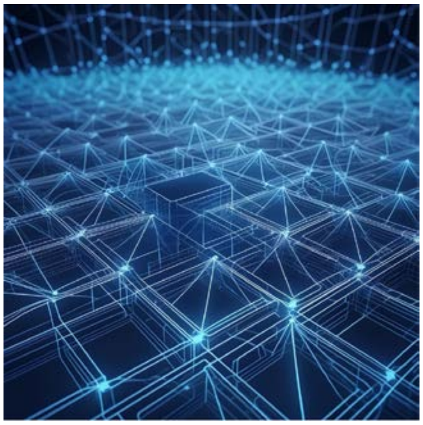
Specification

EA-PDF

An archival email format based on PDF/A

Version 1.0
2025-02



EA-PDF LWG

The research and development of this document was supported by a grant from the Institute of Museum and Library Services administered by the University of Illinois Urbana-Champaign.

The views, findings, conclusions or recommendations expressed in this publication do not necessarily represent those of the Institute of Museum and Library Services or the University of Illinois Urbana-Champaign.

PDF Association Inc.

E-mail: copyright@pdfa.org

Web: https://www.pdfa.org

Vendors own their respective copyrights and trademarks wherever they are mentioned. Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The PDF Association shall not be held responsible for identifying any or all such patent rights. Users of this document are responsible for independently verifying any and all information. Any mention of companies or products does not imply endorsement or support of any of the mentioned information, services, products, or providers.

Table of Contents

Table of Tables

List of Examples

List of Figures

1 Introduction

This document defines a specification and best practice for a vendor- and platform-neutral archival email format based on PDF/A known as EA-PDF for the long-term preservation of email. This file format is compatible with either PDF/A-3 (ISO 19005-3, *PDF 1.7*) or PDF/A-4 (ISO/DIS 19005-4 dated revision, *PDF 2.0*). In all cases, a valid EA-PDF file is always a valid PDF according to ISO 32000.

The primary audience of this document are preservationists with an understanding of the needs and use-cases for preserving email, but with limited PDF technical knowledge. This document is thus structured differently from purely technical ISO PDF subset standards and includes best practice, background, and pragmatic field experience information to directly support end users of EA-PDF and developers with expertise across both email and PDF technologies. This document records the reasoning behind design choices or limitations to ensure EA-PDF can meet the goals of end users now (with current generation PDF software) and into the future (with EA-PDF aware software).

2 Document conventions

This document is written for preservationists, many of whom are not PDF technologists. Thus, the structure, layout, and style of writing differs from a traditional ISO PDF subset standard to support end users of the format.

- **Bold** words indicate PDF dictionary key names (just like in other PDF ISO specifications). PDF keys are always case-sensitive. The leading SOLIDUS "/" is not used by convention in PDF specification documents or this document but is always required in real PDF files.
- *Italic* words indicate PDF key values (just like in other PDF ISO specifications). Sometimes a key value is the same as a key name and the formatting differences can help distinguish this usage.
- Blue italic *formatting* indicates a specific EA-PDF "term-of-art" that is defined and used consistently through this document.
- The uppercase word "SHALL" indicates a mandatory technical requirement. It is always required to be done.
- The uppercase word "SHOULD" indicates a recommendation. Whatever is stated is not mandatory but is recommended. There are no gradations, level of importance, or hierarchy associated to different "should" statements. A recommendation may be because of a technical reason (e.g., a corner case), because technical file format validation would require comparing with original source email assets, because it is "best practice", or that it might enable a better experience with non-EA-PDF aware legacy software.
- The phrase "*strongly recommended*" is used in this document to indicate something that is very important, but there may be a corner case which means the use of "SHALL" is not technically possible. These statements are intended as being *more important* than "SHOULD" statements.

  Note: Ignoring these statements in other than the corner case is to the potential detriment to the goals and end users of EA-PDF!
- "Can" and "may" are permissive statements, meaning that something is allowed.
- "Will" and "are" indicate factual statements, possibly because one or more other mandated requirements are necessary (ipso facto).
- Many ISO PDF specifications do not mention features unless they are prohibited or constrained, whereas this document may reference PDF features that can be utilized by EA-PDF software but for which there are no associated formal "SHALL" requirements or "SHOULD" recommendations.
- "Note" statements are intended as explanation. These occur *below* the relevant statements. They are meant to assist the reader and do not add any additional formal requirements.

- "Reason" statements provide an explanation behind each technical statement. These occur below the relevant statements. They are meant to assist the reader and do not add any additional formal requirements.
- "EA-PDF Writer", "EA-PDF Creator" and "EA-PDF Reader" statements provide information and hints to implementers, possibly about details found in other PDF specifications or nuances of features. These occur *below* relevant statements.
- Footnotes are also used to provide additional contextual information for those less versed in the details of PDF technology. They are meant to assist the reader and do not add any additional formal requirements.
- EA-PDF profile identifiers are all lowercase letters and occur after the `PDF/mail-1` moniker. A set-like syntax using curly braces, such as `PDF/mail-1 {s, m}`, is used to indicate when multiple profiles are being referenced. This example is equivalent to "`PDF/mail-1s` and `PDF/mail-1m`". If no profile identifiers are stated (as in `PDF/mail-1`), then all profiles are being referenced.

3 Background

This section provides a brief discussion on the background and context of the EA-PDF v1.0 file format as solicited via the EA-PDF Liaison Working Group (LWG). It does not contain the formal PDF technical requirements; however it needs to be read by all users of this document as it explains how EA-PDF achieves its goals and the agreed decisions made by the LWG.

> *Note: this section is descriptive in nature and there are no "SHALL" or "SHOULD" statements. Use of the words "must" or "will" in this section implies something that is mandated without prescribing exactly how the mandate is technically achieved.*

The foremost goal of EA-PDF is as a faithful and verifiable long-term preservation format for email assets, including a reliable static appearance of a static rendition of the email that is technically achieved by utilizing PDF/A-3 or PDF/A-4. This requires a reliable and verifiable link between the original raw source email asset(s) and the PDF representation of those emails. EA-PDF also supports "isolated" use-cases where this linkage between original raw source email asset(s) and the PDF content is weakened or broken – for example due to the use of non-EA-PDF aware software or because of workflow requirements (such as FOIA).

Because EA-PDF is a flexible format, it is fully expected that the application of EA-PDF in each organization is supported in context by archival policies and procedures and will evolve as EA-PDF aware software becomes available.

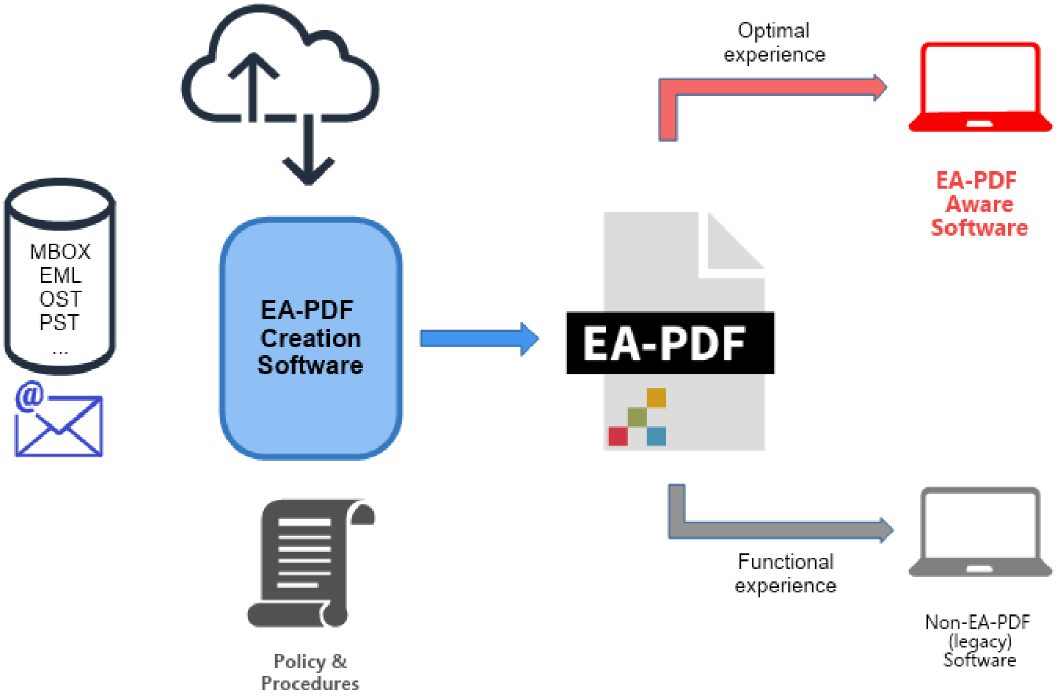


Figure 1: Context of EA-PDF

The EA-PDF v1.0 file format has been defined such that it supports two main usage contexts, as shown in *Figure 1: Context of EA-PDF* above:

1. An **Optimal experience** when using EA-PDF aware software, regardless of whether this software is interactive (e.g., viewer GUI) or non-interactive (e.g., batch processing, web server);
2. A **Functional Experience** with the potential for a limited or less-than-optimal experience when using non-EA-PDF aware legacy software. Software not written to support EA-PDF or PDF/A will work but may result in restrictions on viewing, search, semantics, and/or navigation due to lack of specific EA-PDF knowledge. Although this experience will vary across implementations, various EA-PDF v1.0 file format design choices have been made by the LWG to try and ensure the widest possible support with current non-EA-PDF aware legacy software (such as non-PDF/A conformant software).

   Reason: at the time of writing a lot of current generation PDF software only provides limited support for many PDF 2.0 features and for compliant PDF/A rendering. This is a major consideration in the technical design of EA-PDF v1.0.

The term “EA-PDF” is used as an all-encompassing term and includes the file format, software, use-cases/scenarios, etc. The moniker “`PDF/mail`” represents just the file format defined by this industry specification and any later editions. The term “`PDF/mail-1`” refers to this first edition of `PDF/mail` - future versions may use other versions (such as `PDF/mail-2`, `PDF/mail-3`, etc.) in a similar manner to the way that ISO subsets are versioned. If this industry specification is ever standardized by ISO, “`PDF/M`” would be the most likely equivalent ISO moniker following the current principles of PDF naming conventions (cf. “`PDF/raster`” and “`PDF/R`”).

Reason: “PDF/E” cannot be used as the abbreviation for email, as this is already used by an existing ISO standard for engineering: “ISO 24517-1:2008 Document management — Engineering document format using PDF — Part 1: Use of PDF 1.6 (PDF/E-1)”.

Creation of EA-PDF files from email-specific formats such as EML, MSG, MBOX, Lotus Notes (`.nsf`), or Microsoft OST/PST requires specialized *EA-PDF Creation Software*, as shown above – current PDF/A creation software is insufficient. Many design and implementation choices of EA-PDF software are intentionally *not* explicitly defined by this document to allow flexibility and innovation from implementers (e.g., how to convert email bodies to PDF content streams is not explicitly defined; how to cope with errors in emails; additional content sets; etc.).

Note: requirements on specific (visual) appearance of EA-PDF files are avoided and only restricted via PDF/A (and the assumption that PDF/A conforming software will be used).

Note: this document does try to provide some informative high-level guidance and background explanation for software developers for certain features, even if these features are not formally required (“SHALL”) or recommended (“SHOULD”). These are merely guidance, but their use can provide better context and a better user experience for users of EA-PDF files.

Additional user interface requirements for EA-PDF aware software (beyond those already required by PDF/A) are also not explicitly prescribed but stated only in terms of high-level functional requirements (for example, providing access to all the embedded files in `PDF/mail` files). This is identical to the way ISO 19005 PDF/A conforming processor requirements are defined. EA-PDF does not have any *interactive* processor specific requirements as it is assumed that headless server-based ingestion systems have similar needs to individual users.

`PDF/mail` files are *always* conforming PDF/A files and will therefore *always* pass PDF/A validation. The additional requirements defined in EA-PDF enable both humans and automated systems (i.e., EA-PDF aware software) to share a common understanding of the significant properties of faithfully preserved email and their static renderings – and to perform additional validation but without requiring reprocessing of original source email assets for that validation. Note that non-EA-PDF aware legacy software (including PDF/A conforming software) may not achieve some levels of support due to the lack of specific EA-PDF feature awareness.

This document uses notes in addition to requirements or recommendations for EA-PDF developers to guide them on practices that may assist with improved user experience with non-EA-PDF aware legacy software.

EA-PDF is designed to support many differing preservation-oriented workflows. EA-PDF files created to support Freedom of Information (FOIA) and similar access requests may not be suitable for long-term preservation due to either the absence of the source email asset(s), redaction of original text, or metadata scrubbing conducted to support dissemination[1]. As described later in this document (see section *7.2 Preserving source email assets*), EA-PDF compliant files can include the original raw source email assets, along with extensive metadata, and other necessary information which would need to be redacted resulting in a PDF that is no longer a verifiable preservation asset *and* a reliable rendition of email *and* free of PII. These PDFs will still contain many EA-PDF features (such as email-specific semantic tagging or rich email metadata) but because they are modified, they are no longer a full and faithful preservation of the original raw source email assets and must be specifically indicated as an “isolated” EA-PDF file.

The provenance metadata information in PDF/A can be used to indicate the conversion from one EA-PDF profile to another, or back to PDF/A. This is an intentional design decision to ensure that EA-PDF files that are a faithful and verifiable preservation of the original raw source email assets are cleared distinguishable and identifiable as such.

[1] A modification use-case discussed in the EA-PDF LWG is the extraction of selected pages from an EA-PDF file while retaining some limited (possibly semi-redacted) context (e.g. the username portion of an email address is redacted, but not the email provider: `fred@email.com` becomes `xxxx@email.com`). The resultant PDF/A file is no longer a full and faithful preservation of the original raw source email and thus must be identified differently.

Although the PDF and PDF/A specifications, this document, and related examples all use English content, EA-PDF fully supports non-English and multi-lingual emails, without resorting to rendering emails to bitmaps.

3.1 EA-PDF Profile identifiers

This document describes multiple EA-PDF profiles[2] to meet the various requirements discussed in the LWG that reflected specific email archival scenarios and use-cases. All profiles are compliant with PDF/A and use one or more lowercase letters to indicate conformance:

EA-PDF Profile identifier (*lowercase letter*)	**Meaning**
`c`	A structured **container** that can include hierarchical folders of emails.
`i`	**Isolated** – this means that the faithful original raw source email assets are missing, and the EA-PDF file is not a verifiable preservation asset.
`m`	**Multiple** emails without folders
`s`	**Single** email message (without a folder)

Table 1: EA-PDF profile indicators

EA-PDF profiles <u>*without*</u> the "isolated" profile identifier `i` are suitable as verifiable preservation assets as these files contain the original raw source email assets as well as PDF/A compliant renderings of the emails and related context.

EA-PDF profiles <u>*with*</u> the "isolated" profile identifier `i` may be unsuitable as preservation assets as these files do <u>*not*</u> contain all the original raw source email assets, or possibly contain modified or redacted versions of those original raw source email assets. However, "isolated" EA-PDF profiles may be smaller files as a result. Thus, the suitability of "isolated" EA-PDF profiles for preservation will depend on the precise archival and preservation strategies and policies that provide the context for the application of EA-PDF as shown in *Figure 1: Context of EA-PDF*.

[2] The term "*profile*" is specifically chosen to avoid confusion or overlap with the existing "*conformance levels*" used by PDF/A and other PDF ISO subset standards.

3.2 EA-PDF Profiles

PDF/mail-1 Profile	PDF/A Conformance Levels	Description
PDF/mail-1s **“Single”**	PDF/A-3a PDF/A-3u PDF/A-4f PDF/A-4e [3]	**Single**: A single email message preserved as a single EA-PDF file, where pages in the PDF are a visual representation of only that email’s content or context. The original raw source email asset(s) are always embedded and other files, such as email attachments, are also embedded (if present). *Note: this is the basic 1-to-1 verifiable preservation of an email message as EA-PDF (PDF/A) that non-EA-PDF aware legacy software can very reasonably be expected to support.* *Note: the embedded original raw email asset may contain additional emails or other information, but all PDF pages in PDF/mail-1s reflect only the single email message.*
PDF/mail-1si **“Single, isolated”**	PDF/A-3a PDF/A-3u PDF/A-4 PDF/A-4f PDF/A-4e	**Single, isolated**: A single email message preserved as a single EA-PDF file, where pages in the PDF are a visual representation of the email’s content or context, but the original raw source email asset(s) are *not* embedded (i.e., are not faithfully preserved). Other files may be embedded (e.g. email attachments). *Note: this is a special EA-PDF profile primarily intended for use in PDF/mail-1{c, ci} “containers” (see below) or FOIA.* *Note: “isolated” means that this class of EA-PDF file is isolated from its original raw source email asset and thus is not a self-contained preservation with the associated original raw source email assets.*

[3] PDF/A-4e conformance is only required if 3D or rich media content is required to represent email which is highly unlikely. In ISO/DIS 19005-4, PDF/A-4e is formalized as an extension of PDFA-4f.

PDF/mail-1 Profile	PDF/A Conformance Levels	Description
`PDF/mail-1m` **“Multiple”** *Note: this is unrelated to the email term “multipart”.*	PDF/A-3a PDF/A-3u PDF/A-4f PDF/A-4fe	**Multiple**: Multiple emails preserved as a single EA-PDF file *without* folder structure[4], where pages in the PDF are a visual representation of the content or context of the emails. The original raw source email assets are always embedded (faithfully preserved). *Note: this is a “flat” preservation of multiple emails. Although non-EA-PDF aware legacy software will operate with `PDF/mail-1m` files, the level of context available may limit the experience (e.g., associating PDF pages with specific emails or email attachments, etc.).* *Note: a `PDF/mail-1m` file with only a single email message is effectively a `PDF/mail-1s` file. It is required that such files are indicated as `PDF/mail-1s`.*
`PDF/mail-1mi` **“Multiple, isolated”**	PDF/A-3a PDF/A-3u PDF/A-4 PDF/A-4f PDF/A-4e	**Multiple, isolated**: Multiple emails preserved as a single EA-PDF file *without* folder structure, where pages in the PDF are a visual representation of the content or context of the emails. The original raw source email assets are *not* embedded, however other files may be embedded (e.g. email attachments). *Note: “isolated” means that this class of EA-PDF file is disassociated from its original raw source email asset and thus is not a self-contained preservation with associated original raw source email assets.* *Note: this is a special EA-PDF profile primarily intended for use in `PDF/mail-1{c, ci}` “containers” (see below).*

[4] A future version of EA-PDF might consider how to add folder-like structure to `PDF/mail-1m` files via new PDF capabilities. This was not considered at this time as this would not work with any legacy software.

PDF/mail-1 Profile	**PDF/A Conformance Levels**	**Description**
PDF/mail-1c **“Container”**	PDF/A-3a PDF/A-3u PDF/A-4f[5]	**Container**: An EA-PDF “structured container” for one or more embedded EA-PDF files, each of which may be any other PDF/mail-1 profile. PDF/mail-1c files can replicate complex folder hierarchies typically found in modern email clients, email formats, or file systems. The container that is the PDF/mail-1c file does _not_ contain pages representing content of preserved emails – all PDF representations of email content are in the _embedded_ EA-PDF files stored within the container PDF collection. Pages in the container PDF represent the context of the collection. *Note: there is no requirement that hierarchical folders must be used, in which case a PDF/mail-1c file is like PDF/mail-1m, except that the rendition of emails are stored as separate embedded EA-PDF files.* *Note: non-EA-PDF aware legacy software needs to support PDF Collections (PDF 1.7) as otherwise the level of context and semantics available will severely limit the experience and understanding of the content.*

[5] PDF/mail-1{c, ci} files with hierarchical folder structure relies on the vendor neutral **Folders** feature of PDF Collections. **Folders** were initially defined as part of Adobe Extension Level 3 to ISO 32000-1:2008 (PDF 1.7) and thus can validly exist in PDF/A-3 as they do not impact static rendering of PDF page content. They were later adopted into PDF 2.0 with ISO 32000-2 (see §12.3.5, [ISO 32000-2]). PDF/A standards do not define dynamic features such as interaction with PDF collections. PDF/mail-1{c, ci} files are not required to always use **Folders**, in which case PDF/mail-1{m, mi} may be a better choice. PDF/A-4e is not listed as there is no identified requirement for including JavaScript or 3D content into the container PDF.

PDF/mail-1 Profile	PDF/A Conformance Levels	Description
`PDF/mail-1ci` **“Container, isolated”**	PDF/A-3a PDF/A-3u PDF/A-4f [6]	**Container, isolated**: An EA-PDF “structured container” for one or more embedded EA-PDF files, each of which may be any other `PDF/mail-1` profile. `PDF/mail-1ci` files can replicate complex folder hierarchies typically found in modern email clients, email formats, or file systems. The container that is the `PDF/mail-1ci` file does *not* contain pages representing content of preserved emails – all email content is in the *embedded* EA-PDF files stored within the container PDF collection. Pages in the container PDF represent the context of the collection. This isolated profile indicates that this EA-PDF does *not* contain a *full* set of preserved original raw source email assets for all emails in the collection and is thus not a full preservation format. This may be because a `PDF/mail-1{si, mi}` is included, or that the `PDF/mail-1ci` container itself does not contain the original raw source email assets. *Note: there is no requirement that hierarchical folders must be used, in which case a `PDF/mail-1ci` file is like `PDF/mail-1mi`, except that the rendition of emails are stored as separate embedded EA-PDF files.* *Note: non-EA-PDF aware legacy software needs to support PDF Collections (PDF 1.7) as otherwise the level of context and semantics available will severely limit the experience and understanding of the content.*

Table 2: Summary of EA-PDF profiles

As illustrated in *Figure 2: PDF/mail-1s vs. PDF/mail-1m – single email message vs multiple emails in a single PDF with their associated original raw source email assets* and *Figure 3: PDF/mail-1si vs. PDF/mail-1mi – “isolated” single email message vs multiple emails in a single PDF without their associated original raw source email assets* below, the primary difference between `PDF/mail-1{s,`

[6] `PDF/mail-1c` files with hierarchical folder structure relies on the vendor neutral **Folders** feature of PDF Collections. **Folders** were initially defined as part of “Adobe Extension Level 3 to ISO 32000-1:2008 (PDF 1.7)” and thus can validly exist in PDF/A-3 as they do not impact static rendering of the containers’ page content. They were later adopted into PDF 2.0 with ISO 32000-2 (see §12.3.5, [ISO 32000-2]). PDF/A standards do *not* define dynamic features such as interaction with PDF Collections. `PDF/mail-1c` files are not required to always use **Folders**. PDF/A-4e is not listed as there is no identified requirement for including JavaScript or 3D content into the container PDF.

`si}` and PDF`/mail-1{m, mi}` is the *1-to-1* and *M-to-1* relationship between emails and PDF respectively. As a result of containing multiple emails, PDF`/mail-1{m, mi}` will likely be much larger in size, have much larger document XMP metadata, and contain more pages that need to be logically grouped so it is clear which email is being represented. However, PDF`/mail-1{m, mi}` allows resources such as fonts or images to be efficiently shared and reused across multiple emails and thus a single PDF`/mail-1{m, mi}` file will most likely be significantly smaller than the equivalent emails as multiple PDF`/mail-1{s, si}` files. PDF`/mail-1{si, mi}` files may also be smaller in size again, as the original raw source email assets may not be preserved.

None of PDF`/mail-1{s, si, m, mi}` files support emails in folder hierarchies (PDF`/mail-1{c, ci}` is described in more detail below).

> *EA-PDF Writer: it may be possible to convert (split) PDF`/mail-1{m, mi}` files into multiple standalone PDF`/mail-1{s, si}` files, or for multiple PDF`/mail-1si` files to be combined (merged) into a single PDF`/mail-1mi` file, without loss of any information or functionality. It may also be possible to add emails to PDF`/mail-1mi` files at some later time. Creating container PDF`/mail-1{c, ci}` files from one or more existing PDF`/mail-1{s, si, m, mi, c, ci}` files is also possible. Such functionality is not mandated as maintaining trustworthiness (verifiability) of the original raw source email assets and updating certain PDF data structures, such as logical structure and metadata, may be difficult for some EA-PDF implementations.*

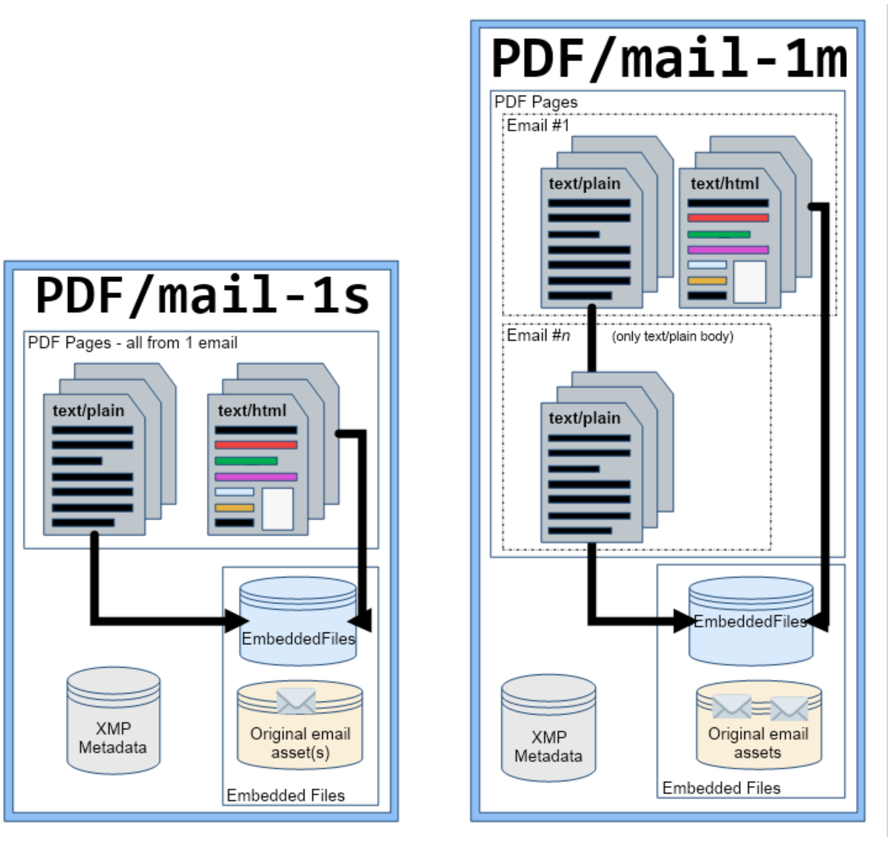


Figure 2: PDF`/mail-1s` vs. PDF`/mail-1m` – single email message vs multiple emails in a single PDF with their associated original raw source email assets

