THE NATIONAL ARCHIVES

# PRONOM Starter Pack

Your Guide to Starting Out with File Format Research

## Introduction

Welcome to PRONOM! We are thrilled that you are interested in file format research and hope that you too enjoy researching some of the fun file formats you find in the wild. You do not need any previous experience to contribute to PRONOM and most of our team also didn't have any before starting out. We want to hear from all our users in order to best serve the PRONOM community and are excited that you wish to join the conversation. You can read further about PRONOM team in our blog, which also explains a little bit about our work, and how file format research is conducted. This is a good starting point as it unpacks PRONOM, what it stands for, and how your work will contribute to wider digital preservation practices.

We have laid out this starter pack in three sections:

**Ready**- which will outline some of the software tools and introductory reading that are valuable for getting started with file format research.

**Set**- this is where we go through the process of file format research which includes analysing hex, the difference between binary and container signature research and how to write a signature.

**Go**- details of how to submit your research, get in touch with the community, check in on our current work and how to contact us if you have any questions.

You may be a first time researcher or may find this guide useful to remind you of a few details . While this can be read in order, we have aimed to make each section stand alone and included a contents table so you can also jump around as you wish

You will notice that there are a lot of links to additional resources in the document and the full list can be found at the end.

**Contents**

**Ready…**

You are probably here because you have some file formats in your collection that are not being identified or do not identify as you think they should. If this is the case, you already have an issue to research. However, if you do not and you still wish to contribute, you will find a list of file formats without signatures here, a list of file formats without descriptions here or you can search the PRONOM database to see if your favourite file formats are already there

## Submission Template

You will find a template for your file format research in this pack. The template is in Microsoft Excel and Word format, please choose whichever is easier for you. The template has a list of fields - this list is the information that we collect before the file format goes into PRONOM. If you cannot fill it all out that is not a problem at all but the more complete the template the quicker we can usually process the submission!

## Introductory Reading

There are a number of useful blogs written about file format research. To start we recommend this blog written by Ross Spencer, this guidance written by Jenny Mitcham and the blog written by PRONOM team mentioned earlier.

## Useful Software

You will need:

- Hex editor
- File format identification tool
- 7ZIP (for certain types of file formats only)

In the introductory reading we found that file format research involves analysing hex and looking for byte sequences within the file format. Internally we use HxD **hex editor**, which is a free software and can be downloaded here. Other software is available also.

You will also need a **file format identification tool**, at The National Archives we use DROID to analyse files. Please download a file format identification tool prior to starting your research, you can use DROID or other tools such as Siegfried or FIDO. For this exercise the tool should use the data from the PRONOM database.

You may also need 7Zip tool for researching container signatures and extracting information from them (more on this later).

**Set…**

In this section we will talk you through the following aspects of file format research:

- Finding information about your file format.
- How to find the best identification method for your file format (there are three types).
- Creating the signature.

To start analysing your file format there are few steps you should follow. These steps do not necessarily need to be completed in a set order and you can adjust how you conduct your research to make it most suitable to your individual working methods.

Steps that we need to take:

- Things to check before starting. Verifying the problem you need to solve and double checking your file formats are not already in the database.
- Finding file samples (maybe you have these already), the more the better from lots of different places!
- The type of file format identification in PRONOM.
- How to find a file format signature and analyse your file format using hex and other tools.
- Checking your signature is successful!
- Finding additional information about the file format.

## Check Before Starting

We recommend as the first step running your files through a file format identification tool. To do this, you should have a file format identification tool that uses the data from the PRONOM database, such as DROID, Siegfried and FIDO. Secondly, we recommend searching the PRONOM database for your file format using the extension-only search and the free text search.

If your file is neither in the PRONOM database nor identifying as it should be using the software then it will need a new entry. Follow the steps ahead as proposed. There is an exception to this rule. If your file identifies unexpectedly as a ZIP or OLE2 file then there is a possibility it is a container file. See the next section **Types of Identification in PRONOM**.

If your file can be found in the PRONOM database but is not identifying using the software then there is a good chance that the signature needs improving or there is a missing file extension. Follow the steps ahead and research your file format, however, try and work out what the differences are between what is in the PRONOM entry and in your files. See if you can investigate what has gone wrong. To help take a look at the method by which it is identifying (Signature, Container or Extension).

If your file identifies correctly but also identifies as other file types then it has the characteristics of two or more signatures in the database. Take a look at how it is identifying. If it is by Extension then there is very little we can do (though do tell us about it and jump ahead to the **GO!** section if you think it is important). However if it is by Signature or Container see if you can investigate why your file may identify as both types by following the steps ahead. Is there a way that one signature can be improved so that this doesn't happen? Should one signature be given a priority over the other (so that if it identifies as both only one appears)?

If your file identifies correctly in the identification software but has an extension mismatch error and is therefore hard to search in the database jump ahead to the **GO!** section. Tell us about the missing extension.

## Finding File Samples

You should aim to have a substantial number of samples, the more the better (five to twenty examples is a good range)! The file samples should preferably be from more than one source. i.e. if all your file samples have come from the same computer then consistencies in the file format may be due to the computer you're using and not the file format itself.  A warning that downloading anything onto your machine comes with its own risks. Having cyber security protection installed and a good anti-virus software could mitigate any potential malware risks.

To find file samples online you can search online using resources such as:

### GitHub

You can search the GitHub repository for more samples, to do this you will require an account. The most effective search technique for GitHub is to search all repositories. If you are trying to find files with the extension .cbz you would search the following *Filename:cbz extension:cbz*. Use these steps to download the sample files:

1. Go to file that you want to download
2. Click it to view the contents within GitHub
3. In the top right, right click the raw button

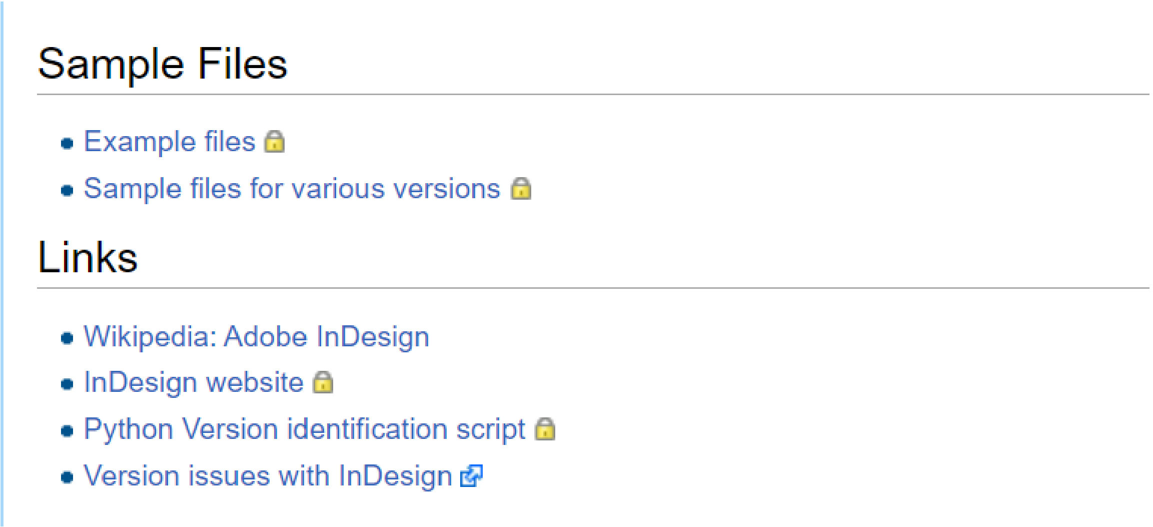
### Search Engines

You can use your search engine to look for samples. Depending on the search engine you use, the ways to find sample will vary. You can try by typing the name of file format name and its extension and samples at the end. An effective way to look for samples using Google is by typing any of the below syntax:

- "parent directory" .cbz = good google search
- Filetype:cbz  -facebook (or +facebook to tell it that it must include facebook)
- filetype:jpg 'filetype:<fileextension>'

### Just Solve the File Format Problem

Another method to find sample files is using the Just Solve the File Format Problem wiki entry. Your format may have already been researched by someone else and more information about it can be found on this site. Depending on the format you are researching you can find webpages which might have sample files for your format, as shown below:



## Sample Files

- Example files
- Sample files for various versions

## Links

- Wikipedia: Adobe InDesign
- InDesign website
- Python Version identification script
- Version issues with InDesign

### Be Aware

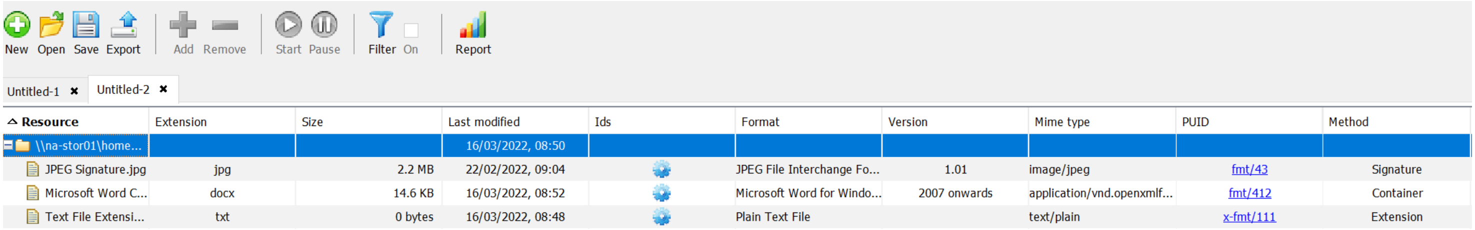
The internet is the wild west of extensions and there are no rules. Something that can be seen often on GitHub but something to be aware of on other sites too. Developers can assign a random extension to their code that may be the same as the one you are looking for.

