Mobile Forensics
Lecture 6

Property Lists

Introduction

- Property List files (*.plist) are one of the widely used data storage formats used by Apple software .

- Most of the system properties are stored in plists (many of them are located in /Library/Preferences/), but many apps store their configuration in plist-files.

- property lists can be found in various places on Apple systems.

- Property lists offer a structured and efficient way to represent and persist hierarchies of objects to disk

Binary plist Structure

- Apple disclosed the structure of the binary property list format;

- it is documented in the comments of the Apple-provided open-source CFBinaryPList.c and declarations of the ForFoundationOnly.h.

- Every binary plist file comprises four sections: a header, an object table, an offset table and a trailer

- Each bplist file begins with an 6-byte header, containing the magic bplist (Hex: 0x62706C697374).

- The header is followed by a 2-bye version

- The most common version on Apple devices is 00, but there are at least two other versions of binary property lists, too; bplist15 or bplist16 occur.

Table 6.1: Structure of a bplist file.

Offset	Size	Description
0x00	6	bplist header (0x62706C697374)
0x06	2	format version
0x08	LEN1	object table
0x08 + LEN1	LEN2	offset table
0x08 + LEN1 + LEN2	32	trailer

- The bplist file ends with a 32-byte long trailer.

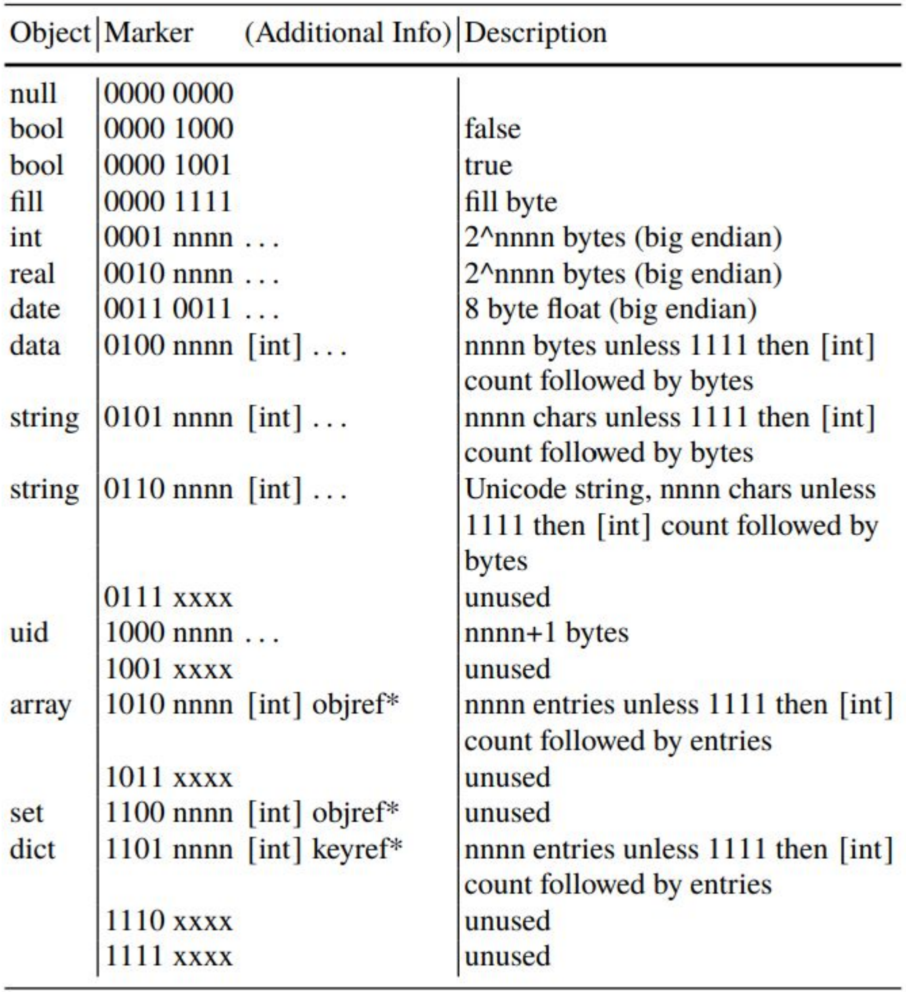
- The structure of the trailer is shown in table 6.2.

