Writing Binary By Hand

An introduction to binary file formats

|  |
| --- |
| **Martin Hoppenheit** |
| *Landesarchiv Nordrhein-Westfalen**Germany**martin.hoppenheit@lav.nrw.de* |
|  |

**Abstract – In this tutorial participants will learn how to approach (i.e., read) a file format specification and based on this specification create binary files by hand. This is useful when learning about, researching and experimenting with file formats as well as to create test or example files.**

**Keywords – file format, binary**

**Conference Topics – innovation; resilience**

# Introduction

Binary file formats are often believed to be complicated, inapproachable, and somewhat arcane. This is because, other than text-based formats, they require specialized, sometimes proprietary viewers and editors to interpret their structure and content.

While it is possible to open any binary file in a hex editor, the vast sequence of hex characters that it throws at its users does not always help to establish understanding – hex-encoded gibberish is still gibberish.

This tutorial follows another approach: Instead of looking at existing binary files, participants will create them from scratch. Starting from a file format specification, they will write small example files by hand, thus gaining a practical understanding of the file format and a basis for further exploration and experiments.

# Tutorial Overview

After a short introduction explaining the basics of binary files, hexadecimal notation and the tools used in the tutorial, participants will be pointed at the essential parts of the TIFF file format specification [1] and create a minimal TIFF file along the way. After that, they will be able to adapt and extend their example files diving deeper into the TIFF specification. Alternatively, they may move on to other format specifications and create other example files themselves.

Although binary files will be created writing hex code (like in a hex editor), their readability will be greatly improved using a notation and tool called Literate Binary [2]. This notation allows combining binary (hex) and textual content in a single text file (Markdown, to be precise) from which both a binary file and corresponding documentation in formats like HTML or PDF can be generated. This is particularly useful when documenting example files.

# Goals

Participants will realize that file format specifications can be surprisingly readable and there is no need to shy away from them. Reading a format specification is useful not only to create example files but also to understand error messages from file format validation.

Participants will learn to create binary files by hand. This helps when learning about a file format, but it also allows reproducing problems in the form of minimal examples (without additional content blurring the crucial aspects or violating copyright) or providing test files for file format identification and validation tools.

# Intended Audience

Everybody can participate; there are no strict prerequisites. Programming skills are not required. It helps though if participants have a basic understanding of hexadecimal notation (e.g., as described in [3] or from working with PRONOM signatures) and are not afraid of occasional command line use. To get most out of the tutorial participants should bring a laptop with their favorite text editor and the Literate Binary tool installed.

# REFERENCES

1. TIFF 6.0 Specification, Adobe Developers Association. https://www.adobe.io/content/dam/udp/en/open/standards/tiff/TIFF6.pdf
2. Literate Binary. https://github.com/marhop/literate-binary
3. Representing binary data, Martin Hoppenheit. https://martin.hoppenheit.info/blog/2017/representing-binary-data/