Unfortunately your file format may not be the only file format with this extension. You can also use the file-extension seeker or TrID database to check if there are any other formats associated with the same extension. This can help avoid samples of different files with same extension as you are looking for.

With all these take a look at the file you have downloaded. If it has a similar structure to other file samples you already have or matches elements of the specification then it is likely to be what you are looking for. If it is completely different it could be another file format entirely.

## Types of Identification in PRONOM

There are currently three types of file format identification. Extension-only, Binary Signature and Container Signature. The majority of file types are identified by binary signature or extension-only. It is comparatively rare to find a container signature.

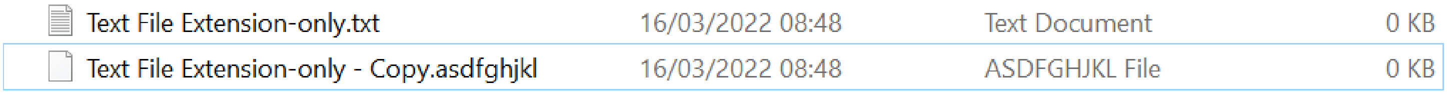


| Resource | Extension | Size | Last modified | Ids | Format | Version | Mime type | PUID | Method |
|---|---|---|---|---|---|---|---|---|---|
| \\na-stor01\home... | | | 16/03/2022, 08:50 | | | | | | |
| JPEG Signature.jpg | jpg | 2.2 MB | 22/02/2022, 09:04 | | JPEG File Interchange Fo... | 1.01 | image/jpeg | fmt/43 | Signature |
| Microsoft Word C... | docx | 14.6 KB | 16/03/2022, 08:52 | | Microsoft Word for Windo... | 2007 onwards | application/vnd.openxmlf... | fmt/412 | Container |
| Text File Extensi... | txt | 0 bytes | 16/03/2022, 08:48 | | Plain Text File | | text/plain | x-fmt/111 | Extension |

In DROID the method of identification appears under the Method column. Here the 'Signature' category applies to binary signatures.

### Extension-only

This type of identification is the least secure method of identifying a file format. If possible we try to apply a binary or container signature to the file format as file format extensions can be easily changed. It is by extension that your windows PC identifies file formats.

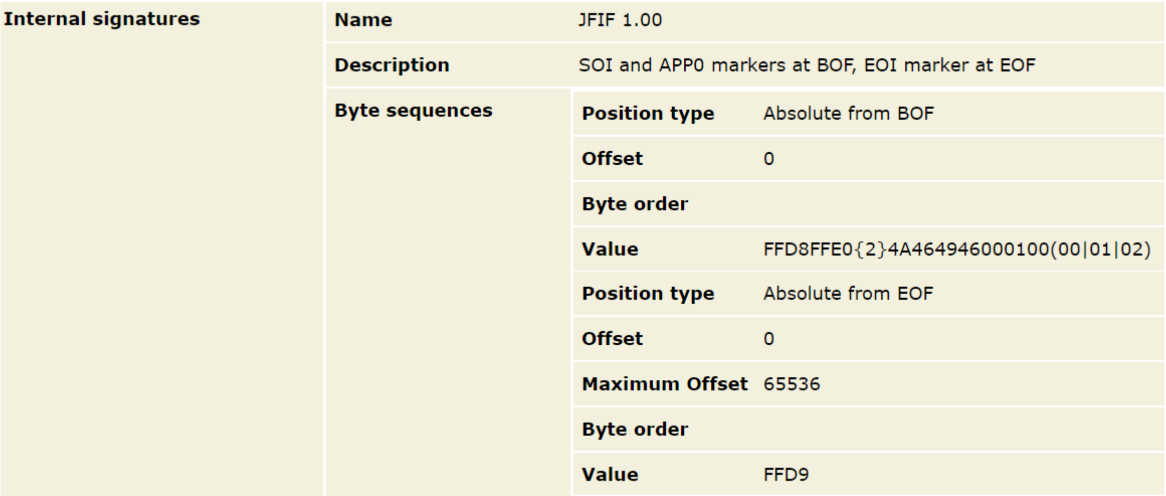


| | | | |
|---|---|---|---|
| Text File Extension-only.txt | 16/03/2022 08:48 | Text Document | 0 KB |
| Text File Extension-only - Copy.asdfghjkl | 16/03/2022 08:48 | ASDFGHJKL File | 0 KB |

For instance copying and pasting this Text Document and then changing the extension (which is done by renaming the file) results in windows identifying the Text Document as an ASDFGHJKL File.

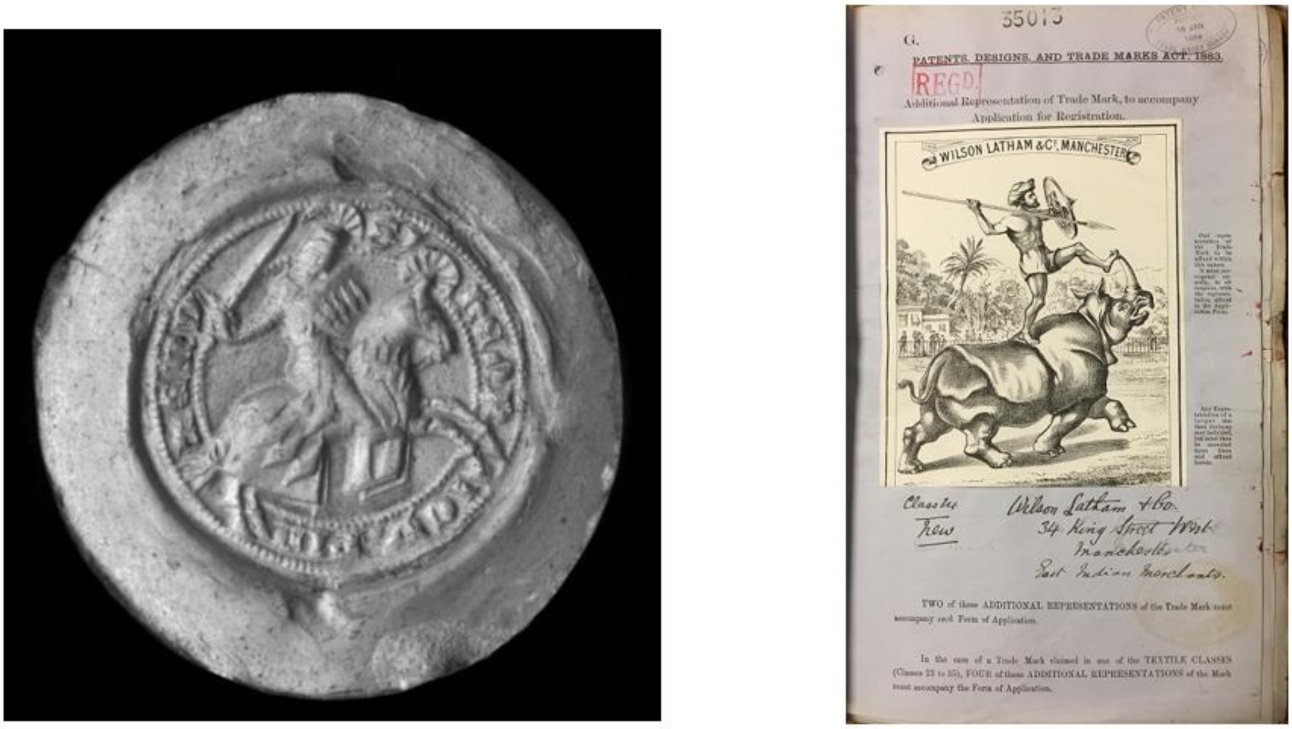
### Binary Signatures

All files and systems on a computer are made up of binary (the 01100010 01101001 01101110 01100001 01110010 01111001 often seen in images associated with computers), which in turn can be translated into another 'computer language' called hex (or 68 65 78). To find a binary signature we use a hex editor to look for repeated patterns across files of the same file type. We then input these patterns into PRONOM. For example an internal signature on the website is displayed as below:



| Internal signatures | Name | JFIF 1.00 | |
|---|---|---|---|
| | Description | SOI and APP0 markers at BOF, EOI marker at EOF | |
| | Byte sequences | **Position type** | Absolute from BOF |
| | | **Offset** | 0 |
| | | **Byte order** | |
| | | **Value** | FFD8FFE0{2}4A464946000100(00\|01\|02) |
| | | **Position type** | Absolute from EOF |
| | | **Offset** | 0 |
| | | **Maximum Offset** | 65536 |
| | | **Byte order** | |
| | | **Value** | FFD9 |

