# Introduction

The Metadata Manager is a utility designed for Records Clerks managing custom metadata properties embedded files formatted as .pdf documents. The utility displays existing custom metadata properties in a grid control and allows editing of individual metadata properties stored within selected file or files. Extensible Metadata Platform (XMP) templates are supported and used for bulk modification of selected files using data stored in Excel. Basic reporting features that exports properties used for Destruction Logs and Record Inventories to Excel.

## Scope

|  |  |
| --- | --- |
| **In Scope** | **Out of Scope** |
| Windows Forms Application | There will be no web based interface or mobile access |
| Persistent data stored as embedded metadata within Adobe PDF documents. | There will be no persistent data stored in an external database. |
| Display and Edit metadata embedded within pdf files in a grid format. | A representative set of metadata was chosen for this project. I leave it to others to define the agency standard metadata. |
| Supported PDF format: PDF/A  ISO 19005-1  Format designed for long-term preservation | The PDFs produced are not guaranteed to conform to other specifications. These formats add additional accessibility features that may (or may not) be useful to support in the future. |
| Open, Display, Edit and Export existing XMP templates. | Although XMP templates are standard and can be used by many formats, only .pdf files will be supported at this time. |
| Metadata is extracted to an Excel File | There is no ability to analyze or filter data extracts. You get what you get and the intention is to do manipulations (sort, filter, group, etc.) using Excel |
|  |  |

## Audience

This document is targeted to three audiences. This is my capstone project which will be presented to Jesse Harlen of Centralia College who is the instructor for the course. The utility was written to help organize archival electronic records, and I present this to my record manager colleagues for review. I hope you find it useful. The third audience is my compatriots in information technology and my goal is to present useful system documentation and source code in order to secure their blessings.

## Challenges

This project was full of firsts: my first experience with a Windows form application; a deep dive into metadata standards; understanding Adobe’s implementation of metadata; and finding and learning iText which is an open source library for manipulating PDFs. I have developed new technical skills, a better understanding of metadata implementation, and gained an understanding of published standards used by records managers and archivists including Dublin Core. Fun stuff.

## Operating Environment

The Metadata Manager will run as a Windows Forms Application on an average Windows 10 PC.

## Security

This utility is intended to manage metadata embedded in Public Records and will run under the context of the currently logged in windows user. It is intended to manage public records which is category 1 data. However there is no technical reason the utility will not work on documents containing private data. Consider adding “privacy level” as Metadata on the record – what would the default be?

## Development Environment

This project is built in C# using Visual Studio 2022 and .net 6. Additional third party libraries were used to manage Acrobat and Excel files.

**Adobe Acrobat SDK 2021** : <https://opensource.adobe.com/dc-acrobat-sdk-docs/acrobatsdk/>

**iText7 v.7.1.8 :** <https://itextpdf.com/en>

Core 7.1 package: <https://www.nuget.org/packages/itext7>

API Reference : <https://api.itextpdf.com/iText7/dotnet/7.1.8/>

**GitHub :** <https://github.com/PowellAndrea/Capstone>

## XMP Reference

Adobe’s Extensible Metadata Platform (XMP) is a file labeling technology that enables the embedding of metadata into the digital files themselves. XMP is open source and an ISO standard(16684-1) that many application developers are now supporting.

**XMP Specification & SDK :** <https://www.adobe.com/devnet/xmp.html>

## Dublin Core Reference

**Adobe’s XMP implementation of Dublin Core**<https://developer.adobe.com/xmp/docs/XMPNamespaces/dc/>

Dublin Core is the most common metadata schema for web content. The specification is maintained by the Dublin Core Metadata Initiative and includes properties, vocabulary, schemes, syntax encoding schemes, and classes.

<https://www.dublincore.org/specifications/dublin-core/dcmi-terms/>

|  |  |
| --- | --- |
| **Namespace** | **Schema** |
| <http://purl.org/dc/elements/1.1/> | <http://dublincore.org/schemas/xmls/qdc/dc.xsd> |
| <http://purl.org/dc/terms/> | <http://dublincore.org/schemas/xmls/qdc/dcterms.xsd> |
| <http://purl.org/dc/dcmitype/> | <http://dublincore.org/schemas/xmls/qdc/dcmitype.xsd> |

# Features

1. Management of metadata properties embedded in .pdf records. (phase 1) [(AD: 1)](#_Activity_:_Edit)
   1. Display pre-existing metadata properties from selected .pdfs within a grid control [(UC: 1.1)](#_UC_1.1_–)
   2. Ability to modify metadata properties within a grid control for a single file
   3. Ability for the user to select multiple .pdf files for editing
   4. Ability to apply bulk metadata to user selected .pdf files
2. Reports (phase 2)
   1. Export embedded metadata fields to an Excel file
   2. (Future) Produce formatted Destruction Request Forms as pdf files
   3. (Future) Formatted Archival Transmittal Forms as pdf files
3. Management XMP Templates (phase 3)
   1. Apply existing template metadata to the Grid Control
   2. Modify pre-existing properties from XMP templates
   3. Ability to update multiple custom metadata properties in selected .pdf files by applying an XMP template. Properties that do not exist are created.
   4. Export modified metadata properties as XMP formatted template
   5. Export modified metadata properties as user readable property tree formatted for Excel

# Selected Metadata

A minimal set of metadata properties has been selected for testing proof of concept. Adobe supports Dublin Core metadata. The selected metadata includes Dublin Core and custom retention metadata.

