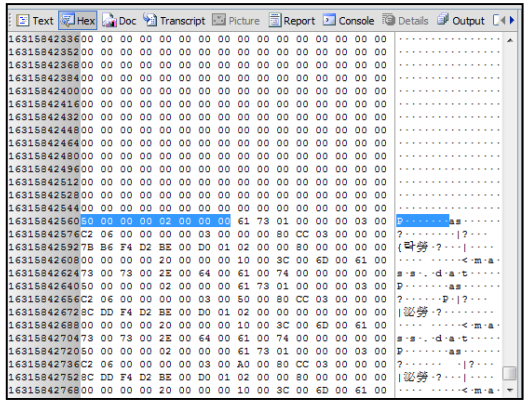
File Attribute (http://msdn.microsoft.com/en-us/library/gg258117.aspx)

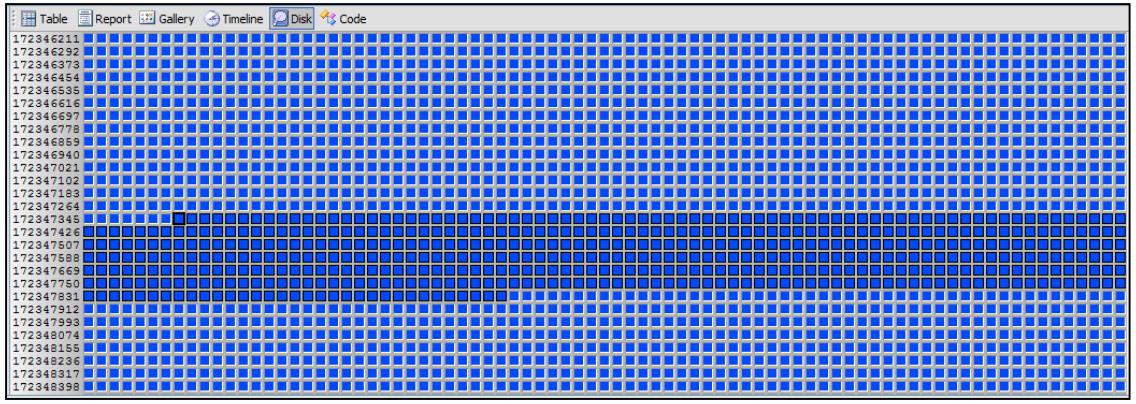
Value	Description
0x01	A file that is read-only.
0x02	The file or directory is hidden
0x04	A file or directory that the operating system uses a part of, or uses exclusively.
0x10	The handle that identifies a directory.
0x20	An archive file or directory.
0x40	This value is reserved for system use
0x80	A file that does not have other attributes set.
0x100	A file that is being used for temporary storage.
0x200	A file that is a sparse file.
0x400	A file or directory that has an associated reparse point, or a file that is a symbolic link.
0x800	A file or directory that is compressed.
0x1000	This attribute indicates that the file data is physically moved to offline storage.
0x2000	The file or directory is not to be indexed by the content indexing service.
0x4000	A file or directory that is encrypted.
0x8000	The directory or user data stream is configured with integrity (only supported on ReFS volumes).
0x10000	0 This value is reserved for system use.
0x20000	The user data stream not to be read by the background data integrity scanner (AKA scrubber).

