Skeleton Test Suite: Testing Results
Software Version: skeleton-suite-generator-v0.2-BETA by Ross Spencer

This reports follows the first results of testing Skeleton-suite-generator-v0.1.

The results discussed here are based on updated versions of the tool and DROID signature file:

- skeleton-test-suite-generator v0.2-BETA
- skeleton-suite-v0.2-BETA
- DROID 6.1 was used alongside Signature File v65

As of 25 October 2012, PRONOM is on Signature file v65. The database contains **934 records** which can be exported. 528 of these contain one or more internal signatures, meaning **528 discrete formats** that can be identified using DROID standard signatures. **661 files** are created by the tool; taking into account formats with multiple internal signatures, e.g. MPEG 1/2 Audio Layer 3 (MP3). The tool takes between approximately 20 and 50 seconds to create these files using the default settings.

Before discussing the output of the second version of the Skeleton-suite-generator I will briefly outline changes made since the previous report:

Attribution from The National Archives

Following the previous report The National Archives were able to take the results and correct a number of issues in PRONOM. These issues were fixed for signature file v65. The changes made to the PRONOM database are as follows.

- fmt/13: Portable Network Graphics 1.2. Gave explicit priority over fmt/11 Portable Network Graphics 1.0 to remove multiple identification conflict.

- fmt/51: Rich Text Format 1.6. Deprecated in favour of fmt/50 Rich Text Format 1.5-1.6 to remove multiple identification conflict.

- fmt/92: Scalable Vector Graphics 1.1. Gave explicit priority over fmt/91 Scalable Vector Graphics 1.0 to remove multiple identification conflict.

- fmt/142: Waveform Audio (WAVEFORMATEX). Modified signature to remove multiple identification conflict with fmt/141 Waveform Audio (PCMWAVEFORMAT).

- fmt/381: FoxPro Project. Gave explicit priority over fmt/373 FoxPro Database (2.x) to remove multiple identification conflict.

- fmt/436: Digital Negative Format (DNG) 1.0. Gave explicit priority over fmt/353 Tagged Image File Format to remove multiple identification conflict.

- fmt/441: Windows Media Video 9 Advanced Profile (WVC1). Gave explicit priority over fmt/131 Advanced Systems Format to remove multiple identification conflict.

- x-fmt/135: Audio Interchange File Format 1.3. Deprecated in favour of fmt/414 Audio Interchange File Format 1.2 to remove multiple identification conflict.

- x-fmt/212: Lotus 1-2-3 (5.0). Deprecated in favour of x-fmt/116 Lotus 1-2-3 (4-5) to remove multiple identification conflict.

- x-fmt/452: SketchUp Document. Deprecated in favour of x-fmt/451 SketchUp Document to remove multiple identification conflict.

This accounts for a large number of the issues I highlighted in the previous report and goes some way to stabilising the PRONOM database.

The National Archives made important changes to the identification of RTF files (Rich Text Format) in the previous release. Organising two records, creating RTF 1.0 – 1.4 and RTF 1.5 – 1.6 and merging the discrete records e.g. 1.2, 1.3 etc. they have removed the cause of a large number of multiple identifications users often found when using the tool.

Open Document based formats

In the previous report I highlighted the non-identification of Open Document based formats. Open Document Text/Presentation/Spreadsheet/Database. It was recognised that a limitation of the skeleton-suite-generator currently is inability to package OLE2 and ZIP files, such as those which characterize Open Office formats.

A concern for those testing with the skeleton-suite is whether the standard identification engine can identify non-container signatures provided for container-based formats as we are testing the integrity of the *standard identification engine* and its ability to match files described by its own regular expression syntax. While the DROID standard configuration returns a non-identification; modification of the container signature file to remove Open Document based formats results in all of these files being identified correctly with standard signatures.

The only question up for debate is whether DROID should, on non-identification through container-signatures, attempt the identification of a file through the standard identification engine and signature file.

v0.2-BETA: Testing and Results

In describing the results and the output of the tool we discuss sequence positions using the following abbreviations:

- BOF: Beginning of file
- EOF: End of file
- VAR: Variable

Currently the tool outputs error messages where it may encounter issues. These messages are split into INFO messages and WARNING messages. INFO messages describe where I believe the output should still be correct despite working around issues in the code. WARNING messages describe where skeleton files output should be treated with caution when looking at test results - the hex sequence in these skeleton files might want to be checked manually.

An aim of the project will be to pre-process signatures where these warnings currently appear. This is to handle them intelligently so that *all* files described by DROID signatures are output correctly. The current warnings and results are described below:

Attempting to write BOF with BOF written. Attempting to correct: offset > current BOF file pointer

WARNING: (fmt/363)	Attempting to write BOF with BOF written. Attempting to correct: offset > current BOF file pointer...
WARNING: (x-fmt/387)	
WARNING: (x-fmt/388)	
WARNING: (x-fmt/399)	

Each of these files are written and identified correctly.

Attempting to write BOF with BOF written. Attempting to correct: offset == zero so writing after

WARNING: (fmt/189)	Attempting to write BOF with BOF written. Attempting to correct: offset == zero so writing after...
WARNING: (fmt/358)	
WARNING: (x-fmt/388)	

Fmt/189 is not identified. The other formats all seem to be written correctly. Pre-processing of the fmt/189 byte sequences to re-order will allow this file to be generated correctly.

