The International Journal of Digital Curation
Volume 8, Issue 1 | 2013

Generation of a Skeleton Corpus of Digital Objects for the Validation and Evaluation of Format Identification Tools and Signatures

Ross Spencer,
Independent Digital Preservation Researcher

Abstract

To preserve digital information it is vital that the format of that information can be identified, in-perpetuity. This is the major focus of research within the field of Digital Preservation. The National Archives of the UK called for the Digital Preservation and Digital Curation communities to develop a test corpus of digital objects to help further develop tools to aid this purpose. Following that call, an attempt has been made to develop the suite.

This paper initially outlines a methodology to generate a skeleton corpus using simple user-generated digital objects. It then explores the lessons learnt in the generation of a corpus using scripting language techniques from the file format signatures described in The National Archives PRONOM technical registry. It will also discuss the use of the digital signature for this purpose, the benefits of developing a test corpus using this technique. Finally, this paper will outline a methodology for future research before exploring how the community can best make use of the output of this project and how this project needs to be taken forward to completion.

International Journal of Digital Curation (2013), *8*(1), 120–130. http://dx.doi.org/10.2218/ijdc.v8i1.249

Introduction

The original 'call-to-arms' by The National Archives (Fetherston and Gollins, 2012) described the development of a test corpus of digital objects that could be used for the validation and evaluation of format identification tools and techniques. The benefits of such a suite for the community are numerous. The suite would create a baseline for testing format identification tools against each other, create a baseline for improving performance, and generate a mechanism by which identification and validation techniques could be improved. Like the PUID in PRONOM[1]. It would allow developers and researchers alike to speak unambiguously about the formats they have used to create new techniques, offering greater transparency for those who want to accurately recreate and verify results and measurements.

Fetherston and Gollins highlighted the difficulty of setting up a fully-formed test corpus and the considerations anyone taking up the challenge would need to make. While this paper makes no attempt to describe the successful generation and publication of a fully-formed test corpus, it does describe the generation of a skeleton corpus of objects for testing format identification tools, which is expected to provide a sustainable and practical solution for a number of issues described in the original paper. While this paper offers an interim solution for the 'call-to-arms' it is expected that it will continue to compliment the existence of the fully-formed suite, should such a corpus of objects become available.

The Skeleton Corpus

During the development of the DROID 6.1 project at The National Archives, mechanisms were required for the testing of the identification engine. This would take the form of unit tests developed in JUnit.[2] For example, a unit test might be written to test the identification engine's ability to parse syntactical constructs. Such a test would involve the matching of a digital object against a digital signature. Where bugs had previously been raised, the best digital object to test against would be the original that caused the problem. Using a test-driven development approach allows the developer to confirm the existence of the bug and demonstrate it by creating a unit test that fails under the initial conditions. Once modifications have been made to the engine to correct the issue, the unit test will eventually pass.

The digital object needs to be distributed with the unit test. The package, therefore, gets distributed with the source code. This scenario faces two of the issues highlighted by Fetherston and Gollins (2012). The **usability** of the source code is impacted, as the size of the source code becomes bloated by the size of the digital objects that need to be distributed with DROID. Further, **intellectual property rights (IPR) and copyright** are not easily addressed. In this instance, The National Archives needs to hold the IPR and copyright for any problem files they might want to distribute with the source code. This isn't always possible if files from long obsolete systems have

[1] Description of the PUID scheme in Wikipedia: http://en.wikipedia.org/wiki/PRONOM_technical_registry#The_PRONOM_Persistent_Unique_Identifier_.28PUID.29_scheme

[2] Java unit testing framework: http://junit.org

been provided to the organization for testing or if it doesn't prove easy to recreate those files in modern technical environments.

The approach taken by the developers of DROID to avoid these issues was to extract the relevant bytes from the format stream and place them in a new binary object; essentially creating a skeleton of the original format which demonstrated the properties required for a DROID signature to match against. This new object has a smaller footprint and, having been generated by The National Archives, becomes their IPR and therefore free for them to distribute with the DROID source code.