$UsnJrnl Record Carving

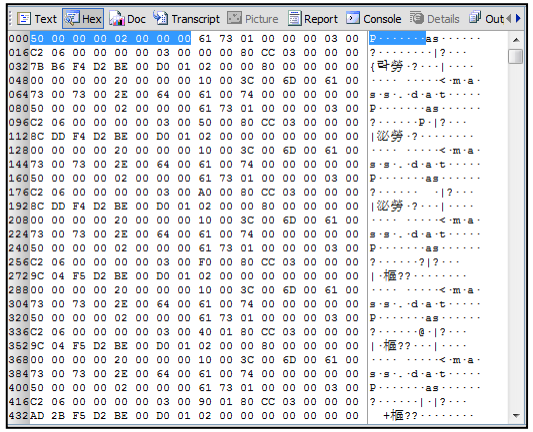
$UsnJrnl records in Unallocated Area

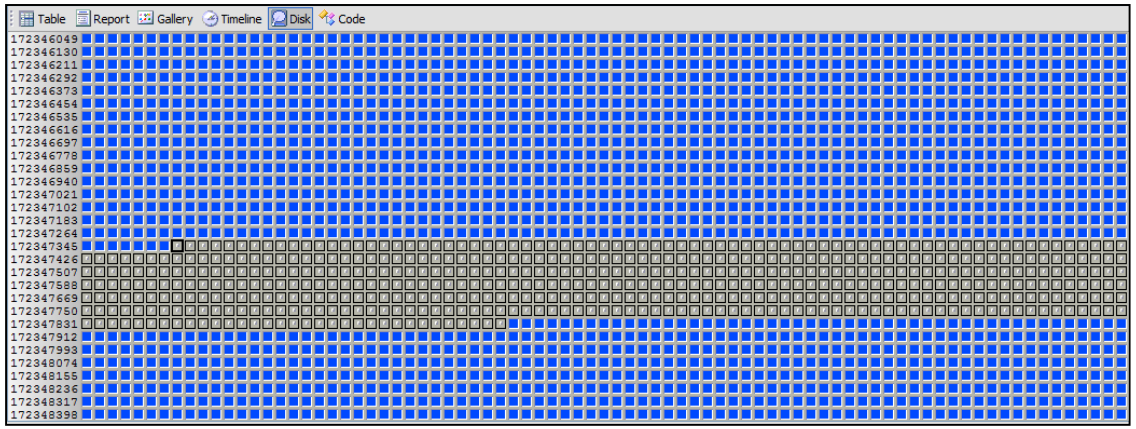
- **The location of first $UsnJrnl record in disk : 172347352 sector**





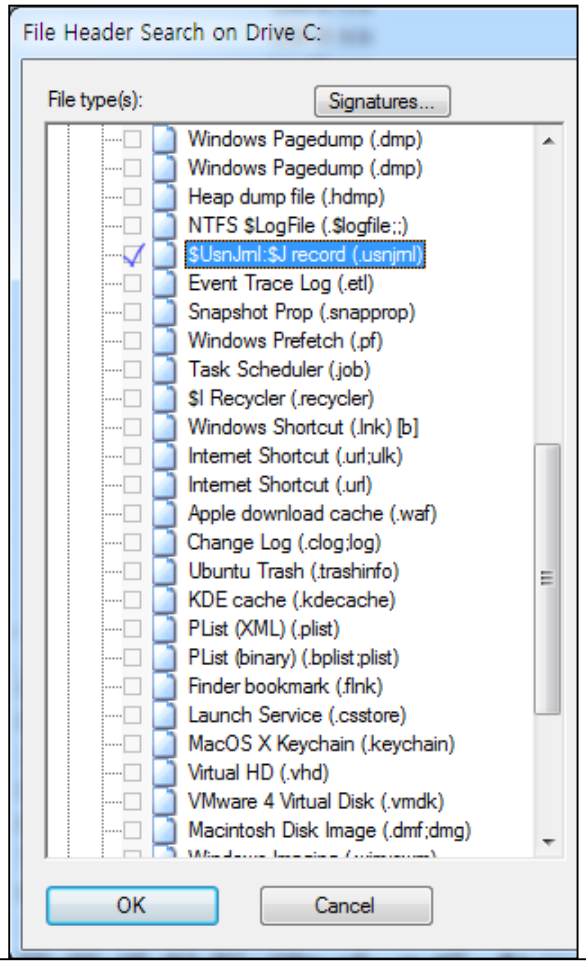
- **In a couple of hours… ➔ the space which saves $Usnjrnl record is changed to unallocated space. ➔ There are valid data in this area…**