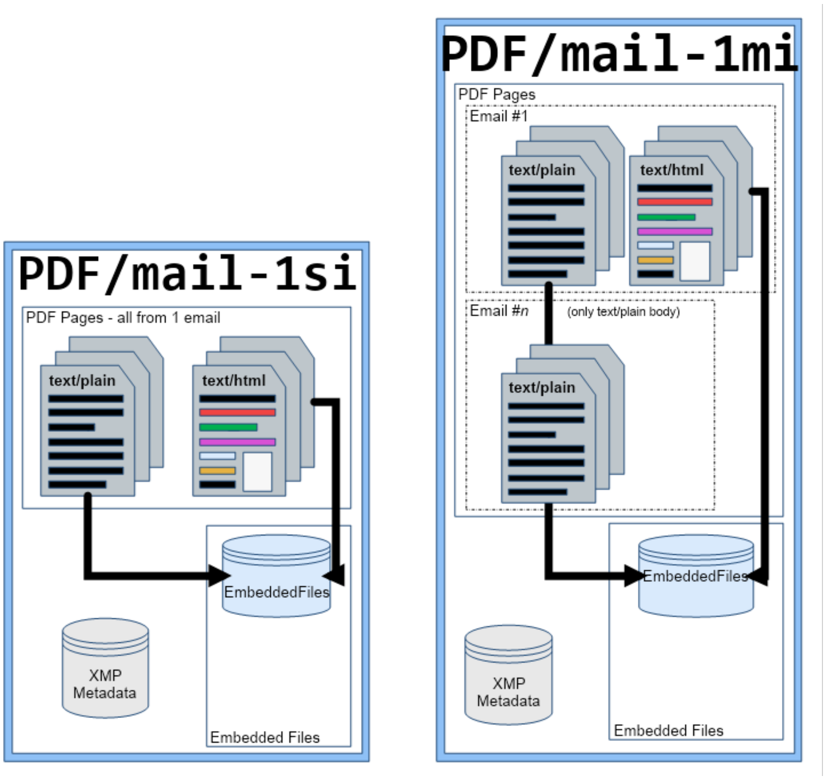


Figure 3: PDF/`mail-1si` vs. PDF/`mail-1mi` – “isolated” single email message vs multiple emails in a single PDF *without* their associated original raw source email assets

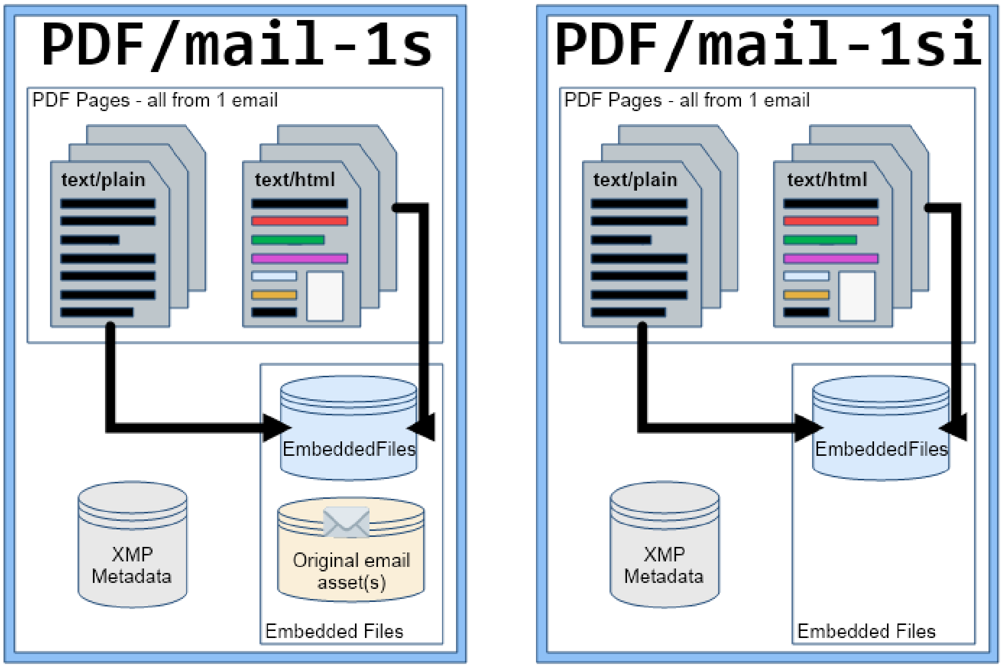
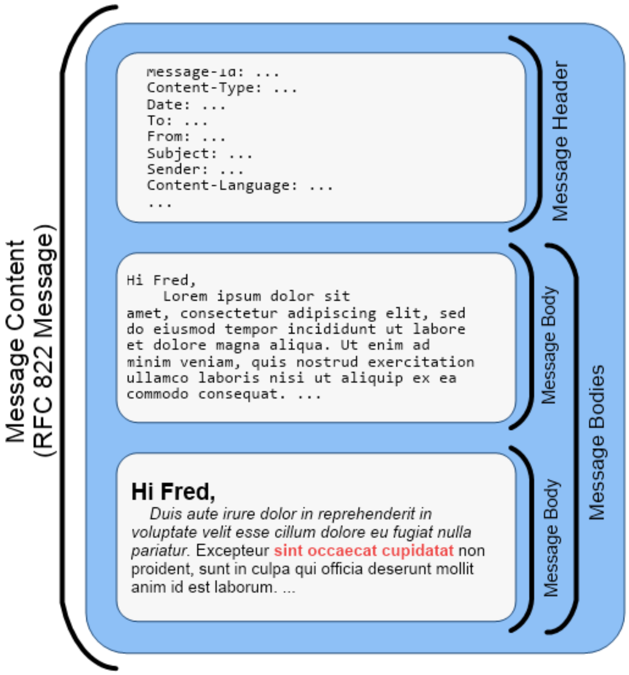


Figure 4: Comparison of PDF/`mail-1s` vs. PDF/`mail-1si` (isolated), where the isolated EA-PDF file is missing the original email asset(s).

EA-PDF aware software is _not_ required to support all EA-PDF profiles. For example, simple *EA-PDF Creation Software* might only create PDF`/mail-1{s, si}` files on a 1-for-1 basis with email or be limited to processing certain simple email formats. EA-PDF aware consumption software (such as EA-PDF viewers or web servers) however must be able to _detect_ all PDF`/mail` profiles[7]. If a detected PDF`/mail` profile is not implemented or fully supported, then the EA-PDF aware consumption software must inform the user.

This document uses the following terminology in relation to email, regardless of the physical storage format (see *Figure 5: Email terminology* below):



```
Message-ID: ...
Content-Type: ...
Date: ...
To: ...
From: ...
Subject: ...
Sender: ...
Content-Language: ...
...
```

```
Hi Fred,
    Lorem ipsum dolor sit
amet, consectetur adipiscing elit, sed
do eiusmod tempor incididunt ut labore
et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud exercitation
ullamco laboris nisi ut aliquip ex ea
commodo consequat. ...
```

Hi Fred,
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur **sint occaecat cupidatat** non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ...

Figure 5: Email terminology

The *Message Header* includes both structured and unstructured header fields. A single email message may have zero or more *Message Body* sections. A draft email is an example where a *Message Body* may not exist, and when many email headers may also be blank (empty). In addition, EA-PDF defines some additional fields which are calculated based on email content (e.g., number of attachments).

> *Note: this document is not concerned with the transmission of email between systems beyond what gets recorded in email message header fields by email systems.*

[7] Note this says "PDF`/mail`", not "PDF`/mail-1`": EA-PDF aware consumption software developed to support PDF`/mail-1` files defined by this specification must also be able to detect any future versions of this specification (e.g., PDF`/mail-2`, PDF`/mail-3`) and any newer profiles (e.g., PDF`/mail-2z`) and warn users appropriately. This is so users are clearly informed of potential limitations. This ensures preservationists are aware when using outdated software with future EA-PDF files.

4 References

This list of references includes both normative and informative references to aid readers.

[PDF/A-3] "ISO 19005-3, *Document management — Electronic document file format for long-term preservation — Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3)"* (including any Amendments and errata). https://pdfa.org/resource/iso-19005-3-pdf-a-3/

Note: all conformance levels of both PDF/A-1 (ISO 19005-1) and PDF/A-2 (ISO 19005-2) are unsuitable for EA-PDF because they do not allow arbitrary embedded files which are required to preserve email attachments and email asset(s). Conformance level PDF/A-3b is explicitly excluded from EA-PDF as it is a reasonable expectation that EA-PDF Creation Software *can always create Unicode content from emails, even if logical structure is not readily available. Thus EA-PDF files may be PDF/A-3a (preferred) or PDF/A-3u compliant.*

Note: the description of associated files and marked content in Annex E.5 is technically incorrect in PDF/A-3 and thus cannot be used. See PDF Errata #374.

Reason: This PDF/A-3 reference is undated so any future dated revisions or amendments to PDF/A-3 can be included as part of EA-PDF.

[PDF/A-4] "ISO/DIS 19005-4:2024 *Document management — Electronic document file format for long-term preservation — Part 4: Use of ISO 32000-2 (PDF/A-4)",* https://pdfa.org/resource/iso-19005-4-pdf-a-4/

Reason: conformance level PDF/A-4f is required by most EA-PDF profiles as it supports arbitrary embedded files necessary to preserve original raw source email assets, email attachments, etc. Only certain PDF/mail files that are isolated and need PDF 2.0 features, do not have any email attachments do not need to be PDF/A-4f. Note that PDF/A-4 does not define conformance levels A, B, or U like earlier PDF/A versions.

Note: this is a specific dated reference to the 2024 dated revision of PDF/A-4 which resolves various issues related to embedded files in PDF/A-4. When ISO/DIS 19005-4:2024 is completed and published by ISO this reference will be updated.

[ISO 32000-2] "ISO 32000-2, *Document management — Portable document format — Part 2: PDF 2.0"* including all Amendments and errata. https://pdfa.org/resource/iso-32000-2/

Note: this document only references the PDF 2.0 ISO specification as it has greatly improved wording over ISO 32000-1:2008 (PDF 1.7). All features used in EA-PDF are described by their definitions in ISO 32000-2:2020. This is also an undated reference so any future dated revisions or amendments can be included as part of EA-PDF, such as the forthcoming Amendment 1, as well as all industry-agreed errata resolutions.

[ADBE-Extn-L5] "Adobe® Supplement to ISO 32000-1 BaseVersion: 1.7 ExtensionLevel: 5 (Adobe® Acrobat® SDK, Version 9.1)", dated June 2009. Available via https://pdfa.org/resource/pdf-specification-archive/

[PDF-Declarations] "*PDF Declarations*", PDF Association, https://www.pdfa.org/resource/pdf-declarations/

Note: PDF Declarations is an industry standardized way that XMP metadata can be extended to declare that a PDF file or object also conforms with 3rd party specifications or profiles that are unrelated to PDF technology (e.g., PREMIS or EAXS).

[AssociatedFiles] “*PDF 2.0 Application Note 002: Associated Files*”, PDF Association, https://www.pdfa.org/resource/pdf-2-0-application-note-002-associated-files/

Note: this application note is not entirely technically correct with respect to PDF 2.0 and does not include PDF/A-4. It will be updated soon by the PDF Association.

[C2PA] “*C2PA Technical Specification*”, Coalition for Content Provenance and Authenticity (C2PA), https://c2pa.org/specifications/specifications/2.0/index.html

[CLIR] “*The Future of Email Archives. A Report from the Task Force on Technical Approaches for Email Archives*” Council on Library and Information Resources (CLIR), CLIR, vol. 1, 1 vols. Washington DC, USA, 2018. https://www.clir.org/pubs/reports/pub175/

[EA-PDF-TR] “*A Specification for Using PDF to Package and Represent Email Technical Report*”, EA-PDF Working Group, University of Illinois at Urbana-Champaign, p. 34, Jan. 2021. https://www.ideals.illinois.edu/handle/2142/109251

[EAXS] “*Email Account XML Schema (EAXS*)”, State Archives of North Carolina and the Smithsonian Institution Archives https://github.com/StateArchivesOfNorthCarolina/tomes-eaxs/blob/master/versions/1/eaxs_schema_v1.xsd

[EAXSPREMISCrosswalk] “*Email EAXS/PREMIS Crosswalks for collection/account*”, https://docs.google.com/spreadsheets/d/1GoLnA0tkslkfYXIDM6k5wM8KU8YhfUMXWgYg7Sz99s0/edit#gid=0

[ePADDMeta] “*EA PDF proposed metadata profile - based on ePADD*”, https://docs.google.com/spreadsheets/d/1bckPBvYEeoBT0x2OeNH-dO6JoyWAolTu/edit#gid=175841588

[FOAF] “*FOAF Friend Of A Friend Vocabulary Specification*”, http://xmlns.com/foaf/0.1/

[HeaderCharacteristics] *“Email Header Significant Characteristics”*, Kevin De Vorsey, NARA, https://docs.google.com/spreadsheets/d/16uSKduxhS7Z8GJ4AKEQ6hFM7hXhsal8KQ-_jMUcdMWA/edit#gid=0

[MetadataStreams] “*PDF 2.0 Application Note 003: Use of object metadata streams*”, PDF Association, https://www.pdfa.org/resource/pdf-2-0-application-note-003-use-of-object-metadata-streams/

[Narrative] “*EA-PDF Project Narrative*”, https://www.pdfa.org/resource/ea-pdf/

[PREMIS] “*PREMIS Data Dictionary for Preservation Metadata*”, https://www.loc.gov/standards/premis/

[RFC-8118] “*RFC 8118: The application/pdf Media Typ*e”, https://www.rfc-editor.org/rfc/rfc8118.html

[RFC-822] “*RFC 822: Standard for ARPA Internet Text Messages*”, https://www.w3.org/Protocols/rfc822/ plus related RFCs (e.g., “RFC 2076: *Common Internet Message Headers”*; “RFC 4021: *Registration of Mail and MIME Header Fields”*; “RFC 6532: *Internationalized Email Headers”, etc.*)

[UIUC-EAPDF] “*UIUC Library’s prototype EA-PDF implementation”*. https://github.com/UIUCLibrary/ea-pdf

[XMP] “ISO 16684-1, *Extensible metadata platform (XMP) specification — Part 1: Data model, serialization and core properties”,* https://pdfa.org/resource/iso-16684-1/.

[XMPNamespaces] https://developer.adobe.com/xmp/docs/XMPNamespaces/

5 Terms, definitions, and acronyms

Term	Description
Asset Original raw source email asset	An input file needed by *EA-PDF Creation Software* when creating EA-PDF files, such as the original source email file(s) (EML, MSG, MBOX, NSF, PST/OST, etc.), a referenced image, or a file referred to by HTML, etc. There is no requirement in EA-PDF that the original raw source email asset only represents the email(s) that are rendered into EA-PDF files – additional information, including other emails, may get preserved. The phrase "original raw source email asset" is used in reference to the preservation of the original email data format (as a bitstream) in its unmodified (raw) state. *Any* change to the original format (for whatever reason) does not meet this definition as it could potentially impact the reliability and trustworthiness (verifiability) of the preservation and its associated renderings (as PDF/A pages). *Note: the generic word "file" is avoided as it can get confusing with so many different files: email files, PDF files, embedded files, file attachments, the EA-PDF file, email attachments, files in a collection, etc.* *Note: this is an EA-PDF specific term and is independent from the use of "asset" in C2PA (https://c2pa.org/).*
Attachment, Embedded file	In the context of email, PDF and common usage, the terms "attachment" and "embedded file" are heavily overloaded and potentially ambiguous or confusing. To avoid ambiguities the follow clarified terms are always used in this document: • Email attachment – a file attached (embedded) in an email; • File attachment annotation – a specific user-visible PDF feature for visually referencing a file (see [ISO 32000-2], §12.5.6.15) from PDF pages; • Embedded file – the technical PDF objects used to embed the contents of a file within a PDF via a File Specification dictionary and associated Embedded File stream object (see [ISO 32000-2], §7.11.3 and §7.11.4); • **EmbeddedFiles** name tree – the PDF feature used to list *some* embedded files in a PDF (see [ISO 32000-2], §7.7.4); • PDF Collection – see PDF Collection/PDF Portable Collection below. All files in a collection are required by ISO 32000-2 to be listed in the **EmbeddedFiles** name tree; • Associated File – a PDF feature that associates content in arbitrary formats with an object in a PDF file and to identify a basic semantic relationship between them via the **AFRelationship** key (see [PDF/A-3], [PDF/A-4] and [ISO 32000-2], §14.13).

Term	Description
Content Set	EA-PDF specific term referring to a consecutive set of PDF page(s) associated with a common source (e.g., from the HTML body of an email, from the text body of an email, a list of email attachments, a conversion report, etc.). A content set will always be at least 1 page in an EA-PDF file, as page boundaries demarcate content sets.
Core Fields	EA-PDF specific term referring to a core set of email headers and related fields agreed by the EA-PDF LWG as highly relevant to the management and long-term reliable preservation of emails. Technically a few core fields are not email headers but calculated or derived from emails (e.g. the number of attachments). *Note: Core Fields from email may not all have values – for example in draft, unsent, or corrupted emails. Some EA-PDF Core Fields are also derived (e.g. size of the email file in bytes) or calculated (e.g., number of attachments).*
DC, DCMI	Dublin Core Metadata Initiative, https://www.dublincore.org/
DPart, DPM	Document Part / Document Part Metadata. See [ISO 32000-2] §14.12.
EA-PDF Creation Software	Software that creates EA-PDF files according to this specification by reading and processing one or more email assets as input. This creation software may be interactive or non-interactive. A software application may have a mode of operation for EA-PDF Creation as well as other modes. *EA-PDF Creation Software* is not required to create all PDF/`mail-1` profiles.
EML	A text-based email file format representing a single email message, defined by [RFC-822] and various other RFCs. Most likely an email asset used with PDF/`mail-1{s, si}` files. See also https://www.loc.gov/preservation/digital/formats/fdd/fdd000388.shtml.
FOAF	Friend Of A Friend ontology. See [FOAF].
FOIA	Freedom of Information Act. Used as a general term to refer to any workflow where redacted information of a preservation artifact may need to be disclosed.
Isolated	In an EA-PDF context, an "isolated" PDF/mail file refers to a file including an `i` profile identifier. This indicates that the file does *not* contain a trustworthy original raw source email asset and thus may not be appropriate for long-term verifiable preservation. Such files do contain the PDF/A reliable rendering of emails and associated metadata without a verifiable linkage to the original raw (unmodified) email assets.
MBOX	Text-based MailBox email file formats that support multiple emails in a single file. See https://en.wikipedia.org/wiki/Mbox. Most likely an email asset used with PDF/`mail-1m` files. See https://www.loc.gov/preservation/digital/formats/fdd/fdd000383.shtml

Term	Description
Media Type	IANA Media Type. Previously referred to as MIME type. See https://www.iana.org/assignments/media-types/media-types.xhtml
MSG	Microsoft™ Outlook Item file format for a single email message, see https://learn.microsoft.com/en-us/openspecs/exchange_server_protocols/ms-oxmsg. Most likely an email asset used with PDF`/mail-1{s, si}` files.
Non-EA-PDF aware legacy software	This refers to a broad range of PDF software that does not implement any specific support for EA-PDF. This may range from software that is otherwise PDF/A compliant (*strongly recommended*!) to software that is neither PDF/A nor EA-PDF aware.
NSF	IBM Lotus Notes™ email file extension. See https://www.loc.gov/preservation/digital/formats/fdd/fdd000433.shtml.
OST, PST	Microsoft™ proprietary binary file formats that can store multiple emails and other types of data (e.g. calendar entries) in hierarchical folder-like structures. Most likely an email asset used with PDF`/mail-1c` files so that the folder hierarchy can be represented and preserved. See https://www.loc.gov/preservation/digital/formats/fdd/fdd000378.shtml.
PDF Collection, PDF Portable Collection	The PDF feature supporting collections of files - as defined in [ISO 32000-2], §12.3.5 Collections. Originally introduced in PDF 1.7 and extended in PDF 2.0. Also known as “Portfolios”, “Packages”, or “Binders”. *Note: this PDF-specific term is completely independent from an “archival collection”.*
PDF`/mail-1c`	EA-PDF v1.0 container file format profile that contains other EA-PDF files arranged in hierarchical folder structures and where original raw source email assets are faithfully preserved. “Container”. Based on PDF Collections.
PDF`/mail-1ci`	EA-PDF v1.0 container file format profile that contains other EA-PDF files arranged in hierarchical folder structures, but where some original raw source email assets are missing or modified. “Container, isolated”. Based on PDF Collections.
PDF`/mail-1m`	EA-PDF v1.0 file format profile supporting multiple emails with embedded original raw source email assets. “Multiple”. *Note: this profile is unrelated to “multipart”.*
PDF`/mail-1mi`	EA-PDF v1.0 file format profile supporting multiple emails but *without* embedded original raw source email assets. “Multiple, isolated”. Intended use is within PDF`/mail-1c` containers for complex email formats that contain internal folder-like hierarchies (such as OST/PST). *Note: this profile is unrelated to “multipart”.*
PDF`/mail-1s`	EA-PDF v1.0 file format profile supporting a single email message with an embedded original raw source email asset. “Single”.

Term	Description
`PDF/mail-1si`	EA-PDF v1.0 file format profile supporting a single email message but *without* the embedded original raw source email asset. "Single, isolated". Intended use is within PDF`/mail-1c` containers for complex email formats that contain internal folder-like hierarchies (such as OST/PST).
PREMIS	PREMIS is an international standard for preservation metadata to support the preservation of digital objects, https://www.loc.gov/standards/premis/. EA-PDF does not mandate the use of any PREMIS elements. See [PREMIS].
Profile	The kind of PDF/mail-1 file: PDF`/mail-1{s`, `si`, `m`, `mi`, `c`, `ci}`. This term was specifically chosen to avoid confusion and overlap with PDF/A "conformance levels", which are also highly relevant to EA-PDF.
Render, Rendering	When referring to email, the term "render" in this document means the process by which an email (in an original source email format) is converted to one or more PDF pages for visualization. This document does not prescribe any specific conversion processes.
XMP	Extensible Metadata Platform. XML-based metadata used with PDF and other formats. Standardized by ISO 16684-1 – see [XMP]. Comprises both various XMP Standard Namespaces and XMP Specialized Namespaces – see https://developer.adobe.com/xmp/docs/XMPNamespaces/.

6 Out of scope

This specification defines the PDF/`mail-1` file format as a faithful and verifiable long-term preservation of email, including a reliable static appearance of renditions of the email and associated assets that is technically achieved by utilizing PDF/A-3 or PDF/A-4. EA-PDF is designed to be fully machine validatable (i.e., without requiring human checks) and without referencing the original source email asset(s).

The following aspects are not defined by this document, but limited guidance or suggestions may be given in the form of notes:

- Line wrapping, pagination, layout, and visual appearance of content in EA-PDF files;
- Conversion of email body formats to PDF content streams;
- Handling and recovery of errors in email file formats encountered by *EA-PDF Creation Software* when converting to EA-PDF (beyond the fact that an output EA-PDF file must comply with this specification, PDF/A, and ISO 32000);
- Processing and/or display of invalid EA-PDF files;
- Methods to create Tagged PDF and/or PDF Logical Structure;
- The user interface of EA-PDF aware software;
- Cyber-security factors related to email or PDF (e.g., excluding certain kinds of attachments, confirming hyperlinks, etc.);
- The policies and procedural frameworks in which EA-PDF is expected to operate;
- Preservation of non-digital emails (e.g., hardcopy prints of emails);
- Protocols and transmission of email, beyond what is recorded in email files;
- Redaction of email;
- Precise software algorithms, user interface choices, and implementation design decisions.

7 From an email and end-user perspective

This section provides a description of the PDF`/mail-1` file format from the perspective of readers familiar with email such as [RFC-822] and related RFCs, however it needs to be read by everyone.

This section is written and structured from an *input* point of view and contains additional explanation and reasoning, including recommendations that would normally be found in Application Notes or similar supporting publications. In addition, testing of a selection of existing PDF applications was performed to support LWG requests for reasonable support in existing non-EA-PDF aware legacy software.

In this section, the term "render" implies the PDF content that *EA-PDF Creation Software* creates as PDF content streams (operators and operands, using associated resources and other PDF objects such as images and fonts) such that the resulting PDF pages have the "look and feel" of email such as might be seen in an email client.

Many of the technical requirements for EA-PDF are achieved by direct reference to PDF/A.

It is expected that extant email files will have their own errors, extensions, or other deviations from official specifications. How this is handled by *EA-PDF Creation Software* is beyond the scope of this document.

> *EA-PDF Writer:* EA-PDF Creation Software *may choose to log processing errors encountered during conversion into EA-PDF files as either machine-readable metadata, additional rendered content (i.e., as a multi-page report for humans), or both. However, even with invalid emails, an output EA-PDF file must always be valid according to this specification, the appropriate PDF/A conformance level, and ISO 32000.*

7.1 Key PDF features

By way of a gentle introduction to PDF, certain PDF features are described below at a high level so readers unfamiliar with PDF can understand their relevance in achieving the goals of faithful and verifiable email preservation and representation as PDF pages. This section summarizes these key features but does not state any formal requirements (i.e., there are no technical "SHALL" or "SHOULD" statements).

7.1.1 Metadata

Metadata is critically important to email preservation and management using EA-PDF. It both presumptively identifies EA-PDF files and declares the necessary conformance as well as providing definitive descriptive information about the preserved emails. Additional requirements and best practices for metadata (beyond what is required by PDF/A and this EA-PDF specification) are defined by the policy and procedure frameworks in which EA-PDF is used. EA-PDF follows the metadata conventions used in the ISO 19005 family of standards.

EA-PDF also supports the representation of metadata or any other information as human readable content on PDF pages to better support non-EA-PDF legacy software that cannot access XMP metadata (generically referred to as "conversion reports" in this document).

Metadata in PDF can be stored in two formats: XML-based XMP in **Metadata** streams, and as PDF string objects in the traditional Document Information PDF dictionary. PDF/A and the other modern PDF ISO subsets all use XMP Metadata to store their conformance level and other information. In accordance with recommendations in the [XMP] standard and to facilitate interoperability with the broadest range of software, EA-PDF leverages existing XMP schemas such as Dublin Core, XMP Basic Schema, and the Adobe PDF Schema for certain basic information (see [XMPNamespaces]). XMP supports custom data and, where possible, EA-PDF will extend XMP by utilizing existing XMP or RDF ontologies, such as Dublin Core, Friend Of A Friend [FOAF], [PREMIS], [EAXS], etc.

> *Note: leveraging existing specifications for metadata will hopefully make EA-PDF files more compatible with existing document and archival management systems that already support XMP.*

All XMP metadata stored in EA-PDF files must be encoded as UTF-8.

> *Note: ISO 19005-3 PDF/A-3 did not mandate UTF-8 based XMP whereas the latest dated revision of PDF/A-4 adds this requirement. Testing of legacy viewers indicates problematic support for XMP with non-UTF-8 content. EA-PDF thus mandates encoding all XMP as UTF-8 for the widest possible support and matches with the latest dated revision of PDF/A-4.*

XMP Metadata used in EA-PDF files such as PDF`/mail-1m` and PDF`/mail-1c` may be very large as it covers many emails, attachments, etc. and potentially XMP Extension Schemas. For this reason, *EA-PDF Creation Software* may wish to compress the XMP Metadata, however this may reduce interoperability with some document and archive management systems.

Any use of custom XMP data in PDF`/mail-1` files that conform to PDF/A-3 must also include the related XMP extension schema as defined in PDF/A-3, §6.6.2.3.2.

> *Reason: ISO 19005-3 PDF/A-3 mandates the use of XMP extension schemas either in the same XMP metadata stream as the custom data, or in the main document catalog **Metadata** stream. PDF/A-4 does not mandate XMP extension schemas, while also supporting Associated Files with an **AFRelationship** value of* Schema.