Attempting to write VAR with VAR written

WARNING: (fmt/39)	Attempting to write VAR with VAR written.
WARNING: (fmt/40)	
WARNING: (fmt/125)	

WARNING: (x-fmt/88)	
WARNING: (x-fmt/430)	

Fmt/39, fmt/40 and x-fmt/430 are identified as fmt/111 in DROID 6 variants, however, these are identified correctly in DROID 4.0 and 5.0. This is because of the container handling in DROID. Fmt/111 describes the generic OLE2 container.

These files shouldn't be identified using DROID container signatures as they are not valid OLE2 files, however, they do match the standard signatures described in PRONOM when the container identification mechanism is removed from DROID. As with Open Document based formats described above the only question up for debate is whether DROID should, on non-identification through container-signatures, attempt the identification of a file through the standard identification engine and signature file.

NOTE: This is only a pertinent question while The National Archives continues to commit to supporting DROID 4.0 and 5.0. Should it commit to DROID 6 onwards it could afford to remove the redundancy associated with standard and container signatures that conflict with each other in the most recent versions of the tool.

Attempting to write BOF with EOF written and **Attempting to write VAR with EOF written**

INFO: (fmt/134)	Attempting to write BOF with EOF written.
INFO: (fmt/161)	Attempting to write VAR with EOF written.

Finally, the last messages reported from the skeleton-suite-generator - both formats output here are written and identified correctly.

Multiple Identifications returned from skeleton-suite

Fmt 437/438 returns the duplicate identification of fmt/353 – Digital Negative (DNG) and TIFF. This issue is known by The National Archives, and relates to the priorities given to DNG and TIFF where TIFF will always be a subsequence of the DNG signature.

Fmt/60 is identified as fmt/59 – Excel 95 and 5.0 workbook. These both share the same signature, the priorities ensure that one result is returned, in this case we expect fmt/60 but see fmt/59. This might indicate some housekeeping that needs to be done on the PRONOM database to remove the redundancy or to complete additional work necessary to identify what makes either of these two formats distinct from one another.

Skeleton files not identified by DROID

Fmt/435 – Drawing Interchange File Format. It is unclear why this isn't identified by DROID as manual

checking shows the byte sequences to be consistent. This file isn't identified in either DROID 4.0 variants or DROID 6.0. I have completed an analysis of the bit-stream and logged this issue in GitHub for The National Archives to investigate further.

X-fmt/412 – Java Archive File (JAR). This file will identify in DROID 4.0 variants but not DROID 6.1. This indicates something wrong with the pattern matching in this instance and so should be investigated further. I have completed an analysis of the bit-stream and logged this issue in GitHub for The National Archives to investigate further.

In both cases logged above it is recognised that we're working with 'artificial' data and so it might be important to find 'real' files where DROID is also returning a null identification. It is also recognised, however, that on a purely technical level it is important that the DROID regular expression engine works on all bit-streams that a regular-expression pattern *should* match. This may prevent issues for users further down the line and therefore should be treated on the same level as any other file.

Conclusion

Having improved the code base and ironed out a handful of issues writing skeleton files we're now beginning to see much better coverage and more reliable results. The National Archives have dealt with most of the previously identified issues which means that the we can identify more deeply-set issues in PRONOM and DROID much easier.

A question I raise above is whether DROID should return standard identifications, where available, for container-based formats that haven't been identified through the DROID 6 container identification mechanism. What could this approach tell us?

- We have a clue that we have files that belong to a particular format family?
- We have potentially corrupt or truncated bit streams?
- Nothing at all?!

While both signatures exist I feel that there is additional information to be gleamed from a fall back identification such as this, however, there is an effort trade-off to be considered as I expect container signatures to become de-facto standard as the current generation of the DROID tool continues to improve therefore allowing The National Archives to remove standard signatures used only in older generations of DROID from PRONOM.

During this work we've also identified two files which might demonstrate an important bug or bugs in the DROID identification engine. Certainly two files that warrant further investigation and understanding. It is paramount to to those using DROID that files are identified accurately against the signatures described in the

PRONOM database.

Following the development of v0.2-BETA of the skeleton-suite-generator, the tool can now output **645** out of **661** described files correctly. This gives us access to skeleton-files matching **97.6%** of the *standard* signatures from the PRONOM database.

This tool has demonstrated its value through a single iteration alone but the long-term value of it still needs to be understood. This will depend on community feedback and how effectively this tool can be used to generate skeleton-files, and then, how conveniently these files can be incorporated into testing procedures, and the results of those testing procedures analysed.

Work now needs to be done to output files based on standard DROID signatures with 100% reliability. This then needs to be followed up by incorporating methods of outputting skeleton-files for container-based formats. This is complex and relies on software engines capable of writing ZIP- and OLE2-based files. The community requirement for the skeleton-test-suite will dictate the amount of effort that should be put into this.

Once the tool is stabilized completely, or perhaps sometime before, we can then move on to discussing other issues raised in the first results paper such as hosting a skeleton-suite online and how users should move forward adding *manually* created skeleton-files for each new DROID signature created.