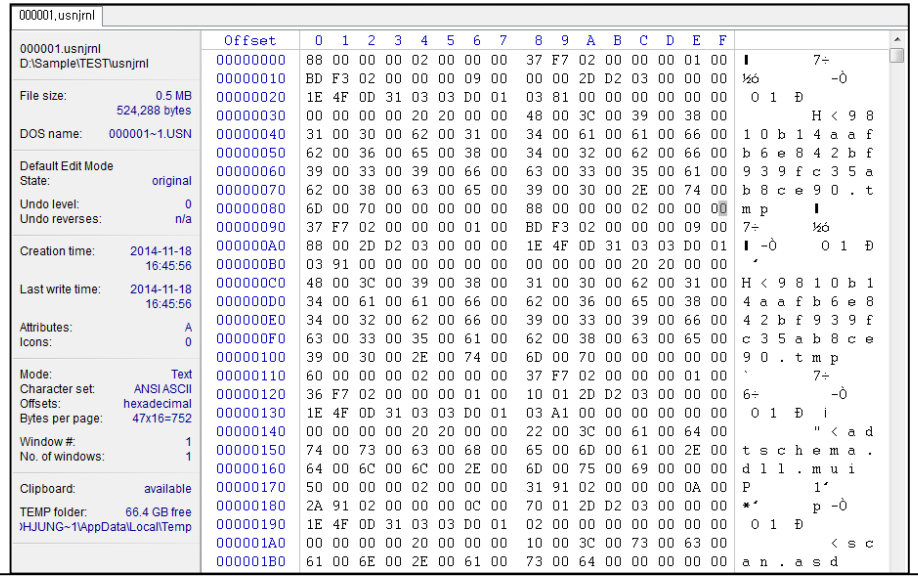


Existing Tool 1

- **Carving function of USN Record in X-way forensics(Tool➔Disk Tool➔File Recovery by Type)**
  - After record carving, the tool saves the records into many files.

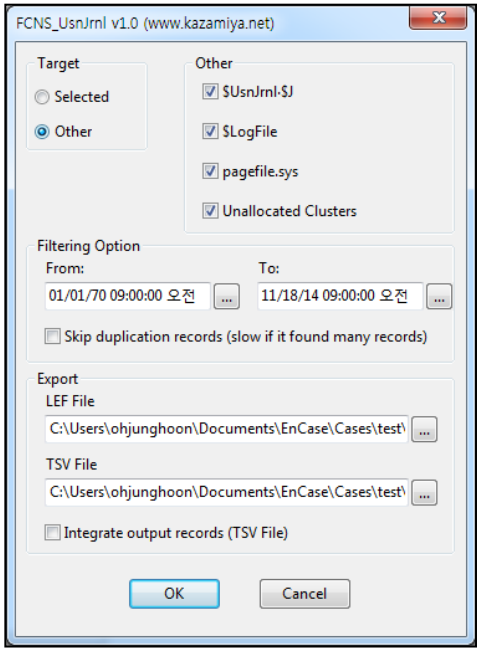


이름	수정한 날짜	유형	크기
000001.usnjrnl	2014-11-18 오후...	USNJRNL 파일	512KB
000002.usnjrnl	2014-11-18 오후...	USNJRNL 파일	512KB
000003.usnjrnl	2014-11-18 오후...	USNJRNL 파일	56KB
000004.usnjrnl	2014-11-18 오후...	USNJRNL 파일	528KB
000005.usnjrnl	2014-11-18 오후...	USNJRNL 파일	556KB
000006.usnjrnl	2014-11-18 오후...	USNJRNL 파일	452KB
000007.usnjrnl	2014-11-18 오후...	USNJRNL 파일	128KB
000008.usnjrnl	2014-11-18 오후...	USNJRNL 파일	748KB
000009.usnjrnl	2014-11-18 오후...	USNJRNL 파일	372KB
000010.usnjrnl	2014-11-18 오후...	USNJRNL 파일	512KB
000011.usnjrnl	2014-11-18 오후...	USNJRNL 파일	12KB



Existing Tool 2

- **FCNS_UsnJrnl EnScript**
  - EnPack for Encase 7
  - There is no full path information of file or directory.
  - L01, CSV output



