CCIT
CULTURE CONTENTS &
INFORMATION TECHNOLOGY

010 Editor Prefetch 템플릿 개발

김현진, 지창환, 허송이

JOONGBU UNIVERSITY
JBU
중부대학교

1. 프로젝트 개요

1. 프로젝트 소개
2. 목표 및 기대 효과

2. 수행 과정

1. 참고 자료
2. 코드 구현

3. 산출물

1. PF.bt
2. 공식 템플릿 등록
3. 향후 계획

1. 프로젝트 개요

1. 프로젝트 소개
2. 목표 및 기대 효과

1-1. 프로젝트 소개 - 배경

① 010Editor는 검찰, 경찰 등 국가 주요기관에서 널리 사용하며 인지도가 높은 도구!

② Prefetch 파일로 악성코드 탐지 가능!

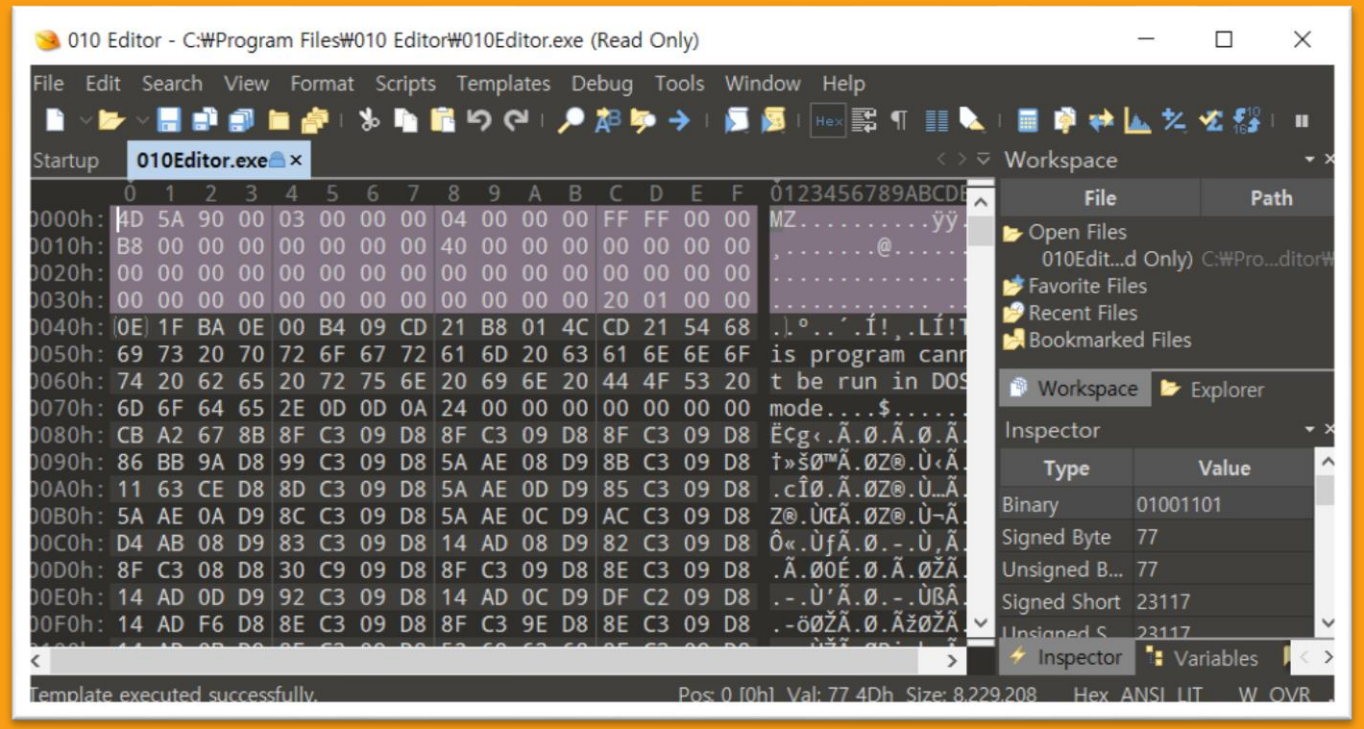
③ Prefetch 템플릿 존재 x

010 Editor
프리패치 템플릿 개발

CCIT CULTURE CONTENTS & INFORMATION TECHNOLOGY
JOONGBU UNIVERSITY JBU 중부대학교

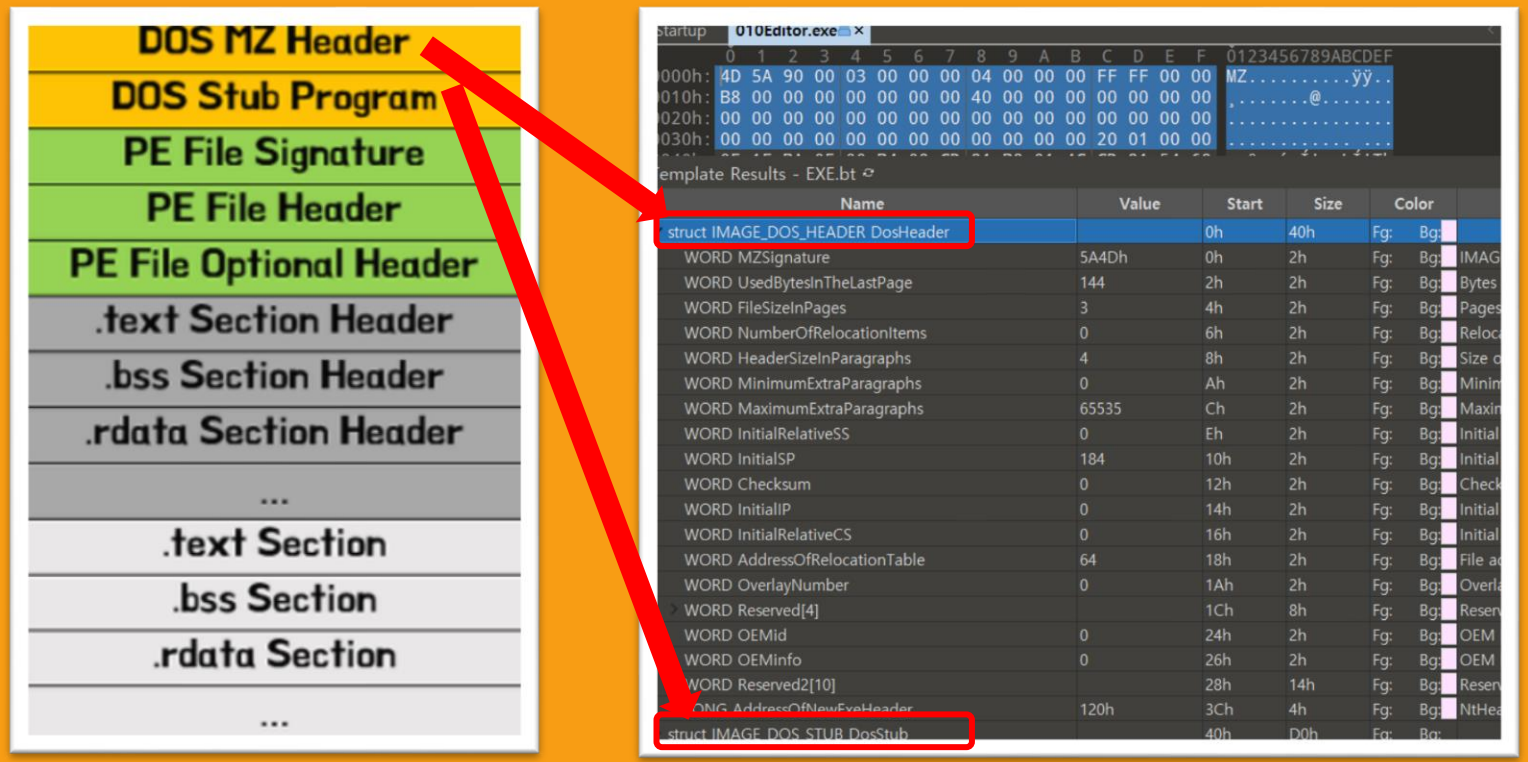
010 Editor? Hex editor

➔이진 파일안의 데이터를
16진수의 형태로 보여주고,
이를 편집할 수 있는 프로그램



010 Editor의 기능 中 템플릿 기능?

➔ 파일 데이터를 파일 구조대로 정리하여 나열해주는 기능



Prefetch file?