PRONOM has its own syntax, similar to regex which it uses to describe the specific pattern (more on this later). Some of the patterns are purposefully left there by developers for identification purposes and are called Magic Bytes. Think of it as a trademark or a stamp that a developer left behind when creating a file format.



35013

G.

PATENTS, DESIGNS, AND TRADE MARKS ACT, 1883.

REGD.

Additional Representation of Trade Mark, to accompany Application for Registration.

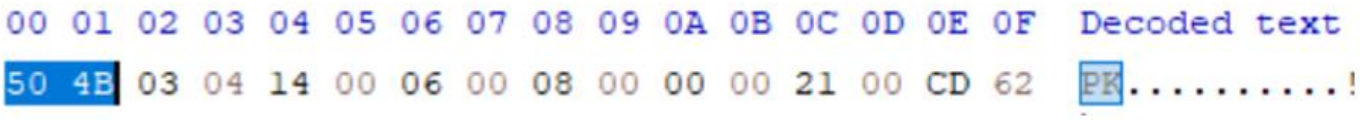
WILSON LATHAM & Co. MANCHESTER

Class 24 New

Wilson Latham & Co.
34 King Street West
Manchester
East Indian Merchants.

TWO of these ADDITIONAL REPRESENTATIONS of the Trade Mark must accompany each Form of Application.

In the case of a Trade Mark claimed in one of the TEXTILE CLASSES (Classes 23 to 35), FOUR of these ADDITIONAL REPRESENTATIONS of the Mark must accompany the Form of Application.



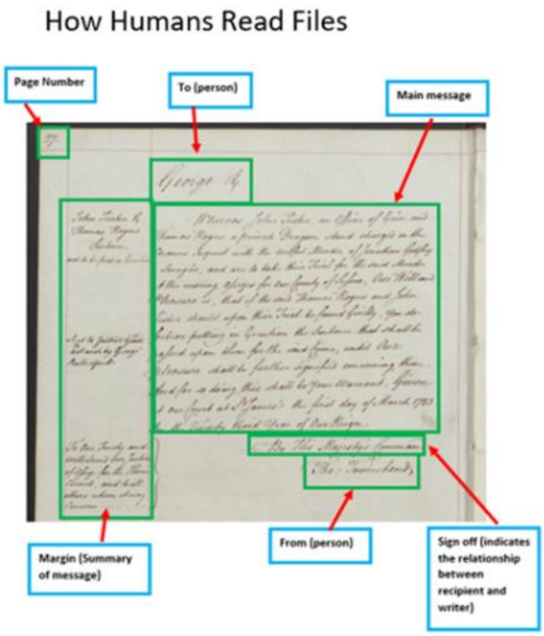
```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
50 4B 03 04 14 00 06 00 08 00 00 00 21 00 CD 62  PK..........!
```

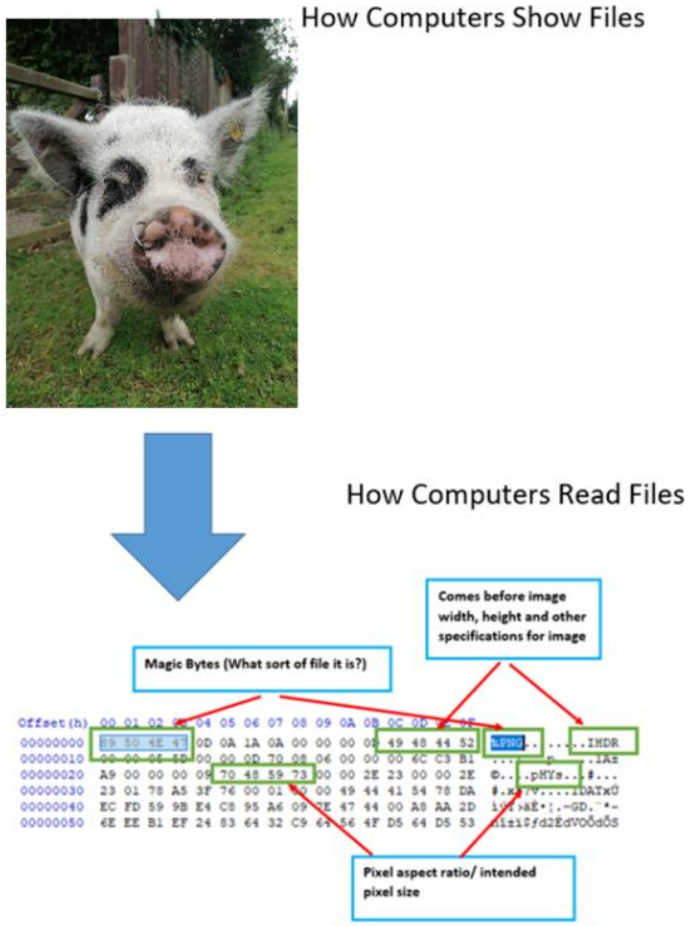
*Trademarks and seals from The National Archives Collection, alongside the magic bytes of a zip file.*

Whilst other patterns can be requirements in the specification that certain information has to be stated in a certain place.

Software is programmed to understand the structure of the file format and therefore show a file as an image, and not a stream of binary. A direct comparison can be seen with the human brain reading a letter. We can disseminate what text relates to what information easily, with an understanding that the top right will contain the sender's address and the end of the letter will be signed by the sender. So in the same way that software will render files, a human will read and understand a physical record.

When your file format identification software recognises a file type it is because it has looked through the hex of your files to see if it matches any of the patterns we have in the PRONOM database.

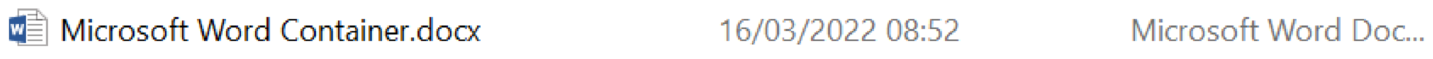

How Humans Read Files
Page Number
To (person)
Main message
Margin (Summary of message)
From (person)
Sign off (indicates the relationship between recipient and writer)


How Computers Show Files
How Computers Read Files
Magic Bytes (What sort of file it is?)
Comes before image width, height and other specifications for image
Pixel aspect ratio/ intended pixel size