이름	수정한 날짜	유형	크기
FCNS_UsnJrnl_Data.L01	2014-11-18 오후...	EnCase Logical Evidence File	2,096,995KB
FCNS_UsnJrnl_Data.L02	2014-11-18 오후...	L02 파일	19,570,272KB
FCNS_UsnJrnl_Records.csv	2014-11-18 오후...	Microsoft Excel 쉼표로 구분된 값 파일	1,177,528KB

ItemPath	PS	SO	TimeStamp	FileName	FileID	ParentID	Reason	Reason(String)
C:\Unallocated Clusters	0	0		1				
pnidevdolocsemfrmotblopoahshgupgua	6881385	655471	000a0030	ADS				
C:\Unallocated Clusters	3459424	400	11/17/14 03:15:05 오후	aosmon.log	162023	25466	80000002	DATA
C:\Unallocated Clusters	3459426	176	11/17/14 03:15:05 오후	PaLogU.log	141286	141610	80000002	DATA
C:\Unallocated Clusters	3459426	448	11/17/14 03:15:05 오후	agent.dbf-journal	39515	141611	103	CREATE
C:\Unallocated Clusters	3459427	32	11/17/14 03:15:05 오후	agent.dbf	124844	141611	1	DATA
C:\Unallocated Clusters	3459427	480	11/17/14 03:15:05 오후	SendLog.apc	39515	141611	80000102	CREATE
C:\Unallocated Clusters	3459428	136	11/17/14 03:15:05 오후	PaSvc.log	192015	141610	3	DATA
C:\Unallocated Clusters	3459428	216	11/17/14 03:15:06 오후	SendLog.apc	39515	141611	80000200	DELETE
C:\Unallocated Clusters	3459429	32	11/17/14 03:15:06 오후	PaLogU.log	141286	141610	80000002	DATA
C:\Unallocated Clusters	3459429	112	11/17/14 03:15:06 오후	agent.dbf	124844	141611	80000001	DATA
C:\Unallocated Clusters	3459430	400	11/17/14 03:15:06 오후	PaLogU.log	141286	141610	80000002	DATA
C:\Unallocated Clusters	3459431	160	11/17/14 03:15:06 오후	agent.dbf-journal	39515	141611	103	CREATE

Record Carving

Offset	Size	Stored Information	Detail
0x00	4	Size of Record	
0x04	2	Major Version	2(Change Journal Software's major version)
0x06	2	Minor Version	2(Change Journal Software's major version)
0x08	8	MFT Reference Number	"MFT Reference Number" of file or directory that effected by currently change event.
0x10	8	Parent MFT Reference Number	"MFT Reference Number" of parent directory of file and directory that effected by currently change event. The full path information can be obtained with this information and $MFT.
0x18	8	USN	Update Sequence Number
0x20	8	TimeStamp(FILETIME)	Event Time(UTC +0)
0x28	4	Reason Flag	The flag of change event
0x2C	4	Source Information	The subject that triggers change of event
0x30	4	Security ID	
0x34	4	File Attributes	The attribute information of the object effected by current event. Generally, it is used for classifying the object into a file or directory.
0x38	2	Size of Filename	The size of object name effected by current event
0x3A	2	Offset to Filename	The offset of object name within record
0x3C	N	Filename	The object(file or directory) name effected by current event

- Signature : ₩x??₩x??₩x00₩x00₩x02₩x00₩x00₩x00
- Sub-checking Point : USN, TimeStamp, Source Information, Size/Offset of Filename

The Result of Record Carving

System	The size of Unallocated Area	The number of recovered records(De-Duplication)	The period of recovered records
A(Win7 64bit)	72G(HDD)	32,379,635	2014-02-10 ~ 2014-11-03
B(Win7 64bit)	120G(HDD)	36,650,278	2014-01-28 ~ 2014-11-10
C(Win7 64bit)	269G(HDD)	24,907,010	2014-01-28 ~ 2014-11-13
D(Win7 64bit)	120G(HDD)	22,310,563	2013-10-27 ~ 2014-12-23