➔ Windows 운영체제에서 부팅/응용 프로세스가 사용하는 시스템 자원을 미리 저장하여 실행파일 로딩 시 효율을 높이는 파일

Prefetch file을 통해 알 수 있는 정보?

➔ 응용프로그램 이름, 실행 횟수, 마지막 실행 시각, 참조 파일/디렉토리 …

➔ 부팅 과정에서 로드되는 악성코드

➔ 악성코드가 활용하는 리소스 목록

1-2. 목표 및 기대 효과

목표	기대 효과
공식 템플릿 등록	프리패치 파일 분석 소요시간 단축

2. 수행 과정

1. 참고 자료
2. 코드 구현

2-1. 참고 자료

https://www.sweetscape.com/010editor/repository/templates/

010 Editor - Binary Template Repository

This page contains a repository of Binary Templates for use with 010 Editor. For more information on templates see the Binary Templates page and for information on how to install these files see the Installing page.

Sort by: Category | Alphabetic | Newest

Template File	Description	More Info
Archive		
7ZIP.bt	Parse 7-Zip archive files.	More Info...
AR.bt	Parse ar archives used for .a, .lib, .ar and .deb files.	More Info...
CAB.bt	Template for Microsoft cabinet format files.	More Info...
GZip.bt	Quick template for parsing GZip data/files.	More Info...
LZ4.bt	Template for LZ4 Framing Format.	More Info...
Nus3Audio.bt	Parse nus3 music archives.	More Info...
OrochiDAT.bt	Analyzing Silicon Studio's Orochi / Mizuchi Engine data archiving format	More Info...
PAK.bt	Parsing chrome pak files.	More Info...

```
//------------------------------------------------
//--- 010 Editor v9.0 Binary Template
//
//      File: 7ZIP.bt
//    Author: Richard Perrott
//   Version: 0.1
//   Purpose: Parse 7-Zip archive files.
//  Category: Archive
// File Mask: *.7z
//  ID Bytes: 37 7A BC AF 27 1C
//   History:
//   0.1   2019-01-28 Richard Perrott: Initial release
//
// More information available at:
// https://www.7-zip.org/7z.html
// https://infogalactic.com/info/7z
// https://sourceforge.net/projects/sevenzip/files/7-Zip/
//    in the org.apache.commons.compress.archivers.sevenz.SevenZFile call and associated classes.
//------------------------------------------------

//
// This is a tricky file format to built a template for, because of the confusing official docs
// and the structure optimisations for compression; source code looks better for clarifying how
// to actually comprehend the structures and their use.
//
// The lack of support for local structs, local struct arrays, and local references in the template language is really annoying;
// I had to needlessly use multiple, transient variables and presized arrays, including arrays of FTell values!
//

LittleEndian();

local uint64 offsetPos;

// Variables for CRC
local uint64 nextHeaderPos;
local uint64 nextHeaderSize;
local uint32 nextHeaderCRC;
local ubyte HeadersScanned = 0;

void initCRC(uint64 _nextHeaderPos, uint64 _nextHeaderSize, uint32 _nextHeaderCRC) {
    nextHeaderPos = _nextHeaderPos;
    nextHeaderSize = _nextHeaderSize;
    nextHeaderCRC = _nextHeaderCRC;
}
```

2-1. 참고 자료

libscca/Windows Prefetch File (PF) format.asciidoc at main · libyal/libscca · GitHub

Windows Prefetch File (PF) format

Summary

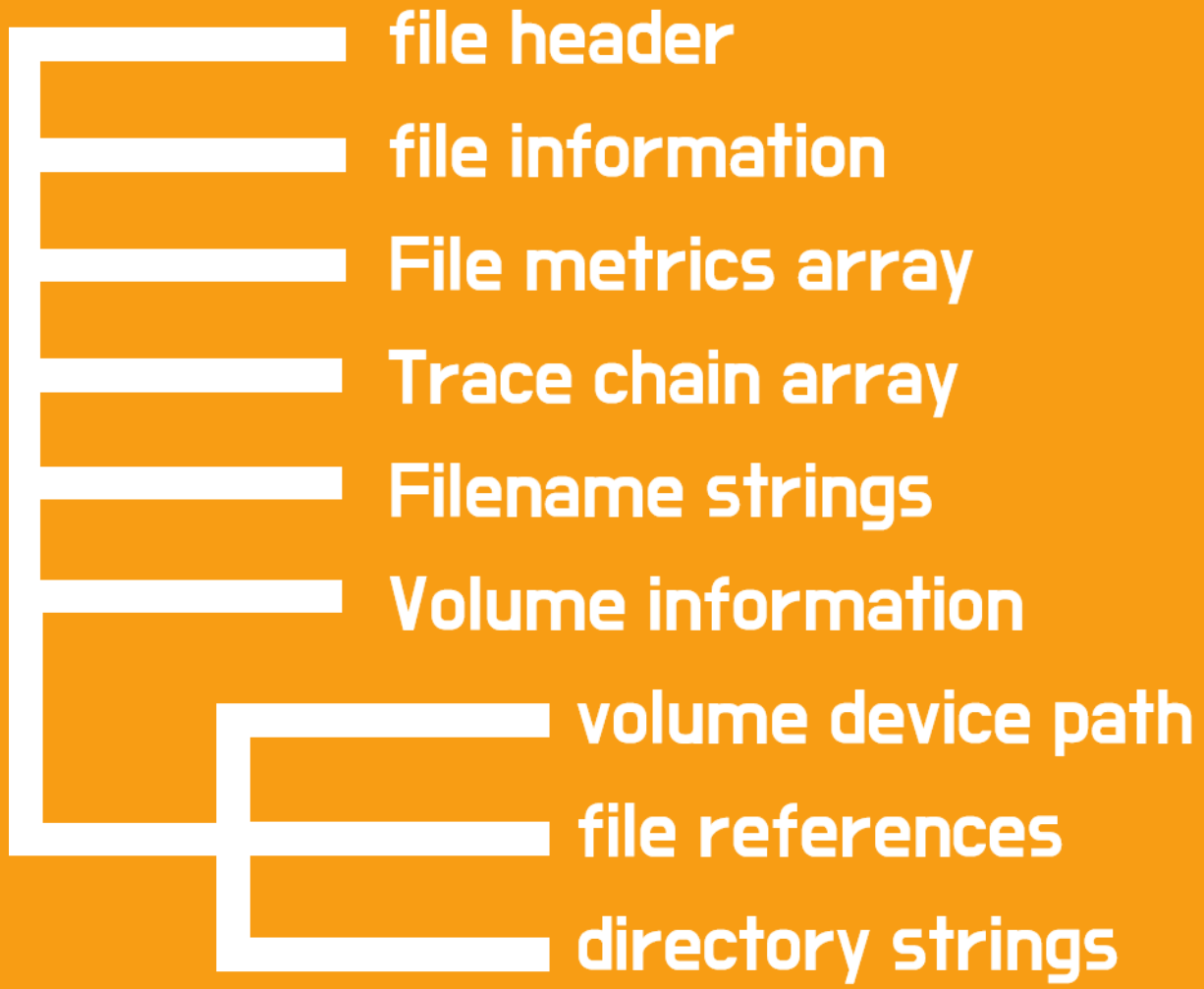
A Windows Prefetch File (PF) is used by Microsoft Windows to improve Windows and application start-up performance by loading application data into memory before it is demanded. This specification is based on earlier work on the format and was enhanced by analyzing test data.