Table 6.2: Structure of the bplist trailer.

Offset	Length	Description
0x0	5	unused
0x5	1	sort version
0x6	1	size per offset in offset table in bytes
0x7	1	size per object reference in a container
0x8	8	number of objects in object table (big endian)
0x10	8	offset of the first offset in the offset table (big endian)
0x18	8	offset of the offset table (big endian)

- The bytes 0 to 4 of the trailer are unused.
- Byte 5 contains the sort version.
- Byte 6 stores the information of the size in byte of each offset entry in the offset table.
- byte 7 stores the information of the size of each object reference in a container. At offset 0x8, there is an 8-byte entry that saves the number of objects that are encoded inside the object table.
- The following 8 bytes save the offset of the first offset in the offset table (usually zero).
- The last 8 bytes of the trailer denotes the start of the offset table, counting from the start of the bplist.

Table 6.3: Format of object types.



Object	Marker (Additional Info)	Description
null	0000 0000	
bool	0000 1000	false
bool	0000 1001	true
fill	0000 1111	fill byte
int	0001 nnnn . . .	2^nnnn bytes (big endian)
real	0010 nnnn . . .	2^nnnn bytes (big endian)
date	0011 0011 . . .	8 byte float (big endian)
data	0100 nnnn [int] . . .	nnnn bytes unless 1111 then [int] count followed by bytes
string	0101 nnnn [int] . . .	nnnn chars unless 1111 then [int] count followed by bytes
string	0110 nnnn [int] . . .	Unicode string, nnnn chars unless 1111 then [int] count followed by bytes
	0111 xxxx	unused
uid	1000 nnnn . . .	nnnn+1 bytes
	1001 xxxx	unused
array	1010 nnnn [int] objref*	nnnn entries unless 1111 then [int] count followed by entries
	1011 xxxx	unused
set	1100 nnnn [int] objref*	unused
dict	1101 nnnn [int] keyref*	nnnn entries unless 1111 then [int] count followed by entries
	1110 xxxx	unused
	1111 xxxx	unused

- The marker is the binary representation of a single byte.

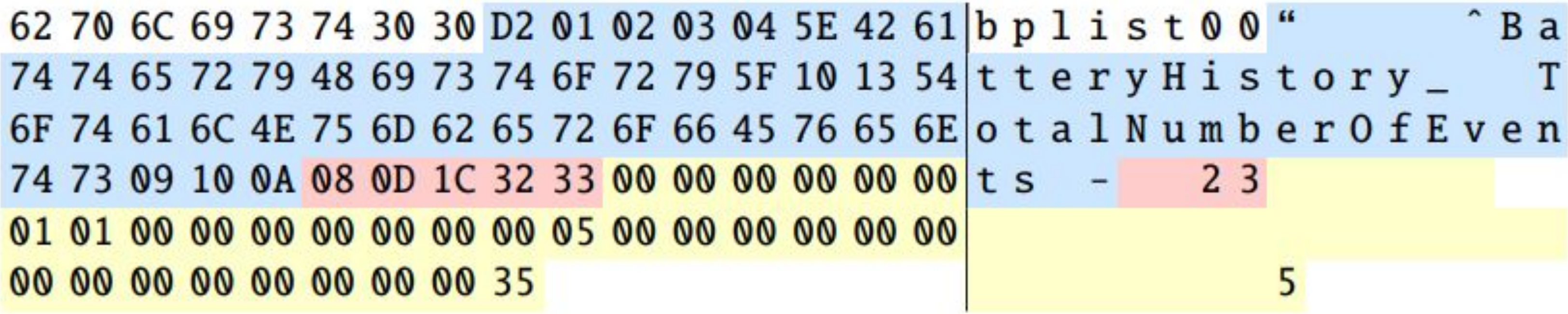
- All other objects can be uniquely identified by the marker byte’s 4 most significant bits (MSB).

- At the same time, the least significant bits (LSB) of the marker byte denotes sizing information.