Taking this approach, it became apparent that it would be possible to create a set of objects like this for *all* formats with signatures in the PRONOM database. The immediate advantages established are:

- Such a suite will help identify collisions between two format signatures where a digital object is matched by more than one signature. This approximates the concept of false positives in Fetherston and Gollins (2012).
- Enable users to test the stability of DROID signature files over time and understand the differences in identification capability between releases.
- Provide a sanity check for signature developers and software developers. A unit test that gets executed during the compilation of DROID that runs across these test files should see an expected number of results each and every time. Currently, the DROID source code only contains tests for targeted formats, as alluded to above.

The skeleton corpus has the potential to provide the community with a set of objects, in short order, while the issues surrounding a fully featured test suite are considered in more detail. The suite may help us to understand the challenges outlined by Fetherston and Gollins (2012) by addressing some of the same questions, and by providing immediate and pragmatic answers which will be discussed in more detail throughout this paper.

Creating a Skeleton File

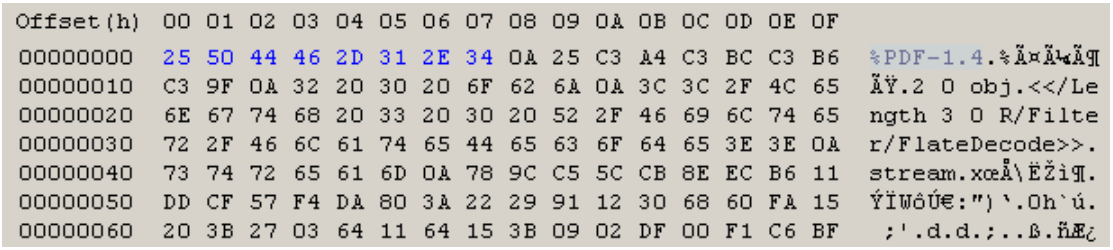
The current PRONOM signature for PDF 1.4 consists of two sequences at the head and the foot of the file.[3] They look like this:

```
Position: Absolute from beginning of file (BOF)
Sequence: 255044462D312E34
Max offset: 0

Position: Absolute from end of file (EOF)
Sequence: 2525454F46
Max offset: 4
```

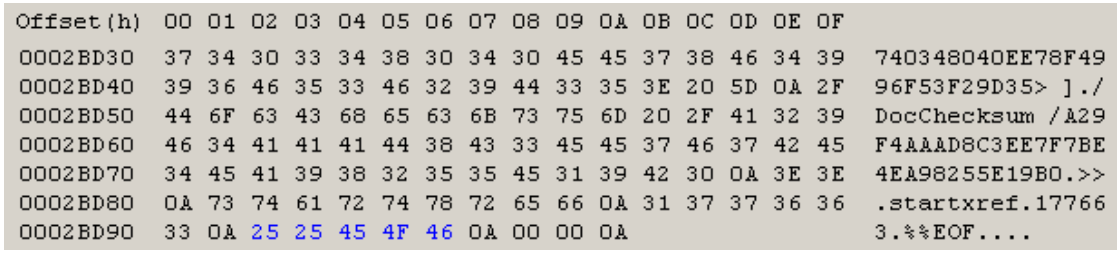
[3] Entry for the PDF 1.4 signature in the PRONOM database: http://www.nationalarchives.gov.uk/PRONOM/Format/proFormatSearch.aspx?status=detailReport&id=617&strPageToDisplay=signatures