> *EA-PDF Writer: as discussed in the LWG, PDF/A-3 XMP extension schemas can be very complex while the use of PDF/A-4 allows alternate schema formats to be included.*

An EA-PDF file will contain many logical groupings of XMP metadata as illustrated in Figure 6: logical groupings of different categories of XMP metadata and schemas in EA-PDF, with linkage via Message-ID/Mail_GUID properties below

> *Note: Figure 6: logical groupings of different categories of XMP metadata and schemas in EA-PDF, with linkage via Message-ID/Mail_GUID properties below does not necessarily represent the physical ordering of the document level XMP metadata. XMP Extension Schemas are only required if the EA-PDF file conforms to PDF/A-3. PDF/A-4 files may alternatively optionally embed schemas using Associated Files with an **AFRelationship** value of Schema.*

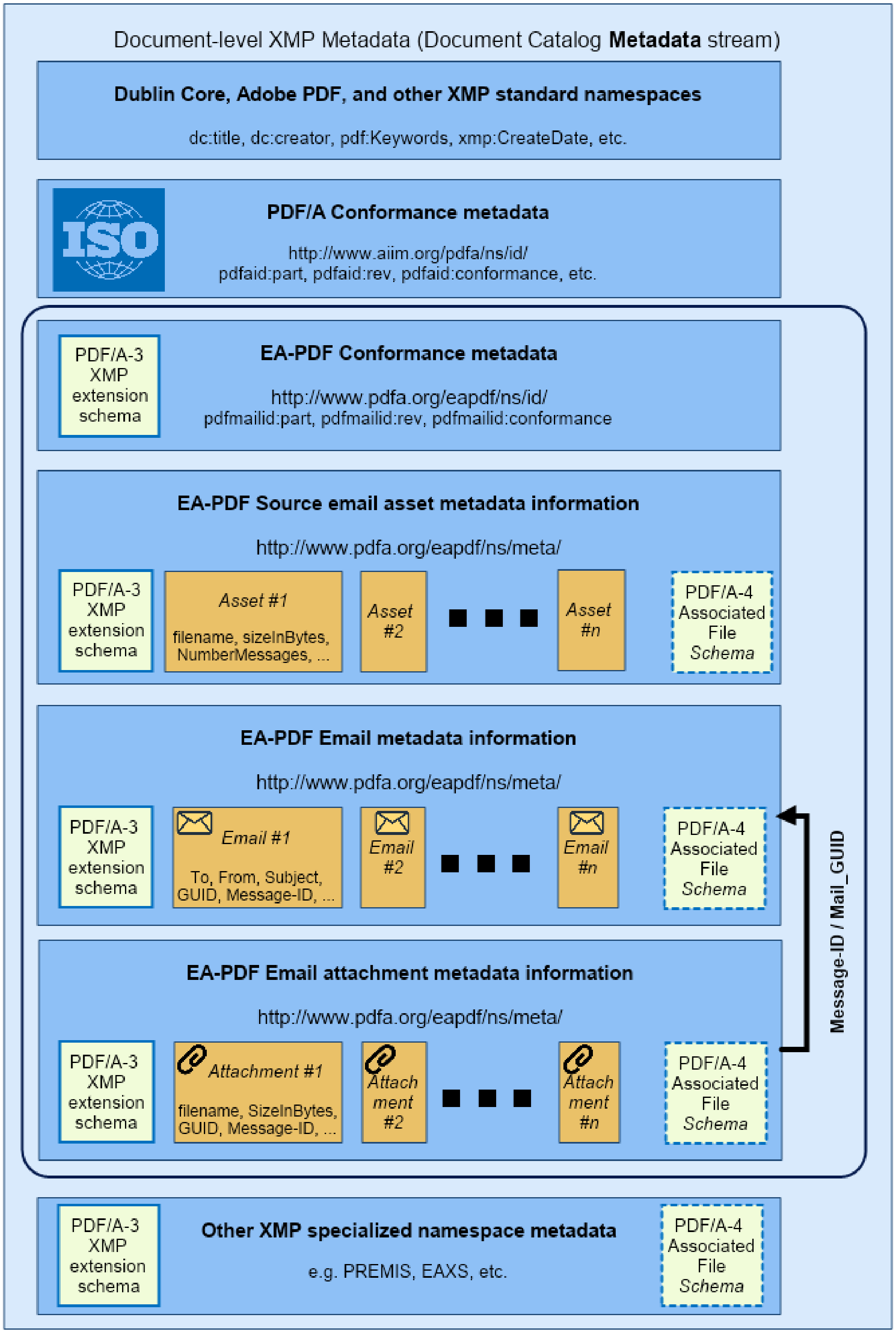

Document-level XMP Metadata (Document Catalog **Metadata** stream)
Dublin Core, Adobe PDF, and other XMP standard namespaces
dc:title, dc:creator, pdf:Keywords, xmp:CreateDate, etc.
ISO
PDF/A Conformance metadata
http://www.aiim.org/pdfa/ns/id/
pdfaid:part, pdfaid:rev, pdfaid:conformance, etc.
PDF/A-3 XMP extension schema
EA-PDF Conformance metadata
http://www.pdfa.org/eapdf/ns/id/
pdfmailid:part, pdfmailid:rev, pdfmailid:conformance
EA-PDF Source email asset metadata information
http://www.pdfa.org/eapdf/ns/meta/
PDF/A-3 XMP extension schema
Asset #1
filename, sizeInBytes, NumberMessages, ...
Asset #2
Asset #n
PDF/A-4 Associated File Schema
EA-PDF Email metadata information
http://www.pdfa.org/eapdf/ns/meta/
PDF/A-3 XMP extension schema
Email #1
To, From, Subject, GUID, Message-ID, ...
Email #2
Email #n
PDF/A-4 Associated File Schema
EA-PDF Email attachment metadata information
http://www.pdfa.org/eapdf/ns/meta/
PDF/A-3 XMP extension schema
Attachment #1
filename, SizeInBytes, GUID, Message-ID, ...
Attach ment #2
Attach ment #n
PDF/A-4 Associated File Schema
Message-ID / Mail_GUID
PDF/A-3 XMP extension schema
Other XMP specialized namespace metadata
e.g. PREMIS, EAXS, etc.
PDF/A-4 Associated File Schema

Figure 6: logical groupings of different categories of XMP metadata and schemas in EA-PDF, with linkage via `Message-ID`/`Mail_GUID` properties

Any PDF dictionary or stream object may also contain a **Metadata** entry referencing an XML-based XMP metadata stream allowing metadata to be logically and semantically associated with different objects in an EA-PDF file. Example use-cases of this are discussed below. See [MetadataStreams] and §14.3.2 [ISO 32000-2].

All EA-PDF files are required to have a Document Catalog **Metadata** stream that declares the EA-PDF profile and a corresponding PDF/A conformance level and a set of email-related *Core Fields*. Additional metadata in any **Metadata** stream may also be stored, in addition to what EA-PDF and PDF/A explicitly require. *EA-PDF Creation Software* must also record **dc:Creator**, **pdf:Producer** and **xmp:CreateDate** information in the document-level XMP metadata for all EA-PDF files.

An EA-PDF file may also use [PDF-Declarations] in any **Metadata** XMP stream to specify compliance with other 3rd party specifications or profiles, such as the EAXS specification [EAXS] or PREMIS [PREMIS]. It is *strongly recommended* that all Declarations be publicly recorded at https://pdfa.org/declarations/.

PDF/A-3 [PDF/A-3, §6.1.5] permits the use of the Document Information dictionary (§14.3.3 [ISO 32000-2], as referenced by the document trailer **Info** entry), however PDF/A-4 prohibits the general use of this dictionary[8]. To support the widest range of non-EA-PDF aware legacy software, EA-PDF files that are not required to use any PDF 2.0 features and thus can be PDF/A-3 compatible may wish to use PDF/A-3 in order to include a Document Information dictionary and provide a better experience with non-EA-PDF aware legacy software.

Reason: a lot of legacy software has no functionality to display XMP metadata and can only report strings from the conventional PDF Document Information dictionary. Furthermore, software that can display XMP metadata may not have appropriate user interfaces for large XMP metadata such as required by EA-PDF.

EA-PDF Writer: Table 7 in PDF/A-3 (ISO 19005-3) defines a "crosswalk" of how the XMP and Document Information metadata ought to be aligned.

*EA-PDF Writer: when choosing between PDF/A-3 and PDF/A-4, it is necessary to appreciate the PDF 2.0 specific features. PDF 2.0 specific features include PDF Unicode strings using UTF-8 (convert to UTF-16BE for compatibility with PDF 1.7). Certain other PDF 2.0 features such as Collection **Folders**, Document Part Metadata and new logical structure features (e.g. namespaces) can be written to PDF 1.7 but will be ignored (treated as unknown private data) by legacy software.*

7.1.2 Pages and content sets

Page content comprises the graphic operators and operands that describe the painting operations necessary to draw a PDF page. For the purposes of EA-PDF, PDF/A defines all necessary requirements to ensure a fully device-independent and reliable static page appearance for the visual representation of emails. PDF/A also defines all additional font requirements to ensure extractable and searchable Unicode text are present.

[8] ISO 32000-2:2020 deprecated the Document Information dictionary in preference for XMP metadata streams while PDF/A-4 only permits Document Information dictionaries with **ModDate** entries.

EA-PDF Creation Software is responsible for ensuring that the page content and related font information is correct according to PDF/A requirements. EA-PDF does not otherwise prescribe how emails are to be rendered.

EA-PDF files will have multiple "sets" of pages that are related – for example, a set of pages for a `text/plain` email body, a different set of pages for a `text/html` body of the same email, front matter, conversion report(s), sets of pages for lists of attachments or embedded files, etc. Although PDF`/mail-1{c,ci}` container files do not directly contain pages from email, they may contain front matter, conversion reports, etc. In EA-PDF each of these sets is referred to as a *Content Set*, where each *Content Set* is a sequential set of pages that always start on a new page. Each *Content Set* is always at least 1 page in length resulting in EA-PDF files always having at least one page.

Reason: having each Content Set start on a new page makes the page extraction of specific emails (or renderings of emails) easier, simpler and supported by more software, as well as supporting less capable non-EA-PDF aware legacy viewers that do not support all forms of destination and sub-page navigation.

Note: there are no limits to the number of pages that a PDF file may contain, however some legacy software may have restrictions.

Note: PDF`/mail-1{c,ci}` files require PDF software that supports PDF Collections (PDF 1.7), otherwise only the container PDF may be accessible and no folder hierarchical will be visible. Not all legacy software can display an embedded files list.

There are three distinct PDF features EA-PDF leverages to support *Content Set* understanding and machine validation of EA-PDF:

1. Outlines (also known as "bookmarks") – for user navigation;
2. Logical structure and Tagged PDF[9] – content semantics with limited validation when present;
3. Document Part Metadata (also known as DPart/DPM) – for richer capabilities and stronger validation.

The PDF outline feature (commonly known as "bookmarks") must mirror the *Content Set* hierarchy for easy navigation in most interactive PDF viewers. *EA-PDF Creation Software* may also decide to add additional outline entries for longer emails, semantically rich emails (e.g., those with headings), or other use cases that support users navigating an EA-PDF file in more detail in interactive viewers. However, this is insufficiently deterministic for software, not the least because the text of each bookmark is flexible and might be localized.

EA-PDF Writer: this specification does not prescribe the ***Title*** *text to use in outline nodes, not the least because of localization and the variety of emails. To support the widest range of non-EA-PDF aware legacy interactive viewers, it is recommended to use either* PDFDocEncoding *(effectively US ASCII) or UTF-16BE encoding and to keep text relatively short (as some viewers do not support resizing or wrapping of bookmark text). Color (****C****) and styling (****F****) can also be used, but support in legacy software varies.*

[9] The PDF 1.7 standard structure elements are valid in both PDF 1.7 (PDF/A-3) and PDF 2.0 (PDF/A-4) files.

To support a sematic understanding of *Content Sets* by EA-PDF aware software, logical structure and Tagged PDF can also be used. These PDF features are not direct user navigational features and are not widely supported in legacy software. The logical structure requirements specified in this document are optional, limited to email header representation, and only reference the PDF 1.7 standard structure elements (which is also the default standard structure set in PDF 2.0). More detailed additional logical structure and semantic tagging may be added by EA-PDF Writers, including the use of a custom tag set defined by this specification.

The addition of standard tagged PDF and logical structure semantics is *strongly recommended* for richly formatted emails, such as HTML bodies. For PDF/A-4 files, EA-PDF also defines a custom namespace.

When present, EA-PDF (like PDF) requires that the logical structure tree root structure element is always a *single* *Document* structure element, representing the entire EA-PDF file:

- For `PDF/mail-1{s, si, m, mi}` files, every email is represented by a nested *Document* structure element, directly nested below the top-level *Document* structure element that represents the `PDF/mail-1` file itself. Again, each email may use the `Mail_Message` custom EA-PDF tag which is always role mapped to *Document*.

PDF/mail-1{s, si}	**PDF/mail-1{m, mi}**
`Document` -- the EA-PDF file	`Document` -- the EA-PDF file
`Part`	`Part`
`Art` -- front matter for single email	`Art` -- front matter for MBOX
`Document` -- the only email	`Document` – 1st email
`Art` -- email headers	`Art` -- email headers for 1st email
`Art` -- HTML message body	`Art` -- HTML message body
`Art` -- plain text message body	`Art` -- plain text message body
`Part`	`Document` – 2nd email
`Art` -- attachments list	`Art` -- email headers for 2nd email
`Art` -- conversion report	`Art` -- plain text message body
	`Art` -- special report for 2nd email
	`Document` – 3rd email
	`Art` -- …
	`Part`
	`Art` -- attachments list for all emails

Table 3: Example logical structuring of PDF/`mail-1{s, si, m, mi}` files

- In `PDF/mail-1{c, ci}` container files this top-level *Document* structure element represents the container PDF and its related *Content Sets* (*not* emails, as these are in the embedded files in the collection). Thus the `Mail_Message` custom EA-PDF tag will never occur in `PDF/mail-1{c, ci}` container files.

Optional *Part* structure elements or `Mail_ContentGroup` custom EA-PDF tags (which is always role mapped to *Part*) may also be used to represent additional hierarchical structure, but there will always be *Art* (article) child structure elements or `Mail_ContentSet` custom EA-PDF tags (which are always role mapped to *Art*) of the implicit or explicit *Document* or *Part* structure elements that can be used to semantically encapsulate each *Content Set* in all EA-PDF profiles.

> *Note: the structure element sets formally defined in ISO 32000-1:2008 (PDF 1.7) and ISO 32000-2 (PDF 2.0) are different, however for the purpose of EA-PDF Content Sets only nested implicit or explicit* Document, Part and Art *structure elements from the PDF 1.7 standard structure element set are utilized. To enable the broadest range of non-EA-PDF aware legacy software, PDF 2.0-only standard structure elements (e.g.,* DocumentPart*) are not mandated.*

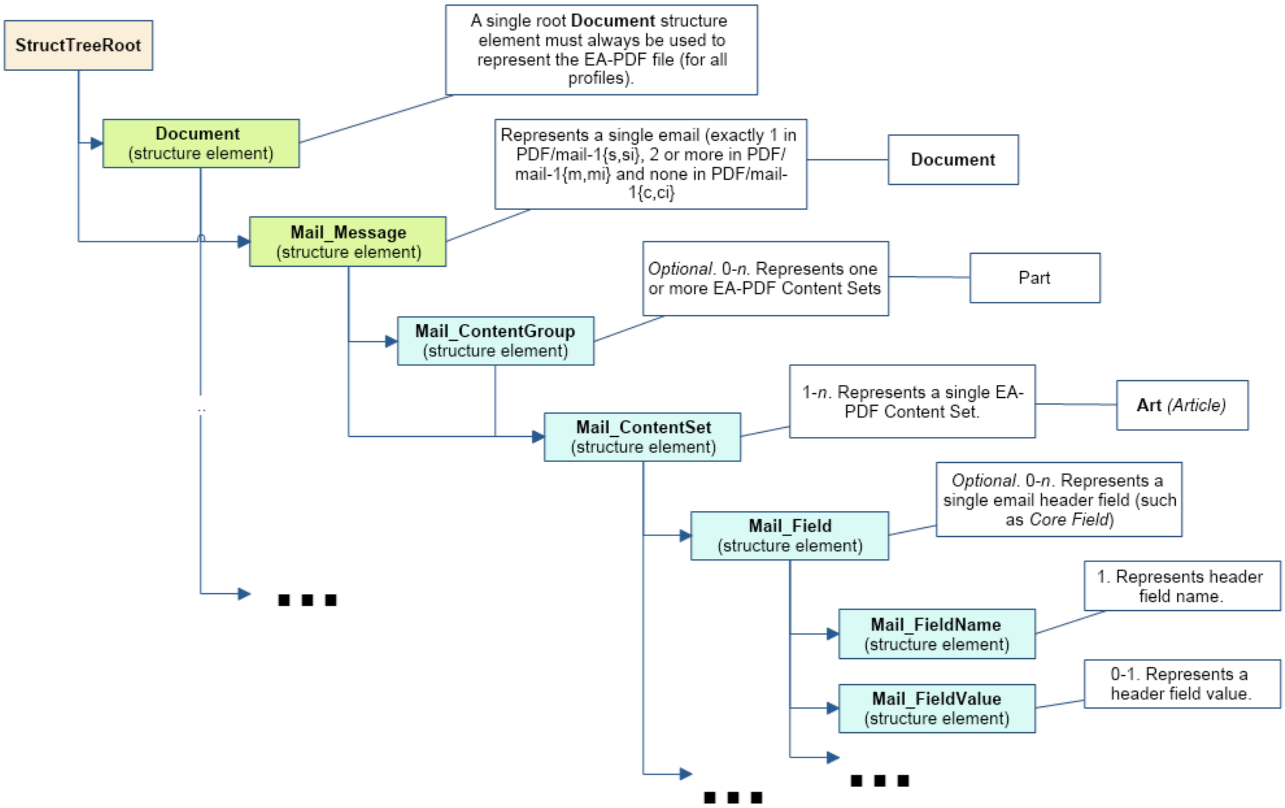

StructTreeRoot
A single root **Document** structure element must always be used to represent the EA-PDF file (for all profiles).
Document (structure element)
Represents a single email (exactly 1 in PDF/mail-1{s,si}, 2 or more in PDF/mail-1{m,mi} and none in PDF/mail-1{c,ci}
Document
Mail_Message (structure element)
Optional. 0-*n*. Represents one or more EA-PDF Content Sets
Part
Mail_ContentGroup (structure element)
1-*n*. Represents a single EA-PDF Content Set.
Art *(Article)*
Mail_ContentSet (structure element)
Optional. 0-*n*. Represents a single email header field (such as *Core Field*)
Mail_Field (structure element)
1. Represents header field name.
Mail_FieldName (structure element)
0-1. Represents a header field value.
Mail_FieldValue (structure element)
■ ■ ■
■ ■ ■
■ ■ ■

Figure 7: Conceptual illustration of logical structure using the custom EA-PDF tag set

To enable improved reuse and accessibility, EA-PDF additionally defines a set of custom tags (structure element types) for optionally semantically marking up PDF page content streams to identify *Core Field* information using a custom PDF 2.0 namespace. All keys are purposely prefixed with the registered second-class PDF name "`Mail`" followed by an underscore (i.e. `/Mail_`…) so that identifiable semantics remain after content or page extraction or processing by legacy software which may not support or maintain custom logical structure. The role-mapping of some custom EA-PDF structure elements back to approximate standard structure types is *not* defined, allowing flexibility in the way that *Core Field* information is presented.

EA-PDF Writer: different writers may choose to present the Core Field information as spans, paragraphs, lists, or in a tabular format. Having flexibility in the role-map allows EA-PDF writers to most appropriately provide the "approximate equivalents" in the PDF 1.7 standard structure elements for their presentation choice.

The PDF 2.0 feature called Document Part Metadata (§14.12, [ISO 32000-2] – also referred to as **DPart**/**DPM**) is also used by EA-PDF[10]. This is a tree-like data structure using PDF object syntax, like the logical structure tree (but smaller), that can additionally express high-level semantics or associate data about page ranges. For EA-PDF aware software, the Document Part Metadata feature provides an additional rich programmatic (deterministic) understanding of *Content Sets* and can be used for improved validation capabilities. Only the unique email identifiers (such as `Message-ID`/`Mail_GUID`) are stored in the Document Part Metadata allowing EA-PDF aware software to deterministically and reliably map *Content Sets* back to the definitive XMP metadata.

Reason: duplicating email metadata in the DPart tree is unnecessary and makes files larger.

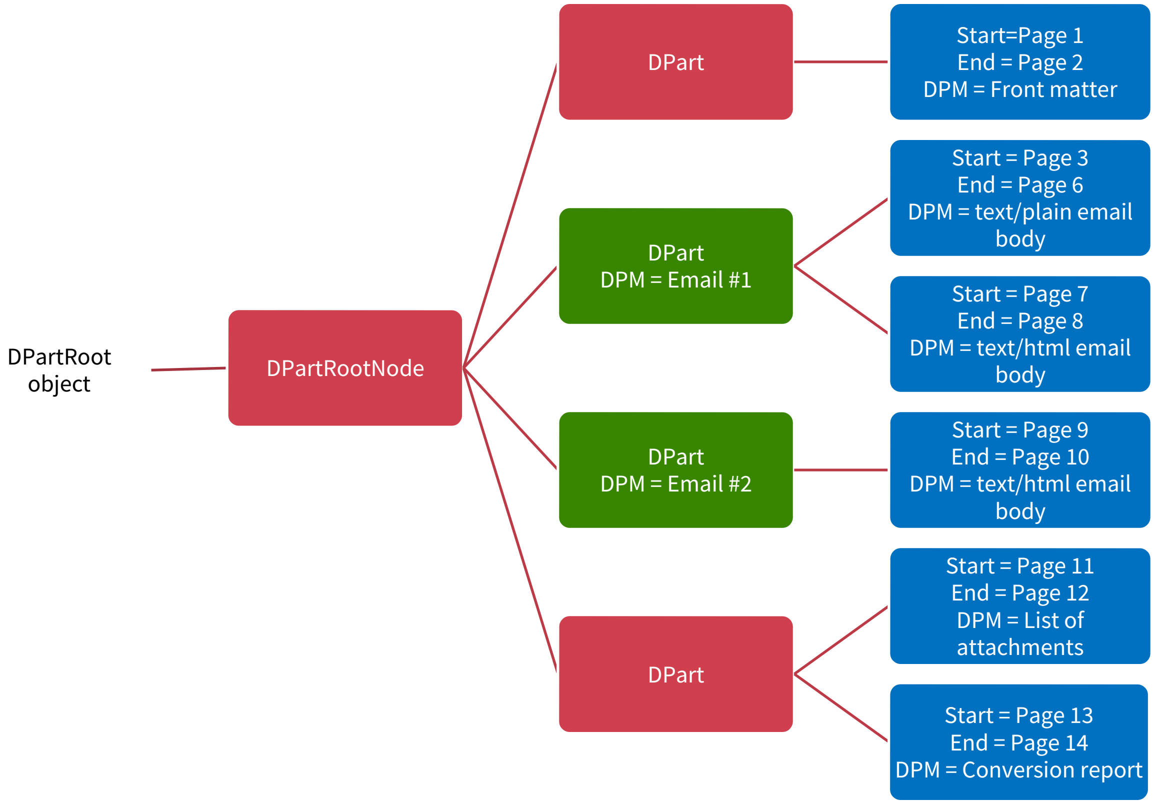

DPartRoot object
DPartRootNode
DPart
DPart DPM = Email #1
DPart DPM = Email #2
DPart
Start=Page 1 End = Page 2 DPM = Front matter
Start = Page 3 End = Page 6 DPM = text/plain email body
Start = Page 7 End = Page 8 DPM = text/html email body
Start = Page 9 End = Page 10 DPM = text/html email body
Start = Page 11 End = Page 12 DPM = List of attachments
Start = Page 13 End = Page 14 DPM = Conversion report

Figure 8: Conceptual illustration of a simplified Document Part Metadata tree

[10] Although Document Part Metadata was formally documented as a part of core PDF 2.0 in ISO 32000-2, its initial use was defined in PDF 1.6 with PDF/VT-2 files (ISO 16612-2:2010). PDF/A does not prohibit the inclusion of private data so long as that private data does not impact rendering – and Document Part Metadata has no influence on rendering and is thus acceptable in all PDF/A files utilized by EA-PDF.

The use of Document Part Metadata allows EA-PDF aware software to easily identify or logically group pages into like sets, since EA-PDF does not (and cannot) prescribe page counts for every possible *Content Set*. The **DPM** dictionaries in Document Part Metadata can also have Associated Files and **Metadata** streams to provide additional data. This is achievable *without* needing to parse page content streams and might be used to improve email-centric navigation in interactive EA-PDF aware software or to batch-process multiple EA-PDF files (e.g., extract all pages that represent a specific kind of *Content Set*, identify EA-PDF files with `text/html` page renderings, find XMP metadata streams or associated files linked to specific kinds of *Content Sets*, etc.).

7.1.3 Embedded files

PDF files can contain multiple embedded files which are often presented in a dedicated pane in the user interface of interactive PDF viewers or reported by console or server applications. The PDF file format supports various methods (data structures) for referencing embedded files in PDF including file attachment annotations, files in PDF collections, associated files, and assets associated with multimedia.

PDF does not internally contain a file system and simulates embedded files with PDF objects (specifically a file specification dictionary and associated embedded file stream). As a result, PDF supports *different* embedded files having the *same filename* and multiple references with different filenames to the same embedded data. No PDF specification or standard specifies how applications should curate the list of embedded files in PDFs nor how the list is to be presented (for example indicating different filenames for the same embedded data or contextualizing duplicate filenames). As a result, legacy PDF applications vary greatly in which files are displayed and the context of the files listed.

> *Note: some legacy applications only display embedded files associated with File Attachment annotations, other legacy applications list only those in the **EmbeddedFiles** name tree, while some list files from both sources (sometimes resulting in duplicate entries). Other viewers may require a PDF page to be viewed before embedded files associated with file attachment annotations are listed.*

Embedded files in a PDF can be utilized by several distinct PDF features relevant to email:

- File attachment annotations (§12.5.6.15, [ISO 32000-2]), typically represented as paperclip icons on pages that when clicked open the file. In an email context, these are contextually like email attachments. File attachment annotations are always associated with PDF pages and have a visual on-page representation via the annotation appearance stream.

  > *Note: embedded files associated with PDF File attachment annotations do not have to be listed in the **EmbeddedFiles** name tree, however this can mean that some legacy software applications no not detect their presence until the page is viewed.*

- Rich media assets, like 3D, movies, animations, or audio files. In an email context, these may originate as an embedded asset in the source email – or are external assets referenced by the body of a source email – and thus are needed by EA-PDF to render and accurately preserve the appearance of an email as PDF page content. As a result, they will need to be preserved as embedded file streams in EA-PDF files (note that this is a rare but distinct use-case from email attachments above).

> *Note: both PDF/A-3 and PDF/A-4 prohibit Sound, Screen, and Movie annotations, while PDF/A-4e limits support to 3D and RichMedia annotation subtypes and the capabilities of PDF/A-4f. PDF 2.0 RichMedia annotations list their assets in a separate asset name tree in the RichMedia Content Dictionary (**Assets** entry), however most legacy software applications do not make visible this list of files.*

> *EA-PDF Writer: if 3D or RichMedia content is required to represent email, then the EA-PDF file will need to conform to PDF/A-4e, which is a superset of PDF/A-4f (and thus allows other embedded files).*

- PDF Collections (§12.3.5, [ISO 32000-2], also known as "Portable Collections", "Portfolios", "Packages", or "Binders" presents hierarchical folder-like views of sets (collections) of embedded files. In an email context, this is contextually like how typical email client software presents folders and sub-folders of emails for organizing email.

> *Note: ISO 32000 PDF Collections require that all files comprising the collection are listed in the **EmbeddedFiles** name tree.*

- PDF's Associated Files feature allows specific embedded files to have a defined simple semantic relationship with a specific PDF object via the **AFRelationship** entry. For example, a GIF or PNG image used in an email cannot be directly used by PDF and must first be converted to an alternate format such as an Image XObject. The raw GIF/PNG however can still be faithfully preserved by embedding it directly into the PDF as an associated file of the Image XObject with a *Source* relationship. Associated Files were first added to PDF 1.7 by PDF/A-3 and later adopted into PDF 2.0 with an increased set of **AFRelationship** key values[11]. See [AssociatedFiles] and §14.13 [ISO 32000-2].