Future implementations may support the MARC formats which are standards for the representation and communication of bibliographic and related information in machine-readable form which are used by the Digital Archives and Washington State Library.

Graphical user interface, text, application

Description automatically generated

# Classes

## Record – as of April 19, 2022

/// Record object uuid

internal Guid FileId;

/// PDF Internal version ID xmpMM:InstanceID / xmpMM:DocumentID

public string PdfInstanceId;

public string FilePath;

public string FileName;

public string FileSize; // System managed

/// Dublin Core 1.1 Namespace (DCMI)

/// https://developer.adobe.com/xmp/docs/XMPNamespaces/dc/

public string Title; //dc:title

public string Author; //dc:creator

public string Description; //dc:description

/// Pdfx 1.3 namespace - Custom Metdata

public string RecordSeries; //pdfx:RecordSeries

public string YearStart; //pdfx:YearStart

public string YearEnd; //pdfx:YearEnd

public string Published; //pdfx:Published

/// XMP Rights Management namespace --

/// xmpRights:Marked = False for Public Records

public string CopyrightNotice; //xmpRights:Marked = False

Author – not yet implemented, as of April 19, 2022

Andrea Notes: Several options – figure out how DCMI is storing multiple authors.

Guid AuthorId = Guid.NewGuid();

private string AuthorFirstName;

private string AuthorLastName;

## Class Relationships

Andrea Note: Retention objects not yet implemented

Diagram

Description automatically generated

# Use Case Diagrams

Andrea Note: This is more like 3 use cases – split up into different graphics?

## UC 1.1 – Updating Metadata embedded in .PDF files

See Activity Diagram [(AD: 1)](#_Activity_:_Edit)

Record Manager needs the ability to display existing metadata in a grid view in order to easily view and manage properties embedded in .pdf files stored in the local file system.

Diagram

Description automatically generated

# Activity Diagrams

Andrea Note: Not a lot of user feedback at this point. Add fail path.

## Activity : Edit Metadata(AD:1)

Diagram

Description automatically generated

# Reports

Not yet implemented. Planning for metadata extracted as .csv, xlsx, or pdf formatted document. See references reports below for an example of data included in forms for the State Archives.

Andrea Task Reports : .csv report should be trivial;

Andrea Task: Reports : Excel reports moderate effort

Andrea Task: Reports: forms are epics that will not be implemented at this time.

# Reference Forms

The ability to create forms is out of scope for the current project, however the example forms are included here as reference to required Archival metadata properties used by these reports.

## Public Records Destruction Log

Table

Description automatically generated

## Digital Records Inventory

The record inventory form will be similar to the format used by the Digital Archives Transfer. In the future, the Metadata Manager will produce this type of report for both internal record inventory and digital archive transfer inventory sheets.

Text

Description automatically generated with medium confidence

# Project Management

**Change Log**

|  |  |
| --- | --- |
| 2022-04-24 | Code cleanup, update documentation |
| 2022-04-19 | **Milestone:** Can open file; edit metadata in grid control; save updated metadata to a copy of the original file |
| 2022-04-18 | Moved variables into classes; updated columns in dataGridView to use selected metadata set |
| 2022-04-16 | Implemented iText7; can read from existing file & create a new file with the copied metadata. |
|  | Enabled multi-select; resize File Name column to fit & freeze full name; clear dataGridView before openFileDialog runs. |
| 2022-04-01 | First upload with documentation, Ability to run window form application, select file from windows openFileDialog, display the file name and path in a dataGridView control, and exit form. |

**Parking lot of the Titans**

* Reports – not yet implemented
* XMP Templates – not yet implemented
* Forms – not planned

**Backlog**

* Pop out thumbnail window (from data grid)
* Display properties in a tree based on namespace, possibly from popup window.

**Future Notes**

[Andrea Notes: Several options – figure out how DCMI is storing multiple authors. 6](#_Toc101700328)

[Andrea Note: Retention objects not yet implemented 7](#_Toc101700329)

[Andrea Note: The use cases diagrams are not great – need to update & add to 8](#_Toc101700330)

[Andrea Note: Not a lot of user feedback at this point. Add fail path. 9](#_Toc101700331)

[Andrea Task Reports : .csv report should be trivial; 10](#_Toc101700332)

[Andrea Task: Reports : Excel reports moderate effort 10](#_Toc101700333)

[Andrea Task: Reports: forms are epics that will not be implemented at this time. 10](#_Toc101700334)