Notice that there is no additional syntax involved in this signature, just two plain hexadecimal byte sequences. The offset in the second sequence enables the sequence to be found within a maximum threshold of four bytes at the end of a file. The two sequences as they look in a hex editor are shown in Figures 1 and 2 below.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	25	50	44	46	2D	31	2E	34	0A	25	C3	A4	C3	BC	C3	B6	%PDF-1.4.%Ã¤Ã¼Ã¶
00000010	C3	9F	0A	32	20	30	20	6F	62	6A	0A	3C	3C	2F	4C	65	ÃŸ.2 0 obj.<</Le
00000020	6E	67	74	68	20	33	20	30	20	52	2F	46	69	6C	74	65	ngth 3 0 R/Filte
00000030	72	2F	46	6C	61	74	65	44	65	63	6F	64	65	3E	3E	0A	r/FlateDecode>>.
00000040	73	74	72	65	61	6D	0A	78	9C	C5	5C	CB	8E	EC	B6	11	stream.xœÅ\ËŽì¶.
00000050	DD	CF	57	F4	DA	80	3A	22	29	91	12	30	68	60	FA	15	ÝÏWôÚ€:")'.0h`ú.
00000060	20	3B	27	03	64	11	64	15	3B	09	02	DF	00	F1	C6	BF	;'.d.d.;..ß.ñÆ¿

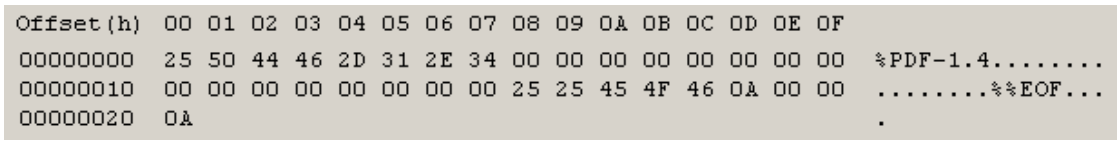
Figure 1. Beginning of file sequence highlighted in hex editor.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
0002BD30	37	34	30	33	34	38	30	34	30	45	45	37	38	46	34	39	740348040EE78F49
0002BD40	39	36	46	35	33	46	32	39	44	33	35	3E	20	5D	0A	2F	96F53F29D35>]./
0002BD50	44	6F	63	43	68	65	63	6B	73	75	6D	20	2F	41	32	39	DocChecksum /A29
0002BD60	46	34	41	41	41	44	38	43	33	45	45	37	46	37	42	45	F4AAAD8C3EE7F7BE
0002BD70	34	45	41	39	38	32	35	35	45	31	39	42	30	0A	3E	3E	4EA98255E19B0.>>
0002BD80	0A	73	74	61	72	74	78	72	65	66	0A	31	37	37	36	36	.startxref.17766
0002BD90	33	0A	25	25	45	4F	46	0A	00	00	0A						3.%%EOF....

Figure 2. End of file sequence highlighted in hex editor.

Taking the two sequences, we can open a new hex editor window and paste them directly into it, as shown in Figure 3.



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	25	50	44	46	2D	31	2E	34	00	00	00	00	00	00	00	00	%PDF-1.4........
00000010	00	00	00	00	00	00	00	00	25	25	45	4F	46	0A	00	00	........%%EOF...
00000020	0A																.

Figure 3. Sample skeleton file for PDF 1.4.

These sequences alone would generate a file 17 bytes long. I have padded the file in Figure 3 with an extra 16 null bytes. The benefit of this is that we can eyeball the file and easily distinguish between the two discrete patterns at either end. Using 16 bytes is a somewhat arbitrary decision; the exact structure of the files created for the skeleton corpus should ultimately be considered in more detail.

The file described in Figure 3 is enough for DROID to think that it is a PDF 1.4 file and will match against that signature. Creating a file manually like this is the easiest way to generate test files and is easily incorporated into a signature development methodology because one can simply copy and paste the bytes used from an exemplar of the file format used to develop the signature.

A second approach is to generate the skeleton files automatically. The PRONOM technical documentation describes the syntax used to create file format signatures

(Brown, 2006). As well as being a syntax to create signatures, we can use it to deconstruct signatures into one or more byte sequences that can then be fed into a file, which will then become a skeleton file for the test corpus.