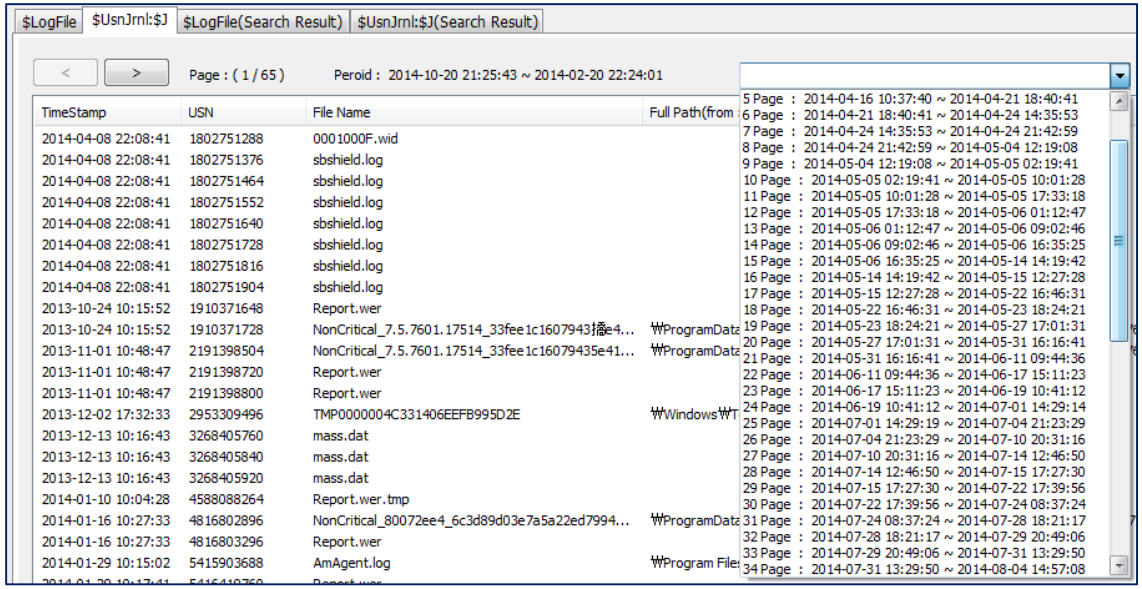
- **There are about 30,000,000 records in unallocated space.**
  - There are some records before 10~11 months.
  - Generally, there are 300,000 records in $UsnJrnl:$J on average.