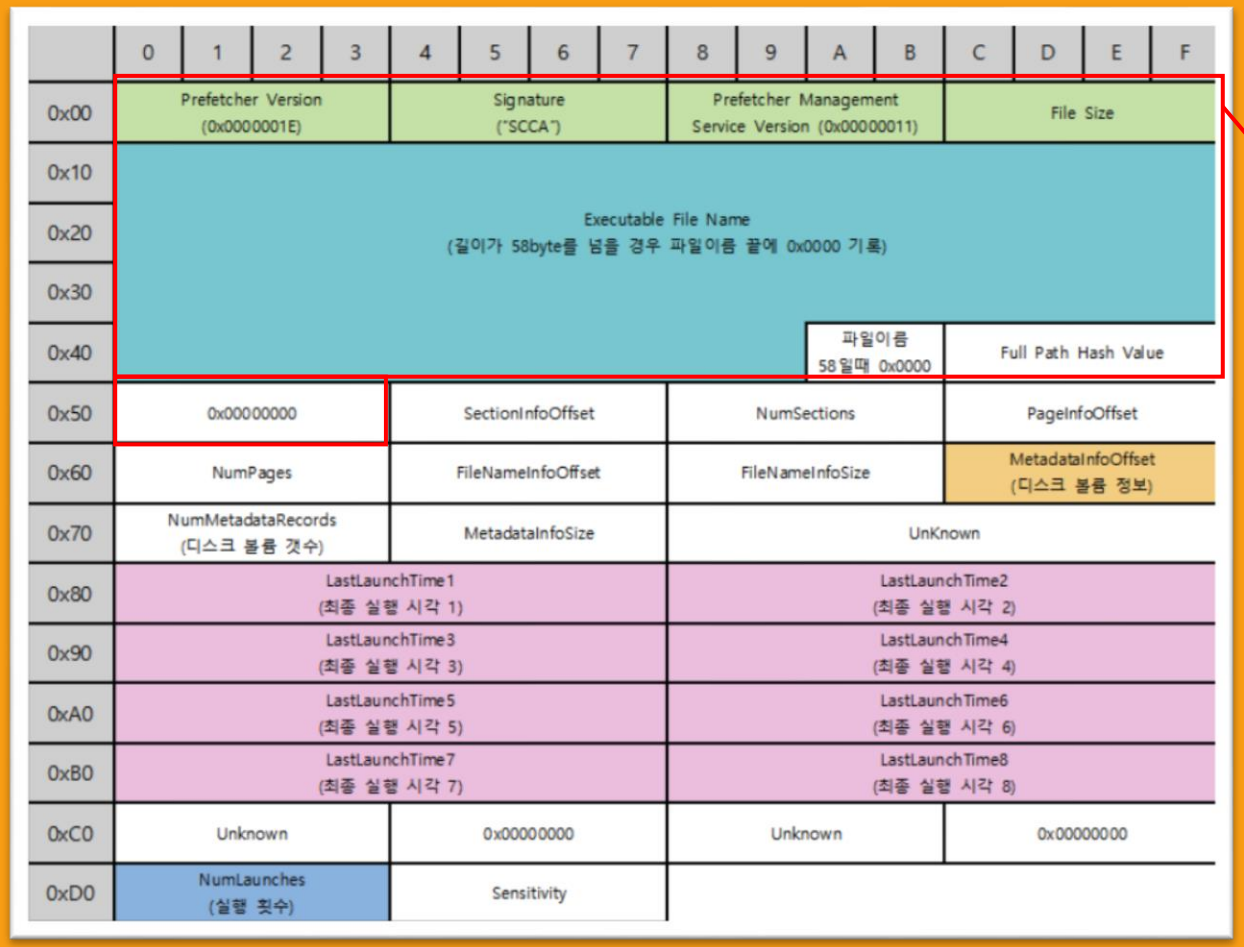
This document is intended as a working document of the data format specification for the libscca project.

Document information

Author(s):	Joachim Metz <joachim.metz@gmail.com>
Abstract:	This document contains information about the Windows Prefetch File (PF) format
Classification:	Public
Keywords:	Prefetch File, PF, SCCA

1. 파일 구조 파악


file header
file information
File metrics array
Trace chain array
Filename strings
Volume information
volume device path
file references
directory strings



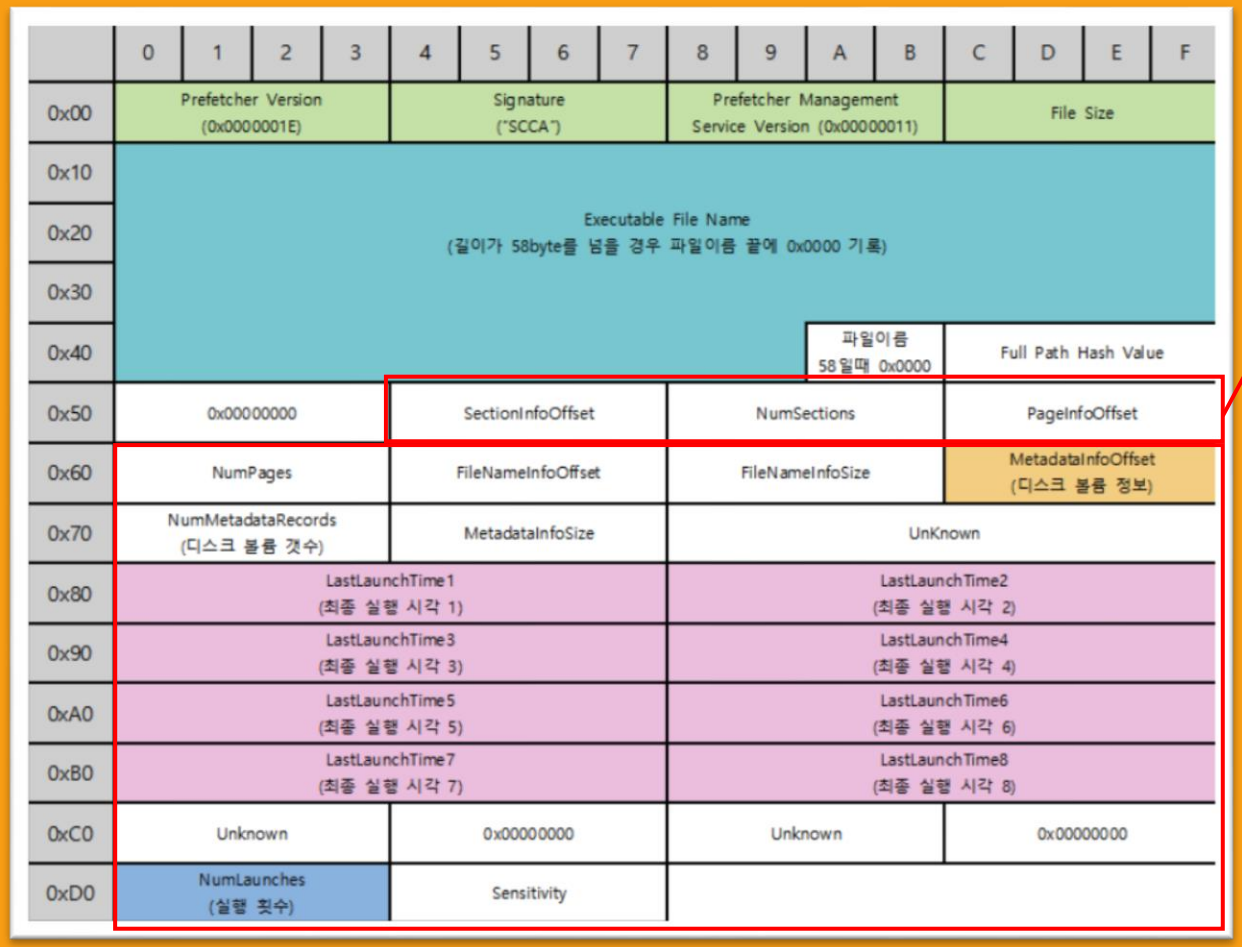
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Prefetcher Version (0x0000001E)				Signature ("SCCA")				Prefetcher Management Service Version (0x00000011)				File Size			
0x10	Executable File Name (길이가 58byte를 넘을 경우 파일이름 끝에 0x0000 기록)															
0x20																
0x30																
0x40													파일이름 58일때 0x0000		Full Path Hash Value	
0x50	0x00000000				SectionInfoOffset				NumSections				PageInfoOffset			
0x60	NumPages				FileNameInfoOffset				FileNameInfoSize				MetadataInfoOffset (디스크 볼륨 정보)			
0x70	NumMetadataRecords (디스크 볼륨 갯수)				MetadataInfoSize				UnKnown							
0x80	LastLaunchTime1 (최종 실행 시각 1)								LastLaunchTime2 (최종 실행 시각 2)							
0x90	LastLaunchTime3 (최종 실행 시각 3)								LastLaunchTime4 (최종 실행 시각 4)							
0xA0	LastLaunchTime5 (최종 실행 시각 5)								LastLaunchTime6 (최종 실행 시각 6)							
0xB0	LastLaunchTime7 (최종 실행 시각 7)								LastLaunchTime8 (최종 실행 시각 8)							
0xC0	Unknown				0x00000000				Unknown				0x00000000			
0xD0	NumLaunches (실행 횟수)				Sensitivity											