To illustrate, two basic signatures might look like this:

1. `CAFEBABE[01:0F]CAFED00D`
2. `BAADCAFE(0A|0D0A)BAADF00D`

The syntax in the first example describes a "sequence of bytes which lies lexicographically between `01` and `0F`," that is, a run of numerical values. If we deconstruct the signature, the bytes `CAFEBABE` and `CAFED00D` can be placed directly into a new file. The bracketed syntax needs to be parsed and a decision made in the code about outputting an integer between one and 15 between the two sequences.

In the second example, the syntax describes a "wildcard matching one from a list of values," that is, a list of optional values separated by the OR '`|`' symbol. Again, deconstructing this signature sees the sequences of `BAADCAFE` and `BAADF00D` output directly to the new file. The bracketed syntax in this case needs to be parsed and a decision made by the processing engine over which *additional* sequence to select to generate the skeleton file. We need to understand if the processing engine should output files *for each* option documented in a signature in this way.

Generating files automatically isn't recommended for each file created for the suite, as the concept of these test files is for them to be handwritten by developers of signatures, closely mirroring the work done to generate a signature, to provide a solid human-based provenance trail for each file in the suite. The recommended use of this approach is to provide a baseline for the test suite by generating the suite from the PRONOM database as it is at present and then seeking for it to be supported by handmade files in future. The automated tool should exist to allow for bulk generation of signatures where a lot of work on signatures has already been completed but without the files being created alongside. Its use should be minimized to improve the quality of the test suite and ensure its independence from the data that is used by DROID.

Structure of the File

The tool described above has been developed as a prototype to enable this paper to be written and to enable the challenges involved to be better understood.[4] Initially, it was thought to be extremely clear how skeleton files should look. Their structure should be simple, and to support the sequences in the file they should be padded by null bytes to clearly distinguish the different sections that make up a signature. Furthermore, the files created should be identifiable as their *own format.* We could give files their own magic number to identify the file type, and also create a lightweight structure inside the file to store metadata, such as the PUID for the format that the skeleton file represents.

[4] Python scripts for the Skeleton-test-suite-generator hosted on GitHub: https://github.com/exponential-decay/skeleton-test-suite-generator

Through this development, a number of issues came up which need to be considered before finalizing a suitable structure. One concern includes the multiple internal signatures that can belong to a single format in PRONOM. This means that the relationship between a PUID and a skeleton file is one-to-many (1:*); this needs to be recorded in metadata if we want self-describing files. We must also consider what to do with multiple combinations of byte-sequences. In the example above a 'Max offset' of four means that there are four other combinations of sequence (and therefore file) that can match the signature. The purest skeleton-suite would contain all variants described by a signature, but what are the practicalities of this when we get into offsets in the hundreds or thousands of byte ranges? What should we do pragmatically that gives us the best return for our effort?

Deconstruction of signatures and generation of files proved to be more complex than anticipated. As well as dealing with multiple combinations of sequence, the code for outputting the skeleton files needs to consider what to do with wildcards, how many extra bytes to pad files with, and how it fills bytes between sequences that are statically positioned (i.e. Beginning of File (BOF) and End of File (EOF) sequences), and those which are variably positioned but bounded by two offsets.

At the time of writing, v0.2-BETA[5] of the skeleton-test-suite-generator outputs only the most basic skeleton-file, i.e. files matching byte-sequences described in PRONOM 'as-is' with minimal padding between variable positioned sequences and no additional metadata. To identify individual files we adopt the naming convention '*[puid]-[pronom-signature-id].[file-extension]*'. This makes it easier to debug output and keep track of permutations of file described by multiple internal signatures.

Over the two iterations of development thus far, the focus has been on the accurate output of internal signatures as skeleton bitstreams. Further development of the skeleton-test-suite-generator will help us to understand the flexibility we have in order to include greater structure, output identifying sequences for skeleton-files and additional self-describing metadata without interfering with the essence of the files themselves.