Example

Given is the following plist (Table 6.4) from a MacBook Pro:

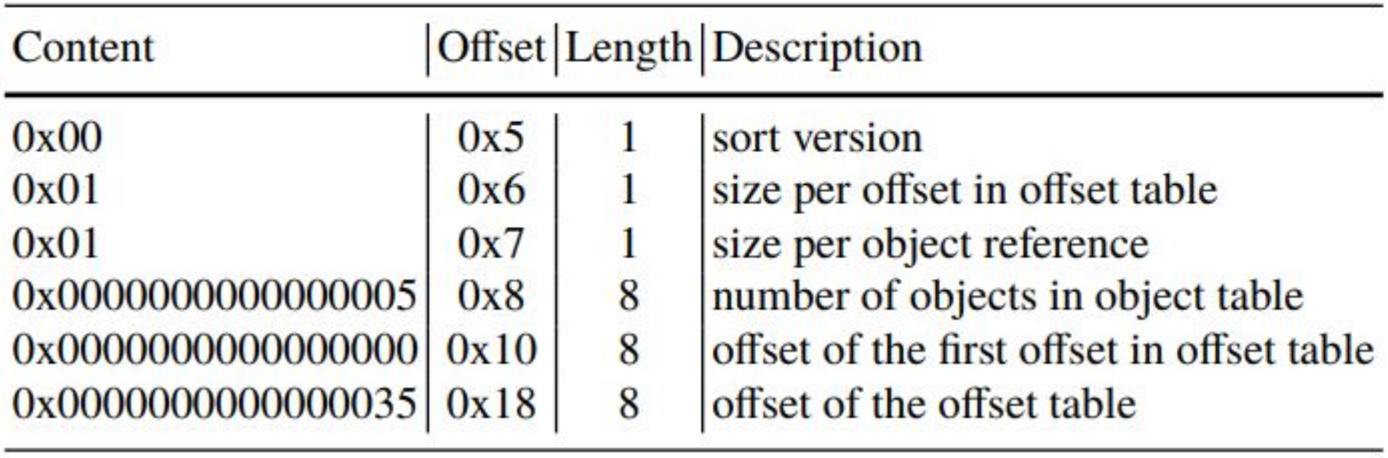
Table 6.4: Example plist (object table colored in blue, offset table colored in red, trailer colored in yellow).



```
62 70 6C 69 73 74 30 30 D2 01 02 03 04 5E 42 61|b p l i s t 0 0 “         ^ B a
74 74 65 72 79 48 69 73 74 6F 72 79 5F 10 13 54|t t e r y H i s t o r y _     T
6F 74 61 6C 4E 75 6D 62 65 72 4F 66 45 76 65 6E|o t a l N u m b e r O f E v e n
74 73 09 10 0A 08 0D 1C 32 33 00 00 00 00 00 00|t s   -     2 3
01 01 00 00 00 00 00 00 00 05 00 00 00 00 00 00|
00 00 00 00 00 00 00 00 35                     |        5
```

To analyze the bplist in a first step the trailer is marked (here yellow) the trailer comprises the last 32 bytes of the file. Now the trailer can be decoded (result in Table 6.5):

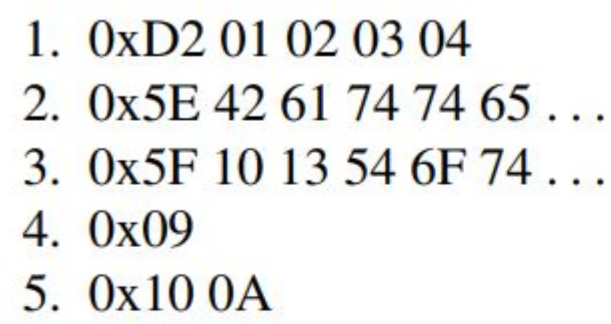
Table 6.5: Decoded example bplist trailer.