> *Note: Associated Files are not technically required to be listed in the **EmbeddedFiles** name tree, however for interoperability this is strongly recommended as all PDF/A conforming interactive processors must be capable of displaying information from the **EmbeddedFiles** name tree. For PDF/A-4f there must also be an **EmbeddedFiles** name tree present.*

PDF/A only permits embedded files in formats other than PDF/A in PDF/A-3 and the PDF/A-4f and PDF/A-4e conformance levels. Thus, specific EA-PDF profiles have limitations on which PDF/A version and conformance levels may be used – see *Table 2: Summary of EA-PDF profiles* above.

Note that PDF embedded files in the **EmbeddedFiles** name tree, Associated Files, or via file attachment annotations form a flat list (i.e., there is no folder or inherent hierarchy), and there is no requirement to have unique filenames, so navigation with legacy software of large PDF/`mail-1{m`, `mi`, `c`, `ci}` files may be less than ideal. EA-PDF aware software can additionally utilize other EA-PDF data to provide a far better navigation experience relevant to archival management of email.

[11] C2PA further extends the set of **AFRelationship** values in ISO 32000-2 with a value of *C2PA_Manifest.* C2PA is not precluded from use in EA-PDF.

*Note: some non-EA-PDF aware legacy software displays additional information such as the page or the value of the **AFRelationship** key. Based on experimentation with current interactive software, embedded files that have a page number will generally relate to file attachment annotations, whereas those without are likely related to the entire document – but this is not mandated.*

PDF file specification strings with absolute or relative paths (see §7.11.2 [ISO 32000-2]) must not be used.

*Note: although PDF file specification strings can specify both absolute and relative paths with filenames, most existing legacy PDF viewers do not support this feature and thus these features are prohibited in EA-PDF. The PDF Portable Collections **Folders** feature is used to preserve folder hierarchies in* `PDF/mail-1{ci, ci}` *files.*

The PDF embedded file stream dictionary ([ISO 32000-2], Table 44) only records the IANA media type as the top-level media type and its description as the **Subtype** value, with IANA media type parameters not permitted. Thus EA-PDF adds an optional custom entry **Mail_MediaTypeParameters** to record any media type parameters in the embedded file stream dictionary.

Note: see PDF Errata #155.

EA-PDF Creation Software may also wish to identify duplicated attachments and store the embedded file data just once in the EA-PDF file to optimize file size. Common scenarios include `winmail.dat` files and email threads where the same attachment may be included multiple times. This document does not specify any specific algorithm, but any such optimization must be based on the binary content of the embedded files and not just the filename. Since PDF also allows referencing the same embedded file data stream from multiple file specification dictionaries, even if the same file is differently named across one or more emails, EA-PDF can embed it only once and efficiently reference via different filenames.

7.1.4 URLs and hyperlinks

URLs or hyperlinks in PDF content that are intended to be actionable by end users need to created using PDF Link annotations (§12.5.6.5 [ISO 32000-2]).

EA-PDF Reader: although a good security practice, not all PDF viewing software explicitly confirms URL links with the user before activation.

Note that some software may also automatically detect other links in content and make them actionable, even though no PDF Link annotation is explicitly present for that content.

EA-PDF Writer: if this automatic behavior is undesirable and the choice of viewing software cannot be controlled, then possible options available to EA-PDF Creation Software *include altering the PDF page content so that standard URL link detection fails (e.g., replace “*`https:`*” with “*`hxxps:`*”) or to add dummy Link annotations (e.g., using a **Dest** rather than URI action via the **A** entry). Such methods are not guaranteed.*

7.2 Preserving source email assets

A critical aspect of establishing EA-PDF as a suitable preservation format is the mandated preservation and embedding of the original raw source email asset(s) in the EA-PDF file.

When PDF/`mail-1{s`, `m`, `c}` files are directly created from an email source asset (e.g., EML, MSG, MBOX, NSF, OST/PST, etc.), that *exact unmodified asset* must also be embedded via the Document Catalog **AF** (Associated Files) array entry in order to create a verifiable preservation asset. This results in an Associated Files array element referring to a file specification dictionary with an **AFRelationship** entry of *Source* (see Table 43, [ISO 32000-2]) and with a **Subtype** value appropriate for Media Type of the original email file format. In non-isolated profiles, there must be at least one array element in the Document Catalog where **AFRelationship** is *Source*.

> *Reason: This ensures that the original email file used at EA-PDF creation is always faithfully preserved and clearly identifiable. Note that **AF** is always an array (even if there is only a single associated file), even though several PDF producers currently generate malformed PDFs using a dictionary!*

> *Note: EA-PDF does not require that only the rendered emails are present in the source email assets. Thus, preserving an MBOX or PST file with multiple emails in PDF/`mail-1s` where only a single email is rendered is valid although potentially inefficient.*

If multiple emails from different original email assets need to be preserved as a single EA-PDF file (e.g. a folder with multiple EML or MBOX files), then *EA-PDF Creation Software* has different options:

- create a PDF/`mail-1c` collection with individual EA-PDF files (PDF/`mail-1{s, m}`), each with its own independent trusted original source email asset. This is the preferred approach with the best support across legacy software and ensures the extracted EA-PDF files also retain preservation qualities;
- combine the email assets into a single new email asset (e.g., concatenating EML or MBOX files or merge all OST/PST files) prior to creating an EA-PDF file (such as a PDF/`mail-1m`);
- embed all original source email assets, each with an **AFRelationship** value of *Source*, and rely on EA-PDF aware software to use `Message-ID`/`Mail_GUID` metadata to map each email to the correct original email source asset; or
- use an EA-PDF isolated (`i`) profile.

In all cases, all pre-processing steps ought to be recorded in the provenance metadata.

There is no requirement that *only* the email that is being preserved must be in the original raw email asset - additional data may also be present that is not represented by any pages in the EA-PDF file (e.g. calendar or contact entries in OST/PST, additional emails in MBOX).

> *EA-PDF Writer: if the source email data format is a multiple-email format (such as MBOX, OST/PST or NSF) and PDF/`mail-1s` files are being created with single email messages, then the EA-PDF Writer may elect to either export*

each email to a singular format (e.g. EML) and then convert that export to PDF/`mail-1s`, or to embed the full multi-email file as-is (however this may be large). This is a policy decision outside of the EA-PDF file format.

If a source email format requires multiple files to be preserved as a set, then a Related Files array might be used (see §7.11.4.2, [ISO 32000-2]) or the additional related files can be stored as separate embedded files in the Document Catalog **AF** array with an **AFRelationship** of *Supplement*.

EA-PDF Writer: it is not desirable to use a format such as ZIP to store a set of source email asset(s), as the ***Subtype*** *of the embedded file stream would then be `application/zip`, and the actual data formats of the original source email assets inside the ZIP is hidden. Instead, use the standard PDF FLATE and LZW compression filters that can losslessly compress the raw original email source data.*

The preserved source email asset(s) in PDF/`mail-1{s, m, c}` files must reflect <u>at least all</u> emails that are represented in that EA-PDF file. If only a subset of emails in an MBOX, PST/OST, NSF, etc. are converted to EA-PDF, then the full original raw source email asset (with the additional information) can be embedded. However, it is <u>not</u> valid to store a source email assets with an **AFRelationship** of *Source* that do not contain <u>*all*</u> the source email content in a PDF/`mail-1{s, m, c}` file (i.e., every email represented in PDF/`mail-1{s, m}` files must be associated with data in the embedded source original email asset(s)) – otherwise the "isolated" profiles must be used (PDF/`mail-1{si, mi}`) to indicate that not all original source email assets are faithfully preserved.

All embedded source email assets need to be listed in the Document Catalog **Names** name-tree **EmbeddedFiles** entry (§7.7.4, [ISO 32000-2]). All PDF file specification dictionaries in EA-PDF files ought to include meaningful descriptive text.

Reason: Many legacy software viewers use the Document Catalog Names name-tree ***EmbeddedFiles*** *entry to display and access embedded files. Embedded files not listed in the* ***EmbeddedFiles*** *entry may or may not get displayed in some legacy viewers. In some cases, embedded files associated with file attachment annotations may not show at all or may not appear until the associated page is scrolled into view.*

```
10 0 obj
<< /Type /Catalog
   /Metadata 20 0 R % the required XMP metadata for this EA-PDF file
   … other document catalog key/values …
   /AF [ 11 0 R ] % associated files array containing the source email
   /Names <<
      /EmbeddedFiles << % name tree mapping strings to File specification dicts
         /Names [
            …
            (93910.msg) 11 0 R % associated file listed somewhere in name tree
            …
         ]
      >>
      … % other PDF name trees as per PDF specifications
   >>
>>
endobj
```

```
11 0 obj
<< /Type /Filespec
   /Desc (Preserved source email file 93910.msg)
   /F (93910.msg)
   /UF (93910.msg)
   /AFRelationship /Source % identified as the faithfully preserved original email asset
   /EF << /F 12 0 R >>
endobj

12 0 obj
<< /Type    /EmbeddedFile
   /Subtype /application#2Fvnd.ms-outlook % IANA Media Type for ".msg". No parameters
   /Params << % required for embedded files used as Associated Files
      /ModDate  (D:20000901104905) % required for embedded files used as Associated Files
      /CheckSum <f1e884313db0d133ea409b7043c35288>  % 16 byte MD5 as PDF hex string
      /Size     342324
      …
   >>
   /Filter /FlateDecode % compressed to save space in PDF
   /Length …            % compressed length inside PDF file
   /DL 342324           % decompressed length - same as Params/Size
>>
stream
… FLATE compressed .msg data file (binary) …
endstream
endobj
```

Example 1: Document Catalog referencing the source email file as an Associated File and also listed in the **EmbeddedFiles** name tree.

7.3 Email headers

7.3.1 Core fields

EA-PDF defines a set of common email header fields and related attributes of each email as *Core Fields*. Each email will have its *Core Fields* reflected in the document-level XMP of the PDF `/mail` file that contains the email. The values used in the XMP must be as equivalent as possible to the full value in the source email assets, subject to representation/encoding differences between the source email and UTF-8 based XMP.

Reason: as discussed in the LWG, the document level XMP metadata is the primary and definitive source of email metadata used by document and archival management systems to manage email archives. Core Field names must be consistent to enable the most reliable searching across a diverse corpus of EA-PDF files and hence must not be localized or vary between EA-PDF Writers. EA-PDF Writers may however localize core fields when it renders this same information to PDF page content (e.g., "Subject:" might be rendered as "Objet:" for French emails, but in XMP it will always be "Subject" – the value of the field in both cases needs to be the same and not localized).

Note: because the rendering of Core Fields in page content may be truncated, cropped, wrapped, localized, etc., the XMP Core Fields are to be considered the definitive "source of truth" for all EA-PDF aware software.

The set of *Core Fields* defined for each email stored in EA-PDF are listed in *Table 4: EA-PDF Core Fields* below. *Core Fields* names usually correspond to the matching email header field name, however *EA-PDF Creation Software* may add additional email header fields prefixed with "`Raw-`" to indicate a raw value from the email that would otherwise be an error when using a more rigid or structured XMP data type (see `Sent` vs. `Raw-Sent` below). To help minimize XMP Metadata size, most blank fields do not have to be stored.

Core Field Property	Source	Condition
`To`	Email header	If present and not blank/empty.
`From`	Email header	If present and not blank/empty.
`Sent`	Conversion from email header	**Required** (even if blank/empty) *Note: RFC 822 defines this field as "Date" which can be ambiguous when out of context. EA-PDF uses the term "Sent".*
`Raw-Sent`	Email header	**Conditionally Required** if the email header `Sent` field date value contains an error that cannot be identically represented as a valid XMP date/time. Optional otherwise. *Note: RFC 822 defines this field as "Date" which can be ambiguous when out of context. EA-PDF uses the term "Sent".*
`Subject`	Email header	If present and not blank/empty.
`Message-ID`	Email header	If present and not blank/empty.
`Cc`	Email header	If present and not blank/empty.
`Bcc`	Email header	If present and not blank/empty.
`In-Reply-To`	Email header	If present and not blank/empty.
`Content-Type`	Email header	**Required** (even if blank/empty)
Mail_GUID[12]	Created by EA-PDF software	**Required**. Unique for the email and that is generated by the *EA-PDF Creation Software*.
Original email message size (*in bytes*)	Directly from email asset	**Required.**
Number of email attachments	Calculated from email	**Required.**

Table 4: EA-PDF Core Fields[13]

All non-blank *Core Fields* are *strongly recommended* to also be rendered into page content using PDF text objects.

[12] GUID = Global Unique Identifier. EA-PDF does not prescribe how GUIDs are generated and does not distinguish between the terms GUID and UUID (Universally Unique Identifier, see also RFC 4122).

[13] Note that some *Core Fields* in email, such as Message-ID and Subject, are optional according to [RFC-822].

Reason: using PDF text objects in PDF/A compliant files such as EA-PDF ensures that non-EA-PDF aware legacy software with text searching or extraction capabilities ought to be able to find Core Fields in page content, even if they do not support XMP.

Reason: if rendering of Core Fields into visible page content was mandated by a "SHALL" requirement then to make this validatable by software without human checks requires additional complexity including reading the original source email asset. Otherwise, machine validation would be unreliable as emails containing replies, forwarded emails, or other content with similar text may ambiguously look like certain Core Fields. PDF/A files currently do not contain any content requirements and can be fully machine validated by software without human checks and without reference to any original document - EA-PDF aims for this same level of efficiency.

Emails are generally considered identifiable by the standard email header `Message-ID` field, however not all emails in EA-PDF files may have a `Message-ID` (e.g. draft or unsent emails). When present, the `Message-ID` will always be present in the definitive XMP metadata to allow easy discovery (even by non-EA-PDF aware software), but for the purposes of defining links or relationships between PDF data in or between EA-PDF files, the *Core Field* `Mail_GUID` must always be used.

EA-PDF Writer: XMP is very flexible in its definition of GUID (see [XMP], §8.2.2.3 – it is just a string) so the email `Message-ID` field can simply be used when it is present. When not present however, EA-PDF Writer must generate something that is globally/universally unique and this is why `Mail_GUID` is mandated.

EA-PDF Creation Software is otherwise free to choose how to layout and render the appearance of the *Core Fields* into page content, including using advanced renderings or layouts that might simulate rich email client user interfaces with additional graphics or images.

Reason: the rendering of Core Fields can mimic the appearance of rich desktop email clients, but importantly must be text content for non-EA-PDF aware legacy software to find using their text search functionality (this also occurs because of the requirements of the PDF/A conformance levels used with EA-PDF). The choice of where and how the Core Fields are rendered, and their appearance is not further prescribed by EA-PDF.

EA-PDF Writer: as discussed in the LWG, EA-PDF Creation Software *may also decide to repeat some or all Core Fields in the header or footers of PDF pages so that extracted pages may retain some form of human identifiable context information. This is not mandated but recommended.*

Note: For some emails (e.g., unsent drafts), the value of Core Fields may not be present. Requirements are worded such that the name of each Core Field is rendered thus giving a visual indication to users that the corresponding Field Value is blank. This recommendation and PDF/A both require the use of PDF text objects, ensuring that text search in non-EA-PDF aware legacy software will work (as required by the specified PDF/A conformance levels). However, this requirement does not mean that the English field names defined in the email must be used, thus allowing EA-PDF to mimic support of non-English email clients in non-English environments on the rendered pages, but with the XMP metadata being technically equivalent to the original source email asset.

*EA-PDF Writer: text rendering mode 3 (**Tr** operator) supports invisible text that can still be searched and extracted by most PDF software, even if external libraries render email appearances to bitmaps.*

The positioning and layout of any additional rendered *Core Fields* ought to be clearly distinguishable from the rendering of email Message Bodies.

Note: this is a recommendation only, as it would otherwise require a human to validate. EA-PDF files need to be fully machine validatable like PDF/A. Note that the document XMP metadata for all email header fields (including Core Fields) can be machine validated.

7.3.2 Other header fields

EA-PDF Creation Software may choose to render other email *Header Fields* to PDF pages or add other email *Header Fields* to the document-level XMP metadata referenced from the document catalog **Metadata** entry (Table 29, [ISO 32000-2]).

EA-PDF Writer: Since each EA-PDF Creation Software *is free to choose how to render* Header Fields*, the appearance of an EA-PDF email may vary between implementations.*

EA-PDF also defines a custom PDF 2.0 namespace with structure elements to additionally support richer tagging and association of email header fields and their values within content. EA-PDF does not mandate the precise role mapping back to the PDF 1.7 standard structure elements.

```
110 0 obj  % PDF 1.7 standard structure namespace as per 14.8.6 in ISO 32000-2
<< /Type /Namespace /NS (http://iso.org/pdf/ssn) >>
endobj

111 0 obj  % see Table 356 in ISO 32000-2:2020
<< /Type /Namespace
   /NS (https://pdfa.org/ns/ea-pdf/mail-1) % required namespace URI for PDF/mail-1
   /RoleMapNS <<     % Role map from EA-PDF back to PDF 1.7 standard structure elements
      …
   >>
>>
endobj
```

Example 2: PDF 2.0 logical structure namespace dictionary for EA-PDF custom structure elements.

7.4 Message body(s)

7.4.1 Common requirements

EA-PDF does not prescribe page sizes, page layout, content reflow, wrapping, page breaks, appearance, etc. within *Content Sets*. All textual content in the email must be represented using PDF text objects and, because EA-PDF files also conform to specific PDF/A conformance levels, all such text will have identifiable Unicode codepoints.

EA-PDF Writer: EA-PDF Creation Software *is free to choose page size (media size), margins, scaling factors, headers, footers, wrapping, etc. However, all textual content in the email body must be represented in EA-PDF files as text objects so that text search is possible across all software, which is also enforced by requiring PDF/A conformance. This does not preclude the rendering of email to images so long as the textual content is also added (like is often done with scan-to-PDF and OCR solutions that then use text rendering mode **Tr** 3 [§9.3.6, ISO 32000-2]). This rendering of email to images is undesirable and inefficient but is not technically prohibited by EA-PDF or PDF/A conformance.*

All pages in PDF`/mail-1` files ought to have a visible page number with equivalent page label (see §12.4.2 [ISO 32000-2]). If Tagged PDF is also used, then these page labels can be tagged as artifacts.

Reason: As desired by the LWG, this allows individually extracted pages from an EA-PDF file to remain visibly identifiable even when extracted with non-EA-PDF aware software. Equivalent page labels also assist with logical navigation in legacy viewers. This specification does not specify the style, where, or how page numbering or identification labelling is placed on a page (such as the use of some email header fields), but typically headers or footers are used.

7.4.2 Richly formatted email body formats

Richly formatted email body formats such as HTML and RTF define formatting and semantics that can be re-applied in PDF using known techniques. *EA-PDF Creation Software* will need to make its own reflow, wrapping, pagination and layout decisions as well as mapping email semantics to Tagged PDF and Logical Structure if so desired.

Note: semantics here refer to the type of content, such as a heading, paragraph, table, ordered or unordered list, etc. Semantics are represented with tags and attributes in HTML, whereas PDF uses Logical Structure and Tagged PDF.

Where the color space of email message body content is not explicitly defined in the source email format, sRGB is to be assumed and appropriate device-independent PDF color space objects defined (as required by PDF/A).

Reason: PDF/A requires the use of device independent color spaces to ensure a consistent and reliable appearance across devices.

*EA-PDF Writer: sRGB support in PDF/A is best done via an **ICCBased** color space object, which only ever needs to be embedded once in a PDF file and can be reused as necessary. See also §8.6.5.6 Default colour spaces in [ISO 32000-2].*

For PDF/`mail-1{s, si}` and PDF/`mail-1{m, mi}` files that contain pages representing richly formatted email bodies with existing semantic information, PDF logical structure and Tagged PDF is *strongly recommended*.

Reason: Richly formatted emails such as HTML and RTF contain their own semantics, so creation of equivalent PDF data is possible by EA-PDF Creation Software*, but this is not mandated since it cannot be meaningfully validated without reference to the original source email data.*

Note: it is not mandated that EA-PDF files are also PDF/UA compliant (in addition to the requirement to be PDF/A compliant) as this imposes additional requirements. However, EA-PDF Creation Software *ought to retain an equivalent level of semantics of the source email content (e.g., any semantics represented by the HTML tags of HTML email bodies ought to be retained when converted to PDF), however determining appropriate semantics from plain text emails is implementation dependent.*

7.4.2.1 Referenced assets

Subject to the policy environment, *EA-PDF Creation Software* may additionally decide to preserve assets referenced by the original raw source email (e.g., images, SVG, rich media, etc.), including fetching and preserving external assets from the internet. When such assets are not natively compatible with PDF, the raw asset may also be saved (preserved) in the EA-PDF file as an Associated File (in an **AF** array) to the most relevant PDF object (e.g., Image XObject, Form XObject, font, etc., but <u>not</u> the Document Catalog[14]) with an **AFRelationship** of *Source*. See [AssociatedFiles].

[14] Associated Files with **AFRelationship** *Source* in the Document Catalog **AF** array refer to the original raw source email assets for non-isolated PDF/`mail-1{s, m, c}` files.

Missing or corrupted assets may be visually indicated in EA-PDF output, but this is not mandated.

EA-PDF Writer: for example, if an image is not to be included in the PDF page rendering, then a bounding box rectangle might be shown along with other information as to the reason. It is not mandated because it cannot be validated without reference to the original source email asset and policy environment.

Additionally, conversion information or other provenance information about referenced assets (such as set by an archival policy) may also be included in the EA-PDF file ideally as an XMP **Metadata** stream associated to the most relevant PDF object (e.g., Image XObject, Form XObject). See [MetadataStreams] and §14.3.2 [ISO 32000-2].

*EA-PDF Writer: as illustrated below, an animated GIF image in an email might be converted to a static JPEG (***DCTDecode***) image for inclusion in the rendered representation in PDF. SVG might be converted to a Form XObject or rendered to a canvas and embedded into the EA-PDF as a bitmap (Image XObject). The source asset and XMP metadata would then be associated with those PDF objects.*

Reason: Fetching and/or storing of external assets referenced from email bodies into EA-PDF is an archival policy matter (and not a file format requirement) due to overhead and potential tracking and privacy issues. Storing of all assets is not mandated because of the impact to file size – embedded assets in the source email will already be preserved in the source email assets and thus do not have to be embedded again. Externally referenced assets might also be fetched to assist with page layout algorithms but not saved into the EA-PDF. In such cases, a proxy graphic or image of similar dimensions might be added by the EA-PDF Creation software*. Machine validation is also not possible.*

Referenced assets that need conversion and result in an image ought to use Image XObjects and not inline images.

Note: avoiding inline images in PDF content streams for converted assets means that Image XObjects must be used (inline images can still be used for other purposes however). This is because a lot of legacy software only supports Image XObjects when looking for metadata, performing image extraction, etc. As per §8.9.7, [ISO 32000-2] inline images are also only appropriate for very small images (4096 bytes or less).

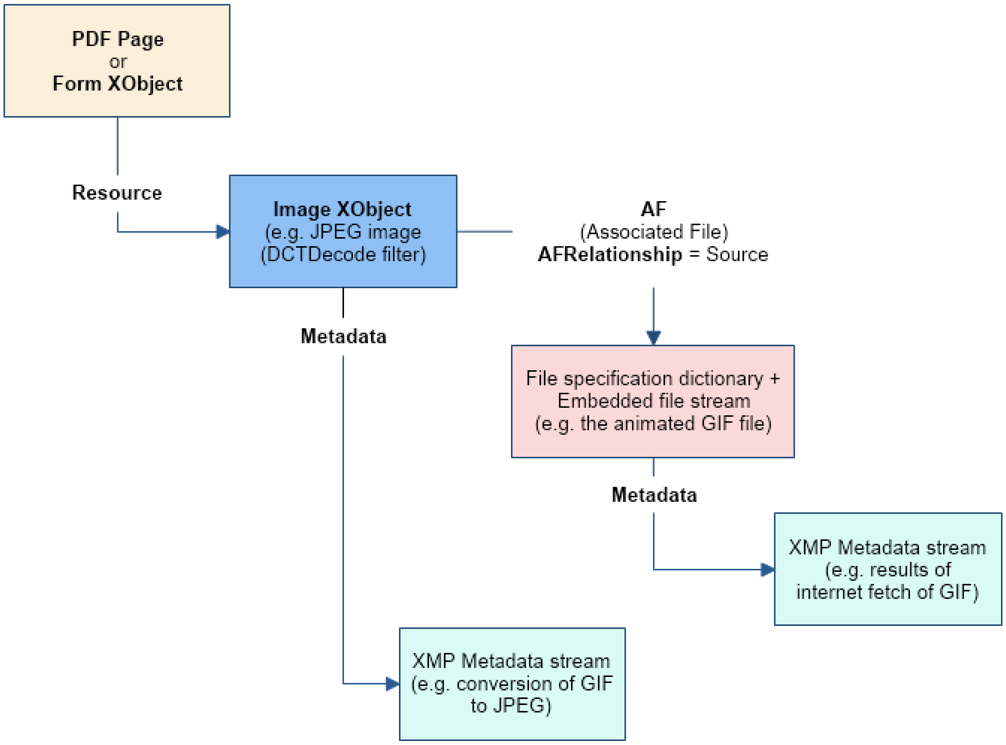


Figure 9: Example of multiple Associated Files and Metadata streams associated with a converted asset (e.g. animated GIF converted to Image XObject)

7.4.3 Plain text emails

Email bodies which are plain text (e.g., "`Content-type: text/plain; charset=…`" or equivalent) lack the native formatting and semantics present in rich email body formats such as HTML or RTF. When creating PDF`/mail-1{s, si, m, mi}` files, *EA-PDF Creation Software* must make additional formatting decisions such as typeface selection, font size, text color, as well as other layout and pagination decisions to convert the plain text email to typeset and formatted PDF page content that is PDF/A compliant. These decisions are not mandated by this specification.

Note: by convention plain text emails are often displayed in email clients using monospaced fonts, such as Courier. This document does not mandate this convention and thus users and EA-PDF aware software are advised not to assume that a monospace appearance implies a plain text email – but other data structures used by EA-PDF can ensure that users know if an email rendering is of a text-based email body. PDF/A compliance does however require that all referenced fonts are always embedded.

Note: although many email clients allow users to select text foreground and background colors, font size, and possibly other font properties when displaying plain text emails, the use of black text on the default white page background that PDF defines is preferable for EA-PDF use-cases but is also not mandated. PDF/A compliance however does require that device independent color is always used to ensure a reliable device-independent appearance.

For PDF`/mail-1{m, mi}` files with multiple emails, *EA-PDF Creation Software* ought to optimize font usage and other resources across multiple emails to reduce the overhead of embedded fonts in the resultant EA-PDF file.

EA-PDF Creation Software may add semantic markup (via Logical Structure and Tagged PDF) for plain text emails through various additional means that are beyond the scope of this document.

> *Note: some email clients have their own algorithms for special processing of additional end-of-line characters in text emails which can result in different displays across email clients.*

7.4.4 URL hyperlinks

Although richly formatted email bodies may contain explicitly marked-up URLs (such as using HTML `<a href="…">…</a>` tags), many modern email clients also detect and make active (clickable) other URLs found in emails, including plain text email bodies which do not contain any markup. Active URLs in email clients are commonly shown as underlined colored text. This difference in behavior is oftentimes not obvious to users and can mean that the presentation and functionality of email can look different across different email clients (i.e., which URLs are active and which are not).