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  ‰PNG........IHDR
00000010  00 00 05 5D 00 00 0D 70 08 06 00 00 00 6C C3 B1  .......p.....lÃ±
00000020  A9 00 00 00 09 70 48 59 73 00 00 2E 23 00 00 2E  ©....pHYs...#...
00000030  23 01 78 A5 3F 76 00 01 00 00 49 44 41 54 78 DA  #.x¥?v....IDATxÚ
00000040  EC FD 59 9B E4 C8 95 A6 09 7E 47 44 00 A8 AA 2D  ìýY›äÈ•¦.~GD.¨ª-
00000050  6E EE B1 EF 24 83 64 32 C9 64 56 4F D5 64 D5 53  nî±ï$ƒd2ÉdVOÕdÕS
```

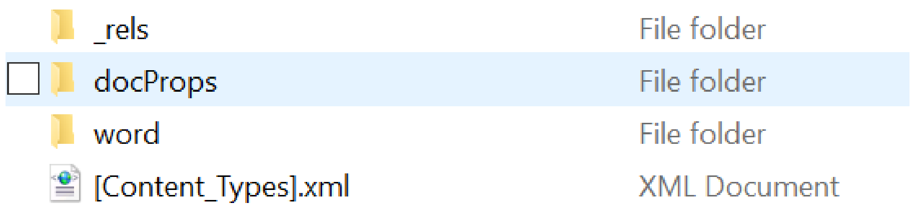
## Container Signatures

These are file formats that are compressed or zipped. Though you won't often see this immediately just by looking at the file. For example, most Microsoft file formats are actually zipped folders. As the file is zipped it contains other files and possibly folders inside it. For instance a Microsoft Word file in windows normally looks like this:



Microsoft Word Container.docx 16/03/2022 08:52 Microsoft Word Doc...

But when you unzip or extract the file it looks like this:



| | |
|---|---|
| _rels | File folder |
| □ docProps | File folder |
| word | File folder |
| [Content_Types].xml | XML Document |

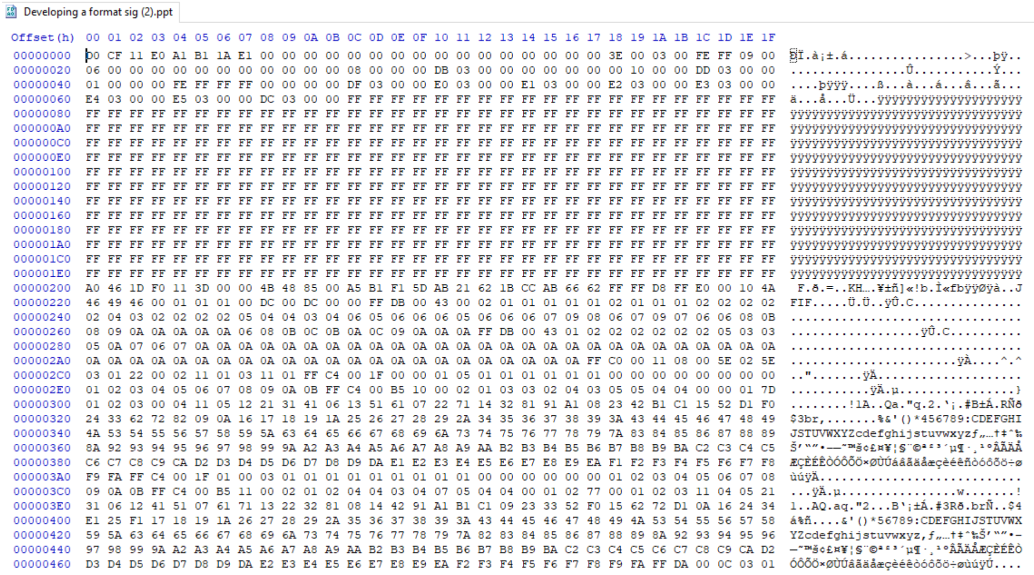
Think of container files as stacking boxes or a matryoshka doll.

Each container file contains other file formats and possibly folders which we aim to extract and analyse. We use information such as the internal file paths within the file format and the binary signatures within individual files to create a signature. We look for file paths that multiple samples have in common as well as binary signatures within those files that multiple files have in common. Think of it like the binary signature identification but with the additional consideration of file structure. You can also read more information about container signatures in Ross Spencer's blog.

## Researching Binary Signatures

Binary signatures are often more common and easier to find than container signatures. To find out their signatures we look for the repeated sequence within format's hex. Hexadecimal (or hex) is a system that simplifies and translates how binary is represented within file format. We use the hex editor to tell us the repeated information that is within each file format, which we record as a file format signature.

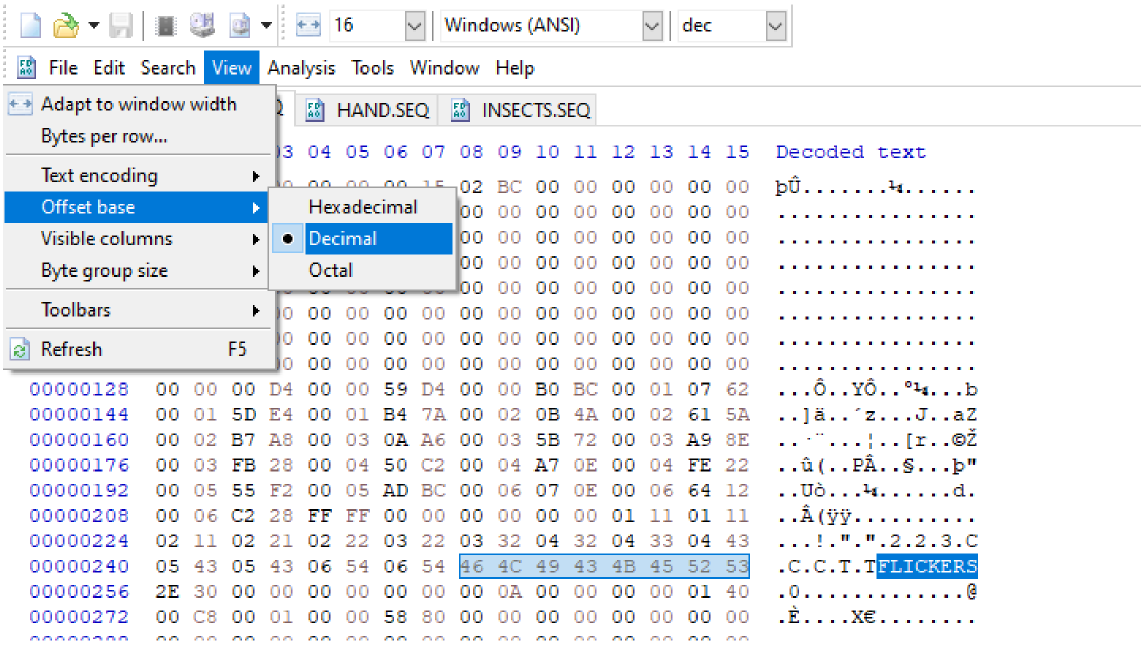
With your sufficient sample files, you can run them through file format identification tool, and then input/drag and drop your files into a hex editor and start analysing the hex sequence. Usually, this sequence is either at the beginning of file format (BOF) or at the end of file format (EOF) but can sometimes be variable (anywhere within the file).



Developing a format sig (2).ppt

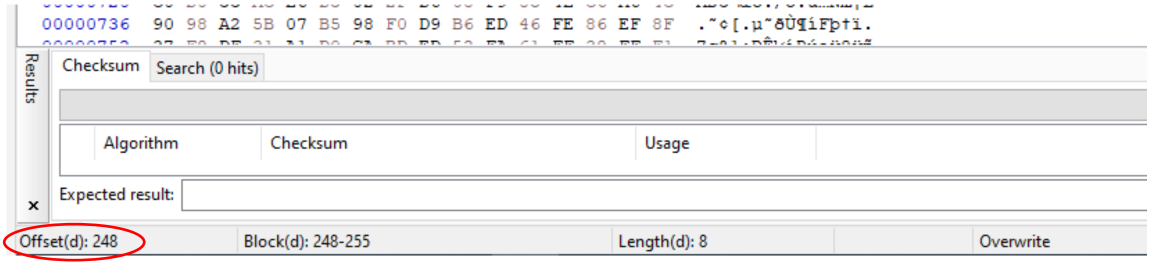
```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F