Proposed Methodology for Organizations Contributing Signatures

Once a baseline suite exists, the proposed methodology will be for developers creating signatures to create matching skeleton files. When testing signatures it is important for the developer to test signatures against their own sample files, the skeleton file, and the complete skeleton test corpus. This will enable them to check for collisions in the work they are doing. New signatures should not match any existing file in the test suite. Should this happen, the developer should think about modifying their signature or look for changes that can be submitted for the existing signature and corresponding skeleton file causing the issue.

Once the testing has been completed, the DROID signature and skeleton file can then be submitted to The National Archives and to the organization/community controlling the test suite so it can be added to the PRONOM database *and* skeleton corpus respectively.

[5] Skeleton-test-suite-generator v0.2-BETA on GitHub: https://github.com/exponential-decay/skeleton-test-suite-generator/tree/v0.2-BETA

 doi:10.2218/ijdc.v8i1.249

For The National Archives developing signatures on a roughly bi-monthly release cycle, every attempt should be made for a skeleton file to be uploaded to the corpus for each new signature created or each signature modified. Where large numbers of signatures are created in a single release, the automated generation of these digital objects may be appropriate. However, this approach should be made clear in the provenance trail of the test suite and every attempt should be made to generate files by hand in all other cases.

Licensing the Skeleton Suite

The issues surrounding IPR, copyright and access rights expressed by Fetherston and Gollins (2012) are perhaps the biggest concerns for a developer hoping to benefit from a fully-featured corpus of digital objects. These concerns stem from the perceived lack of digital preservation researchers with legal experience. It is clear that for a full test suite to succeed, the committee or organization maintaining it needs a clearly defined **written** policy which states who owns the suite, what can be done with the suite, and how it can be reused. It must also clearly describe the process for contributing to the suite and the transfer of ownership of the files to the suite holders.

The skeleton suite being much simpler in nature makes the legal issues surrounding it easier to comprehend. In adopting the open source community as a model, we are provided with licenses that will be appropriate for this work.

As the skeleton suite should have minimal restrictions placed upon it, the proposed license under which the suite should be placed is 'Creative Commons Share Alike' (CC BY-SA)[6]. This should be supported by a simple waiver that is accepted by anyone contributing simple skeleton files to the suite, allowing the resource to then be licensed in this manner.

Summarizing various licenses under which research data can be released, the Digital Curation Centre (DCC) provides some useful guidance that allows us to be confident of the selection of CC BY-SA (Ball, 2012). Firstly, CC BY-SA *"allows others to copy, distribute, display and perform the work as long as the creator is given due credit."* Secondly, Share Alike *"inserts a strong copyleft clause into the license. This means that all derivative works must be released under the same license as the original work"*. Finally, the DCC note that: *"a quite general problem is that the licenses are aimed at homogeneous works."*

The attribution clause in CC BY-SA means that the provenance of the suite should remain strong through potentially numerous generations of the suite by many different organizations. The 'Share Alike' clause ensures that organizations are free to share and add to the suite. They are also able to release it for themselves, as long as they adopt the same license. This clause is expected to benefit the community, as it allows anyone to adopt and expand the skeleton suite for other research purposes, and to re-release it without worrying about doing so. The rest of the community can then benefit from the improved collection, provided the license is followed correctly. Further still, as we are creating large volumes of homogeneous objects – objects that do not conform to a single specification individually, but all conform to the

[6] Creative Commons CC BY-SA license text: http://creativecommons.org/licenses/by-sa/3.0/

specification of a DROID skeleton test-suite file – the latter point by the DCC should not impact the collection, and supports the adoption of the license.

DROID is a community-based product, but it is heavily controlled by The National Archives. As the National Archives is a government body, the Open Government License (OGL) should perhaps be considered for this data. The DCC summary is again comprehensive in its coverage. In summary the DCC highlight the following:

OGL is good for:

- UK public sector databases and datasets
- Data to be used automatically

Watch out for:

- Attribution stacking if used with differently licensed data
- Categories of data that cannot be licensed in this way
- Ties to the UK legal context

Given that the test suite data is not public sector information and there is no need for attribution stacking in this dataset, nor any need for the license to be tied into the UK legal context, the OGL license provides additional features that are not required by the skeleton test-suite, with no additional benefits.