- **There are some records before formatting current system.**

NTFS Log Tracker v1.4

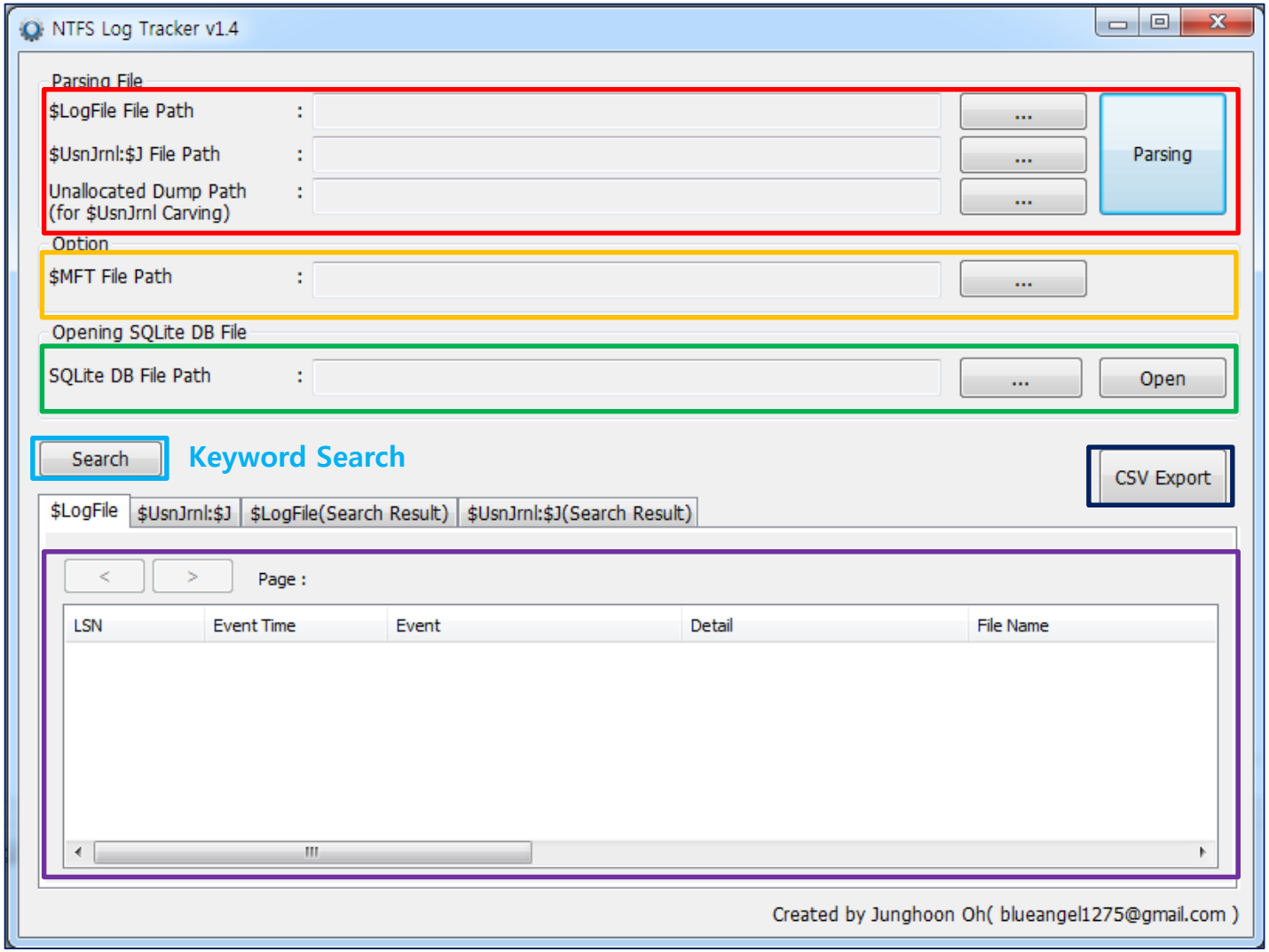
Updated List (https://sites.google.com/site/forensicnote/ntfs-log-tracker)

1. **$UsnJrnl record carving from unallocated space**
   - Carving result is printed out by page unit. (500,000 records by one page)
   - The full path information is printed out. (from $MFT)
   - Indexing page information(In case of more than 3 pages)
     - ✓ After ordering by USN, first and last record's time information are recorded.



2. **Supporting source file extracted by "ExtractUsnJrnl" tool**
3. **Changing interface of keyword search**
4. **Supporting Korean keyword search**
5. **Tab bug is fixed.**