2. 파일 구조를 구조체로 구현

File Header

```
25  typedef struct {
26      Version version;
27      char Signature[4]; //SCCA
28      byte Unknown[4];
29      int FileSize;
30      wchar_t FileName[30];
31      int Hash <format=hex>;
32      Flag flag;
33  } FileHeader <optimize=true>;
```

2-2. 코드 구현 - 구조체 구현



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Prefetcher Version (0x0000001E)				Signature ("SCCA")				Prefetcher Management Service Version (0x00000011)				File Size			
0x10	Executable File Name (길이가 58byte를 넘을 경우 파일이름 끝에 0x0000 기록)															
0x20																
0x30																
0x40											파일이름 58일때 0x0000		Full Path Hash Value			
0x50	0x00000000				SectionInfoOffset				NumSections				PageInfoOffset			
0x60	NumPages				FileNameInfoOffset				FileNameInfoSize				MetadataInfoOffset (디스크 볼륨 정보)			
0x70	NumMetadataRecords (디스크 볼륨 갯수)				MetadataInfoSize				UnKnown							
0x80	LastLaunchTime1 (최종 실행 시각 1)								LastLaunchTime2 (최종 실행 시각 2)							
0x90	LastLaunchTime3 (최종 실행 시각 3)								LastLaunchTime4 (최종 실행 시각 4)							
0xA0	LastLaunchTime5 (최종 실행 시각 5)								LastLaunchTime6 (최종 실행 시각 6)							
0xB0	LastLaunchTime7 (최종 실행 시각 7)								LastLaunchTime8 (최종 실행 시각 8)							
0xC0	Unknown				0x00000000				Unknown				0x00000000			
0xD0	NumLaunches (실행 횟수)				Sensitivity											

File Information

```
typedef struct {
    uint MetricsOffset <format=hex>;
    uint MetricsvolumeentryNum;
    uint ChainsOffset <format=hex>;
    uint TraceChainsvolumeentryNum;
    uint NameStringOffset <format=hex>;
    uint NameStringSize;
    uint VolumesInformationOffset <format=hex>;
    uint VolumesNum;
    uint VolumesInformarionSize;
    uint64 Unknown1;
    FILETIME LastRunTime[8];
    uint64 Unknown2;
    uint RunCount;
    uint Unknown3;
    uint Unknown4;
    uint64 Unknown5[11];
} Win10_FileInfomation;
```

2-2. 코드 구현 - 구조체 구현

Offset	Size	Value	Description
0	4		Unknown (Prefetch start time in ms?) Could be the index into the trace chain array as well, is this relationship implic
4	4		Unknown (Prefetch duration in ms?) Could be the number of entries in the trace chain as well, is this relationship i explicit?
8	4		Unknown (Average prefetch duration in ms?)
12	4		Filename string offset The offset is relative to the start of the filename strings
16	4		Filename string number of characters Does not include the end-of-string character
20	4		Unknown (Flags?) Seen: 0x00000001, 0x00000002, 0x00000003, 0x00000200, 0x00000202
24	8		File reference Contains a file reference of the file corresponding to the filename string or 0 if not set See section: The file reference

File metrics array

```
93  typedef struct {
94      uint PrefetchStartTime;
95      uint PrefetchDuration;
96      uint AveragePrefetchDuration;
97      uint FileNameStringOffset;
98      uint FileNameStringNumber;
99      uint Flags;
100     FileReference Data;
101 } Win10_Entry_1;
102
103 typedef struct {
104     local int num1;
105     for (num1 =0; num1 < FileInfo.MetricsvolumeentryNum; num1++)
106         Win10_Entry_1 array;
107 } Win10_FileMetricsArray;
108
```

Trace Chain Array

```
109   typedef struct {
110       uint TotalBlockLoadCount;
111       uint UnKnown;
112   } Win10_Entry_2;
113
114   typedef struct {
115       local int num2;
116       for (num2 =0; num2 < FileInfo.TraceChainsvolumeentryNum; num2++)
117           Win10_Entry_2 array;
118   } Win10 TraceChainArray;
```

File Name Strings

```
35   typedef struct {
36       wstring read;
37   } Entry_3 <read=ReadWstring>;
120  typedef struct {
121      while (FTell() - FileInfo.NameStringOffset < FileInfo.NameStringSize)
122          Entry_3 Array;
123  } Win10 FileNameStrings;
```

Volume Information

```
125  typedef struct {
126      uint UnKnown;
127      uint NumberOfFileReference;
128      uint64 Unknown;
129      local int num = VolumeInfo.VolumeEntry.FileReferencesDataSize - 16;
130      byte ArrayOfFileReference[num];
131  } Win10_FileReferences;
132
133  typedef struct {
134      local int64 now = FTell();
135      uint VolumeDevicePathOffset <format=hex>;
136      uint NumberVolumeDevicePath;
137      FILETIME VolumeCreateTime;
138      uint VolumeSerialNumber <format=hex>;
139      uint FileReferencesOffset <format=hex>;
140      uint FileReferencesDataSize;
141      uint DirectoryStringsOffset <format=hex>;
142      uint NumberDirectoryStrings;
143      uint Unknown1;
144      uint64 UnKnown2[3];
145      uint CopyNumberDirectoryStrings;
146      uint64 UnKnown3[3];
147      uint Unknown;
148  } Win10_VolumeInformationEntry;
149
150  typedef struct {
151      FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.VolumeDevicePathOffset);
152      wstring VolumeDevicePath;
153      FSeek(VolumeInfo.VolumeEntry.now+ VolumeInfo.VolumeEntry.FileReferencesOffset);
154      Win10_FileReferences FileReferences;
155      FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.DirectoryStringsOffset);
156      DirectoryStrings DirStrings;
157  } Win10_VolumeInformationData;
158
159  typedef struct {
160      Win10_VolumeInformationEntry VolumeEntry;
161      Win10_VolumeInformationData VolumeData;
```

3. FTell() 함수를 이용하여 위치 정보 획득

Offset을 이용하여 파일의 정보를 찾아가는 일이 흔함

➔따라서 현재 파일의 위치를 알고 있어야 함

➔이때, 현재의 파일 위치가 파일의 시작부터 얼마나 떨어져 있는지 알려주는 FTell() 함수 사용!