Content	Offset	Length	Description
0x00	0x5	1	sort version
0x01	0x6	1	size per offset in offset table
0x01	0x7	1	size per object reference
0x0000000000000005	0x8	8	number of objects in object table
0x0000000000000000	0x10	8	offset of the first offset in offset table
0x0000000000000035	0x18	8	offset of the offset table

- it is clear that the bplist contains five objects in the object table, and the offset table starts at 0x35, whereas the first object-offset starts at 0x35 + 0x00, and each offset has the size of one single byte.

the offsets from the offset table are 0x08, 0x0D, 0x1C, 0x32 and 0x33, which leads to the following four objects from the object table:



1. 0xD2 01 02 03 04
2. 0x5E 42 61 74 74 65 . . .
3. 0x5F 10 13 54 6F 74 . . .
4. 0x09
5. 0x10 0A

- The first object starts with 0xD2, which is 1101 0010 in binary. The MSB (1101) shows that the object-type is a dictionary.

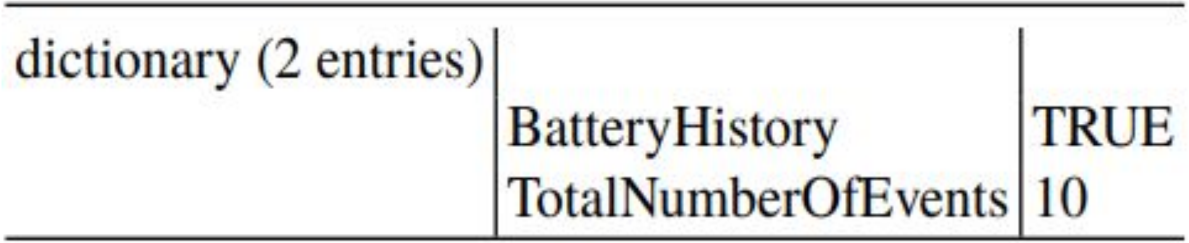
- The first object starts with 0xD2, which is 1101 0010 in binary. The MSB (1101) shows that the object-type is a dictionary.

- The third object starts with 0x5F, which is 0101 1111 in binary

- The fourth object starts with 0x09, which is 0000 1001 in binary.

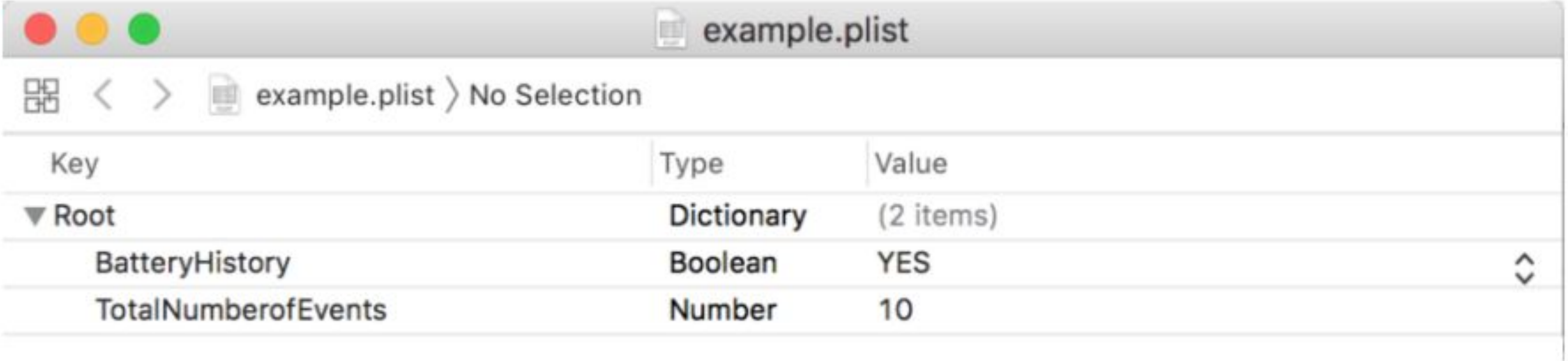
- The fifth object starts with 0x10, which is 0001 0000 in binary.

Table 6.6: Decoded example bplist.



dictionary (2 entries)		
	BatteryHistory	TRUE
	TotalNumberOfEvents	10

Fig. 6.1 shows the same plist file decoded with Apples XCode IDE and confirms the correct decoding.