00000000  D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09 00  ÐÏ.à¡±.á................>...þÿ..
00000020  06 00 00 00 00 00 00 00 00 00 00 00 08 00 00 00 DB 03 00 00 00 00 00 00 00 10 00 00 DD 03 00 00  ................Û...........Ý...
00000040  01 00 00 00 FE FF FF FF 00 00 00 00 DF 03 00 00 E0 03 00 00 E1 03 00 00 E2 03 00 00 E3 03 00 00  ....þÿÿÿ....ß...à...á...â...ã...
00000060  E4 03 00 00 E5 03 00 00 DC 03 00 00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ä...å...Ü...ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000080  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000A0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000C0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000000E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000100  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000120  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000140  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000160  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000180  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001A0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001C0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000001E0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00000200  A0 46 1D F0 11 3D 00 00 4B 48 85 00 A5 B1 F1 5D AB 21 62 1B CC AB 66 62 FF FF D8 FF E0 00 10 4A   F.ð.=..KH….¥±ñ]«!b.Ì«fbÿÿØÿà..J
00000220  46 49 46 00 01 01 01 00 DC 00 DC 00 00 FF DB 00 43 00 02 01 01 01 01 01 02 01 01 01 02 02 02 02  FIF.....Ü.Ü..ÿÛ.C...............
00000240  02 04 03 02 02 02 02 05 04 04 03 04 06 05 06 06 06 05 06 06 06 07 09 08 06 07 09 07 06 06 08 0B  ................................
00000260  08 09 0A 0A 0A 0A 0A 06 08 0B 0C 0B 0A 0C 09 0A 0A 0A FF DB 00 43 01 02 02 02 02 02 02 05 03 03  ..................ÿÛ.C..........
00000280  05 0A 07 06 07 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A  ................................
000002A0  0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A 0A FF C0 00 11 08 00 5E 02 5E  .......................ÿÀ....^.^
000002C0  03 01 22 00 02 11 01 03 11 01 FF C4 00 1F 00 00 01 05 01 01 01 01 01 01 00 00 00 00 00 00 00 00  ..".......ÿÄ....................
000002E0  01 02 03 04 05 06 07 08 09 0A 0B FF C4 00 B5 10 00 02 01 03 03 02 04 03 05 05 04 04 00 00 01 7D  ...........ÿÄ.µ.................}
00000300  01 02 03 00 04 11 05 12 21 31 41 06 13 51 61 07 22 71 14 32 81 91 A1 08 23 42 B1 C1 15 52 D1 F0  ........!1A..Qa."q.2.‘¡.#B±Á.RÑð
00000320  24 33 62 72 82 09 0A 16 17 18 19 1A 25 26 27 28 29 2A 34 35 36 37 38 39 3A 43 44 45 46 47 48 49  $3br‚......%&'()*456789:CDEFGHI
00000340  4A 53 54 55 56 57 58 59 5A 63 64 65 66 67 68 69 6A 73 74 75 76 77 78 79 7A 83 84 85 86 87 88 89  JSTUVWXYZcdefghijstuvwxyzƒ„…†‡ˆ‰
00000360  8A 92 93 94 95 96 97 98 99 9A A2 A3 A4 A5 A6 A7 A8 A9 AA B2 B3 B4 B5 B6 B7 B8 B9 BA C2 C3 C4 C5  Š’“”•–—˜™š¢£¤¥¦§¨©ª²³´µ¶·¸¹ºÂÃÄÅ
00000380  C6 C7 C8 C9 CA D2 D3 D4 D5 D6 D7 D8 D9 DA E1 E2 E3 E4 E5 E6 E7 E8 E9 EA F1 F2 F3 F4 F5 F6 F7 F8  ÆÇÈÉÊÒÓÔÕÖ×ØÙÚáâãäåæçèéêñòóôõö÷ø
000003A0  F9 FA FF C4 00 1F 01 00 03 01 01 01 01 01 01 01 01 01 00 00 00 00 00 00 01 02 03 04 05 06 07 08  ùúÿÄ............................
000003C0  09 0A 0B FF C4 00 B5 11 00 02 01 02 04 04 03 04 07 05 04 04 00 01 02 77 00 01 02 03 11 04 05 21  ...ÿÄ.µ................w.......!
000003E0  31 06 12 41 51 07 61 71 13 22 32 81 08 14 42 91 A1 B1 C1 09 23 33 52 F0 15 62 72 D1 0A 16 24 34  1..AQ.aq."2...B‘¡±Á.#3Rð.brÑ..$4
00000400  E1 25 F1 17 18 19 1A 26 27 28 29 2A 35 36 37 38 39 3A 43 44 45 46 47 48 49 4A 53 54 55 56 57 58  á%ñ....&'()*56789:CDEFGHIJSTUVWX
00000420  59 5A 63 64 65 66 67 68 69 6A 73 74 75 76 77 78 79 7A 82 83 84 85 86 87 88 89 8A 92 93 94 95 96  YZcdefghijstuvwxyz‚ƒ„…†‡ˆ‰Š’“”•–
00000440  97 98 99 9A A2 A3 A4 A5 A6 A7 A8 A9 AA B2 B3 B4 B5 B6 B7 B8 B9 BA C2 C3 C4 C5 C6 C7 C8 C9 CA D2  —˜™š¢£¤¥¦§¨©ª²³´µ¶·¸¹ºÂÃÄÅÆÇÈÉÊÒ
00000460  D3 D4 D5 D6 D7 D8 D9 DA E2 E3 E4 E5 E6 E7 E8 E9 EA F2 F3 F4 F5 F6 F7 F8 F9 FA FF DA 00 0C 03 01  ÓÔÕÖ×ØÙÚâãäåæçèéêòóôõö÷øùúÿÚ....
```

You will need to scroll through the hex sequence to find the common pattern between your samples of the file format you are researching. Once you have found the pattern, you can record it and there you have it, you found your first file format signature! You will also need to record the relevant information about your sequence which is where within the file is placed, so the BOF; EOF; or Variable. Additionally, we also look for a number where the sequence starts, and we call this an **Offset** number. You can find this number by selecting your sequence and adjusting your hex editor to show offset base as a decimal number. Below you can see an example of a sequence and how we found its offset number.



```
00000128  00 00 00 D4 00 00 59 D4 00 00 B0 BC 00 01 07 62  ...Ô..YÔ..°¼...b
00000144  00 01 5D E4 00 01 B4 7A 00 02 0B 4A 00 02 61 5A  ..]ä..´z...J..aZ
00000160  00 02 B7 A8 00 03 0A A6 00 03 5B 72 00 03 A9 8E  ..·¨...¦..[r..©Ž
00000176  00 03 FB 28 00 04 50 C2 00 04 A7 0E 00 04 FE 22  ..û(..PÂ..§...þ"
00000192  00 05 55 F2 00 05 AD BC 00 06 07 0E 00 06 64 12  ..Uò...¼......d.
00000208  00 06 C2 28 FF FF 00 00 00 00 00 00 01 11 01 11  ..Â(ÿÿ..........
00000224  02 11 02 21 02 22 03 22 03 32 04 32 04 33 04 43  ...!.".".2.2.3.C
00000240  05 43 05 43 06 54 06 54 46 4C 49 43 4B 45 52 53  .C.C.T.TFLICKERS
00000256  2E 30 00 00 00 00 00 00 00 0A 00 00 00 00 01 40  .0.............@
00000272  00 C8 00 01 00 00 58 80 00 00 00 00 00 00 00 00  .È....X€........
```

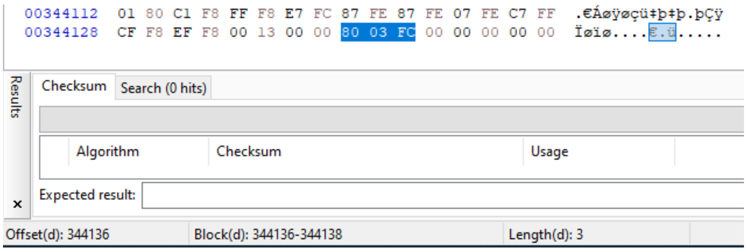
Below number tells us that the offset is 248, meaning that the sequence starts at 248th hex.



```
00000736  90 98 A2 5B 07 B5 98 F0 D9 B6 ED 46 FE 86 EF 8F  .˜¢[.µ˜ðÙ¶íFþ†ï.
```

Offset(d): 248 Block(d): 248-255 Length(d): 8

For EOF sequences we calculate the offset number from the end of the file format and record that number.

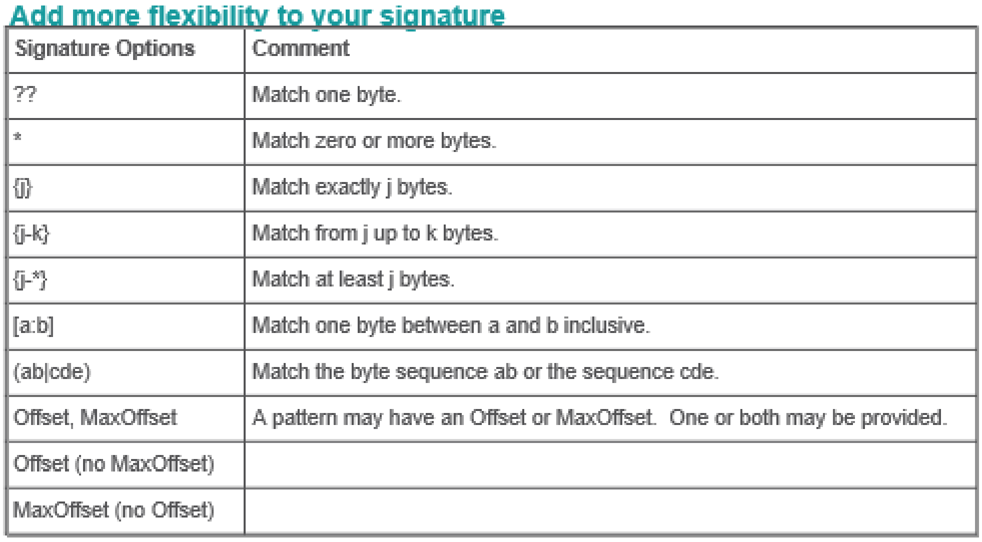