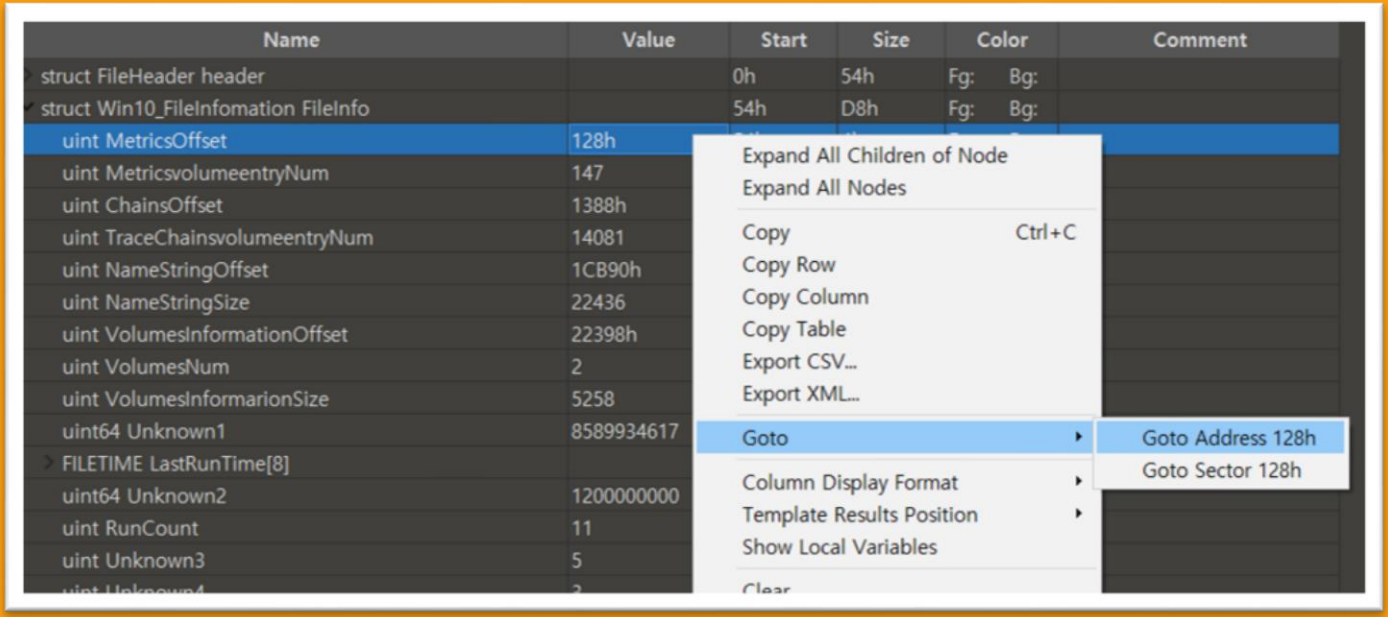
0x50	0x00000000	SectionInfoOffset	NumSections	PageInfoOffset
0x60	NumPages	FileNameInfoOffset	FileNameInfoSize	MetadataInfoOffset (디스크 볼륨 정보)
0x70	NumMetadataRecords (디스크 볼륨 갯수)	MetadataInfoSize	UnKnown	

4. FSeek() 함수를 이용하여 원하는 위치로 이동

Int fseek(FILE* stream, long offset, int start) 함수는

start부터 offset까지 스트림을 이동시키는 함수로,

FTell()로 안 현재 위치와 Offset 값을 이용하여 원하는 위치로 이동 가능!



5. If문을 이용하여 Version에 따른 파일 구조 구현

4.3.1. File metrics array entry - version 17

The file metrics array entry - version 17 is 20 bytes in size and consists of:

Offset	Size	Value	Description
0	4		Unknown (Prefetch start time in ms?) Could be the index into the trace chain array as well, is this relationship implicit or explicit?
4	4		Unknown (Prefetch du Could be the number
8	4		Filename string offset The offset is relative to
12	4		Filename string numbe Does not include the e
16	4		Unknown (Flags?)

4.3.2. File metrics array entry - version 23

The file metrics array entry - version 23 is 32 bytes in

Offset	Size	Value	Description
0	4		Unknown (Prefetch start time in ms?) Could be the index into the trace chain array as well, is this relationship implicit or explicit?
4	4		Unknown (Prefetch duration in ms?) Could be the number of entries in the trace chain as well, is this relationship implicit or explicit?

4.1.1. Format version

Value	Description
17	Used in: Windows XP, Windows 2003
23	Used in: Windows Vista, Windows 7
26	Used in: Windows 8.1
30	Used in: Windows 10

```
448  if(header.version == 30)
449  {
450      Win10_FileInfomation FileInfo;
451      FSeek(FileInfo.MetricsOffset);
452      Win10_FileMetricsArray FileMetrics;
453      FSeek(FileInfo.ChainsOffset);
454      Win10_TraceChainArray TraceChain;
455      FSeek(FileInfo.NameStringOffset);
456      Win10_FileNameStrings FileName;
457      FSeek(FileInfo.VolumesInformationOffset);
458      Win10_VolumeInformation VolumeInfo;
459      Trailing Data;
460  }
461  else if(header.version == 26)
462  {
463      Win8_FileInfomation FileInfo;
464      FSeek(FileInfo.MetricsOffset);
465      Win8_FileMetricsArray FileMetrics;
466      FSeek(FileInfo.ChainsOffset);
467      Win8_TraceChainArray TraceChain;
468      FSeek(FileInfo.NameStringOffset);
469      Win8_FileNameStrings FileName;
470      FSeek(FileInfo.VolumesInformationOffset);
471      Win8_VolumeInformation VolumeInfo;
472      Trailing Data;
```