Key	Type	Value
▼ Root	Dictionary	(2 items)
BatteryHistory	Boolean	YES
TotalNumberofEvents	Number	10

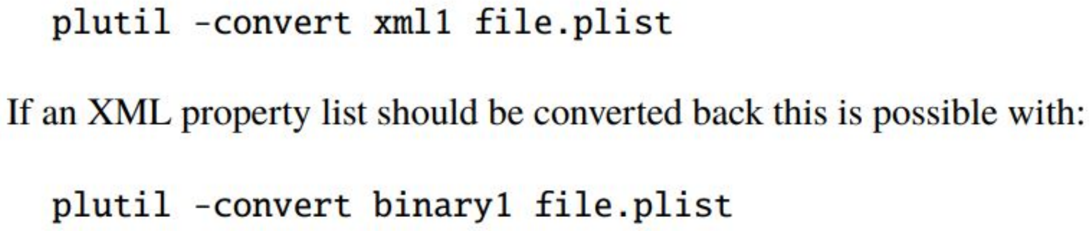
Fig. 6.1: Illustration of the elements of a block group.

Forensic Tools Supporting plists

- There is quite a bunch of tools supporting the decoding of plist files.

- Most of the tools support binary plists as well as XML property lists.

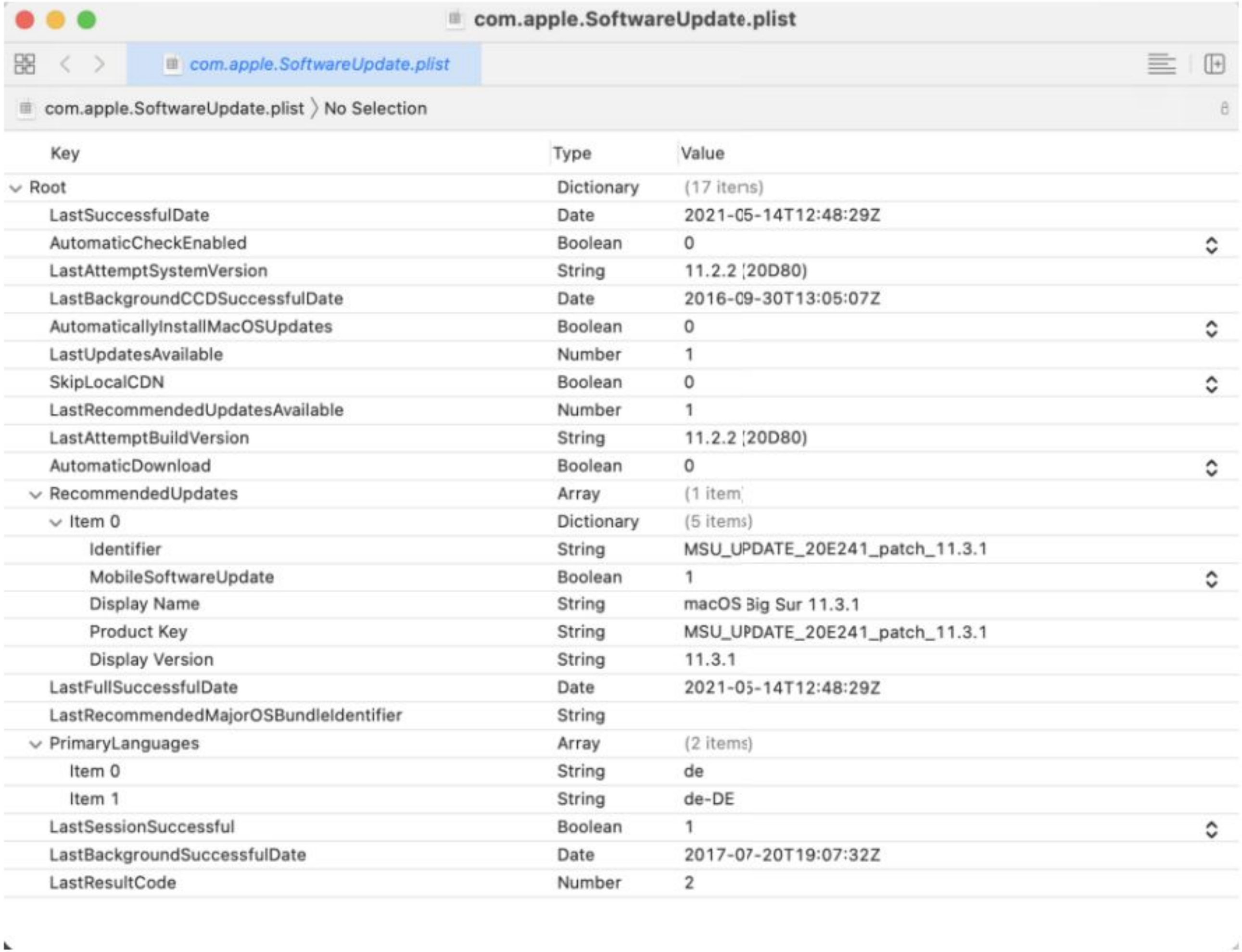
- if the given property list file is in the binary format, it can be converted to XML first by running on the macOS shell:



```
plutil -convert xml1 file.plist
```