Hosting the Skeleton Suite

As with licensing, we can adopt the open source model for hosting the suite. This paper is focused on the generation and development of the test suite, so it is beyond its scope to move into a comprehensive evaluation of hosting options. However, one potential host, given its adoption in the community by organizations such as The National Archives' DROID team[7] and the Open Planets Foundation (OPF), is GitHub.[8]

Referring to the Open Planets Foundation Software Development Guidelines (Tarrant, 2012), GitHub provides the following features required by the skeleton suite:

- Ability to provide software description and README files
- Ability to manage your source code
- Ability to manage branches and tags
- Ability to manage releases and packaging code
- Ability to keep track of issues relating to the code
- Ability to manage a roadmap via the issue tracker

[7] DROID GitHub pages: http://digital-preservation.github.com/droid/

[8] GitHub: https://github.com

With GitHub we can create a centralized host for this work containing a clear description about what the suite does, including licensing information. We are able to manage binary objects in the master branch, and for each release of the suite (per DROID signature file release) we are able to tag the test suite, package it, and enable it to be downloaded by the community. Provenance for the files in the suite is provided by the commit log and commit messages. A clear log must be written for each file committed; e.g. *'[new file] fmt/445 submitted by The National Archives for Signature file V62.'* This approach can also be used to track changes and updates to files. Further, the issues tracker will enable users of the suite to report errors in their findings and monitor the work being done. Supported by a Google Group for the test suite, a community could be built around it, promoting an environment for collaborative work as described by Fetherston and Gollins (2012).

Recommendations: What next?

The next steps will be to complete the tool, to create a full baseline test suite and host the results on GitHub. The code will need to be thoroughly tested and unit tests created to provide assurances that the code outputs files accurately reflecting the PRONOM signatures. The testing of digital objects will also be important, and the end result we will be expecting is a one-to-one relationship between a file and an identification in DROID. Any collisions found will require investigation to determine whether the file is inaccurate or whether the signature as it is stored in PRONOM needs additional work because it is returning false-positives.

Beyond the tool itself, the community *and* The National Archives will need to decide the best way forward to continue to use the results of this work. Incorporation of the skeleton test corpus into the DROID unit tests will be an important first step, as this will give developers extra freedom and confidence to modify code, thus giving developers and DROID users assurances that identification capabilities remain consistent. It will be important for DROID to distinguish the difference between its own license (BSD 3-clause) and that of the data used for the unit tests (i.e. the skeleton corpus), but we do not anticipate any conflicts between the two sets of work.

Once the suite has been perfected for DROID, other identification tools can then be explored and we can begin to understand whether this approach will bring similar advantages to users and developers of those tools as well.

Container formats will be difficult to create in the skeleton suite initially. While creating zip-based analogues and mirroring the structure of those files for identification by DROID will be straightforward due to the accessibility of zip software, the second container type identified by DROID, OLE2, will take additional research to understand. It is conceivable that this cannot be done without the aid of a specialist tool to create the OLE2 structure. This must be understood but should not detract from the other work we are capable of doing.

Finally, during the process of prototyping the automated tool to output byte sequences in PRONOM as skeleton files, it was noticed that small improvements could be made to the quality of the information in the PRONOM database – specifically the names assigned to 'Internal Signatures' as these are different from the

umbrella records that can contain *many* of these. Following this work, and as it becomes more widely adopted, it will be preferable to see the PRONOM team improve signature naming conventions in the PRONOM database. As should be demonstrated once the database is semantically linked,[9] using the internal signature data in this way affords the names of the signatures the same importance as the PUID and the 'Format Name.' For this work it is important that internal signature data is well-formed. If these changes are made this should also benefit the *signature-centric* nature of the work conducted by The National Archives.