```
00344112  01 80 C1 F8 FF F8 E7 FC 87 FE 87 FE 07 FE C7 FF  .€Áøÿøçü‡þ‡þ.þÇÿ
00344128  CF F8 EF F8 00 13 00 00 80 03 FC 00 00 00 00 00  Ïøïø....€.ü.....
```

Offset(d): 344136 Block(d): 344136-344138 Length(d): 3

Often, the offset number can vary for which we record the minimum number (min offset) found in our sample of where the sequence can start from and a maximum number of where it can start from (max offset).

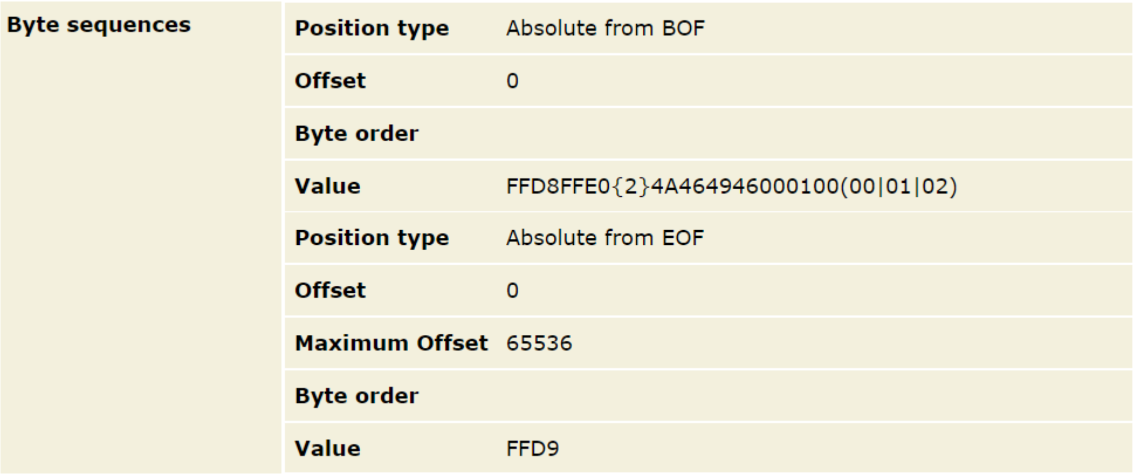
Sometimes, the sequence can vary slightly between samples, and we can add flexibility to our sequence using the below table.



## Add more flexibility to your signature

| Signature Options | Comment |
|---|---|
| ?? | Match one byte. |
| * | Match zero or more bytes. |
| {j} | Match exactly j bytes. |
| {j-k} | Match from j up to k bytes. |
| {j-*} | Match at least j bytes. |
| [a:b] | Match one byte between a and b inclusive. |
| (ab\|cde) | Match the byte sequence ab or the sequence cde. |
| Offset, MaxOffset | A pattern may have an Offset or MaxOffset. One or both may be provided. |
| Offset (no MaxOffset) | |
| MaxOffset (no Offset) | |

Bear in mind your file format could have two or three sequences. The one below looks for a sequence at the beginning of the file AND for another at the end of the file.



| | | |
|---|---|---|
| **Byte sequences** | **Position type** | Absolute from BOF |
| | **Offset** | 0 |
| | **Byte order** | |
| | **Value** | FFD8FFE0{2}4A464946000100(00\|01\|02) |
| | **Position type** | Absolute from EOF |
| | **Offset** | 0 |
| | **Maximum Offset** | 65536 |
| | **Byte order** | |
| | **Value** | FFD9 |

The more information you can provide the more unique the signature will be. This will prevent it from clashing with other existing file formats in the future. However it is a balance, the more specific your signature the less likely it can be that all files of that type are guaranteed to have those patterns.

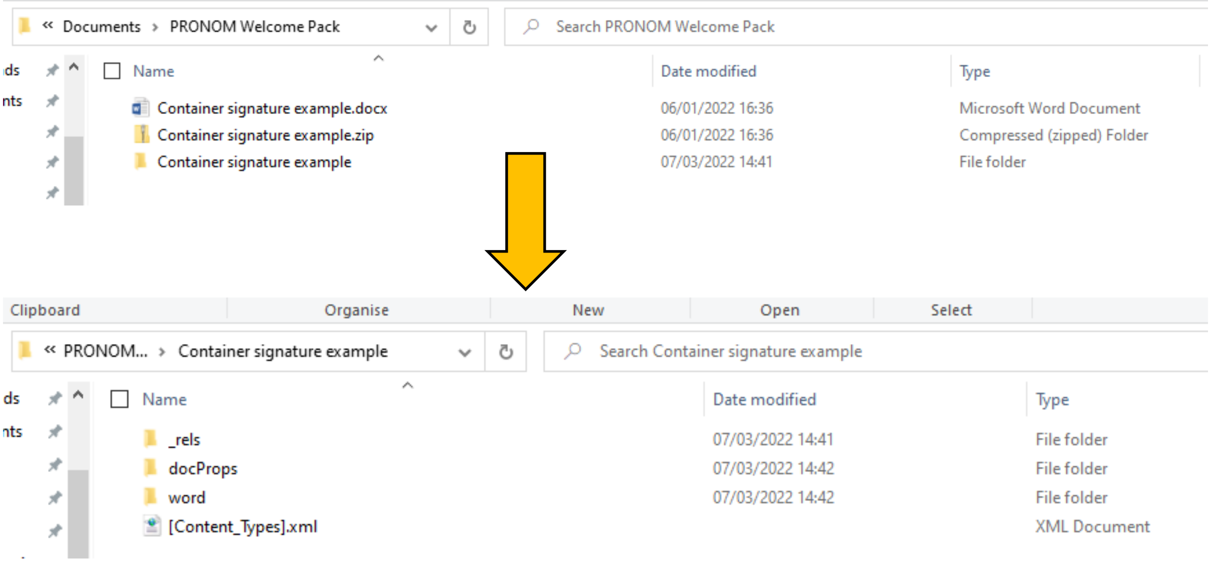
Now you should be ready to record your signature with its relevant information in the template provided.

## Researching Container Signatures

As mentioned above container signatures contains two or more sequences and we aim to find all of them. Currently, PRONOM identifies two types of containers which are either ZIP or OLE2. You can find out which of these formats you are dealing with using the file identification tool. Once you found out that you have a container signature to research. You can make a copy of it on your machine, change the extension to .zip and unzip it using an in-built tool. Alternatively you can use the **7Zip tool** to extract the folder content.

Let us look at an example together:

Microsoft Word is a ZIP file format with an extension .docx. If we create a copy of this file, change its extension to .zip, and extract the file format, we can see the content of this format.

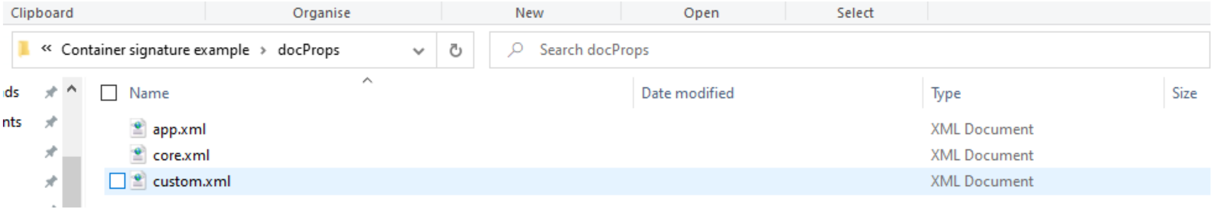

« Documents › PRONOM Welcome Pack
Search PRONOM Welcome Pack
Clipboard
Organise
New
Open
Select
« PRONOM... › Container signature example
Search Container signature example

| Name | Date modified | Type |
|---|---|---|
| Container signature example.docx | 06/01/2022 16:36 | Microsoft Word Document |
| Container signature example.zip | 06/01/2022 16:36 | Compressed (zipped) Folder |
| Container signature example | 07/03/2022 14:41 | File folder |

| Name | Date modified | Type |
|---|---|---|
| _rels | 07/03/2022 14:41 | File folder |
| docProps | 07/03/2022 14:42 | File folder |
| word | 07/03/2022 14:42 | File folder |
| [Content_Types].xml | | XML Document |

You can see from this example that a .docx file is not just a singular file format but rather exists as a set of multiple file formats, each of which we aim to record.

For container signatures, you will need to unzip each of your sample to find out what are the repeated pathways (folder names) are included in this format and then record them. For the example above we would record that the type of format is a ZIP format (this can be identified using file format identification tool), then we would say that it contains folders of _rels, docProps, word and an .xml file format, on which we would need to do a further research and find out its binary signature (more on this later). We would also open each of those formats and then record whether they contain other formats or folders, as shown below.

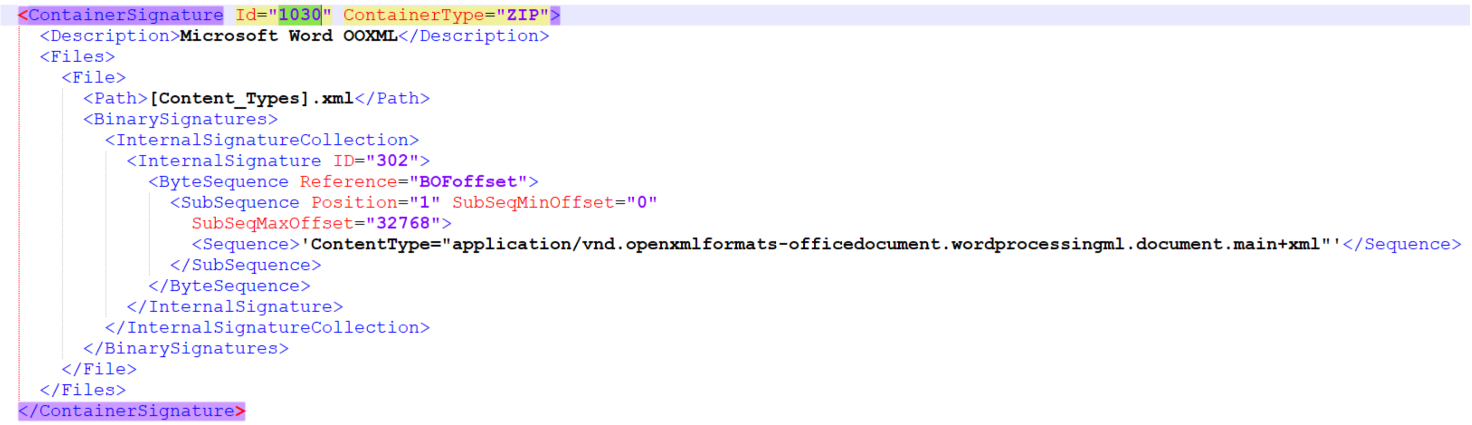


| Name | Date modified | Type | Size |
|---|---|---|---|
| app.xml | | XML Document | |
| core.xml | | XML Document | |
| custom.xml | | XML Document | |

Once you have collected information on the file pathways and binary signatures within them you have found your container signature.

To show you how these file paths and sequences are used here is an example of how container signatures are written in xml. Container signatures are currently stored in a separate file and linked to the main database. To see this, if you are using DROID go to C:/Users/[username]/.droid6/containersigs/ and you will find the xml container signatures that DROID is using to identify your files.

Below is the xml for the Microsoft Word example above:



```
<ContainerSignature Id="1030" ContainerType="ZIP">
  <Description>Microsoft Word OOXML</Description>
  <Files>
    <File>
      <Path>[Content_Types].xml</Path>
      <BinarySignatures>
        <InternalSignatureCollection>
          <InternalSignature ID="302">
            <ByteSequence Reference="BOFoffset">
              <SubSequence Position="1" SubSeqMinOffset="0"
                SubSeqMaxOffset="32768">
                <Sequence>'ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"'</Sequence>
              </SubSequence>
            </ByteSequence>
          </InternalSignature>
        </InternalSignatureCollection>
      </BinarySignatures>
    </File>
  </Files>
</ContainerSignature>
```

This particular file format is identified by opening the file '[Content_Types].xml'. The sequence 'ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"' can then be found between the absolute beginning of the file 0 to byte number 32768. This is shown in the values SubSeqMinOffset and SubSeqMaxOffset. Unlike binary signatures container sequences can be written in both ASCII and hex and we use both in the container signatures interchangeably dependending on what is more human readable.

It is possible to have multiple file paths and multiple sequences  in multiple files for a container signature.

## No Signature? Extension-only Formats

And after all this work, you realise that none of the above methods work, you cannot figure out any signatures. What to do now…

Well, you might be dealing with a format which does not have a specific signature – yes, those sadly do exist. These formats are trickier to be identified but we can still identify them by their extension. This method of identification is not very secure and so we try to avoid it as much as possible. This is because file format extension can be easily altered by anyone (as you may have done when researching container signature), and we also have examples of file formats which can have over fifty different file format extensions. However, if you have analysed your files correctly using the above steps and you can conclude that no clear sequence is present in the file format, you can submit the file format to PRONOM team as an extension only format.

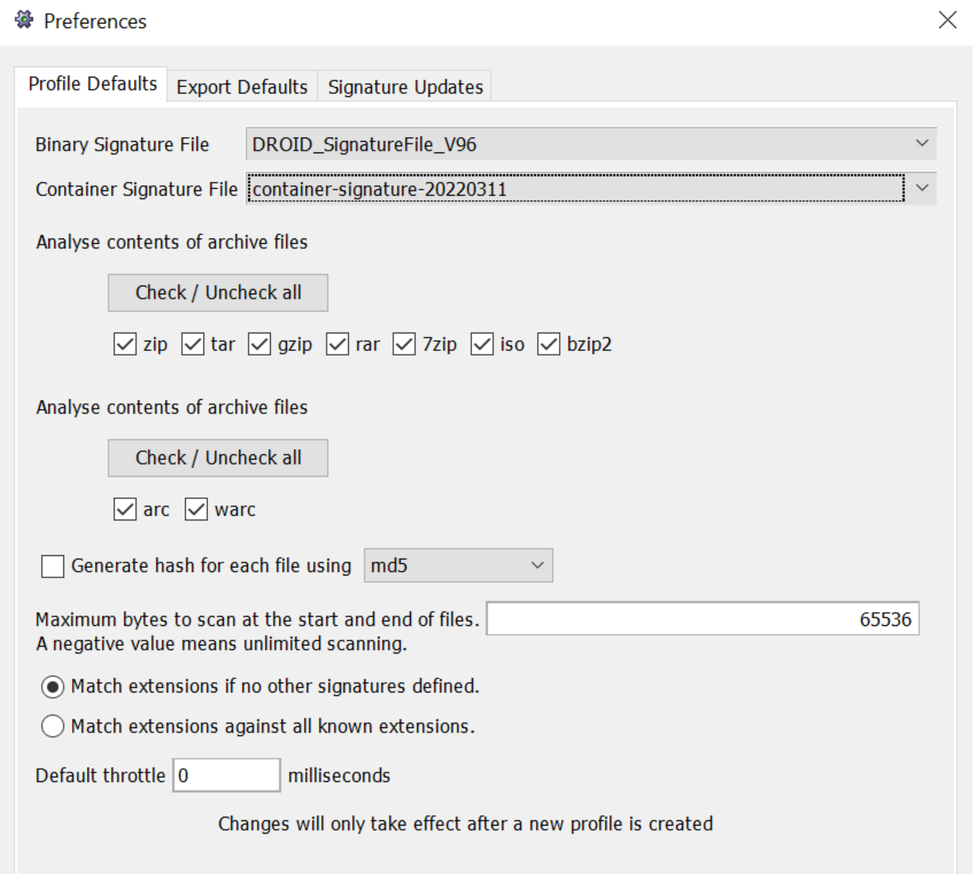
## Checking It Works!

The best way to check if your signature works is to use Ross Spencer’s Signature Development Utility tool. Instructions and a walk through on how to use this can be found here, and the tool itself is found here.

The tool works by inputting your signature and research into the fields provided and then generates a signature that you can download and test using your file format identification tool.

In the case of DROID it is a simple process to add a new signature or container signature to DROID. Simply add the files to the .droid6 folder found in C:/Users/[username]/.droid6 . Place the container signatures in the container_sigs folder and the regular signatures generated in the signature_files folder. Within DROID go to Tools/ Preferences and the binary signature file and/or container signature file you have generated should appear in the drop downs at the top. Select your signature and press OK.

Now you can run DROID over your files and test that they are identifying, don’t forget to create a new profile or the changes you have made won’t take effect.



Binary Signature File: DROID_SignatureFile_V96

Container Signature File: container-signature-20220311

Analyse contents of archive files

☑ zip ☑ tar ☑ gzip ☑ rar ☑ 7zip ☑ iso ☑ bzip2

Analyse contents of archive files

☑ arc ☑ warc

□ Generate hash for each file using md5

Maximum bytes to scan at the start and end of files. A negative value means unlimited scanning. 65536

● Match extensions if no other signatures defined.

○ Match extensions against all known extensions.

Default throttle 0 milliseconds

Changes will only take effect after a new profile is created

Further instruction on how to use DROID can be found here.

## Description, MIME Types and Other Useful Information

Do not forget you may also want to do some additional research to find out relevant information about your file format to create a description; find out if there are more versions of this file formats; additional possible extensions; and any other information included in the template provided.

### Format Name

Use the official name where known. Please capitalise each word unless the format name is stylised in some alternative way, e.g. Apple iBook. Include the version numbers (where relevant). If the file format has other official or common names then that is useful to know too.

### Extensions

Even if we aren't identifying a file by extension-only we collect information about the file extension. This also prevents extension mismatches when identifying in DROID.

### Description

For each file format we write a short description of the file format, usually around 2-4 sentences. We keep our descriptions objective, and avoid commercial statements e.g. 'this software is best for'. Online sources can be used, and we try to use multiple sources, rewritten into our own words. As a general rule if you are unsure about what you are writing, or it is too technical then keep it short and omit it. We believe that if anyone questioned our description we should be able to explain what we meant and the subject area.

Areas that could be covered in the description include a timeline of development and support, the function of the file format and software and details of the format specification. The description should also include information relevant to its identification, preservation and the conditions you might encounter it.

Questions you could ask for example are what different extensions may refer to? Microsoft Word for example has a .docx extension but also a dotx extension which signifies that it is a template file. Whether the size of the file has any relevance? Some file formats are a specific number of bytes and this could signify a property of the file. If other files are normally found in conjunction with this format? Some files are always found with another type of format as the data from one could input into the other.

### MIME Types and Identifiers

Another aspect of the research we do is finding out if there are any MIME types or other identifiers associated with the file format. A brief description of what a MIME type is can be found here. This should be an official Media Type. We only accept MIME types that are either registered and listed via the IANA (https://www.iana.org/assignments/media-types/media-types.xhtml) or listed in official format documentation produced by the vendor. The best way to search IANA is to find in page and search for key words of the format.

We have other types of identifiers such as apple resource forks that are less common but if relevant useful to note down in research.

### File Format Type

The current list of format classifications within PRONOM are:

- Audio Database - the formats of database software, such as MS Access, MySQL
- Email
- GIS - Geographic Information System (geospatial data formats)
- Image (Raster) - images based on pixel grids, such as JPG, GIF, PNG
- Image (Vector) - images based on mathematical primitives, such as SVG, Adobe Illustrator, CorelDraw, WMF
- Page Description - the language of printers (https://en.wikipedia.org/wiki/Page_description_language). Examples include HP-GL, PDF, PostScript
- Presentation - such as Powerpoint, Impress, Apple Keynote Spreadsheet
- Text (Unstructured) - plain text formats with no formal structure
- Text (Structured) - plain text formats with defined, regular structure
- Text (Mark-up) - such as XML, SGML, MD
- Word Processor
- Video
- Aggregate - such as zip, WARC, 7z, rar, iso
- Dataset - structured forms of data
- Model - 3d formats such as CAD and 3d models
- Font

Your format may not easily fit into any of the above categories, so feel free to reach out for advice.

### Vendors

Do you know who supports it? Or who developed it originally?

### Links

It is always good to know where you found all your information. If you want to add the links to your submission then we can reference them for future users.

### Credits

We want to make sure that everyone (who would like to be) is credited for all the work they've put in to researching for us. Usually this is by organisation but if you prefer it can be by individual. We put this information in the source of a new PRONOM entry and in our release notes that can be found here.

**GO!**
Once you have identified all the above information, you can use one of the templates attached with this welcome pack to input the information and share it with PRONOM team.

Alternatively, you can use the Signature Development Utility Tool to create sequences and send it to PRONOM team. More information about this tool can be found here.

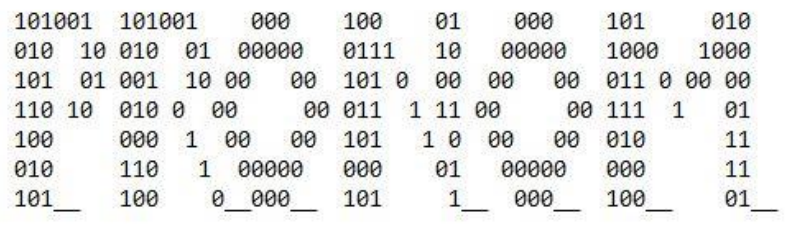
You have not completed your research and it is time to share it with the community! You can post your research to our mailbox or in our GitHub page.

In case of any troubleshooting or more information, please contact the PRONOM team using mailbox pronom@nationalarchives.gov.uk or our dedicated Google groups, PRONOM and DROID.

To check in and see how we are progressing with our work we update an online spreadsheet with our progress towards the next release as we go. This will give you a preview of what will be in the next release. We try and keep each release to around 40-70 changes. We also upload any changes we have made in our development environment to GitHub every Friday here. If you don't want to wait for the official release you can download the signatures and use them (just be warned that these will not have been officially tested and may have errors).

Could we have explained something better in this pack? Was there a part that was difficult to understand? Do you have some tips that you wish we'd added in? Let us know via our mailbox and we will improve it!

Congratulations on completing your file format research, we hope you enjoyed it as much as we do. Welcome to the PRONOM community! 👏