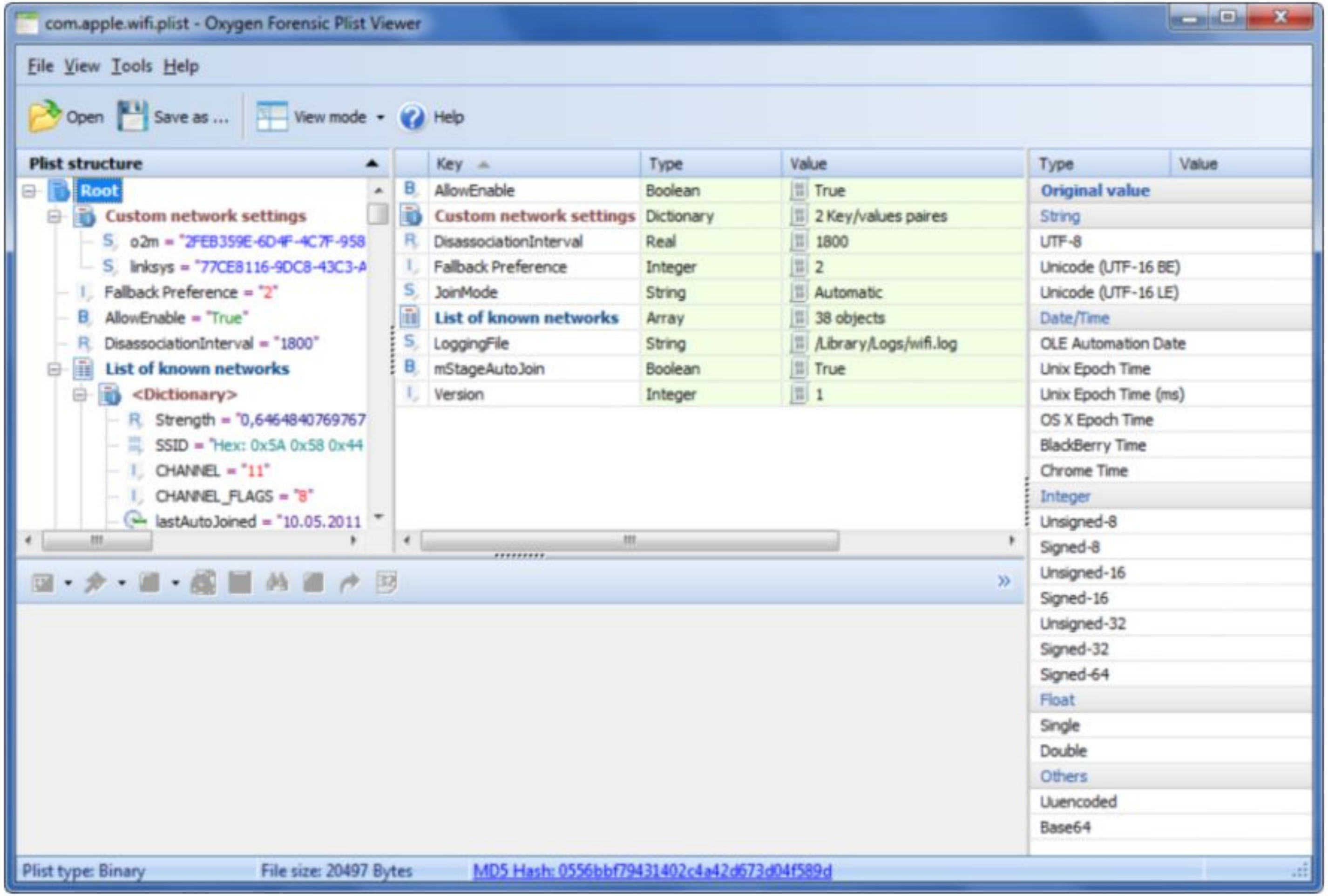
If an XML property list should be converted back this is possible with:

```
plutil -convert binary1 file.plist
```