User interface



Source File for Parsing

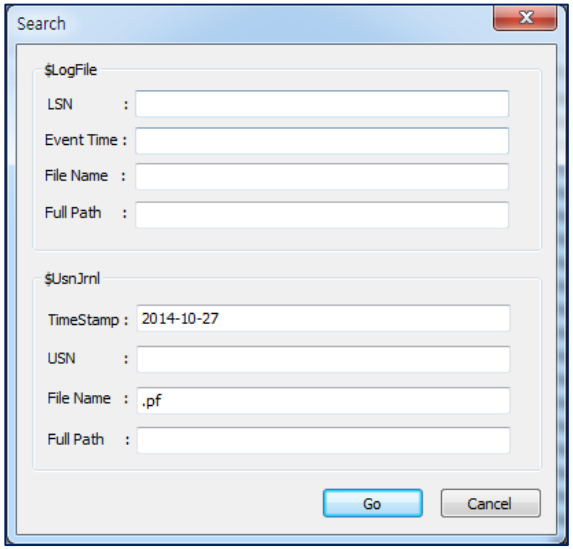
$MFT for Full Path Construction

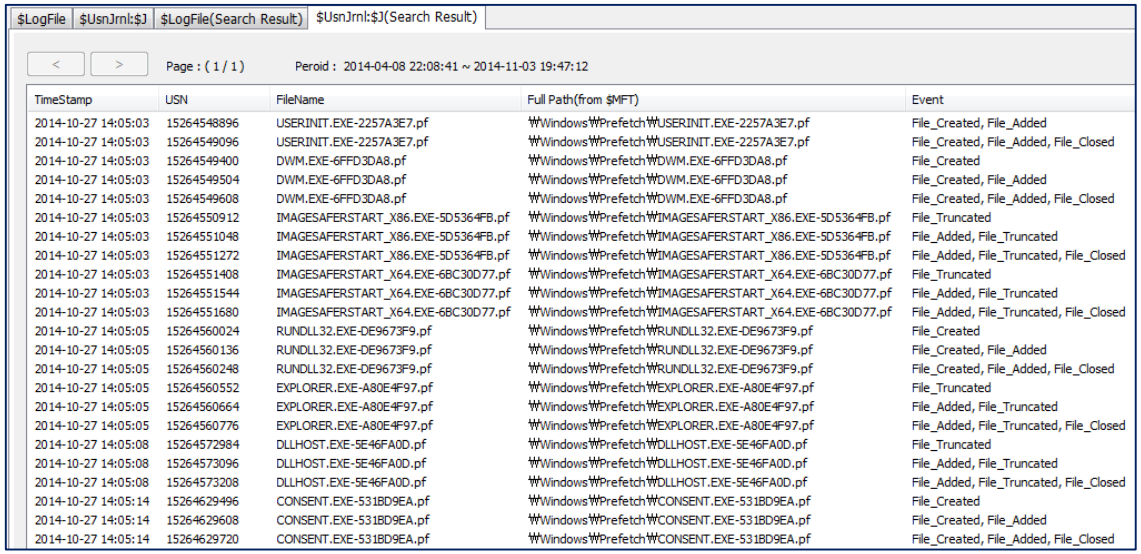
Opening DB file created by this tool

Exporting CSV format

Parsed Data Output

Keyword Search





TimeStamp	USN	FileName	Full Path(from $MFT)	Event
2014-10-27 14:05:03	15264548896	USERINIT.EXE-2257A3E7.pf	₩Windows₩Prefetch₩USERINIT.EXE-2257A3E7.pf	File_Created, File_Added
2014-10-27 14:05:03	15264549096	USERINIT.EXE-2257A3E7.pf	₩Windows₩Prefetch₩USERINIT.EXE-2257A3E7.pf	File_Created, File_Added, File_Closed
2014-10-27 14:05:03	15264549400	DWM.EXE-6FFD3DA8.pf	₩Windows₩Prefetch₩DWM.EXE-6FFD3DA8.pf	File_Created
2014-10-27 14:05:03	15264549504	DWM.EXE-6FFD3DA8.pf	₩Windows₩Prefetch₩DWM.EXE-6FFD3DA8.pf	File_Created, File_Added
2014-10-27 14:05:03	15264549608	DWM.EXE-6FFD3DA8.pf	₩Windows₩Prefetch₩DWM.EXE-6FFD3DA8.pf	File_Created, File_Added, File_Closed
2014-10-27 14:05:03	15264550912	IMAGESAFERSTART_X86.EXE-5D5364FB.pf	₩Windows₩Prefetch₩IMAGESAFERSTART_X86.EXE-5D5364FB.pf	File_Truncated
2014-10-27 14:05:03	15264551048	IMAGESAFERSTART_X86.EXE-5D5364FB.pf	₩Windows₩Prefetch₩IMAGESAFERSTART_X86.EXE-5D5364FB.pf	File_Added, File_Truncated
2014-10-27 14:05:03	15264551272	IMAGESAFERSTART_X86.EXE-5D5364FB.pf	₩Windows₩Prefetch₩IMAGESAFERSTART_X86.EXE-5D5364FB.pf	File_Added, File_Truncated, File_Closed
2014-10-27 14:05:03	15264551408	IMAGESAFERSTART_X64.EXE-6BC30D77.pf	₩Windows₩Prefetch₩IMAGESAFERSTART_X64.EXE-6BC30D77.pf	File_Truncated
2014-10-27 14:05:03	15264551544	IMAGESAFERSTART_X64.EXE-6BC30D77.pf	₩Windows₩Prefetch₩IMAGESAFERSTART_X64.EXE-6BC30D77.pf	File_Added, File_Truncated
2014-10-27 14:05:03	15264551680	IMAGESAFERSTART_X64.EXE-6BC30D77.pf	₩Windows₩Prefetch₩IMAGESAFERSTART_X64.EXE-6BC30D77.pf	File_Added, File_Truncated, File_Closed
2014-10-27 14:05:05	15264560024	RUNDLL32.EXE-DE9673F9.pf	₩Windows₩Prefetch₩RUNDLL32.EXE-DE9673F9.pf	File_Created
2014-10-27 14:05:05	15264560136	RUNDLL32.EXE-DE9673F9.pf	₩Windows₩Prefetch₩RUNDLL32.EXE-DE9673F9.pf	File_Created, File_Added