3. 산출물

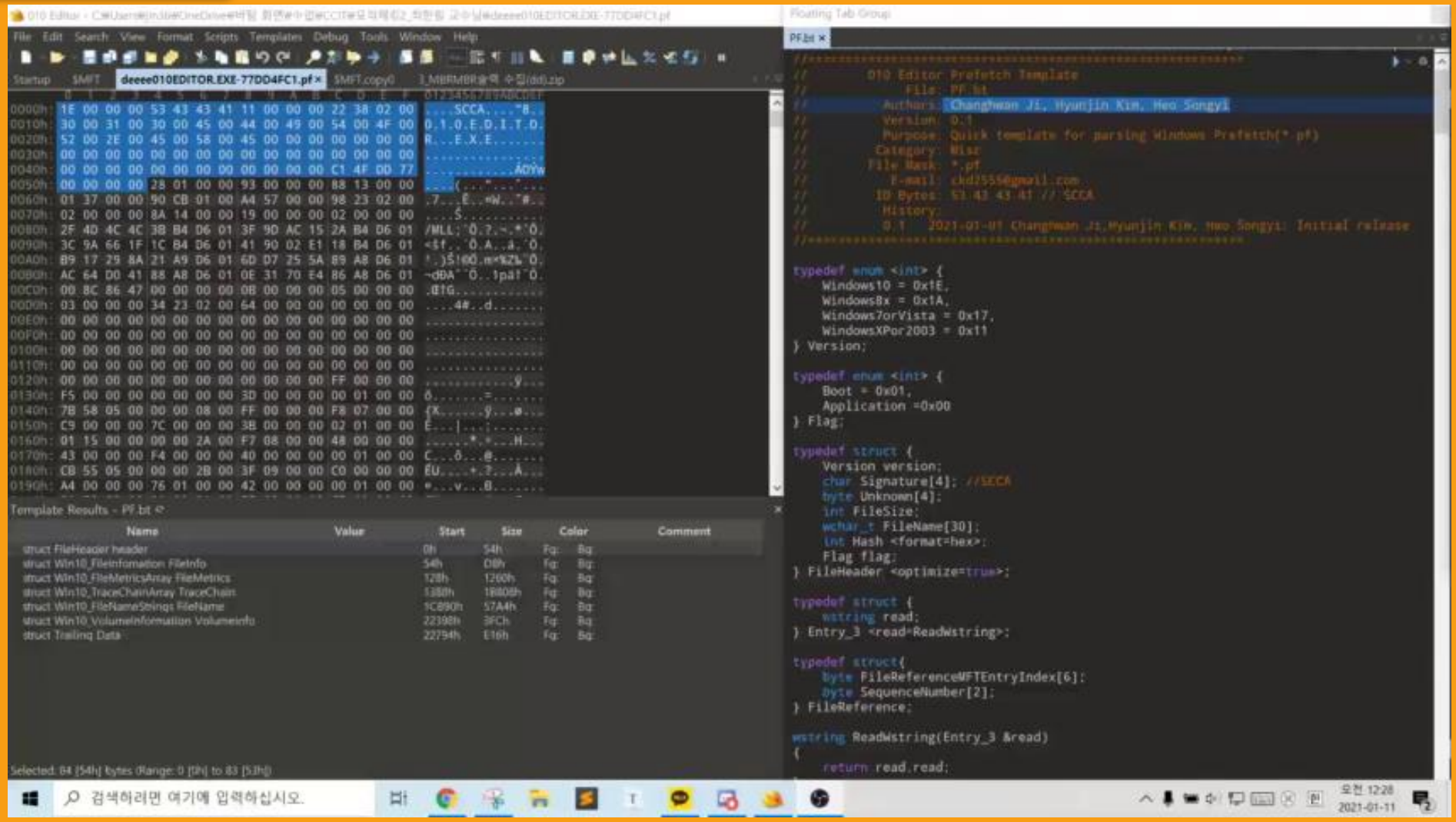
1. PF.bt
2. 공식 템플릿 등록
3. 향후 계획

3-1. 산출물 - PF.bt

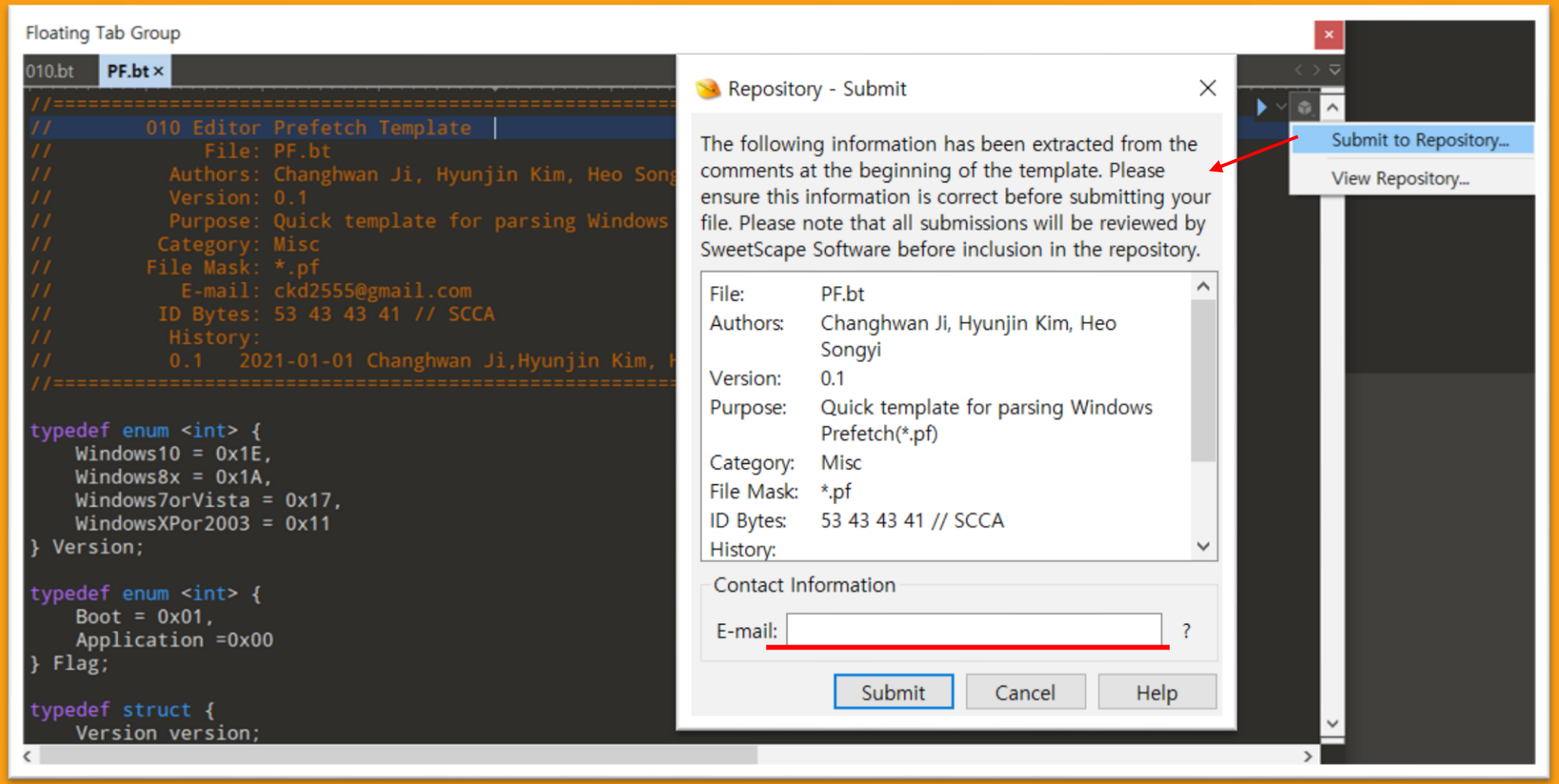
```
//------------------------------------------------
//        010 Editor Prefetch Template
//             File : PF.bt
//          Authors :
//          Version : 0.1
//          Purpose : Quick template for parsing Prefetch
//        File Mask : *.pf
//         ID Bytes : 53 43 43 41 // SCCA
//          History :
//          0.1   2021-01-01 Initial release
//------------------------------------------------

typedef enum <int> {
    Windows10 = 0x1E,
    Windows8x = 0x1A,
    Windows7orVista = 0x17,
    WindowsXPor2003 = 0x11
} Version;

typedef enum <int> {
    Boot = 0x01,
    Application =0x00
} Flag;

typedef struct {
    Version version;
    char Signature[4]; //SCCA
    byte Unknown[4];
    int FileSize;
    wchar_t FileName[30];
    int Hash <format=hex>;
    Flag flag;
} FileHeader <optimize=true>;

typedef struct {
    wstring read;
} Entry_3 <read=ReadWstring>;

typedef struct{
    byte FileReferenceMFTEntryIndex[6];
    byte SequenceNumber[2];
} FileReference;

wstring ReadWstring(Entry_3 &read)
{
    return read.read;
}

typedef struct {
    uint16 StringNumberOfCharacters <format=hex>;
    wstring DirectoryString;
} DirectoryStrings;

typedef struct {
    uint16 StringNumberOfCharacters <format=hex>;
    wstring DirectoryString;
} TrailingData;

typedef struct {
    local int num;
    for (num = 0; num < VolumeInfo.VolumeEntry.NumberDirectoryStrings; num++)
        TrailingData Data;
} Trailing;

typedef struct {

} Error;

//------------------------------------------------
//                  Windows 10
//              header version 30
//------------------------------------------------

typedef struct {
    uint MetricsOffset <format=hex>;
    uint MetricsvolumeentryNum;
    uint ChainsOffset <format=hex>;
    uint TraceChainsvolumeentryNum;
    uint NameStringOffset <format=hex>;
```

```
//------------------------------------------------
//                  Windows 8.1
//              header version 26
//------------------------------------------------

typedef struct {
    uint MetricsOffset <format=hex>;
    uint MetricsvolumeentryNum;
    uint ChainsOffset <format=hex>;
    uint TraceChainsvolumeentryNum;
    uint NameStringOffset <format=hex>;
    uint NameStringSize;
    uint VolumesInformationOffset <format=hex>;
    uint VolumesNum;
    uint VolumesInformarionSize;
    uint64 Unknown1;
    FILETIME LastRunTime[8];
    uint64 Unknown2[2];
    uint RunCount;
    uint Unknown3;
    uint Unknown4;
    uint64 Unknown5[11];
} Win8_FileInfomation;

typedef struct {
    uint PrefetchStartTime;
    uint PrefetchDuration;
    uint AveragePrefetchDuration;
    uint FileNameStringOffset <format=hex>;
    uint FileNameStringNumber;
    uint Flags;
    FileReference Data;
} Win8_Entry_1;

typedef struct {
    local int num1;
    for (num1 =0; num1 < FileInfo.MetricsvolumeentryNum; num1++)
        Win8_Entry_1 array;
```