In PDF, active (clickable) URLs are enabled via the use of PDF Link annotations[15]. *EA-PDF Creation Software* is free to decide not to create any PDF Link annotations (so URLs only appear as text), only create PDF Link annotations for those explicitly marked-up URLs in richly formatted email bodies, detect URLs in the content of email bodies (including plain text emails that contain no explicit markup) and make some or all those links active via PDF Link annotations, or some other strategy.

> *Reason: The archival policy settings ought to define such behavior, since following links may invoke side effects, privacy/PII, tracking, or other undesirable issues.*

> *Note: many legacy PDF viewers will automatically detect and make actionable text that appears to be a URL even if it does not have an associated PDF Link annotation. Such behavior may be controlled by a viewer option, but this is vendor specific and beyond the scope of the ISO 32000 or this specification.*

> *EA-PDF Writer: one approach to try and protect against this behavior is for* EA-PDF Creation Software *to explicitly add dummy Link annotations where the link action is harmless (e.g. goes to a local destination in the same EA-PDF file, rather than a URL on the internet).*

7.5 Email attachments

All email attachments are faithfully preserved in EA-PDF files as file specification dictionaries with an embedded file stream (see §7.11.4, [ISO 32000-2]) linked to one or more File Attachment annotations (§12.5.6.15, [ISO 32000-2]). Each file attachment annotation is associated with a page.

> *Reason: almost all legacy software viewers provide basic support for PDF file attachment annotations. File attachment annotations are typically visually represented by paperclips on pages which is a similar metaphor used by many email clients. Technically, the embedded file stream associated with file attachment annotations are not required to be listed in the Document Catalog Names name-tree* ***EmbeddedFiles*** *entry.*

[15] PDF page content that *looks like* URL text is not active (clickable) unless a PDF Link annotation is also created. The appearance of the URL text in the PDF page content has no influence on whether a URL is active (e.g. blue underlined text is not active unless a PDF Link annotation is also created).

Note: the file specification string associated with the embedded file stream that represents an email attachment is expected not to contain folder names, "`..`", or other path or platform components. PDF file specification strings with absolute or relative paths ([ISO 32000-2], §7.11.2) must not be used with EA-PDF.

Note: email attachment filenames used are not guaranteed to be unique.

Note: by design, multiple file attachment annotations can efficiently refer to the same embedded file stream in the PDF which provides flexibility for additional reports, etc. while optimizing for file size.

The PDF File Attachment annotation must contain a **Contents** entry and the file specification dictionary of the embedded attachment ought to have a meaningful **Desc** entry. For example, *EA-PDF Creation Software* may decide to use these entries to assist in disambiguating email attachments for situations where the filename is not unique in an EA-PDF file.

*Reason: The **Contents** key is often used by legacy viewers when navigating file attachment annotations on a PDF page (such as when hovering over the paperclip icon of the annotation). The embedded file stream dictionary **Desc** key (description) is often used when presenting the list of embedded files in a separate navigation pane. PDF standards do not define this level of user experience. Ensuring both are present provides a wider and hopefully more reliable legacy viewer experience.*

If an attachment exists in the source email but cannot be embedded in the EA-PDF file, then the file attachment annotation and embedded file stream representing that attachment ought to be created, with the stream **Length** set to zero bytes.

Reason: this simulates the preserved representation of the email will still appear to have an attachment (via the File attachment annotation linked to a filename) even in legacy software, even if the bytes of the attachment are excluded from preservation (e.g., by a policy setting). However, this is not mandated as it cannot be machine validated without referencing the original source email asset.

When the length of the stream data of the embedded file stream is non-zero, the decompressed PDF stream data needs to contain the full data of the decompressed email attachment as present in the original source email asset.

Note: PDF embedded file streams can be losslessly compressed to reduce PDF file size. PDF does not support Base64 as is used by email.

Reason: this ensures that the email attachment is faithfully preserved in EA-PDF by not being processed by the EA-PDF Creation Software *which might result in a different bitstream (e.g. line ending changes). However, this is not mandated as it cannot be machine validated without referencing the original source email asset.*

The embedded file parameter dictionary ([ISO 32000-2], Table 45) is required to have a **ModDate** entry for all associated files. *EA-PDF Writers* may also wish to record the MD5 checksum via the **CheckSum** and **Size** entries for all embedded file streams, which can assist correlating with XMP metadata and detecting any issues with file extraction.

The embedded file stream dictionary may also contain an **AF** array with a file specification dictionary having an **AFRelationship** key with a value of *Alternative* for an alternative representation of the attachment (e.g., conversion of the email attachment to PDF or PDF/A). See [AssociatedFiles].

Note: EA-PDF permits renderings of email attachments to be included in an EA-PDF file (e.g. as an embedded PDF/A file or an additional Content Set (set of pages)) even if the attachment itself is not included (zero length). However, these alternate representations are not ambiguous as the actual email attachment itself because of the ***AFRelationship*** *value.*

The embedded file stream dictionary may also contain a **Metadata** entry. This may contain metadata that associates the attachment back to an email (i.e., via `Mail_GUID`), or additional provenance information such as set by the archival policy or captured by the *EA-PDF Creation Software*. See [MetadataStreams] and §14.3.2 [ISO 32000-2].

Reason: email attachments may be malicious, corrupted, or zero bytes in length. EA-PDF desires that a file attachment annotation, file specification dictionary, and embedded file stream are present for every email file attachment but does not require that every attachment must be embedded – the embedded file stream may be zero bytes long. In such cases, a ***Metadata*** *entry might be used to record the reason. See example below. However, this is not mandated as it cannot be machine validated without referencing the original source email asset.*

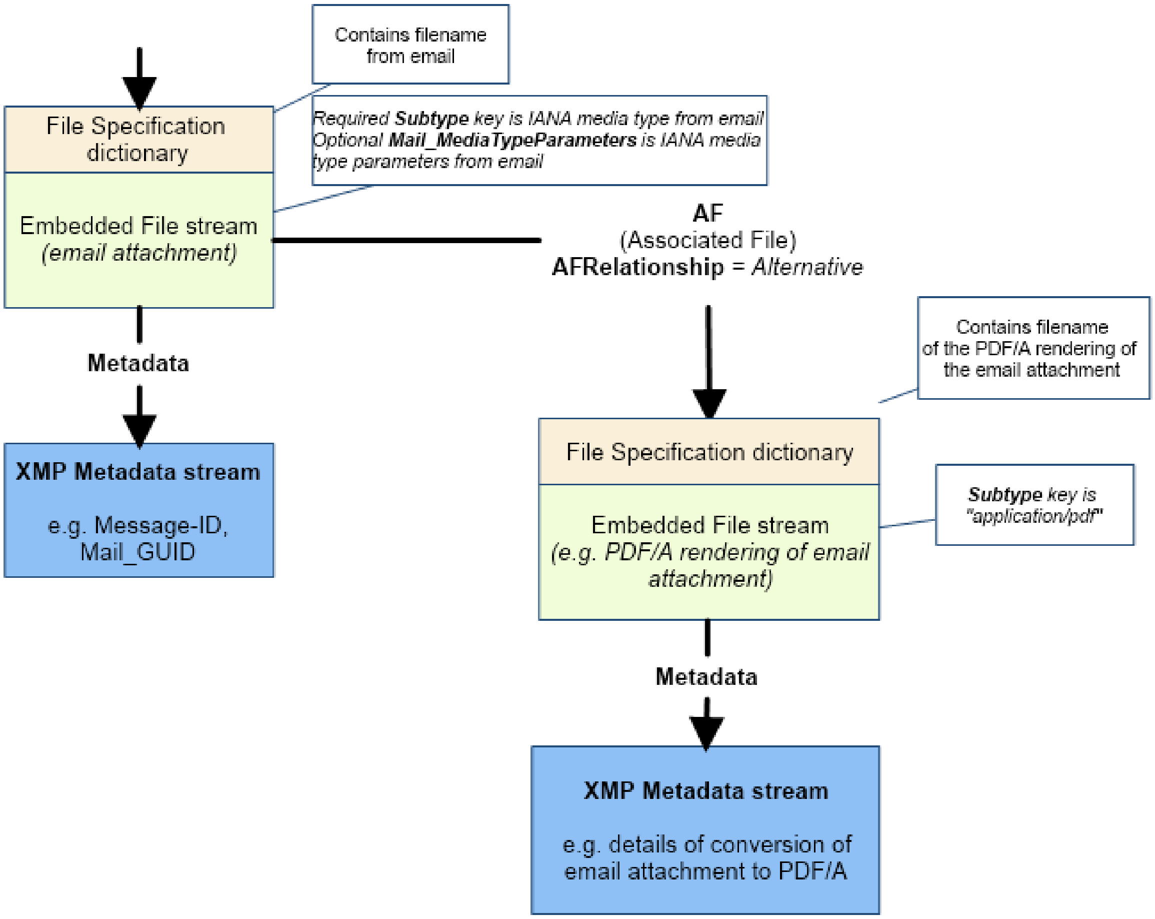

Contains filename from email
File Specification dictionary
Required Subtype key is IANA media type from email
Optional Mail_MediaTypeParameters is IANA media type parameters from email
Embedded File stream (email attachment)
AF (Associated File)
AFRelationship = Alternative
Metadata
XMP Metadata stream
e.g. Message-ID, Mail_GUID
Contains filename of the PDF/A rendering of the email attachment
File Specification dictionary
Subtype key is "application/pdf"
Embedded File stream (e.g. PDF/A rendering of email attachment)
Metadata
XMP Metadata stream
e.g. details of conversion of email attachment to PDF/A

Figure 10: conceptual framework of PDF objects related to email attachments with alternate rendering

Reason: the LWG wanted to associate the email attachment filename, Media Type (from email), associated `Message-ID` (of the email, but this may not always be present hence the need to use `Mail_GUID`), and to support alternative *renderings for non-archival attachment formats. Note that all streams in the above diagram may be losslessly compressed using **FlateDecode**.*

The **Subtype** entry of the embedded file stream dictionary must be present and ought to be set to the IANA Media Type as recorded in the source email (subject to encoding differences), but excluding any parameters that are present in the source email. If required, IANA Media Type parameters can be stored as a PDF string object in the optional **Mail_MediaTypeParameters** entry.

Note: Only some IANA Media Types have parameters. RFC 8118 does not define any IANA Media Type parameters for application/pdf.

Reason: some email clients use additional logic beyond just the attachment Media Type from the source email. To ensure accurate and consistent preservation and representation, the source email Media Type should reflect the original source email asset, rather than a different but arbitrarily determined Media Type. However, this is not mandated as it cannot be machine validated without referencing the original source email asset.

```
2 0 obj                 % see https://www.iana.org/assignments/media-types/text/csv
<< /Type /EmbeddedFile  % see Table 44 in ISO 32000-2:2020
   /Subtype /text#2Fcsv % IANA Media Type “text/csv” as a PDF name object
   /Mail_MediaTypeParameters (charset=utf-8, headers=present) % EA-PDF optional key
                        % supporting defined parameters for text/csv. PDF string.
   /Params <<           % see Table 45 in ISO 32000-2:2020
      /ModDate (D:…)
      /Size …
      /CheckSum (…)
   >>
   /Metadata 3 0 R      % optional XMP metadata about this embedded file
   /AF [ 4 0 R ]        % optional associated file (e.g. a PDF/A rendering of the CSV)
   /Filter /FlateDecode % compressed to save space
   /Length …
   … other keys needed for streams …
>>
stream
… FLATE-compressed binary data …
endstream
endobj
```

Example 3: EA-PDF embedded file stream dictionary with IANA media type including parameters.

The total number of email attachments must be included in the document level XMP metadata for PDF/mail-1{s, si, m, mi} files. This includes all attachments that resulted in zero-sized PDF embedded file streams.

Reason: this explicit request was from the EA-PDF LWG for an inventory of attachments and is limited to PDF/mail-1{s, si, m, mi} only. It is not relevant to PDF/mail-1{c, ci} files since each PDF/mail-1{s, si, m, mi} file in the container will have its own embedded email attachment count and maintaining aggregated counts is error-prone.

7.5.1 Email attachment example

Consider the following source email fragment where the email attachment “Report.doc” is detected by the *EA-PDF Creation Software* to contain content in violation of an archival policy setting (e.g., malicious VBA macros), but the archival policy requires an alternate representation for preservation purposes (e.g., conversion to PDF/A):

```
…
------RGskdOleHeu1K4pe7KmIzUgCk2qkjW8-r2KiIFoM3IJ_Eg0L=_b_
Content-Disposition: attachment;name="Report.doc"
Content-Transfer-Encoding: base64
Content-Type: application/vnd.openxmlformats;name="Report.docx"
X-Attachment-Index: 0
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAACAAAAQgAAAAAAAAAAAA
EAAARAAAAAEAAAD+////AAAAAEEAAAB4AAAA////////////////////////////////////////
…
```

EA-PDF Writer: Base64 is not supported by PDF so *EA-PDF Creation Software* will need to first decode and then re-compress using a suitable PDF compression filter such as FLATE (**FlateDecode**).

The PDF fragment for the file specification dictionary referenced from the file attachment annotation **FS** entry:

```
9 0 obj % File attachment annotation visualized as a paperclip on a PDF page
<< /Type /Annot
   /Subtype /FileAttachment
   /FS 10 0 R
   /Name /Paperclip
   /Contents (Email attachment: Report.docx)
   /Rect [ … ]  % Location on page
   /AP << /N … >>  % Annotation appearance stream for a paperclip
   …
>>
endobj

10 0 obj % File specification dictionary for “Report.doc”
<< /Type /Filespec
   /Desc (Email attachment: Report.doc [VBA macros])
   /F (Report.doc)  % Same as source email
   /UF (Report.doc) % Same as source email
   /EF << /UF 11 0 R /F 11 0 R >>
   …
>>
endobj

11 0 obj % The embedded file stream of “Report.doc” that contained VBA macros
<< /Type /EmbeddedFile
   /Subtype /application#2Fvnd.openxmlformats % Same as email (#2F is “/” in hex)
   % no Media type parameters so no /Mail_MediaTypeParameters entry is required
   /Metadata 12 O R
   /AF [ 13 0 R ]  % Associated Files array for the PDF/A equivalent
   /Length 0 % Email attachment NOT embedded because of a policy setting (no macros)
>>
stream
endstream
endobj
```

```
12 0 obj % The XMP metadata stream about why “Report.docx” was not embedded
<< /Type /Metadata
   /Subtype /XML
   /Length …
>>
stream
… XMP metadata that file contained VBA macros and was converted to PDF/A-4 …
endstream
endobj

13 0 obj  % converted email attachment to PDF/A as a safer alternative
<< /Type /Filespec
   /AFRelationship /Alternative
   /Desc (Email attachment “Report.docx” converted to PDF/A)
   /F (Report.doc.pdf)
   /UF (Report.doc.pdf)
   /EF << /UF 14 0 R /F 14 0 R >>
>>
endobj

14 0 obj
<< /Type /EmbeddedFile
   /Subtype /application#2Fpdf % there are no media type parameters for PDF
   % no Media type parameters so no /Mail_MediaTypeParameters entry is required
   /Metadata 15 O R   % XMP metadata about conversion of DOCX→PDF/A process
   /Length …
   /Filter …
>>
stream
… XMP metadata …
endstream
endobj
```

Example 4: Email attachment and associated file example

In the example above, if the source email Media Type was the generic “`application/octet-stream`”, some email clients might then use additional means to associate the attachment with an application (possibly by examining the file extension or file content). EA-PDF *strongly recommends* avoiding this application- and machine-specific behavior.

Note: this cannot be mandated as it cannot be machine validated without referencing the original source email asset.

Note: the IANA Media Type `application/pdf` is used for every kind of PDF file – there is no specific Media Type for PDF, PDF/A or EA-PDF files and there are no parameters defined. See [RFC-8118].

7.6 Structured containers (PDF/mail-1c)

The LWG recognized that EA-PDF has a requirement to preserve complex hierarchies of folders containing emails in a single PDF, reflecting emails stored in Microsoft™ OST/PST files or as represented in file systems by other email clients such as Mozilla Thunderbird using folders with one or more MBOX files. This requirement is achieved by using PDF Collections (also known as "PDF Portable Collections", "PDF Portfolios", "PDF Packages" or "PDF Binders") that were introduced with PDF 1.7 and extended in PDF 2.0 (§12.3.5 [ISO 32000-2]).

PDF Collections are fully compatible with PDF/A-3 and PDF/A-4f / PDF/A-4e.

> *Note: only some legacy PDF viewers support PDF Collections while those that don't will often fallback with behavior typically resulting in a list of the embedded files (e.g., a flat list of files in a file attachment pane without any folder hierarchy).*

Several scenarios have been identified by the LWG:

1. Creation of a preservation EA-PDF container from monolithic complex email formats (such as Microsoft™ OST/PST that contains internal folder hierarchies) such that the PDF reflects the hierarchical folders with emails[16], as illustrated in the left portion of *Figure 11: PDF/mail-1c "structured container" containing multiple EA-PDF files arranged in hierarchical folders* below. In this scenario, there is typically a *single* original raw source email asset (e.g., OST/PST) embedded in the container PDF, with PDF`/mail-1{si, mi}` files used to represent the emails in a folder hierarchy (these PDFs do *not* contain original raw source email assets and are thus indicated using the "isolated" profile designator `i`). If such embedded EA-PDF files are extracted from the container, then they are *isolated* from their original raw source email asset.

2. Packaging of previously created individual EA-PDF files into a "structured container". The previously created EA-PDF files could be any PDF`/mail` profile, including other PDF`/mail-1c` files recursively. Use cases included packaging of many EA-PDFs reflecting an organizational structure as well as reflecting folders of MBOX (PDF`/mail-1{m, mi}`) or MSG/EML (PDF`/mail-1{s, si}`) files, as illustrated in the right portion of *Figure 11: PDF/mail-1c "structured container" containing multiple EA-PDF files arranged in hierarchical folders* below. This scenario also reflects how Mozilla Thunderbird stores emails as multiple MBOX files arranged in a folder hierarchy on a hard disk and where each embedded file in the container has its own embedded original raw source email asset (i.e., the MBOX file). If such embedded PDF files are then extracted from the container PDF`/mail-1c` file then

[16] EA-PDF does not attempt to preserve (as so-called "portable renderings" such as with PDF pages) other types of items that may be in OST/PST files, such as calendar, contacts, to-do lists, etc. This data is not prohibited from existing in the embedded source assets, or to be included in additional content sets but EA-PDF does not define such details.

they continue to contain their original raw source email asset and remain as valid preservation assets (although without the context of their folder hierarchy).

3. Packaging of one or more redacted, isolated, or otherwise modified EA-PDF files into a container for further non-preservation distribution uses, such as via FOIA requests or access copies. In this case, the container file will be indicated as `PDF/mail-1ci` as one or more of the original raw source email assets or metadata has been removed or modified. The container `PDF/mail-1ci` otherwise retains all other EA-PDF features, except that provenance back to the original raw source email asset for one or more emails has been lost or compromised.

In an EA-PDF context, the embedded files in the PDF Collection of the right scenario in *Figure 11* are described as "isolated" as they meet all EA-PDF requirements when extracted from the `PDF/mail-1c` file *except* they do not contain the original raw source email asset. Those in the left scenario remain as valid standalone `PDF/mail-1{s,m}` files because they *include* their original raw source email asset even when extracted.

Note: EA-PDF preserved emails need to have verifiable provenance back to a preserved original raw source email asset, whether this be in the embedded file for `PDF/mail-1{s,m}` or in the container `PDF/mail-1c` file.

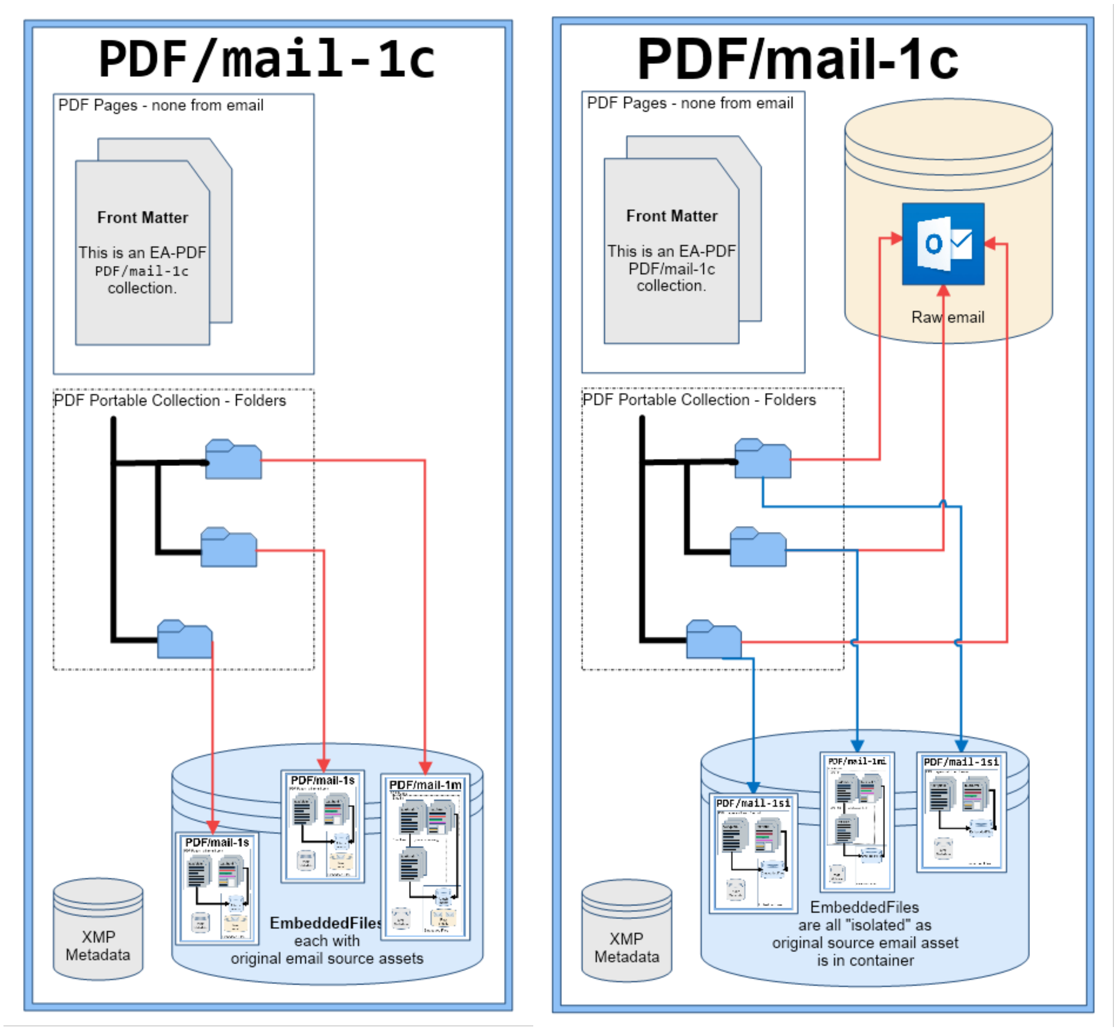


Figure 11: PDF/`mail-1c` “structured container” containing multiple EA-PDF files arranged in hierarchical folders, each with their original raw source email asset (*left*) and using a shared original raw source email asset in the container (*right*).

There is no requirement that PDF/`mail-1c` files _must_ contain hierarchical folders, although it is expected. There are also no specific requirements imposed on any internal folder structure or when/how specific profiles of embedded PDF/`mail` files need to be embedded in PDF/`mail-1c`. Although this provides greater flexibility and supports multiple EA-PDF scenarios and use-cases, some suggestions and guidance may be useful:

- If folder preservation is not required, then it is probably better to consider using PDF/`mail-1m`, as PDF/`mail-1m` will be more widely supported by legacy PDF viewers than PDF/`mail-1c`, but this is not mandated.
- Any EA-PDF PDF/`mail` profile can be stored in a single container PDF/`mail-1{c, ci}` file. An embedded PDF/`mail-1{c, ci}` can thus contain other PDF/`mail-1{c, ci}` files, creating a nesting of files using PDF Collections. Although permitted, this nesting probably does not create

a good end user experience as each PDF Collection needs to be individually opened to examine its contents.

- Empty folders (i.e., folders without any associated embedded files) are allowed in PDF Collections, although some non-EA-PDF aware legacy viewers do not display empty folders.
- As mentioned above, it is preferable to embed multiple PDF`/mail-1m` or PDF`/mail-1s` files in a folder hierarchy so that it can be completely navigated from the container PDF`/mail-1c` (like an email client), rather than embed multiple separate PDF`/mail-1c` files with internal folder hierarchies.
- The folder structure in PDF`/mail-1{c`, `ci}` is not prescribed. The top-level folder structure might reflect an organization structure, users, a single user, different email accounts, or simply folders in a typical desktop email client. Each folder and sub-folder in the PDF Collection might contain an embedded PDF`/mail-1m`[17] file with the user's emails, or multiple PDF`/mail-1s` files if the emails being archived are stored individually (e.g., as MSG or EML). The use of a PDF`/mail-1{m,mi}` file is likely preferable as this also allows sharing of resources such as fonts and images, and will reduce overall file size.
- Using a single PDF`/mail-1{c`, `ci}` container with hierarchical folders, each with either a single PDF`/mail-1{m`, `mi}` or multiple PDF`/mail-1{s`, `si}` is more user friendly than nesting PDF`/mail-1c` files with more subfolders, although this is not prohibited. This is because each nested PDF`/mail-1{c`, `ci}` will need to be opened separately to navigate their content. Using individual PDF`/mail-1{m`, `mi}` files in each folder is also likely more efficient than embedding multiple PDF`/mail-1{s`, `si}` files, although this is not prohibited[18]. Again, the use of a PDF`/mail-1{m,mi}` file is likely preferable as this also allows sharing of resources such as fonts and images, and will reduce overall file size.

The Document Catalog dictionary of a PDF`/mail-1{c`, `ci}` container files will always have a **Collections** entry.

Note: If there was a single source email file (such as OST/PST file) used to create the PDF`/mail-1c` *file, then that OST/PST file will also be listed in the container PDF* ***EmbeddedFiles*** *name tree (as described above). Consequently, it will also be included in the files making up the PDF Collection. This is intentional as it will more likely enable extraction of this source email file by end users using legacy non-EA-PDF aware software, including software that does not support the PDF 1.7 Collections feature. EA-PDF aware software can recognize such a file from its Associated File* ***AFRelationship*** *value of* Source *and present it differently.*

[17] Or, degenerately, a PDF`/mail-1s` if there was just a single email message. Or no embedded PDF`/mail` file if there are no emails. There is no requirement that an embedded EA-PDF file exist in every folder.

[18] Note that resources (fonts, images, etc.) inside one embedded PDF`/mail` file cannot be shared with another embedded PDF`/mail` file. PDF resource reuse is always limited to within a single PDF.

PDF/`mail-1{c, ci}` files will always contain one or more pages (in at least one *Content Set*) that are used as the initial document. These pages do *not* represent the content of email but are front matter or other content or context related to the entire collection of emails (rather than a single email in the collection).

Reason: this ensures that non-EA-PDF aware legacy software does not accidentally present an email from a PDF/`mail-1{c, ci}` file in a way that might confuse a user to think that the EA-PDF only contained a single email. The front matter pages of the container PDF ought to contain content explaining the file is an EA-PDF structured container file.

For this reason, the Collection dictionary **D** entry ([ISO 32000-2], Table 153) must not be present so that the page(s) of the PDF/`mail-1{c, ci}` will be initially shown. These pages may contain one or more *Content Sets* at the discretion of the *EA-PDF Creation Software*. Each *Content Set* will have outline entries (bookmarks) relevant to the content of the container PDF/`mail-1{c, ci}` file as described above. As noted in [ISO 32000-2], such pages ought to include information that "… *helps the user understand what is contained in the collection, such as a title and an introductory paragraph*".