2014-10-27 14:05:05	15264560248	RUNDLL32.EXE-DE9673F9.pf	₩Windows₩Prefetch₩RUNDLL32.EXE-DE9673F9.pf	File_Created, File_Added, File_Closed
2014-10-27 14:05:05	15264560552	EXPLORER.EXE-A80E4F97.pf	₩Windows₩Prefetch₩EXPLORER.EXE-A80E4F97.pf	File_Truncated
2014-10-27 14:05:05	15264560664	EXPLORER.EXE-A80E4F97.pf	₩Windows₩Prefetch₩EXPLORER.EXE-A80E4F97.pf	File_Added, File_Truncated
2014-10-27 14:05:05	15264560776	EXPLORER.EXE-A80E4F97.pf	₩Windows₩Prefetch₩EXPLORER.EXE-A80E4F97.pf	File_Added, File_Truncated, File_Closed
2014-10-27 14:05:08	15264572984	DLLHOST.EXE-5E46FA0D.pf	₩Windows₩Prefetch₩DLLHOST.EXE-5E46FA0D.pf	File_Truncated
2014-10-27 14:05:08	15264573096	DLLHOST.EXE-5E46FA0D.pf	₩Windows₩Prefetch₩DLLHOST.EXE-5E46FA0D.pf	File_Added, File_Truncated
2014-10-27 14:05:08	15264573208	DLLHOST.EXE-5E46FA0D.pf	₩Windows₩Prefetch₩DLLHOST.EXE-5E46FA0D.pf	File_Added, File_Truncated, File_Closed
2014-10-27 14:05:14	15264629496	CONSENT.EXE-531BD9EA.pf	₩Windows₩Prefetch₩CONSENT.EXE-531BD9EA.pf	File_Created
2014-10-27 14:05:14	15264629608	CONSENT.EXE-531BD9EA.pf	₩Windows₩Prefetch₩CONSENT.EXE-531BD9EA.pf	File_Created, File_Added
2014-10-27 14:05:14	15264629720	CONSENT.EXE-531BD9EA.pf	₩Windows₩Prefetch₩CONSENT.EXE-531BD9EA.pf	File_Created, File_Added, File_Closed

- Target field of keyword search
  - ✓ $LogFile : LSN, Event Time, File Name, Full Path
  - ✓ $UsnJrnl : TimeStamp, USN, File Name, Full Path
- Using LIKE operation of SQL
- If multi-keyword are entered, the keywords are used by "AND" operation.

Conclusion

Conclusion

- **Tracking NTFS's history with $UsnJrnl**
  - Creation, deletion, modification, renaming and moving of file and directory
  - It is possible to find trace of deleted file.
  - The event of program execution and opening document can be found through tracking prefetch file and LNK file's history.

- **Collection of $UsnJrnl:$J**
  - Encase or Winhex
  - ExtractUsnJrnl

- **$UsnJrnl record carving from unallocated space**
  - There are mass $UsnJrnl records in unallocated space.
  - Tracking old file system history(before several months) through $UsnJrnl record carving

Question and Answer