com.apple.SoftwareUpdate.plist
com.apple.SoftwareUpdate.plist
com.apple.SoftwareUpdate.plist 〉 No Selection

Key	Type	Value
Root	Dictionary	(17 items)
LastSuccessfulDate	Date	2021-05-14T12:48:29Z
AutomaticCheckEnabled	Boolean	0
LastAttemptSystemVersion	String	11.2.2 (20D80)
LastBackgroundCCDSuccessfulDate	Date	2016-09-30T13:05:07Z
AutomaticallyInstallMacOSUpdates	Boolean	0
LastUpdatesAvailable	Number	1
SkipLocalCDN	Boolean	0
LastRecommendedUpdatesAvailable	Number	1
LastAttemptBuildVersion	String	11.2.2 (20D80)
AutomaticDownload	Boolean	0
RecommendedUpdates	Array	(1 item)
Item 0	Dictionary	(5 items)
Identifier	String	MSU_UPDATE_20E241_patch_11.3.1
MobileSoftwareUpdate	Boolean	1
Display Name	String	macOS Big Sur 11.3.1
Product Key	String	MSU_UPDATE_20E241_patch_11.3.1
Display Version	String	11.3.1
LastFullSuccessfulDate	Date	2021-05-14T12:48:29Z
LastRecommendedMajorOSBundleIdentifier	String	
PrimaryLanguages	Array	(2 items)
Item 0	String	de
Item 1	String	de-DE
LastSessionSuccessful	Boolean	1
LastBackgroundSuccessfulDate	Date	2017-07-20T19:07:32Z
LastResultCode	Number	2

Fig. 6.2: View of an example plist in the Xcode editor.



Root
Custom network settings
o2m = "2FEB359E-6D4F-4C7F-958
linksys = "77CE8116-9DC8-43C3-A
Fallback Preference = "2"
AllowEnable = "True"
DisassociationInterval = "1800"
List of known networks
<Dictionary>
Strength = "0,6464840769767
SSID = "Hex: 0x5A 0x58 0x44
CHANNEL = "11"
CHANNEL_FLAGS = "8"
lastAutoJoined = "10.05.2011

Key	Type	Value
AllowEnable	Boolean	True
Custom network settings	Dictionary	2 Key/values paires
DisassociationInterval	Real	1800
Fallback Preference	Integer	2
JoinMode	String	Automatic
List of known networks	Array	38 objects
LoggingFile	String	/Library/Logs/wifi.log
mStageAutoJoin	Boolean	True
Version	Integer	1

Type	Value
Original value	
String	
UTF-8	
Unicode (UTF-16 BE)	
Unicode (UTF-16 LE)	
Date/Time	
OLE Automation Date	
Unix Epoch Time	
Unix Epoch Time (ms)	
OS X Epoch Time	
BlackBerry Time	
Chrome Time	
Integer	
Unsigned-8	
Signed-8	
Unsigned-16	
Signed-16	
Unsigned-32	
Signed-32	
Signed-64	
Float	
Single	
Double	
Others	
Uuencoded	
Base64	

Plist type: Binary   File size: 20497 Bytes   MD5 Hash: 0556bbf79431402c4a42d673d04f589d

Fig. 6.3: View of an example plist in the Oxygen Forensic Plist Viewer.