Reason: Content Sets in PDF/`mail-1{c, ci}` files are left unconstrained and might include content to help users understand the context of the email collection, conversion reports, policy settings, etc.

The Collection dictionary **View** entry must not be *H*.

Reason: This ensures that the collection view is not hidden by default.

A Collection Schema dictionary must be defined containing at least the following fields listed in *Table 5: Minimum set of Collection Schema entries for PDF/mail-1{c*. This minimum set of collection schema fields reflects a minimal set of EA-PDF *Core Fields* most relevant to human understanding of email archives, however *EA-PDF Creation Software* may add additional fields. See also Example 1 in §12.3.5.2, [ISO 32000-2] which illustrates the use of a simple **CollectionSchema** dictionary for email.

Core Field	**Collection Schema key (*case sensitive*)**
`To`	`To`
`From`	`From`
`Sent`	`Sent`
`Subject`	`Subject`
`Cc`	`Cc`
`Bcc`	`Bcc`
Original message size (*in bytes, ≥ 0*)	`Size`
Number of email attachments (*integer, ≥ 0*)	`Attachments`

Table 5: Minimum set of Collection Schema entries for PDF/`mail-1{c, ci}`

If the Collection dictionary contains a **Navigator** entry (*new in PDF 2.0,* Table 156 [ISO 32000-2]), the Navigator dictionary must not have a **SWF** entry.

Reason: this prohibits the use of vendor-specific and obsoleted FLASH-based (SWF) navigators used by Adobe Extension Level 3 with PDF 1.7 as they are unsuited to the needs of long-term preservation and were not adopted by ISO into ISO 32000-2 (PDF 2.0).

If there is a need to preserve the hierarchical folder structure of a collection of emails, then Collection hierarchical **Folders** (*new in PDF 2.0,* §12.3.5.2 [ISO 32000-2]) must be used.

*Note: PDF 2.0 standardized the **Folders** entry in the Collection dictionary (see Table 153, [ISO 32000-2]) that was initially proposed by Adobe in Adobe Extension Level 3 to ISO 32000-1:2008. Both PDF/A-3 and PDF/A-4f conformance levels can be used as collections as folders do not alter the static page appearance of the PDF/A emails.*

EA-PDF Writer: GMail's use of labels is implemented via custom email headers (e.g., "X-GMailLabel") and only simulates folders in browsers and email clients. EA-PDF Creation Software *is free to select whether such a representation needs to use PDF/`mail-1{c, ci}` hierarchical folders or not.*

All folder and filenames used in PDF/`mail-1{c, ci}` files must always conform to the valid file name restrictions and must be unique after case normalization, as described below Table 159 in [ISO 32000-2]. Conforming EA-PDF aware software that supports PDF/`mail-1{c, ci}` must not support or allow invalid filenames[19].

Besides the original raw source email asset(s), all the "contained" embedded files in a PDF/`mail-1{c, ci}` must be other EA-PDF files.

Reason: this prohibits the addition of miscellaneous embedded files in the collection (such as Associated Files), alongside the EA-PDF files representing the emails in folders, which might cause confusion when using legacy non-EA-PDF aware software.

```
10 0 obj
<< /Type /Collection
   /View /D      % details view
   /Schema <<
      /Type /CollectionSchema
      /To          << /Subtype /S    /N (To)                /O  1 >>
      /From        << /Subtype /S    /N (From)              /O  2 >>
      /Sent        << /Subtype /D    /N (Sent)              /O  3 >>
      /Subject     << /Subtype /S    /N (Subject)           /O  4 >>
      /Size        << /Subtype /Size /N (Size (bytes))      /O  5 >>
      /Attachments << /Subtype /N    /N (No. of attachments) /O  6 >>
      /Cc          << /Subtype /S    /N (Cc)                /O  7 >>
      /Bcc         << /Subtype /S    /N (Bcc)               /O  8 >>
   >>
   /Folders 11 0 R
>>
endobj
```

/N strings may be localized to reflect the "look & feel" of an email client.

The key names in the Collection schema are **not** localized and are fixed for interoperability. They are also ***not*** prefixed with `Mail_` as they are an example in ISO 32000.

[19] Conforming EA-PDF software supporting PDF/`mail-1{c, ci}` needs to always provide a consistent and reliable experience by prohibiting this statement from ISO 32000-2: "*An interactive PDF processor may choose to support invalid names or not*".

```
11 0 obj
<</Type /Folder
  /ID   0               % 1st folder at root level
  /Name (joe@email.org) % name of the folder
  /Desc (Email folders for joe@email.org …) % Description
  /CreationDate (D:…)   % creation date of folder in email system (if one exists)
  /Child 20 0 R         % first sub-folder for this email account
  /Next 12 0 R          % next folder at this root level – maybe a different email account
>>
endobj

20 0 obj
<</Type /Folder
  /ID   2               % 1st folder at level 1 under joe@email.org
  /Name (Inbox)         % name of the folder = joe@email.org/Inbox
  /Desc (Inbox folder for joe@email.org …) % Description
  /CreationDate (D:…)   % creation date of folder in email system (if one exists)
  /Next 21 0 R          % next folder at level 1 – maybe a different standard email folder
>>
endobj

21 0 obj
<</Type /Folder
  /ID   3               % 2nd folder at level 1 under joe@email.org
  /Name (Sent)          % name of the folder = joe@email.org/Sent
  /Desc (Sent folder for joe@email.org …) % Description
  /CreationDate (D:…)   % creation date of folder in email system (if one exists)
  /Next 22 0 R          % next folder at level 1 (not shown) …
>>
endobj

…

11 0 obj
<</Type /Folder
  /ID   1               % 2nd folder at root level
  /Name (joe@home.net)  % name of the folder
  /Desc (Email folders for joe@home.net …) % Description
  /CreationDate (D:…)   % creation date of folder in email system (if one exists)
  /Child 30 0 R         % first sub-folder for this email account (not shown)
>>
endobj

…
```

Example 5: Collection Schema and Folder example

Note that if a PDF/`mail-1{c,ci}` file is opened in a non-EA-PDF aware legacy interactive PDF viewer that does not support PDF Portable Collections, all the EA-PDF files listed in the **EmbeddedFiles** name-tree may appear with a preceding “`<`*xxx*`>`” before the filename, where *xxx* is an integer (e.g., `<2>Inbox.pdf`, `<3>Sent.pdf`). This is due to the way PDF internally associates filenames with folders in PDF Collections and is unavoidable since it facilities a degree of backward compatibility for software that does not support PDF Collections.

> *Note: this also means that file specification strings associated with embedded file streams that use* ***Folders*** *must not be an absolute filename or contain folder names, “*`..`*” or other path components (see §7.11.2 File specification strings, [ISO 32000-2]).*

7.7 User navigation

Navigation of EA-PDF files in interactive PDF viewers was established as an important consideration by the LWG, including in non-EA-PDF aware legacy software. However, some PDF features supporting interactive viewing may not provide sufficient structure and context for reliable machine processing by EA-PDF aware software. This section describes how navigation experiences of EA-PDF files can be optimized.

Note that PDF/A standards do _not_ mandate specific behaviors of interactive PDF viewers.

7.7.1 Initial viewing mode

PDF supports features where a PDF file can hint to an interactive viewer the most appropriate screen layout to be used. Although not always supported, this can provide a nicer viewing experience for complex files such as EA-PDF.

The Document Catalog **PageMode** entry controls the appearance of interactive PDF viewers when a PDF file is first opened. It is widely supported by legacy software. For PDF`/mail-1{s, si}`, the **PageMode** entry is to be *UseOutlines* or, if the email has attachments, *UseAttachments*. For PDF`/mail-1{m, mi}`, **PageMode** entry must be *UseOutlines*. For PDF`/mail-1{c, ci}`, **PageMode** entry needs to be *UseAttachments.*

> *Reason: based on the content of each email, the EA-PDF Creation can set the **PageMode** appropriately for each PDF`/mail-1{s, si}` file. Because PDF`/mail-1{m, mi}` files contain many emails (and potentially many attachments with the same filename), navigating by email is the primary use-case. For PDF`/mail-1{c, ci}` containers, the primary use-case is accessing the embedded EA-PDF file attachments, even if PDF Collections are not supported.*

Viewer preferences (§12.2, [ISO 32000-2]) is widely supported by some legacy software and defines aspects of the interactive PDF viewing experience. The Viewer Preferences **NonFullScreenPageMode** key needs to be *UseOutlines*.

The PDF document information dictionary **Title** or XMP metadata **dc:title** property are strings that might be displayed in the title bar of legacy viewing applications. Thus, these values ought to be set appropriately by *EA-PDF Creation Software*. See also Viewer Preferences **DisplayDocTitle** (Table 147, [ISO 32000-2]), as a common alternative is the PDF filename which is unlikely to be a good user experience. The **DisplayDocTitle** value is mandated in EA-PDF.

7.7.2 Bookmarks

Most interactive PDF viewers provide a “Bookmark” or “Outline” pane for easy navigation commonly associated with headings in documents. PDF document outlines (§12.3.3 [ISO 32000-2]) are commonly referred to as “bookmarks” and represent a hierarchical navigation structure for users.

> *Note: This document does not mandate the **Title** text or formatting (**C** and **F** entries) for outline items. The **Title** text supports the Unicode encodings supported by PDF (UTF-16BE and additionally UTF-8 in PDF 2.0 and PDF/A-4).*

Note: This document does not mandate how each outline item navigates to the appropriate section in an EA-PDF file (by action, by structure element, by destination). However, the PDF/A standards define some constraints and will need to be referenced.

All `PDF/mail-1` files will have at least one outline item, with every *Content Set* (including each message body) having a related outline item for easy navigation.

Example: an empty draft email would have just one outline item to the page with the representation of the Core Fields, whereas an email with multiple bodies will have multiple outlines entries to each of the content sets. In the latter case, the outline items may form a hierarchy (this is not mandated).

Reason: Because emails can have multiple bodies, each resulting in a different number of pages, bookmarks provide a very widely supported user navigation capability that can link to the first page in each content set and other important destinations in EA-PDF files (such as the rendering of Core Fields). It is not required that the Core Fields are always at the top of the first page, although this is conventional in most email clients – EA-PDF Creation Software is free to choose.

EA-PDF Writer: EA-PDF Creation Software is free to generate other PDF page content sets (such as conversion reports, lists of attachments, a rendering of all header fields, a text dump of the raw email, policy settings, etc.) in addition to the message bodies, but it is required to always create an outline item to the first page of each Content Set that is generated. This can be machine validated.

`PDF/mail-1{m, mi}` files ought to use a hierarchical outline structure with each outline item at certain level referencing an individual user, email account, etc. For optimal navigation, large `PDF/mail-1{m, mi}` files ought to collapse the lower-level outline items, so that only the outlines items representing each email are visible on initial viewing.

Reason: because `PDF/mail-1{m, mi}` files can contain many emails (potentially each with multiple message bodies and additional content sets), a fully expanded outline might result in a poor navigation experience with lots of vertical scrolling of the outline tree since each email could have multiple child outline items.

7.7.3 Embedded files

Many interactive PDF viewers provide a dedicated “File Attachment” or “Embedded Files” pane providing a list of certain files that are embedded in the PDF document. Other viewers may include file attachments in their comment pane. Because emails and EA-PDF use many embedded files this is an important feature when selecting non-EA-PDF aware legacy interactive PDF viewers.

All files specification dictionaries need to have an appropriate **Desc** entry.

7.8 Modification workflows

As discussed in the LWG meetings, editing, or modifying EA-PDF files after creation can damage their preservation integrity, which has serious implications for certain workflows:

- An EA-PDF file must always pass validation according to its declared PDF/A conformance level;
- Like PDF ISO subsets, EA-PDF files declare their conformance via the document-level XMP Metadata from the Document Catalog **Metadata** entry. Such data may not be maintained by non-

EA-PDF aware legacy editing software and thus EA-PDF aware software may wish to perform additional validation checks at runtime (*this is not mandated!*).

Note: by making all EA-PDF profiles PDF/A compliant, some non-EA-PDF aware legacy software may detect the PDF/A conformance and help to reduce accidental edits to EA-PDF files. But this is not foolproof!

- Non-isolated EA-PDF files must always have direct provenance to an identifiable embedded original raw source email asset(s). Either this will be in the `PDF/mail` file itself (as the Document Catalog Associated Files with an **AFRelationship** value of *Source*), or in the immediate container `PDF/mail-1c` file (e.g., for monolithic complex OST/PST files). If there is no such traceable linkage to an original raw source email asset (e.g., an embedded `PDF/mail` file in a `PDF/mail-1c` created from a monolithic complex OST/PST that has been extracted) then that extracted file must be indicated as an "isolated" EA-PDF profile as it has lost its provenance to its original raw source email asset.
- Although out of scope for EA-PDF, a redacted `PDF/mail` file can no longer be considered as a trustworthy EA-PDF preservation artifact as it is no longer a faithful rendition of a verifiable preservation of the original raw source email asset – but it might still be PDF/A (subject to the PDF redaction software).

  Reason: redacting only the PDF content but leaving the unredacted embedded original raw source email asset makes no sense – or vice-versa. Redaction can involve anything, including the need to alter one or more embedded files, so this keeps non-isolated EA-PDF files as a clearly demarcated preservation format, while isolated EA-PDF files continue to provide email archival context. Redaction of email is beyond the scope of EA-PDF.

This specification does not define email archival workflows.

8 EA-PDF aware software (normative)

End users experience EA-PDF files via their software so it is not unsurprising that additional requirements must be met by conforming EA-PDF aware software to ensure a consistent, reliable, and trusted user experience.

Non-interactive EA-PDF aware software (e.g., command line or server-based) may perform similar functions, such as bulk extraction of all embedded files or metadata streams, indexing of content, checking links, etc. Wording is intended to reflect both interactive and non-interactive applications.

All conforming EA-PDF software SHALL provide the following functionality *in addition to* the mandated functionality required by PDF/A viewers (see [PDF/A-3] and [PDF/A-4]):

Note: the following requirements and recommendations below are aligned with good, feature-rich compliant PDF/A viewers.

*EA-PDF Reader: EA-PDF is specified to fully support foreign languages and multi-lingual emails. Care needs to be taken not to codify any assumptions about encountering only ASCII (**PDFDocEncoded**) strings including folder or file names, outlines (bookmarks), file annotation **Contents** entries, file specification **Desc** entries, etc. in EA-PDF aware software.*

- All conforming EA-PDF software SHALL render all pages in EA-PDF files according to the appropriate PDF/A requirements.

Note: due to PDF/A-3 conforming reader requirements, an EA-PDF conforming reader must also be able to read all PDF/A files ("Conforming PDF/A-3 readers shall read and process appropriately all PDF/A-3 files. In addition, conforming PDF/A-3 readers shall read and process appropriately all PDF/A-1 files as defined by ISO 19005-1 and PDF/A-2 files as defined by ISO 19005-2." and from PDF/A-4: "Conforming PDF/A-4 processors shall read and process appropriately all conforming PDF/A-4 files.")". There is no intentional difference between the terms "reader" and "processor" used by these PDF/A standards

- All conforming EA-PDF software SHALL display all annotations using their appearance streams.

Note: making annotations visible may involve a user action.

- All conforming EA-PDF software SHALL visually indicate that a PDF file claims conformance as an EA-PDF file.

Note: this is in addition to the current industry practice of visually indicating PDF/A conformance.

- All conforming EA-PDF software SHALL detect unsupported PDF`/mail` versions and/or profiles and suitably inform the user of a potential compatibility issue.
- For all conforming EA-PDF aware software that supports PDF`/mail-1{c,ci}`, invalid filenames of files in a PDF Collection SHALL NOT be supported[20].

[20] Conforming EA-PDF software supporting PDF`/mail-1{c,ci}` needs to always provide a consistent and reliable experience by prohibiting this statement from ISO 32000-2: "*An interactive PDF processor may choose to support invalid names or not*".

- All conforming EA-PDF software SHALL initially open all EA-PDF files in a “read only” mode so they cannot be accidentally altered without an additional explicit user action.

Note: this is fully aligned with current industry best practice for PDF/A.

All conforming EA-PDF software SHALL provide the following functionality without allowing the EA-PDF file to be accidentally modified or edited:

- All conforming EA-PDF software SHALL use the initial viewing mode as specified in the EA-PDF file.
- All conforming EA-PDF software SHALL support navigation via outlines.
- All conforming EA-PDF software SHALL allow extraction of any embedded files in the PDF in either the document catalog **EmbeddedFiles** name tree, as a file attachment annotation on any page or as an Associated File on any PDF object.

Note: in ISO 19005-4 (PDF/A-4) this is also a recommendation: “a conforming interactive PDF/A-4f processor should enable the extraction of any embedded file.”

- All conforming EA-PDF software SHALL display file specification dictionary **Desc** entries as well as filenames.
- All conforming EA-PDF software SHALL display and allow interaction with file attachment annotations on pages to permit to access the referenced embedded file.

Note: PDF/A requirements only cover the static page appearance and do not include document interaction. In the case of EA-PDF and PDF File Attachment annotations, no changes or modifications to the EA-PDF file are required or expected from the above requirement – it is simply a means to navigate to the appropriate referenced embedded file which might then be extracted. Hence this requirement is not onerous.

- All conforming EA-PDF software SHALL display the value of the **Contents** entry of file attachment annotations.
- All conforming EA-PDF software SHALL interact only with link annotations (URLs) explicitly defined in the EA-PDF file.
- All conforming EA-PDF software supporting PDF`/mail-1{c,ci}` SHALL support PDF Collections.
- All conforming EA-PDF software SHALL access and display all XMP Metadata streams.
- All conforming EA-PDF software SHALL allow extraction of all Associated Files and XMP Metadata streams associated with arbitrary PDF objects.

Conforming EA-PDF software SHALL NOT perform additional automatic URL detection that results in an actionable or actioned URL, unless explicitly enabled by a user action for each PDF file.

Reason: This ensures that the policy setting at the time of the EA-PDF file creation is more strictly enforced since actionable URLs that get detected might have unknown or undesirable side effects and relying on all users to not follow certain URLs is unreliable. Wording is to also avoid persistent application settings being used.

9 From a PDF file format perspective (normative)

This section provides a technical description of the EA-PDF v1 file format for readers familiar with the technical details of PDF and PDF/A. This section contains requirements that match those described in the previous sections. However, many aspects discussed above do not lead to formal requirements as they are either covered by existing PDF/A requirements or recommendations, or are explanations for EA-PDF developers or end-users.

All "SHALL" mandatory requirements are worded to ensure that they are entirely machine-validatable. It is assumed that EA-PDF validators do *not* need to parse any original source raw email assets for validation but can parse only PDF data (including content streams) and XMP Metadata to perform all validation defined in this document.

Not every PDF feature is listed: if a PDF feature is not mentioned and is not otherwise constrained by PDF/A, then it may occur in an EA-PDF file. This applies to many PDF features which do not impact the reliable rendering of pages in PDF/A.

Boilerplate and common wording found in PDF ISO subset standards has been intentionally omitted from this section so that the focus can be on EA-PDF-specific requirements. Requirements from PDF/A are not duplicated here since all EA-PDF files are always conforming PDF/A-3 or PDF/A-4 files.

Except for Collection Schema keys, all other PDF key names introduced by EA-PDF start with the registered developer prefix "`Mail`" and followed by an underscore to clearly identify them (i.e. "`/Mail_`…").

Note: technically this is unnecessary in certain situations (such as keys inside new dictionaries), however it can be helpful if non-EA-PDF aware software modifies an EA-PDF file, if pages or content are extracted, etc.

9.1 EA-PDF file

The PDF header SHALL be either "`%PDF-1.7`" or "`%PDF-2.0`" and the declared PDF version number of an EA-PDF file SHALL be either PDF 1.7 or PDF 2.0.

Note: the above PDF version requirement accounts for the Document catalog ***Version*** *key (if present) and file header.*

All EA-PDF files SHALL conform to either PDF/A-3a, PDF/A-3u, PDF/A-4, PDF/A-4f, or PDF/A-4e.

Note: the above requirements for PDF/A conformance incorporate a lot of technical requirements.

All EA-PDF files that also conform to PDF/A-3 SHALL have a Document Information dictionary referenced from the trailer **Info** entry. The information in this document information dictionary SHALL be consistent with the document catalog XMP Metadata entry as defined by *Table 7 - Crosswalk between document information dictionary and XMP properties* in [PDF/A-3].

Reason: because PDF 2.0 has deprecated document information dictionaries, it is an intentional EA-PDF design decision to *__not__* *extend the document information dictionary with custom keys to express email-specific information.*

Document-level XMP metadata is where all EA-PDF definitive metadata information is located. However, the prohibition of a document information dictionary in PDF/A-4 files raises compatibility with a wide range of legacy software that do not support XMP.

All EA-PDF files SHALL NOT contain a **PieceInfo** entry in the document catalog dictionary.

*Note: excluding the use of document catalog **PieceInfo** private data is not viewed as an issue, since Associated Files, 2nd class keys, or page-based **PieceInfo** private data can all be used as alternatives for storing private data. This exclusion ensures that a partial document information dictionary (i.e. only containing a ModDate) is then never present in PDF/A-4 based EA-PDF files, which may otherwise be confusing in legacy viewers that do not understand XMP.*

The XMP **pdf:Keywords** property and, for all PDF-A/3-based EA-PDF files, the PDF document information dictionary **Keywords** entry SHALL each include “EA-PDF”.

Reason: standardized inclusion of the term “EA-PDF” in both the Document Information dictionary and equivalent XMP ensures that EA-PDF users using non-EA-PDF aware legacy software have easy access to see if a PDF file is EA-PDF (since XMP data is not always presented). This is machine validatable.

For PDF`/mail-1{s`, `si}` with no email attachments, the Document Catalog dictionary’s **PageMode** entry SHALL be *UseOutlines* and the Viewer Preferences **NonFullScreenPageMode** entry SHALL be *UseOutlines*.

For PDF`/mail-1{s`, `si}` with email attachments, the Document Catalog **PageMode** entry SHALL be *UseAttachments* and the Viewer Preferences **NonFullScreenPageMode** entry SHALL be *UseAttachments* for PDF 2.0 or *UseOutlines* for PDF 1.7 files.

For PDF`/mail-1{m`, `mi}`, the Document Catalog **PageMode** entry SHALL be *UseOutlines* and the Viewer Preferences **NonFullScreenPageMode** entry SHALL be *UseOutlines*.

For PDF`/mail-1{c`, `ci}`, the Document Catalog **PageMode** entry SHALL be *UseAttachments* and the Viewer Preferences **NonFullScreenPageMode** entry SHALL be *UseAttachments*.

All PDF`/mail-1` files SHALL set the Viewer Preferences **DisplayDocTitle** entry to *true*.

Reason: this tries to ensure that the PDF title metadata is always used in window title bar of viewers.

For non-isolated PDF`/mail-1{s`, `m`, `c}`, the Document Catalog dictionary SHALL have an **AF** entry as an array with at least one array element. At least one array element (file specification dictionary) in this array SHALL have an **AFRelationship** value of *Source*, representing the primary original source email asset(s). If the original email asset comprises a set of files[21], then either:

1. a PDF Related Files (**RF** entry) SHALL be in the file specification dictionary of the primary original email asset; or

[21] A “set of files” is differentiated from multiple original source email assets. A “set” is where several files are needed to function.

2. additional Associated Files SHALL be listed in the Document Catalog **AF** entry, each with an **AFRelationship** value of *Supplement*.

All PDF/`mail-1` files SHALL have an **Outlines** and **DPartRoot** entries in the Document Catalog. As a consequence, all pages in all EA-PDF files will have a **DPart** entry.

Reason: this ensures both PDF document navigation functionality and structured information, which can both be used in validating EA-PDF files.

9.2 Content sets

For the purposes of EA-PDF validation, a blank PDF page is a page without any visible text, visible image, or visible vector as page content or annotations within the page **CropBox** that would result in one or more pixels being painted.

*Note: if **CropBox** is not present, it inherits the value from the **MediaBox**.*

Note: this technical definition for visible is not infallible – a page that marks a single pixel that is white passes this test but will not normally be visible in a viewing application. It is assumed that all EA-PDF Creation Software *is well intentioned.*

Note: the requirements below are worded such that machine-validation can check consistency between XMP metadata, PDF content streams, navigation data, and other PDF data structures without referring to the original source email assets. This assumes EA-PDF validators can parse both PDF and XMP data, which is not unreasonable for PDF/A validators. It is not necessary for a human to perceive the page content to perform EA-PDF validation or to parse source email formats.

All PDF/`mail-1` files SHALL contain at least one *Content Set* of at least 1 page each that SHALL NOT be blank[22]. The first page in each *Content Set* in all PDF/`mail-1` files SHALL be referenced from at least one Outline item referenced from the Document Catalog **Outlines** entry.

Reason: this ensures a basic functional PDF with bookmark (outline) navigation even when the email is empty (e.g., an empty draft).

Note: the form of the outline destination used is not further defined, while still allowing the page to be validated. This is also because logical structure is not mandated in EA-PDF.

PDF/`mail-1{s`, `si}` files SHALL represent only a single email that is the only email referenced in the document-level (document catalog) XMP metadata stream.

PDF/`mail-1{m`, `mi}` files SHALL represent 2 or more emails and SHALL contain a *Content Set* for each email that SHALL NOT be blank. As a result, a PDF/`mail-1{m`, `mi}` file SHALL contain at least 2 pages that are not blank. Each email SHALL be included in the document-level (document catalog) XMP metadata stream.

[22] This requirement is intentionally not narrowed to be the rendering of email headers or of an email body as a zero-length source email asset or a source email asset with no valid emails may also need preserving.

Reason: a requirement such as "`PDF/mail-1{c,ci}` container PDF files SHALL NOT contain any PDF pages that represent the content of an email" is not machine validatable and is thus not stated. Validation is thus limited to outlines (bookmarks), logical structure, and document part metadata (DPM).

EA-PDF files may also contain additional *Content Sets* at the discretion of the *EA-PDF Creation Software* and subject to other requirements, such as:

- multiple renderings of the email bodies;
- a list of email attachments;
- PDF/A renderings of email attachments;
- a comprehensive rendering of all email headers;
- a "raw dump" of the email as a "source code" style text listing of RFC 822 email;
- conversion reports;
- additional rendered or embedded information (e.g., policy settings, etc.).

The first page of every *Content Set* SHALL be referenced by an outline item in the same PDF. Other outline items in the PDF to locations within *Content Sets* may also be present. As a result, all EA-PDF files will have at least one outline item, and the number of outline items will be greater than or equal to the number of *Content Sets*.

All pages in all PDF`/mail-1` files SHALL be included in the Document Part Metadata, referenced by the Document Catalog **DPartRoot** entry.

Note: technically the above requirement is already covered by the previous requirement to have a ***DPartRoot*** *entry in the Document Catalog and the existing ISO 32000-2 requirement to include all pages in DPM.*

This document does not otherwise prescribe how *Content Sets* are organized or how emails are rendered to PDF pages.

9.3 PDF page content

PDF/A standards define requirements for ensuring device independent page content streams and rendering of pages in EA-PDF files. The DPM data, outlines and logical structure can be used by validation software to check some of these requirements.

Content (including email bodies and all header fields) that is text in the source email assets SHOULD use PDF text objects.

Reason: this recommends that email content (including headers) that is text is retained as text so that software can search the content of emails. Text render mode 3 is acceptable. Note that if an email contains an image of text, then there is no requirement for that text to be OCR-ed.