```
101001  101001    000    100    01    000    101     010
010  10 010  01  00000   0111   10   00000   1000   1000
101  01 001  10  00  00  101 0  00  00   00  011 0 00 00
110 10  010 0  00      00 011  1 11 00      00 111  1   01
100     000  1  00   00  101   1 0  00   00  010       11
010     110   1  00000   000    01   00000   000       11
101__   100    0__000__  101     1__  000__  100__    01__
```

**List of Resources and Tools**

| Resource Link | About |
|---|---|
| PRONOM \| Welcome (nationalarchives.gov.uk) | PRONOM home page |
| PRONOM \| Search (nationalarchives.gov.uk) | PRONOM database search page |
| PRONOM: A database centenary - The National Archives blog | Introductory blog post by the PRONOM team introducing the work we do and some of our processes. |
| Five Star File Format Signature Development - Open Preservation Foundation | Blog post written by Ross Spencer outlining researching file formats. |
| Digital Archiving at the University of York: My first file format signature (digital-archiving.blogspot.com) | Blog post by Jenny Mitcham introducing a first time experience conducting file format research. |
| Download DROID: file format identification tool - The National Archives | DROID homepage |
| Siegfried for archivists | Siegfried homepage |
| fido - Open Preservation Foundation | Fido homepage |
| Download (7-zip.org) | Download page for 7-zip tool |
| Just Solve the File Format Problem (archiveteam.org) | The homepage for just solve the file format problem from which you can use the search or browse by extension. |
| File Extension Seeker - Metasearch engine for file extensions (file-extension.net) | |
| Marco Pontello's Home - Software - TrID - Files Extensions and File Type Definitions list (mark0.net) | TrID page for searching by file extension. |
| DROID Container Signature Files: What they are and how to create them: A template and an | Blog post written by Ross Spencer introducing container signature and how to create them. |

| | |
|---|---|
| example, or few... - Open Preservation Foundation | |
| Signature development utillity 2.0 (ffdev.info) | Signature development utility tool which allows you to test your file format signatures |
| digital-preservation/PRONOM_Research (github.com) | PRONOM GitHub page created for |
| pronom-research-week/formats_without_signatures_Oct_2020.csv at master · digital-preservation/pronom-research-week (github.com) | PRONOM list of file formats without signatures as of October 2020 |
| pronom-research-week/formats_with_outline_descriptions_only.csv at master · digital-preservation/pronom-research-week (github.com) | PRONOM list of file formats without descriptions as of October 2020 |
| File Formats in Test Environment (docs.google.com) | Link to the file formats currently in our test environment |
| PRONOM_Research/Test Releases at main · digital-preservation/PRONOM_Research (github.com) | GitHub folder to which we upload test releases on a weekly basis. Users of PRONOM can test and use signatures early before each release. |
| https://www.iana.org/assignments/media-types/media-types.xhtml | List of MIME types officially registered with IANA |