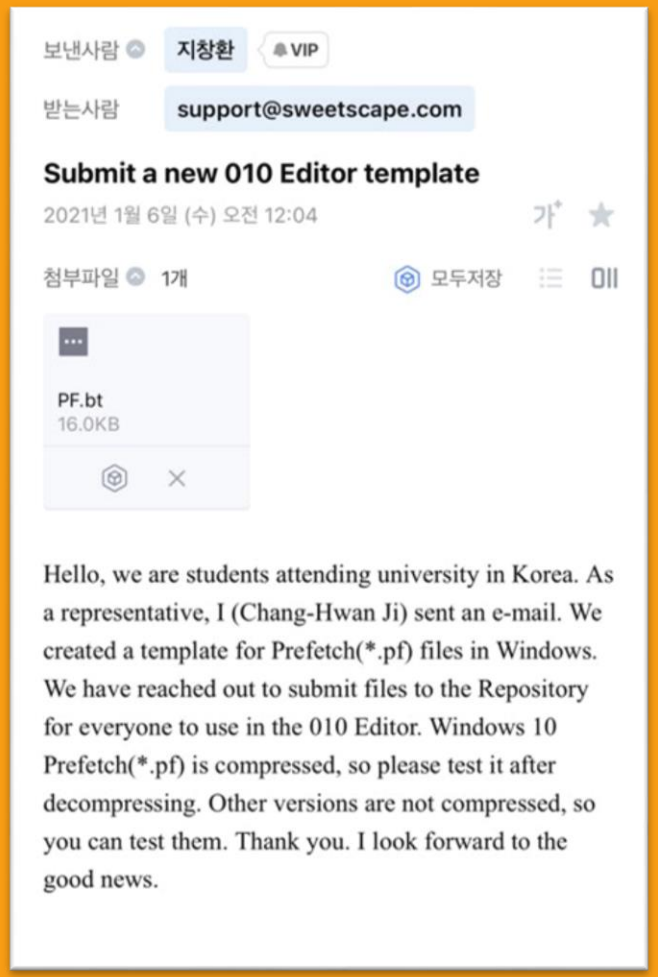
3-1. 산출물 - PF.bt


File Edit Search View Format Scripts Templates Debug Tools Window Help
deeee010EDITOR.EXE-77DD4FC1.pf
Template Results - PF.bt
Name Value Start Size Color Comment
struct FileHeader header 0h 54h
struct Win10_FileInfomation FileInfo 54h D0h
struct Win10_FileMetricsArray FileMetrics 128h 1260h
struct Win10_TraceChainArray TraceChain 1388h 1B808h
struct Win10_FileNameStrings FileName 1C890h 57A4h
struct Win10_VolumeInformation VolumeInfo 22398h 3FCh
struct Trailing Data 22794h E16h
Selected: 84 [54h] bytes (Range: 0 [0h] to 83 [53h])
PF.bt
// 010 Editor Prefetch Template
// Authors: Changwoon Ji, Hyunjin Kim, Heo Songyi
// Version: 0.1
// Purpose: Quick template for parsing Windows Prefetch(*.pf)
// Category: Misc
// File Mask: *.pf
// E-mail: ckd2556@gmail.com
// ID Bytes: 53 43 43 41 // SCCA
// History:
// 0.1 2021-01-07 Changwoon Ji,Hyunjin Kim, Heo Songyi: Initial release
typedef enum <int> {
Windows10 = 0x1E,
Windows8x = 0x1A,
Windows7orVista = 0x17,
WindowsXPor2003 = 0x11
} Version;
typedef enum <int> {
Boot = 0x01,
Application =0x00
} Flag;
typedef struct {
Version version;
char Signature[4]; //SCCA
byte Unknown[4];
int FileSize;
wchar_t FileName[30];
int Hash <format=hex>;
Flag flag;
} FileHeader <optimize=true>;
typedef struct {
wstring read;
} Entry_3 <read=ReadWstring>;
typedef struct{
byte FileReferenceMFTEntryIndex[6];
byte SequenceNumber[2];
} FileReference;
wstring ReadWstring(Entry_3 &read)
{
return read.read;
검색하려면 여기에 입력하십시오.
오전 12:28
2021-01-11

3-2. 산출물 - 공식 템플릿 등록



3-2. 산출물 - 공식 템플릿 등록



보낸사람 지창환 VIP

받는사람 support@sweetscape.com

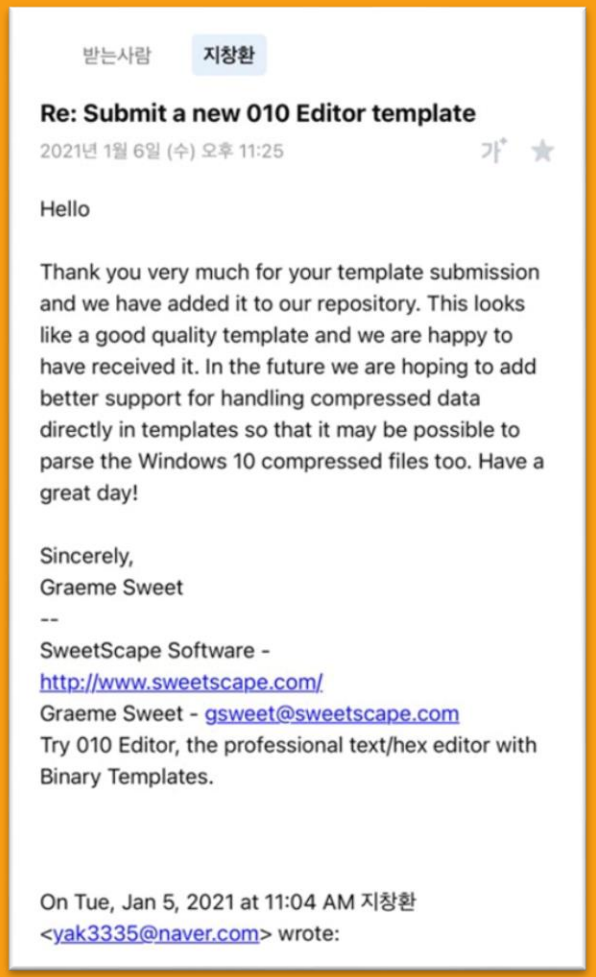
Submit a new 010 Editor template

2021년 1월 6일 (수) 오전 12:04

첨부파일 1개 모두저장

PF.bt
16.0KB

Hello, we are students attending university in Korea. As a representative, I (Chang-Hwan Ji) sent an e-mail. We created a template for Prefetch(*.pf) files in Windows. We have reached out to submit files to the Repository for everyone to use in the 010 Editor. Windows 10 Prefetch(*.pf) is compressed, so please test it after decompressing. Other versions are not compressed, so you can test them. Thank you. I look forward to the good news.