Note: this cannot be validated without referencing the original source email asset(s) and is thus only a recommendation.

All *Core Fields* and their values (when present) SHALL use text objects.

*Reason: this ensures that most legacy software will be to search for the basic email core field information relevant to humans. Text render mode 3 is acceptable. Due to the inclusion of required logical structure (see below) this can also be machine validated (a PDF text object is demarcated by the **BT** and **ET** operators).*

Each email attachment (as indicated in the document-level XMP metadata) SHALL be visibly represented by a File Attachment annotation positioned entirely within the **CropBox** of a PDF page in each *Content Set* that represents an email body of the source email that contained that attachment (as determined by DPM – see below).

*Note: if **CropBox** is not present, it inherits the value from the **MediaBox**.*

Example: if an email has 1 attachment with 2 bodies (HTML and plain text) then there will be at least 2 file attachment annotations – at least one on a page (Content Set) associated with HTML rendering and a separate annotation on a different page (Content Set) associated with the plain text body. There is no requirement that the file attachment annotation is always on the first page of the Content Set, although this is conventional. Note that for efficiency both annotations could refer to the same file specification dictionary and thus the same embedded file stream. Furthermore, both annotations can also share the same appearance stream (e.g., XObject of a paperclip). Additional PDF file attachment annotations may also be added by the EA-PDF Creation Software*.*

Note: this requirement does not constrain how email attachments in other Content Sets (such as an index of email attachments) might be represented.

All pages in an EA-PDF file SHOULD have a unique page number or other identifying context as part of the page content. Such context SHOULD use PDF text objects.

*Reason: this tries to ensure that extracted pages retain some basic context information. It is not a requirement as it cannot be machine validated – however if additional logical structure requirements were mandated then PDF text object (**BT**/**ET**) validation might be possible, but this was viewed as onerous.*

PDF/A always requires the use of device-independent color such as via CIE-based color spaces ([ISO 32000-2], §8.6.5). For unspecified color content in email, sRGB SHOULD be assumed.

Reason: sRGB is the de facto color space of the web, but modern HTML/CSS permits more advanced color spaces. It is unknown if these are (or will be) supported in emails, hence EA-PDF does not place any constraints. The preservation policy may also constrain email conversion to monochrome PDFs (for example).

EA-PDF Writer: many legacy PDF viewers are not "PDF/A conforming processors" and do not support Output Intents ([ISO 32000-2], §14.11.5) while others do not appear to support Default color spaces ([ISO 32000-2], §8.6.5.6) which are common methods utilized by PDF/A files to ensure device-independent color. By using CIE-Based color spaces directly with content streams, such viewer limitations can be overcome.

*EA-PDF Writer: A PDF **CalRGB** color space may be used to <u>approximate</u> sRGB, however various "Tiny sRGB" ICC profiles exist which are very small (~0.5 KB). Because of object reuse in PDF, the sRGB definition need only occur once in each PDF/mail file.*

Note: it cannot be assumed that email is always RGB-based, or that pure text emails are pure black & white. Many email clients do not display plain text emails as pure black so there is no requirement in EA-PDF that plain text emails must be rendered as pure black in a single channel color space. The use of CIE-based color spaces for plain text emails are all suitable as they are all device independent and meet PDF/A requirements.

EA-PDF Writer: PDF graphic state defaults are black content on an assumed white background. EA-PDF Creation Software *does not need to unnecessarily paint white content as a background.*

EA-PDF Writer: if page content stream creation cannot be suitably configured, device independent color in PDF can be achieved using a default color space resources or via PDF output intents at the file level or page-level if PDF 2.0, however several legacy viewers do not appear to support output intents.

Emails with URLs that are intended to be actionable by users or software SHOULD use PDF Link annotations.

Reason: this link requirement cannot be validated without additionally parsing the original source (raw) email assets and understanding the policy environment. This is viewed as onerous for validators and not possible for `PDF/mail-1{c, ci}` so it is stated as a recommendation and not a requirement.

If referenced assets (e.g. file attachments; embedded images) from the source emails needed for page rendering are also being individually preserved (e.g., based on a policy setting), then those referenced assets SHOULD be embedded as Associated Files (**AF** array) with an **AFRelationship** of *Source*. They SHOULD be associated with the most appropriate PDF object (e.g., an Image XObject for images, a Form XObject for SVG vector art, a Font dictionary for font-related data, etc.) Refer to [AssociatedFiles].

Reason: these requirements cannot be validated without additionally parsing the original source (raw) email assets. This is viewed as onerous for validators and not possible for isolated EA-PDF files so they are stated as recommendations and not requirements. It is also not possible to enumerate every possibility and every object to make the above a requirement in all cases.

9.4 Embedded files

All embedded file streams included as Associated Files (i.e. referenced from an **AF** array entry or AF tag in a marked content sequence) SHALL be listed in the Document Catalog **Name** name-tree **EmbeddedFiles** entry.

*Note: a lot of legacy software relies on the **EmbeddedFiles** name-tree to provide basic access to embedded files, while other legacy software may also additionally add file attachment annotations and remove duplicate embedded files. This legacy experience is further constrained as it provides a simple flat list of embedded files using their filename. This can be machine validated.*

PDF file specification strings SHALL NOT include absolute or relative path components (i.e., REVERSE SOLIDUS and two PERIODs SHALL NOT be present).

All embedded file streams SHALL have a **Subtype** key in their embedded file stream dictionary which is the IANA Media (MIME) Type of the file and SHALL match the equivalent document-level (document catalog) XMP metadata. This SHALL NOT include any SEMI-COLONs or IANA Media Type parameters. Where the embedded file originates in a source email asset that also specifies IANA Media Type parameters, the same Media Type parameters SHALL be set as the value of the **Mail_MediaTypeParameters** key (PDF string object).

To reconstruct an email IANA Media Type including parameters the following algorithm SHALL be used:

1. convert the **Subtype** entry (PDF name object) to a UTF-8 equivalent string value as defined in ISO 32000-2, Annex J.3.4;
2. append a SEMI-COLON;
3. append the **Mail_MediaTypeParameters** entry text string (if present) while accounting for any encoding differences (such as UTF-16BE).

Use of generic Media Types (such as `text/plain`, `application/octet-stream`, or similar) SHOULD be avoided.

Reason: these Media Type requirements cannot be validated without additionally parsing the original source (raw) email assets. This is viewed as onerous for validators and not possible for PDF/`mail-1{c, ci}` so they are stated as recommendations and not requirements.

Non-isolated PDF/`mail-1{s, m}` files SHALL embed the original raw source email asset (e.g., EML, MSG, MBOX, OST/PST, etc.) in the Document Catalog Associated File (**AF**) array with array elements referring to an embedded file specification dictionary with an **AFRelationship** entry of *Source*. There SHALL only be at least one such entry in this **AF** array.

The document-level (document catalog) XMP metadata for source email asset containers SHALL match for file size and IANA Media Type.

Note: the XMP Metadata for source email asset container filenames permits the use of absolute and relative paths as well as platform specific components, which is not supported by PDF file specification dictionaries and thus no requirement is specified.

If additional source email assets are required as part of a source email asset file set, then those files can either be added to the Document Catalog Associated File (**AF**) array with array elements referring to file specification dictionaries with an **AFRelationship** value of *Supplement*, or added as a Related Files array (**RF** entry) of the primary original raw source email asset embedded file specification dictionary (which has an **AFRelationship** value of *Source)*.

EA-Writers: PDF lossless compression such as FLATE is recommended for all embedded data streams. If the raw data format is already highly compressed, then the size may grow very slightly.

EA-Readers: for file sets, EA-PDF processors ought to support both the Associated Files Source *with one or more* Supplement *form, as well as the Related Files array form to ensure that any original source email "file sets" always get extracted as a holistic functional set.*

EA-PDF Writers SHOULD add the MD5 checksum and uncompressed size of all embedded files in the embedded file parameter dictionary **CheckSum** and **Size** entries (see [ISO 32000-2], Table 45).

9.5 PDF Collections (PDF/mail-1{c,ci})

The Document Catalog dictionary of all PDF/`mail-1{c, ci}` files SHALL have a **Collections** entry. The **Collections** entry SHALL NOT be present in all other PDF/`mail-1{s, si, m, mi}` files.

PDF/`mail-1{c, ci}` files that are PDF version 1.7 SHALL have an Extensions Dictionary indicating Adobe (**ADBE**) Extension Level 5 with entries **BaseVersion** /1.7 (PDF name object) and **ExtensionLevel** 5, as specified in "*Adobe® Supplement to ISO 32000-1 BaseVersion: 1.7 ExtensionLevel: 5 (Adobe® Acrobat® SDK, Version 9.1)*", dated June 2009 [ADBE-Extn-L5].

The Collection dictionary **D** entry SHALL NOT be present. The Collection dictionary **View** entry SHALL NOT have the value *H*.

Reason: these requirements prohibit opening a specified file in the collection as the default view. EA-PDF relies on opening the container PDF which is what ISO 32000 specifies will happen if the ***D*** *entry is not present.*

If the Collection dictionary contains a **Navigator** entry, it SHALL NOT have either **SWF** or **APIVersion** entries.

Reason: this prohibits vendor-specific support that was not adopted by ISO.

All folder and filenames used in PDF/`mail-1{c,ci}` files SHALL conform to the valid file name restrictions and SHALL be unique after case normalization, as described below Table 159 in [ISO 32000-2].

All PDF/`mail-1{c, ci}` files SHALL have a **Schema** entry in the Collections dictionary. The following custom fields representing the *Core Fields* SHALL all exist in the collection schema dictionary – additional custom fields can also be present:

Key name in Collection Schema	CustomField Subtype
`To`	S (string)
`From`	S (string)
`Sent`	D (date)
`Subject`	S (string)
`Message-ID` [23]	S (string)
`Cc`	S (string)
`Bcc`	S (string)
Size	`Size`
Attachments	N (number)

Table 6: required Collection custom field names and subtypes for Core Fields (PDF/`mail-1{c, ci}`)

If the **E** entry is present in any Collection **CustomField** dictionary it SHALL have the value *false*.

[23] Note that this is the email's `Message-ID` (which may not exist) and is not the synthetically generated `Mail_GUID` used internally.

Reason: this prohibits accidental editing of the CustomField data.

EA-PDF does not further constrain the textual field name presented in the PDF viewer (**N** entry), field ordering (**O** entry), or default visibility (**V** entry) in the Collection schema, or the use of the Collection Sort or Folder dictionaries.

9.6 Document part metadata (DPM)

All EA-PDF files SHALL define document part metadata for all pages via the Document Catalog **DPartRoot** dictionary entry.

Note: consequently, this also requires every page in all EA-PDF files to always have a ***DPart*** *entry.*

Reason: requiring DPM allows additional machine validation of EA-PDF files without needing to process original source email asset file formats, as well as offering richer presentation options in EA-PDF aware software.

All custom EA-PDF keys used in DPM SHALL use the registered prefix "`Mail`" followed by an underscore: "`Mail_`".

Note: some of the same key names introduced for DPM are also used for Logical Structure.

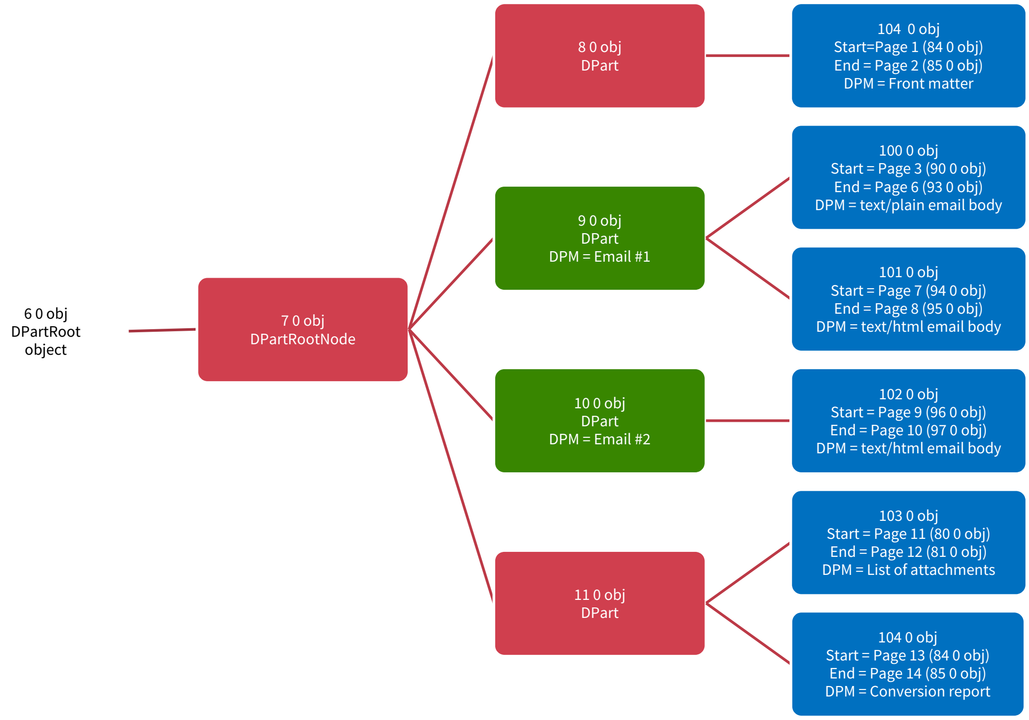


Figure 12: DPart hierarchy for example showing PDF object numbers

Intermediate nodes of the Document Part Metadata tree SHALL be used to associate emails with all *Content Sets* associated with that email, as illustrated by the green nodes (middle column of cells) in *Figure 8: Conceptual illustration of a simplified Document Part Metadata tree.* These intermediate nodes SHALL contain a **DPM** dictionary as specified in *Table 7: Entries in an EA-PDF DPM dictionary for an intermediate node associated with an email* below.

Key	Type	Description
`Mail_MessageID`	text string	Optional. The `Message-ID` from the header of the email and that SHALL match a `pdfmailmeta:messageid` in the document-level XMP metadata. *Note: the Message-ID email header is optional according to [RFC-822], but is extremely common and widely supported with email aware software.*
`Mail_GUID`	text string	**Required**. A GUID that SHALL match a `pdfmailmeta:Mail_GUID` property in the EA-PDF XMP metadata. If **`Mail_MessageID`** is also present, then **`Mail_GUID`** takes precedence in identifying the email. Matching SHALL be performed according to Annex J.3.3 in [ISO 32000-2].

Table 7: Entries in an EA-PDF **DPM** dictionary for an intermediate node associated with an email

PDF/`mail-1{s, si}` files SHALL have exactly one intermediate node in the Document Part Metadata tree with **`Mail_MessageID`** / **`Mail_GUID`** entries.

PDF/`mail-1{m, mi}` files SHALL have at least 2 intermediate nodes in the Document Part Metadata tree with **`Mail_MessageID`** / **`Mail_GUID`** entries.

PDF/`mail-1{c, ci}` files SHALL have no nodes in the Document Part Metadata tree with **`Mail_MessageID`** or **`Mail_GUID`** entries.

Every **`Mail_MessageID`** and **`Mail_GUID`** value SHALL match a corresponding property in the Document Catalog XMP metadata stream.

All leaf nodes of the Document Part Metadata tree SHALL contain a **DPM** dictionary to associate pages derived from a single email message with *Content Sets* as illustrated by the blue nodes (right most column of boxes) in *Figure 8: Conceptual illustration of a simplified Document Part Metadata tree* as defined in *Table 8: Entries in an EA-PDF DPM dictionary for a leaf node associated with pages*.

Key	Type	Description
`Mail_ContentSetType`	name	*Required*. The *Content Set* that all pages in the page range defined between **Start** and **End** (inclusive) of the DPart dictionary represent. Supported values are as follows (other values may be present): • *AttachmentList* – a full list of all email attachments in the EA-PDF file. SHALL NOT occur in PDF/`mail-1{c, ci}` files. • *AttachmentRendering* – see also **Subtype** entry, which SHALL be the Media Type of the original email

Key	Type	Description
		attachment. SHALL NOT occur in PDF`/mail-1{c, ci}` files. • *BodyRendering* – an email body. See also **Subtype** entry which SHALL be the Media Type of the email body. SHALL NOT occur in PDF`/mail-1{c, ci}` files. • *ConversionReport* • *EmailHeaderRendering* – a separate rendering of some or all of the email headers. SHALL NOT occur in PDF`/mail-1{c, ci}` files. • *FrontMatter* - SHALL always occur in a PDF`/mail-1{c, ci}` file, optional in other profiles. • *PolicySettings* • *Provenance* • *RawEmailRendering* – the rendering of some or all of the original source "raw" email. SHALL NOT occur in PDF`/mail-1{c, ci}` files. • *Other* – unspecified
`Mail_Subtype`	name	*Required* if **ContentSetType** is *BodyRendering* or *AttachmentRendering.* Not required otherwise. Indicates the source Media Type and any parameters of the rendered *Content Set.* The value of this entry SHALL conform to the Media Type names defined in Internet RFC 2046, with the provision that characters not permitted in names SHALL use the 2-character hexadecimal code format described in §7.3.5, "Name objects" in [ISO 32000-2]. *Note: this is the same type and definition used by the embedded file stream dictionary **Subtype** entry (see Table 44, ISO 32000-2).*
`Mail_Desc`	text string	*Required*. A human readable text string that describes this *Content Set* that might be presented in the UI of EA-PDF viewers when using DPM.

Table 8: Entries in an EA-PDF **DPM** dictionary for a leaf node associated with pages

An example of the DPart objects in the Document Part Metadata tree representing *Figure 8: Conceptual illustration of a simplified Document Part Metadata tree* (note that the page tree is not shown).

```
6 0 obj
<< /Type /DPartRoot
   /DPartRootNode 7 0 R
>>
endobj
```

```
7 0 obj
<< /Type /DPart
   /Parent 6 0 R
   /DParts [ [ 8 0 R 9 0 R 10 R 11 0 R ] ]
>>
endobj

8 0 obj
<< /Type /DPart
   /Parent 7 0 R
   /DParts [ [ 104 0 R ] ] % the single front matter Content Set
>>
endobj

9 0 obj
<< /Type /DPart
   /Parent 7 0 R
   /DParts [ [ 100 0 R 101 0 R ] ] % the two Content Sets associated with Email #1
   /DPM <<
       /Mail_MessageID (68409d6fde44928ac6cfb79f9c6f23c@email.com)
       /Mail_GUID (68409d6fde44928ac6cfb79f9c6f23c@email.com) % Could be any form of GUID,
                                                              % including email Message-ID
   >>
>>
endobj

10 0 obj
<< /Type /DPart
   /Parent 7 0 R
   /DParts [ [ 102 0 R ] ] % the single Content Set associated with Email #2 (e.g. draft)
   /DPM << /Mail_GUID (6a8420f7-9f9c6f23-ed474811) >>  % generated GUID  as no Message-ID
>>
endobj

11 0 obj
<< /Type /DPart
   /Parent 7 0 R
   /DParts [ [ 103 0 R 104 0 R ] ] % the attachment list and conversion report Content Sets
>>
endobj

99 0 obj
<< /Type /DPart
   /Parent 8 0 R
   /Start 82 0 R  % First page (page 13) of conversion report for full EA-PDF
   /End   83 0 R  % Last page (page 14) of conversion report for full EA-PDF
   /DPM <<
      /Mail_ContentSetType /ConversionReport
      /Mail_Desc (XML conversion report)
   >>
   /AF [ 200 0 R ] % Associated File for Email 1 conversion report (maybe an XML report)
>>
endobj
```

```
100 0 obj
<< /Type /DPart
   /Parent 9 0 R % common intermediate node for Email 1
   /Start 90 0 R % First page (page 3) in plain text email body
   /End   93 0 R % Last page (page 6) in plain text email body
   /DPM <<
      /Mail_ContentSetType /BodyRendering
      /Mail_Subtype /text#2Fplain              % encoded according to PDF name rules
      /Mail_Desc (Plain text email body)       % short human-readable description
   >>
   /Metadata 201 0 R    % metadata concerning Email 1 plain text body only
>>
endobj

101 0 obj
<< /Type /DPart
   /Parent 9 0 R % common intermediate node for Email 1
   /Start 94 0 R % First page (page 7) in HTML email body
   /End   95 0 R % Last page (page 8) in HTML email body
   /DPM <<
      /Mail_ContentSetType /BodyRendering
      /Mail_Subtype /text#2Fhtml          % encoded “/” according to PDF name rules
      /Mail_Desc (HTML email body)        % short human-readable description
   >>
>>
endobj

102 0 obj
<< /Type /DPart
   /Parent 10 0 R % common intermediate node for Email 2
   /Start 96 0 R  % First page (page 9) in HTML email body
   /End   97 0 R  % Last page (page 10) in HTML email body
   /DPM <<
      /Mail_ContentSetType /BodyRendering
      /Mail_Subtype /text#2Fhtml          % encoded “/” according to PDF name rules
      /Mail_Desc (HTML email body)        % short human-readable description
   >>
>>
endobj

103 0 obj
<< /Type /DPart
   /Parent 11 0 R
   /Start 80 0 R  % First page (page 11) of attachment list report for full EA-PDF
   /End   81 0 R  % Last page (page 12) of attachment list report for full EA-PDF
   /DPM <<
      /Mail_ContentSetType /AttachmentList
      /Mail_Desc (List of all email attachments)
   >>
>>
endobj

104 0 obj
<< /Type /DPart
   /Parent 12 0 R
   /Start 84 0 R  % First page (page 1) of front matter
   /End   85 0 R  % Last page (page 2) of front matter
   /DPM <<
      /Mail_ContentSetType /FrontMatter
      /Mail_Desc (Front cover - summary)
   >>
>>
endobj
```

Example 6: document part metadata example

9.7 Logical structure

EA-PDF Creation Software may utilize PDF's Logical Structure and Tagged PDF features to provide enhanced semantics with improved accessibility and reuse of extracted content. EA-PDF defines an optional custom EA-PDF tag-set with specific role mapping back to the PDF 1.7 standard structure tag set.

> *Reason: very few PDF applications appear to process custom PDF tag sets or utilize tagging at all (although adoption is increasing). At the time of writing, the burden of creation for tagging and custom-tagging is left as a design choice for* EA-PDF Creation Software*. Future PDF/mail specifications may mandate the use of a custom EA-PDF tag set.*

> *Note: in this section the word "tag" is used as an informal term for "custom structure element" although it is somewhat ambiguous with the tag operand of certain marked content operators. Where the marked content operators are discussed, the italic term "tag" specifically refers to that operand.*

EA-PDF Creation Software SHOULD create equivalent PDF logical structure to that already present in original source email content (such as encoded in HTML email bodies).

> *Reason: many logical structure requirements cannot be validated without additionally parsing the original source (raw) email assets. This is viewed as onerous for validators and not possible for isolated EA-PDF files, so they are stated as recommendations and not requirements.*

EA-PDF Creation Software SHOULD create PDF logical structure and use Tagged PDF for the content of all pages in custom *Content Sets* that are included in EA-PDF files.

> *Reason: use of logical structure and Tagged PDF ensures both reuse and accessibility of custom content.*

When PDF 2.0 (PDF/A-4) is used, PDF`/mail-1` files with logical structure SHALL also conform to ISO TS 32005 and SHALL include the PDF Declaration for ISO TS 32005 in the document level XMP metadata (see https://pdfa.org/declarations/#ISO_TS_32005 and [PDF-Declarations]).

> *Reason: ISO TS 32005 standardizes the set of inclusion rules for the PDF 1.7 tag set, ensuring improved consistent behavior for reuse and accessibility across processors.*

For all PDF`/mail-1` files that contain Logical Structure, the logical structure tree root structure element as defined by the **K** entry of the structure tree root SHALL be a single structure element dictionary of structure type *Document*. If an array type for the **K** entry of the structure tree root is used, it SHALL have a single array element which SHALL be the structure type *Document*.

In all PDF/A-4 files with logical structure, the custom EA-PDF-defined semantics SHALL be included using a PDF 2.0 Namespace dictionary with an **NS** entry of *`https://pdfa.org/ns/ea-pdf/mail-1`* and including a **RoleMapNS** entry, as shown in Example 2 above. The following role map SHALL also be included (additional role mappings may also be included).

PDF 2.0 unique tags	Compatible PDF 1.7 tag
`Artifact`	`Private`

PDF 2.0 unique tags	Compatible PDF 1.7 tag
Aside	Div
DocumentFragment	Div
Em	Span
FENote	Note
H*n* when *n* ≥ 7	P
Strong	Span
Sub	Div
Title	H1 *or* P

Table 9: PDF 2.0 to PDF 1.7 backward compatibility role map

*Reason: this **RoleMap** will be ignored by all PDF 2.0 processors for structure elements with an explicit namespace (**NS** entry), such as those defined by EA-PDF. However, PDF 1.7 only processors reading a PDF 2.0 EA-PDF file containing this **RoleMap** will use it to create a standard backward-compatible mapping to PDF 1.7 tags (since PDF 1.7 only processors do not support PDF 2.0 namespaces) providing a more consistent behaviour across processors.*

If an EA-PDF file contains any of the following custom EA-PDF structure elements, then the following role map SHALL be included.

Custom EA-PDF structure element	Role mapped PDF 1.7 tag
Mail_Message	Document
Mail_ContentGroup	Part
Mail_ContentSet	Art
Mail_Field	*The PDF Writer SHALL choose the most appropriate PDF 1.7 structure element, such as P (paragraph), Span, list, or table tags.*
Mail_FieldName	
Mail_FieldValue	

Table 10: custom EA-PDF structure elements and their role mapping to PDF 1.7

Note: the use of Mail_ContentGroup *is to semantically represent the grouping of Content Sets. It does not directly relate to a property of an email message.*

Custom EA-PDF tags SHALL only occur on marked content sequences (i.e. they SHALL NOT occur as marked content points). The *tag* operand of marked content sequence operators SHALL match the corresponding custom EA-PDF structure element name.

Reason: ISO 32000 only has a recommendation for matching operands to structure element names.

Parent and immediate child containment rules for custom EA-PDF tags SHALL be as follows.

Parent	Occurrence	Immediate Child [24]
Mail_Message	0 .. *n*	Mail_ContentGroup

[24] Other nesting requirements are described below.

Parent	Occurrence	Immediate Child
	0 .. *n*	`Mail_ContentSet`
	0	*All other custom EA-PDF tags as defined in Table 10.*
`Mail_ContentGroup`	1 .. *n*	`Mail_ContentSet`
	0	*All other custom EA-PDF tags as defined in Table 10.*
`Mail_ContentSet`	0 .. *n*	`Mail_ContentField`
	0	*All other custom EA-PDF tags as defined in Table 10.*
`Mail_Field`	1 .. *n*	`Mail_FieldName`
	0 .. *n*	`Mail_FieldValue` *Note: email field values may be blank and thus not present.*
	0	*All other custom EA-PDF tags as defined in Table 10.*

Table 11: custom EA-PDF tag parent and immediate child occurrence requirements

In addition, the following requirements also apply if custom EA-PDF tags are used.

- For each `Mail_Message` custom EA-PDF tag, there SHALL be at least 1 `Mail_ContentGroup` or `Mail_ContentSet` immediate child.

  Note: this requirement ensures there are no empty email messages in the logical structure without a related Content Set.

- Every *Content Set* shall be tagged with a `Mail_ContentSet` custom EA-PDF tag.
- All header fields rendered into page content SHOULD be tagged with a `Mail_FieldName` and `Mail_FieldValue` custom EA-PDF tag (if present).