Conclusion

The skeleton test suite attempts to address the challenge set by Fetherston and Gollins (2012) by combining this with the approach championed for targeted files in the DROID 6.1 development. The skeleton test suite has the ability to provide the digital preservation and curation communities with a practical interim solution, as addressing the challenge of the original paper will still take some considerable time and resource for the community to accomplish.

With a complete skeleton test suite we will be able to increase the confidence in tools that are based on PRONOM signatures. This will enable developers to incorporate digital objects as part of their testing process. Unit tests built around the test suite and incorporated into DROID can improve the confidence with which a developer can optimise and improve the DROID code-base. The solution outlined in this paper can answer questions about persistence and provenance using the hosting option available to us. We do not face the same usability concerns raised by Fetherston and Gollins, as the sample suites outputted from the prototype tool are currently under a megabyte in size. The skeleton suite can also move things forward by demonstrating a suitable licensing model that the fully-featured test suite could eventually adopt.

Other advantages may still emerge. Making the suite available as soon as possible and observing how it is used will help us to discover more about the advantages that will come of it. The process of creating the suite alone will help us to understand more about the PRONOM database and its signatures, as well as what the coverage and quality of DROID signatures currently looks like.

The need for a ground-truthed suite of digital objects called for by The National Archives and the Open Planets Foundation[10] is still very much needed. It provides the best mechanism for improving format identification techniques, and in providing exemplars of digital objects with carefully catalogued features and properties, such a suite provides a resource on top of which validation tools can be built where no validation tools exist, or improved where currently they are limited in scope. The fully-featured suite will provide the best mechanism for unambiguous discussion amongst the community about how tools have been developed and tested – allowing all users and developers the ability to test and verify results for themselves.

[9] The National Archives Labs blog post describing Linked Data PRONOM: http://labs.nationalarchives.gov.uk/wordpress/index.php/2011/01/linked-data-and-pronom/

[10] Calls for a test-suite of ground-truthed or 'trusted' files by Dirk von Suchodoletz: http://openplanetsfoundation.org/blogs/2011-02-17-call-test-set-files and Jay Gattuso: http://openplanetsfoundation.org/comment/100#comment-100 of the Open Planets Foundation.

Transparency and communication are perhaps two of the most important things we can strive for in digital preservation to prevent expensive research from being duplicated and to create a trust between different parts of the community. Fetherston and Gollins conclude by stating that the development of the fully-featured test suite will provide an 'environment for collaborative work within this field.' It is hoped that the action of developing the skeleton test corpus of digital objects demonstrates an act of collaboration that provides the community with a platform (or platforms) on which it can better discuss the generation of a fully-featured test suite, while benefiting from the interim advantages that have already been discussed.

Acknowledgements

The author would like to thank The National Archives for their role in providing the background for this research; the DROID 6.1 development team, specifically Richard Brennan, for introducing the approach of developing skeleton formats as part of his unit testing work; and Richard Brennan and Nicola Welch (Digital Archivist at The National Archives) for their role in helping to better define the purpose and outcomes of such a test suite.

References

Ball, A. (2012). How to license research data. *DCC How-to Guides*. Retrieved from http://www.dcc.ac.uk/resources/how-guides/license-research-data

Brown A. (2006). Automatic format identification using PRONOM and DROID. *The National Archives Digital Preservation Technical Paper: 1*. Retrieved from http://www.nationalarchives.gov.uk/aboutapps/fileformat/pdf/automatic_format_identification.pdf

Fetherston, A. & Gollins, T. (2012). Towards the development of a test corpus of digital objects for the evaluation of file format identification tools and signatures. *International Journal of Digital Curation, 7*(2), 16-26. doi:10.2218/ijdc.v7i1.211

Tarrant, D. (2012). Software development guidelines. *Open Planets Foundation Knowledge Base Wiki*. Retrieved from http://wiki.opf-labs.org/display/PT/Software+Development+Guidelines