받는사람 지창환

Re: Submit a new 010 Editor template

2021년 1월 6일 (수) 오후 11:25

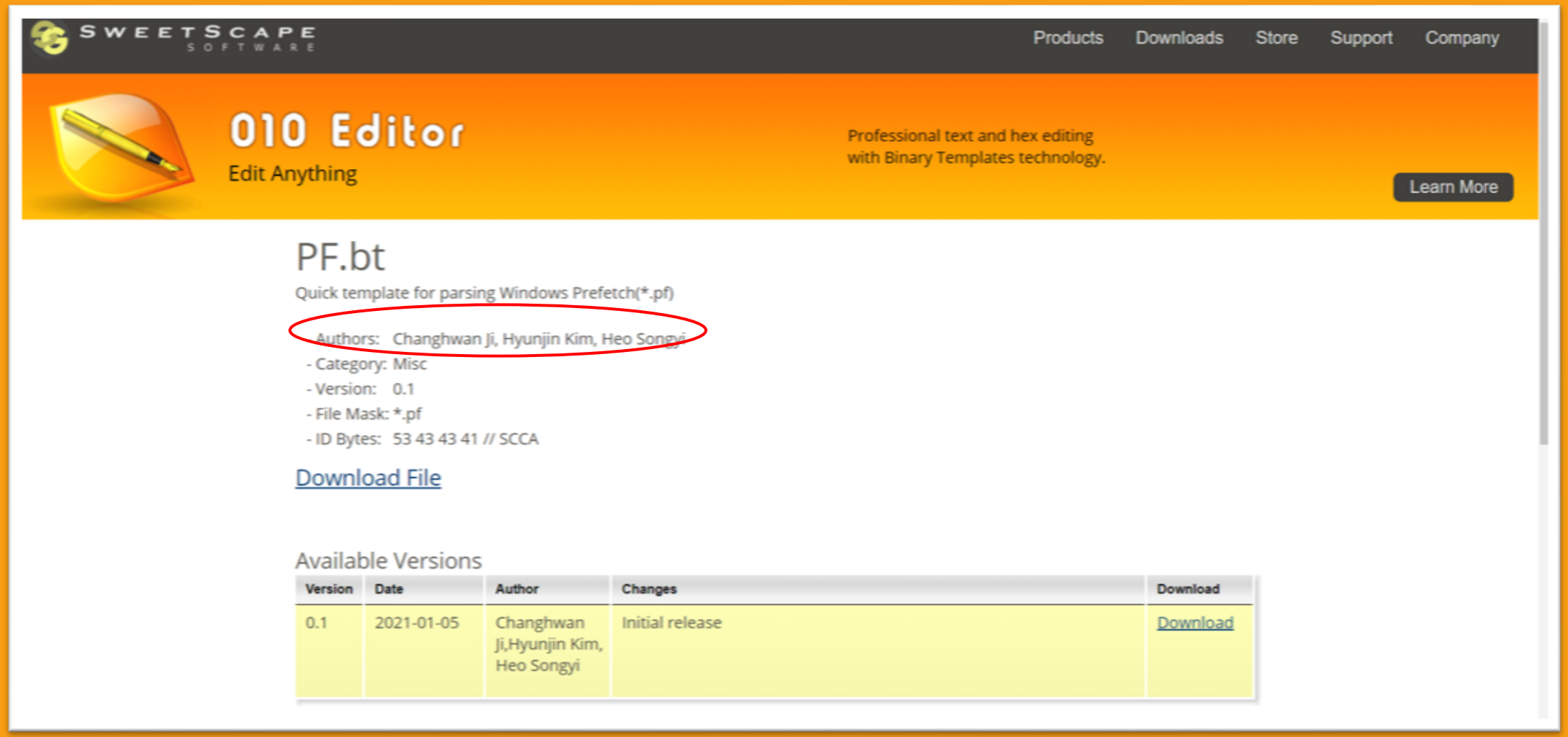
Hello

Thank you very much for your template submission and we have added it to our repository. This looks like a good quality template and we are happy to have received it. In the future we are hoping to add better support for handling compressed data directly in templates so that it may be possible to parse the Windows 10 compressed files too. Have a great day!

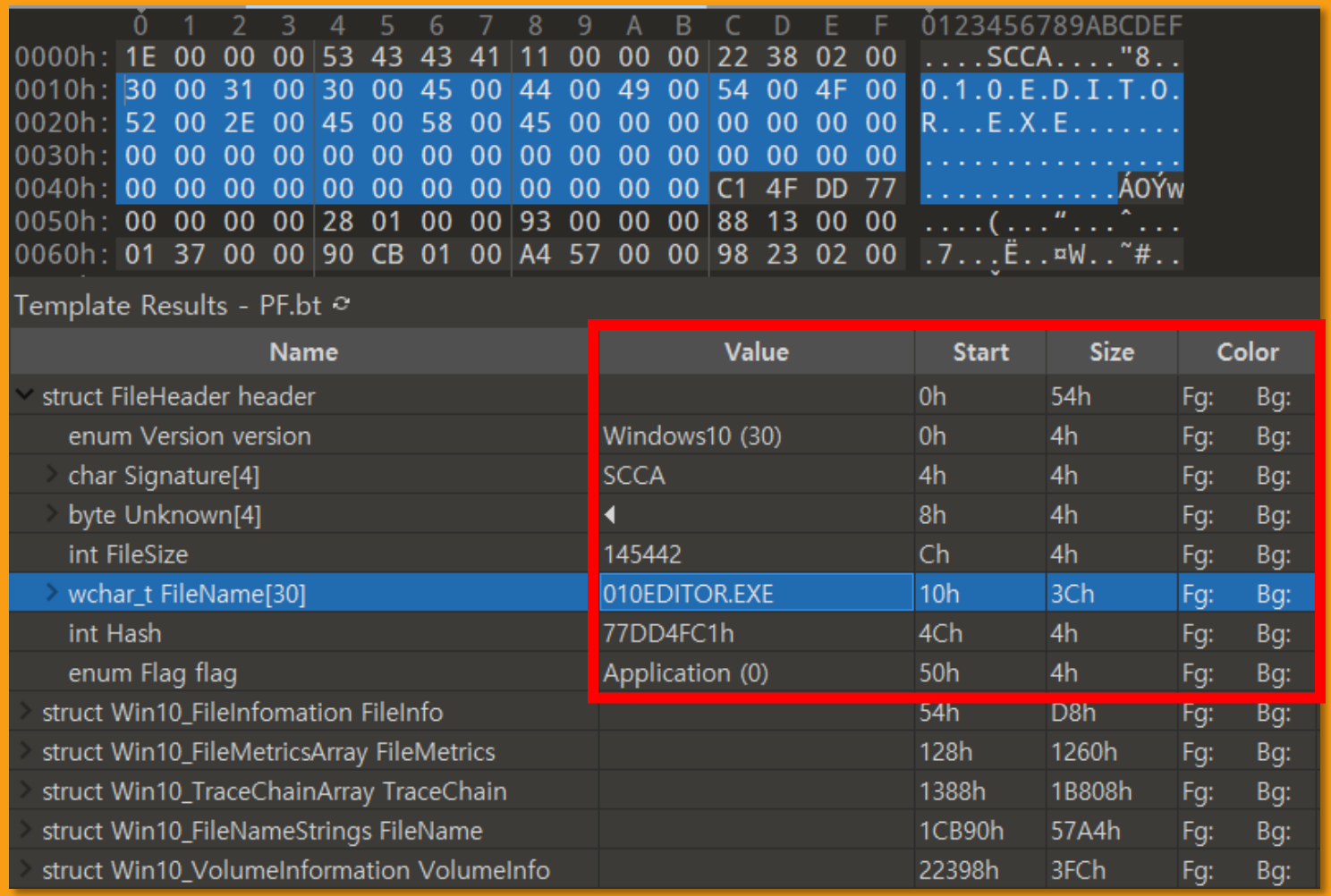
Sincerely,
Graeme Sweet
--
SweetScape Software -
http://www.sweetscape.com/
Graeme Sweet - gsweet@sweetscape.com
Try 010 Editor, the professional text/hex editor with Binary Templates.

On Tue, Jan 5, 2021 at 11:04 AM 지창환 <yak3335@naver.com> wrote:

3-2. 산출물 - 공식 템플릿 등록



3-2. 산출물 - 향후 계획



```
          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
0000h: 1E 00 00 00 53 43 43 41 11 00 00 00 22 38 02 00 ....SCCA....."8..
0010h: 30 00 31 00 30 00 45 00 44 00 49 00 54 00 4F 00 0.1.0.E.D.I.T.O.
0020h: 52 00 2E 00 45 00 58 00 45 00 00 00 00 00 00 00 R...E.X.E.......
0030h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0040h: 00 00 00 00 00 00 00 00 00 00 00 00 C1 4F DD 77 ............ÁOÝw
0050h: 00 00 00 00 28 01 00 00 93 00 00 00 88 13 00 00 ....(...“...ˆ...
0060h: 01 37 00 00 90 CB 01 00 A4 57 00 00 98 23 02 00 .7...Ë..¤W..˜#..
```

Template Results - PF.bt

Name	Value	Start	Size	Color
struct FileHeader header		0h	54h	Fg: Bg:
enum Version version	Windows10 (30)	0h	4h	Fg: Bg:
char Signature[4]	SCCA	4h	4h	Fg: Bg:
byte Unknown[4]	◀	8h	4h	Fg: Bg:
int FileSize	145442	Ch	4h	Fg: Bg:
wchar_t FileName[30]	010EDITOR.EXE	10h	3Ch	Fg: Bg:
int Hash	77DD4FC1h	4Ch	4h	Fg: Bg:
enum Flag flag	Application (0)	50h	4h	Fg: Bg:
struct Win10_FileInfomation FileInfo		54h	D8h	Fg: Bg:
struct Win10_FileMetricsArray FileMetrics		128h	1260h	Fg: Bg:
struct Win10_TraceChainArray TraceChain		1388h	1B808h	Fg: Bg:
struct Win10_FileNameStrings FileName		1CB90h	57A4h	Fg: Bg:
struct Win10_VolumeInformation VolumeInfo		22398h	3FCh	Fg: Bg:

감사합니다