  Note: there are no explicit requirements related to email header fields and the `Mail_FieldName` *or* `Mail_FieldValue` *custom EA-PDF tags as validation requires processing of the original source email assets.*

- A `Mail_FieldValue` custom EA-PDF tag SHALL NOT occur without a corresponding a `Mail_FieldName` tag inside the immediate containing parent `Mail_Field` tag.

  Note: the ordering of the `Mail_FieldName` *and* `Mail_FieldValue` *custom EA-PDF tags is purposely not defined to support right-to-left languages and other layouts. The requirement reflects that an email field needs to be tagged with* `Mail_FieldName` *only.*

Certain custom EA-PDF tags may optionally also contain specific custom user property structure attributes when the associated attribute object dictionary has a value of *UserProperties* for the owner (**O**) entry. In this situation, the user property dictionary **N** SHALL be PDF ASCII strings with **V** entries as defined in *Table 12: EA-PDF UserProperties custom attributes N and V entries* below.

N entry (*PDF ASCII string*)	V entry type	Only with custom EA-PDF tag(s)
`Mail_MessageID`	string	`Mail_Message`
`Mail_GUID`	string	`Mail_Message`
`Mail_ContentSetType`	string	`Mail_ContentGroup`, `Mail_ContentSet`
`Mail_Subtype`	string	`Mail_ContentSet`
`Mail_To`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Raw_To`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_From`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Raw_From`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Sent`	PDF date (string)	`Mail_Message`, `Mail_FieldValue`
`Mail_Raw_Sent`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Subject`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_SizeInBytes`	integer (≥ 0)	`Mail_Message`, `Mail_FieldValue`
`Mail_NumberAttachments`	integer (≥ 0)	`Mail_Message`, `Mail_FieldValue`
`Mail_Cc`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Raw_Cc`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Bcc`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_Raw_Bcc`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_InReplyTo`	string	`Mail_Message`, `Mail_FieldValue`
`Mail_`... (*other email header fields*)	string, number or Boolean only	`Mail_Message`, `Mail_FieldName`, `Mail_FieldValue`

Table 12: EA-PDF UserProperties custom attributes **N** and **V** entries

*Note: the UserProperties **N** entries match the DPM data (see Table 7 and Table 8 above) and XMP metadata fields (see Table 18 below) when prefixed with "`Mail_`" to allow validation. ISO 32000-2 Table 362 recommends only to use strings, Booleans and numbers so name values are represented as strings for UserProperties **V** values.*

When using custom user property structure attributes, the values (**V** entries) SHOULD NOT be truncated or modified from the original source email, even if the page rendering of the equivalent information is cropped, hidden, obscured, or truncated.

Implicit or explicit *Art* (article, implicit via **`Mail_ContentSet`**) child structure elements SHALL only be used to represent each *Content Set* in all PDF`/mail-1` files.

Note: this can be validated for consistency using PDF outlines and the document part metadata.

For PDF`/mail-1{m, mi}` files, each email SHALL be represented by an implicit or explicit *Document* structure element (implicitly via **`Mail_Message`**).

> *Note: this can be validated for consistency using PDF outlines and document part metadata.*

In PDF`/mail-1{c, ci}` container files the explicit top-level *Document* structure element represents the container PDF and its related *Content Sets* (*not* emails, as these are in the embedded files in the collection).

Custom EA-PDF tag	Summary
Mail_Message	SHALL occur once in PDF/mail-1{s, si} files. SHALL occur 2 or more times in PDF/mail-1{m, mi} files. SHALL not occur in PDF/mail-1{c, ci} files. SHALL contain at least one **Mail_ContentGroup** or **Mail_ContentSet** child element. SHALL be role mapped to *Document.* SHALL have attribute **Mail_GUID** that SHALL match a Mail_GUID in the DPM and XMP Metadata (if present).
Mail_ContentGroup	SHALL be role mapped to *Part.* Optional, but SHALL always be nested below **Mail_Message**. SHALL contain at least one **Mail_ContentSet** child element. SHOULD have an attribute **Mail_ContentSetType** that SHALL match the DPM and XMP Metadata if present.
Mail_ContentSet	SHALL be role mapped to *Art.* SHALL always occur at least once in all PDF/mail-1 files. SHALL always be nested below **Mail_Message**. SHOULD have an attribute **Mail_Subtype** that SHALL match the DPM if present.
Mail_Field	SHALL only occur in PDF/mail-1{s, si, m, mi} files. SHOULD always be nested below **Mail_ContentSet**. SHALL contain at least one **Mail_FieldName** child element. SHALL be role mapped appropriately.
Mail_FieldName	SHALL only occur in PDF/mail-1{s, si, m, mi} files. SHOULD always be nested below **Mail_Field**. SHALL be role mapped appropriately.
Mail_FieldValue	SHALL only occur in PDF/mail-1{s, si, m, mi} files. SHOULD always be nested below **Mail_Field**. SHALL only occur if **Mail_FieldName** is also nested in the same parent **Mail_Field**. SHALL be role mapped appropriately.

Table 13: Summary of custom EA-PDF tagging main requirements

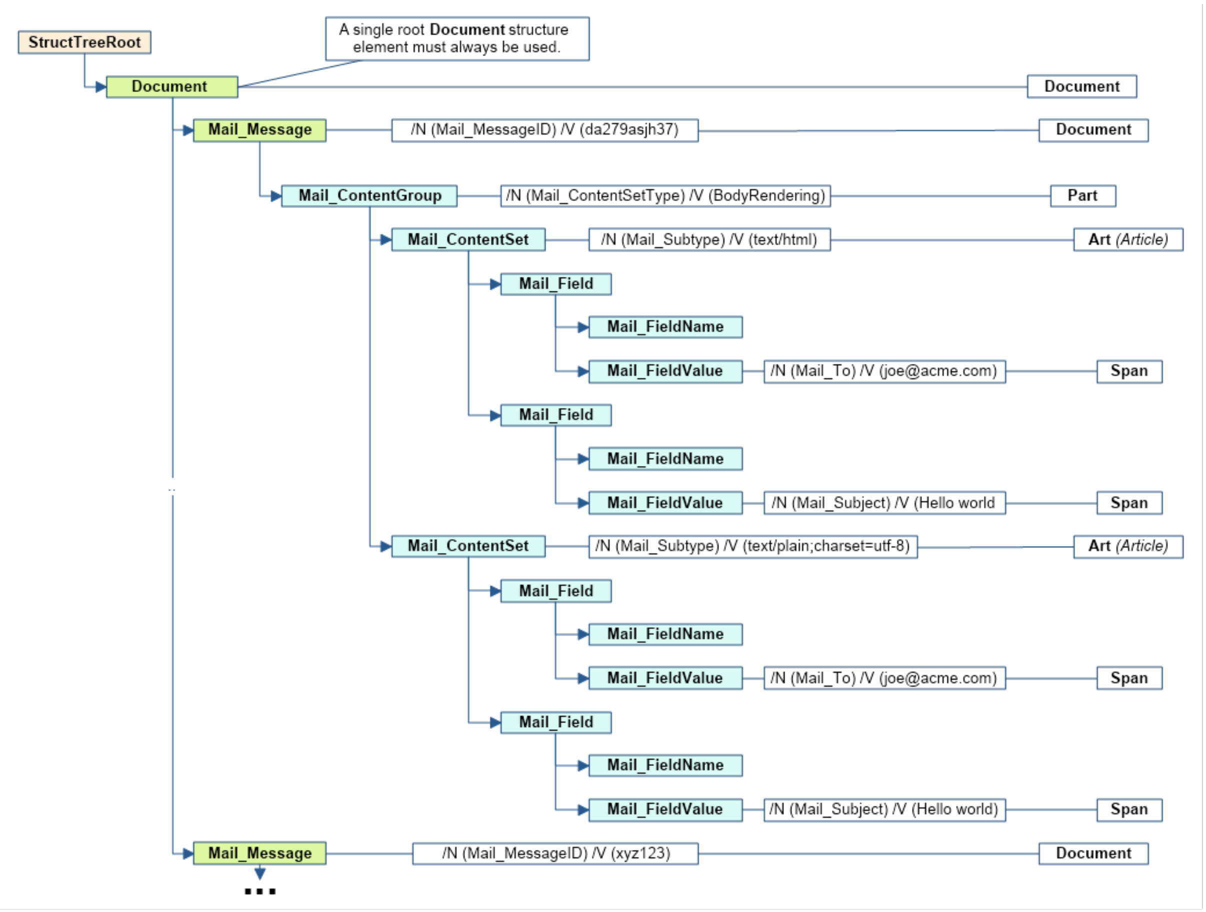

StructTreeRoot
A single root **Document** structure element must always be used.
Document
Document
Mail_Message
/N (Mail_MessageID) /V (da279asjh37)
Document
Mail_ContentGroup
/N (Mail_ContentSetType) /V (BodyRendering)
Part
Mail_ContentSet
/N (Mail_Subtype) /V (text/html)
Art *(Article)*
Mail_Field
Mail_FieldName
Mail_FieldValue
/N (Mail_To) /V (joe@acme.com)
Span
Mail_Field
Mail_FieldName
Mail_FieldValue
/N (Mail_Subject) /V (Hello world
Span
Mail_ContentSet
/N (Mail_Subtype) /V (text/plain;charset=utf-8)
Art *(Article)*
Mail_Field
Mail_FieldName
Mail_FieldValue
/N (Mail_To) /V (joe@acme.com)
Span
Mail_Field
Mail_FieldName
Mail_FieldValue
/N (Mail_Subject) /V (Hello world)
Span
Mail_Message
/N (Mail_MessageID) /V (xyz123)
Document
...

Figure 13: Illustration of custom EA-PDF structure elements and attributes with role mappings to PDF 1.7

```
10 0 obj                           % Structure tree root
<< /Type /StructTreeRoot
   /K 100 0 R                      % Single child → Document. NOT ARRAY!
   /RoleMap <<
     /Artifact          /Private   % Backward compatibility PDF 2.0 → PDF 1.7 role mappings
     /Aside             /Div
     /DocumentFragment  /Div
     /Em                /Span
     /FENote            /Note
     /H7                /P
     /Strong            /Span
     /Sub               /Div
     /Title             /H1
     /Mail_Message      /Document  % EA-PDF custom role mappings → PDF 1.7
     /Mail_ContentGroup /Part
     /Mail_ContentSet   /Art
     /Mail_Field        /Span
     /Mail_FieldName    /Span
     /Mail_FieldValue   /Span
     …                             % Any other role mappings as required
   >>
   /ClassMap << … >>               % Class map containing attribute classes (if used)
   /ParentTree …                   % Number tree for parent elements
   /ParentTreeNextKey …            % Next key to use in parent tree
   /IDTree …                       % Name tree for element identifiers
   /Namespaces [ 11 0 R … ]        % Array for all PDF 2.0 Namespaces. Ignored by PDF 1.7.
>>
endobj
11 0 obj                           % see Table 356 in ISO 32000-2:2020 (PDF 2.0)
<< /Type /Namespace
   /NS (https://pdfa.org/ns/ea-pdf/mail-1) % URI for EA-PDF namespace for PDF/mail-1
   /RoleMapNS <<                   % Role map back to PDF 1.7 standard structure elements
     /Mail_Message      /Document
     /Mail_ContentGroup /Part
     /Mail_ContentSet   /Art
     /Mail_Field        /Span
     /Mail_FieldName    /Span
     /Mail_FieldValue   /Span
   >>
>>
endobj
100 0 obj                          % Structure tree root element for EA-PDF Document
<< /Type /StructElem
   /S    /Document
   /ID   (EA-PDF1)                 % Element identifier
   /T    (EA-PDF for …)            % Human-readable title for the document
   /P    10 0 R                    % Parent is the structure tree root above
   /K    [ … 101 0 R … ]           % Child includes Mail_Message
>>
endobj
```

```
101 0 obj                          % Structure element encapsulating the single email
<< /Type /StructElem
   /S    /Mail_Message             % Role mapped back to Document above
   /ID   (MessageID1)              % Element identifier
   /T    (Message-ID: da279asjh37) % Human-readable title
   /P    100 0 R                   % Parent is the Document for the EA-PDF file
   /Pg   500 0 R                   % Page containing content (below)
   /A    <<
       /O /UserProperties          % Custom attributes
       /P [ << /N (Mail_GUID)      % Mail_GUID is always needed
               /V (da279asjh37)    % The GUID that matches DPM and XMP metadata
            >>
            <<
               /N (Mail_SizeInBytes)
               /V 68738
            >>
            <<
               /N (Mail_NumberAttachments)
               /V 2
            >>
            …
       ]
   >>
   /K    [ 102 0 R ]               % Single child Mail_ContentGroup
>>
endobj

102 0 obj                          % Structure element for Mail_ContentGroup
<< /Type /StructElem
   /S    /Mail_ContentGroup
   /ID   (EmailBodyGroup1)         % Element identifier
   /T    (Email bodies)            % Human-readable title
   /P    101 0 R                   % Parent is the Mail_Message
   /Pg   500 0 R                   % Page containing content (below)
   /A    <<
       /O /UserProperties          % Custom attributes
       /P [ << /N (Mail_ContentSetType)
               /V (BodyRendering)  % Match DPM usage to identify kind of content set
            >>
       ]
   >>
   /K    [ … 103 0 R 104 0 R … ]   % 2 children Mail_ContentSets
>>
endobj

103 0 obj                          % Structure element for Mail_ContentGroup
<< /Type /StructElem
   /S    /Mail_ContentGroup
   /ID   (Email1BodyHtml)          % Element identifier
   /T    (HTML email body)         % Human-readable title
   /P    102 0 R                   % Parent is the Mail_ContentGroup
   /Pg   500 0 R                   % Page containing content (below)
   /A    <<
       /O /UserProperties          % Custom attributes
       /P [ << /N (Mail_Subtype)
               /V (text/html; charset=utf-8) % MIME type, matching DPM and XMP metadata
            >>
       ]
   >>
   /K    [ … ]                     % children
>>
endobj
```

```
104 0 obj                          % Structure element for Mail_ContentGroup
<< /Type /StructElem
   /S    /Mail_ContentGroup
   /ID   (Email1BodyText)          % Element identifier
   /T    (Plain text body)         % Human-readable title
   /P    102 0 R                   % Parent is the Mail_ ContentGroup
   /Pg   500 0 R                   % Page containing content (below)
   /A    <<
       /O /UserProperties          % Custom attributes
       /P [ << /N (Mail_Subtype)
               /V (text/plain)     % MIME type, matching email, DPM and XMP metadata
            >>
       ]
   >>
   /K    [ … ]                     % children
>>
endobj

...

500 0 obj
<< … keys for page content stream for an email body content set … >>
stream
  …
  /Mail_Field BMC
     …
     /Mail_FieldName BMC
        /Span << /Lang (fr) >> BDC % Email rendered using French user interface
           BT (Objet:) TJ ET       % French for “subject” as text object
        EMC
     EMC
     …
     /Mail_FieldValue … BDC        % related attributes: /N (Mail_Subject) /V (Hello world)
        BT 3 Tr (Hello world) TJ ET  % as invisible text object
     EMC
     …
  EMC
  …
endstream
endobj
```

Example 7: simplified custom EA-PDF tagging example

9.8 XMP metadata

Like all PDF ISO subsets, EA-PDF files SHALL presumptively declare their full conformance using the document catalog XMP **Metadata** stream. This XMP metadata stream provides the definitive information about the email messages preserved in the EA-PDF file.

All EA-PDF files SHALL contain a Document Catalog **Metadata** XMP stream including the following namespaces and required prefixes, in addition to those required for their appropriate PDF/A conformance:

URI	Required Prefix
`http://www.pdfa.org/eapdf/`	`pdfmail`
`http://www.pdfa.org/eapdf/ns/id/`	`pdfmailid`
`http://www.pdfa.org/eapdf/ns/meta/`	`pdfmailmeta`

Table 14: Custom XMP namespaces and prefixes for EA-PDF

Note: namespace URIs do not need to exist as websites (i.e., clickable URLs).

Note: the PDF/A conformance level is determined by EA-PDF Creation Software *and can be based on features present in the email messages.*

EA-PDF Creation Software SHALL also record the `dc:creator[0]`, `pdf:Producer` and the date of the EA-PDF file creation date (`xmp:CreateDate`) in the standard XMP metadata fields as defined in [PDF/A-3], Table 7. The `dc:title["x-default"]` property SHOULD also be set.

Reason: many PDF viewers display Title information at the top of viewing window.

9.8.1 Conformance and identification

The EA-PDF Identification schema SHALL use the namespace URI `http://www.pdfa.org/eapdf/ns/id/` and SHALL use the schema namespace prefix `pdfmailid` as follows:

XMP Property	XMP Value Type	Value
`pdfmailid:version` [25]	Open choice of integer	`1`
`pdfmailid:rev`	Open choice of integer	`2024` (representing a year)
`pdfmailid:conformance`	Open choice of text. Always lowercase.	'`s`', '`si`', '`m`', '`mi`', '`c`' or '`ci`' The "`i`" designator SHALL always be last if present.

Table 15: EA-PDF identification schema properties

Reason: this is how all PDF ISO subsets declare conformance in XMP, with similarly named properties. The use of a new namespace for EA-PDF ensures that an EA-PDF file can also be marked as conforming with PDF/A and PDF/UA, and that future versions or revisions of EA-PDF can also be supported. This also forces PDF files to conform to a single conformance level for such standard, including EA-PDF.

As a result of PDF/A-3 requirements, an XMP extension schema for EA-PDF must also be included according to PDF/A-3, §6.6.2.3.2 requirements if the EA-PDF conforms to PDF/A-3.

[25] `pdfmailid:part` is reserved for future ISO standardization.

9.8.2 Source email assets

For non-isolated EA-PDF files that include any source email assets (whether original raw (unmodified) source email asset(s) or modified), additional metadata about those source email asset(s) SHALL be recorded in the document-level (Document Catalog) XMP metadata.

The following structured metadata in the document-level XMP **Metadata** stream using the namespace `http://www.pdfa.org/eapdf/ns/meta/` SHALL be included as an unordered set of one or more source email asset container descriptions using `pdfmailmeta:assets` as follows:

Source email asset property	Requirement	XMP Metadata description
`Filename`	*Required*	Text string of the case-sensistive filename of the original email asset. *Note: XMP in PDF is always UTF-8 so the filename can contain Unicode characters. It may also include absolute or relative path components or platform-specific components in the metadata.*
`SizeInBytes`	*Required*	A non-negative integer that is the size (in bytes) of this source email asset.
`ContentType`	*Required*	Free text string representing the IANA Media Type (including any required parameters) of this email asset. Refer to *Table 17: Common source email asset IANA Media Types* below for commonly used source email asset Media Types. Other IANA Media Types are allowed, but non-specific generic Media Types such as `application/octet-stream` , `application/zip`, or `text/plain` SHOULD NOT be used.
`NumberMessages`	*Optional*	A non-negative integer representing the number of individual email messages in this source email asset that have been successfully rendered into this EA-PDF file. *Note: emails with fatal errors and that could not be converted to EA-PDF are not included in this count.* *Note: this value cannot be validated without re-processing the original source email assets. However it can be checked for consistency with the* Content Sets, *outlines, DPM, etc.*
`Checksum`	*Required*	The MD5 checksum of the original email asset, stored as a 32 character long hexadecimal ASCII string (upper and lowecase alphanumerics). *Note: this checksum is not used for security and is only provided to help detect if the embedded data has been modified or corrupted.*

Table 16: EA-PDF source email asset structured XMP metadata

Reason: an unordered set of source email assets supports sets of files, with only a single asset needing to list the number of email messages.

Email file format	IANA Media Type [26]	Number of emails
EML	`message/rfc822` [27]	1
MSG	`application/vnd.ms-outlook` [28]	1
MBOX	`application/mbox` [29]	*1…N*
OST, PST	`application/vnd.ms-outlook` [30]	*1…N*
NSF	`application/vnd.lotus-notes` [31]	*1…N*

Table 17: Common source email asset IANA Media Types. Other Media Types are allowed.

9.8.3 Email headers

All non-empty *Core Fields* for each email in PDF`/mail-1{s,si,m,mi}` files SHALL be recorded as structured metadata in the document-level XMP **Metadata** stream using the namespace "`http://www.pdfa.org/eapdf/ns/meta/`" with the required schema namespace prefix `pdfmailmeta` as an unordered set of email containers as follows.

EA-PDF Creation Software may add additional email header fields using the case-sensitive email header field name. *EA-PDF Creation Software* may also prefix header fields with "`Raw-`" to indicate a raw value from the email that would otherwise be an error when using a more rigid or structured XMP data type (e.g. an XMP dateTime).

Note: these properties cannot be "SHALL" requirements as they cannot be validated without re-processing the original source email. They can however be validated for consistency with Content Sets*, outlines, DPM, and logical structure (if present) in the EA-PDF file.*

Core Field	Requiremen t	XMP Metadata description
`To`	*Optional*	Represented as an ordered sequence of FOAF Agents containers (`foaf:Agent` entries), each with required `foaf:mbox` email address and optional `foaf:name` or other FOAF Agent fields.
`Raw-To`	*Optional*	Free text string, directly reflecting the email header `To:` field value.

[26] See https://www.iana.org/assignments/media-types/media-types.xhtml for registered Media Types. Not all media types for email formats are registered.

[27] https://www.loc.gov/preservation/digital/formats/fdd/fdd000388.shtml

[28] https://www.loc.gov/preservation/digital/formats/fdd/fdd000379.shtml

[29] https://www.loc.gov/preservation/digital/formats/fdd/fdd000383.shtml

[30] https://www.loc.gov/preservation/digital/formats/fdd/fdd000378.shtml

[31] https://www.loc.gov/preservation/digital/formats/fdd/fdd000433.shtml

Core Field	Requirement	XMP Metadata description
`From`	*Optional*	Represented as an ordered sequence of FOAF Agents containers (`foaf:Agent` entries), each with required `foaf:mbox` email address and optional `foaf:name` or other FOAF Agent fields.
`Raw-From`	*Optional*	Free text string, directly reflecting the email header `From:` field value.
`Sent`	*Required*	Represented as a `xsd:dateTime`. Due to potential errors in email header `Sent:` fields, this value may have been normalized to a valid XMP date/time value by *EA-PDF Creation Software*. *Note: RFC 822 defines this field as "Date" which can be ambiguous when out of context. EA-PDF uses the term "Sent".*
`Raw-Sent`	*Sometimes required*	Required when the email header `Sent:` field value has an error that would result in an invalid XMP dateTime. Otherwise SHOULD not be present. Represented as a free text string, directly reflecting the email header Sent field value. *Note: RFC 822 defines this field as "Date" which can be ambiguous when out of context. EA-PDF uses the term "Sent".*
`Subject`	*Optional*	Free text string.
`Message-ID`	*Optional*	Free text string. *Note: some Message IDs may include an "@" sign and domain name and appear as an email address, however Message-IDs are never FOAF Agents.*
`Mail_GUID`	*Required*	An XMP GUID (see ISO 16684-1:2019, §8.2.2.3) that uniquely identifies an email represented as a single free text string. *Note: according to RFC 822 the Message-ID: core field is optional, so this required synthetic unique identifier allows a mapping between XMP metadata and PDF content.* *Note: this property name matches the other uses in EA-PDF, including PDF second-class keys.*

Core Field	Requirement	XMP Metadata description
`SizeInBytes`	*Optional*	A non-negative integer representing the size (in bytes) of the original source of only this email message. *Note: this property is only meaningful for certain email formats such as EML, MSG and MBOX.* *Note: this property includes all headers, all email attachments that may be encoded as Base64, etc. in the original email format.*
`NumberAttachments`	*Required if 1 or more attachments*	A non-negative integer representing the number of email attachments in this email message. Default value is 0.
`Cc`	*Optional*	Represented as an ordered sequence of FOAF Agent containers (`foaf:Agent` entries), each with required `foaf:mbox` email address and optional `foaf:name` or other FOAF Agent fields.
`Raw-Cc`	*Optional*	Free text string, directly reflecting the email header `cc:` field value.
`Bcc`	*Optional*	Represented as an ordered sequence of FOAF Agent containers (`foaf:Agent` entries), each with required `foaf:mbox` email address and optional `foaf:name` or other FOAF Agent fields.
`Raw-Bcc`	*Optional*	Free text string, directly reflecting the email header `bcc:` field value.
`In-Reply-To`	*Optional*	Unordered sequence of free text strings, directly reflecting the email header `In-Reply-To:` field value(s). *Note: some In-Reply-To message IDs may include an "@" sign and domain name and appear as an email address. In-Reply-To message IDs are not represented as FOAF Agents in XMP metadata.* *Note: although this email field can only occur once, it may contain one or more message IDs separated by whitespace.*

Table 18: EA-PDF Core Fields[32] as structured XMP metadata

Note: by convention with other PDF ISO subset standards such as PDF/A, new XMP fields introduced with EA-PDF generally use "camel case", with capitalization matching the [RFC-822] field names. However, to support existing

[32] Note that some Core Fields such as Message-ID and Subject are optional according to [RFC-822]. Other field values may also be empty/blank in draft emails.

third party XMP schemas and correspondence with PDF second-class names, this is not always the case and is not mandated.

9.8.4 Email attachments

For EA-PDF files that include any emails with attachments (whether preserved as PDF file annotations or removed for policy reasons), additional metadata about all email attachments in the original source email assets SHOULD be recorded in the document-level (Document Catalog) XMP metadata.

Note: this cannot be a "SHALL" requirement as it cannot be validated without re-processing the original source email. It can however be validated for consistency with embedded file streams in the EA-PDF file.

The following structured metadata in the document-level XMP **Metadata** stream using the namespace `http://www.pdfa.org/eapdf/ns/meta/` SHALL be included as an unordered set of zero or more source email attachment descriptions `pdfmailmeta:attachments` as follows:

Attachment property	Requirement	XMP Metadata
`Filename`	*Required*	A free text string representing the case-sensistive filename of the email attachment as stored in the original email asset. *Note: XMP in PDF is always UTF-8 so the filename can contain Unicode characters.* *Note: due to limitations with PDF filenames, the corresponding PDF embedded file stream filename may not be an exact match.* *Note: the filename can come from either the Content-Disposition header 'filename' parameter or the Content-Type header 'name' parameter. Both are optional, so it is possible that an email attachment will not have a filename specified in the email headers. In this case a suitable string will need to be generated by the EA-PDF Creation Software.*
`SizeInBytes`	*Required*	A non-negative integer, reflecting the size (in bytes) of this email attachment as stored in the EA-PDF as an embedded file stream. *Note: email attachments that were removed for policy or cybersecurity reasons may have zero length.* *Note: this property is expected to match the corresponding embedded file stream parameter dictionary **Size** entry (if present).*
`Message-ID`	*Optional*	Text string reflecting the `Message-ID` of the original email containing this email attachment.

Attachment property	Requirement	XMP Metadata
`Mail_GUID`	*Required if `Message-ID` is not present*	Text string reflecting the generated GUID of the email containing this email attachment. *Note: this property name matches the DPM entry.*
`Content-Type`	*Required*	A free text string, representing the IANA Media Type (including any media type parameters) of this email attachment as specified in the original email asset. *Note: because of potential errors in the headers of source emails, this string is not defined as an IANA Media Type and must always reflect the raw field in the original source email.* *Note: this property is equivalent to the corresponding embedded file stream dictionary **Subtype** and **Mail_MediaTypeParameters** entries when combined according to the algorithm above.*
`CheckSum`	*Optional*	A 32 character ASCII text string, representing the MD5 checksum (as hexadecimal, upper or lower case alphanumerics) of this email attachment when processed as its raw binary data. *Note: this property is equivalent to the corresponding embedded file parameter dictionary **CheckSum** entry (subject to encoding differences, if present).* *Note: this is not security related.*

Table 19: EA-PDF email attachment structured XMP metadata