
**Information and documentation —
WARC file format**

Information et documentation — Format de fichier WARC

STANDARDSISO.COM : Click to view the full PDF of ISO 28500:2017



STANDARDSISO.COM : Click to view the full PDF of ISO 28500:2017



COPYRIGHT PROTECTED DOCUMENT

© ISO 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

	Page
Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms, definitions and abbreviated terms	2
4 File and record model	3
5 Named fields	5
5.1 General.....	5
5.2 WARC-Record-ID (mandatory).....	5
5.3 Content-Length (mandatory).....	5
5.4 WARC-Date (mandatory).....	6
5.5 WARC-Type (mandatory).....	6
5.6 Content-Type.....	6
5.7 WARC-Concurrent-To.....	7
5.8 WARC-Block-Digest.....	7
5.9 WARC-Payload-Digest.....	7
5.10 WARC-IP-Address.....	8
5.11 WARC-Refers-To.....	8
5.12 WARC-Refers-To-Target-URI.....	8
5.13 WARC-Refers-To-Date.....	8
5.14 WARC-Target-URI.....	9
5.15 WARC-Truncated.....	9
5.16 WARC-Warcinfo-ID.....	9
5.17 WARC-Filename.....	9
5.18 WARC-Profile.....	10
5.19 WARC-Identified-Payload-Type.....	10
5.20 WARC-Segment-Number.....	10
5.21 WARC-Segment-Origin-ID.....	10
5.22 WARC-Segment-Total-Length.....	10
6 WARC record types	11
6.1 General.....	11
6.2 'warcinfo'.....	11
6.3 'response'.....	11
6.3.1 General.....	11
6.3.2 'http' and 'https' schemes.....	12
6.3.3 Other URI schemes.....	12
6.4 'resource'.....	12
6.4.1 General.....	12
6.4.2 'http' and 'https' schemes.....	12
6.4.3 'ftp' scheme.....	12
6.4.4 'dns' scheme.....	13
6.4.5 Other URI schemes.....	13
6.5 'request'.....	13
6.5.1 General.....	13
6.5.2 'http' and 'https' schemes.....	13
6.5.3 Other URI schemes.....	13
6.6 'metadata'.....	13
6.7 'revisit'.....	14
6.7.1 General.....	14
6.7.2 Profile: Identical Payload Digest.....	14
6.7.3 Profile: Server Not Modified.....	15
6.7.4 Other profiles.....	15

6.8	'conversion'	15
6.9	'continuation'	16
7	Record segmentation	16
8	WARC file name, size and compression	16
Annex A	(informative) Use cases for writing WARC records	18
Annex B	(informative) Examples of WARC records	21
Annex C	(informative) WARC file size and name recommendations	24
Annex D	(informative) Compression recommendations	25
Bibliography		26

STANDARDSISO.COM : Click to view the full PDF of ISO 28500:2017

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 46, *Information and documentation*, Subcommittee 4, *Technical interoperability*.

This second edition cancels and replaces the first edition (ISO 28500:2009), which has been technically revised.

Introduction

Websites and web pages emerge and disappear from the World Wide Web every day. For the past 10 years, memory storage organizations have tried to find the most appropriate ways to collect and keep track of this vast quantity of important material using web-scale tools such as web crawlers. A web crawler is a program that browses the web in an automated manner according to a set of policies; starting with a list of URLs, it saves each page identified by a URL, finds all the hyperlinks in the page (e.g. links to other pages, images, videos, scripting or style instructions, etc.), and adds them to the list of URLs to visit recursively. Storing and managing the billions of saved web page objects itself presents a challenge.

At the same time, those same organizations have a rising need to archive large numbers of digital files not necessarily captured from the web (e.g. entire series of electronic journals, or data generated by environmental sensing equipment). A general requirement that appears to be emerging is for a container format that permits one file simply and safely to carry a very large number of constituent data objects for the purpose of storage, management, and exchange. Those data objects (or resources) need to be of unrestricted type (including many binary types for audio, CAD, compressed files, etc.), but fortunately the container needs only minimal knowledge of the nature of the objects.

The WARC (Web ARChive) file format offers a convention for concatenating multiple resource records (data objects), each consisting of a set of simple text headers and an arbitrary data block into one long file. The WARC format is an extension of the ARC file format (ARC) that has traditionally been used to store “web crawls” as sequences of content blocks harvested from the World Wide Web. Each capture in an ARC file is preceded by a one-line header that very briefly describes the harvested content and its length. This is directly followed by the retrieval protocol response messages and content. The original ARC format file has been used by the Internet Archive (IA) since 1996 for managing billions of objects, and by several national libraries.

The motivation to extend the ARC format arose from the discussion and experiences of the International Internet Preservation Consortium (IIPC), whose members include the national libraries of Australia, Canada, Denmark, Finland, France, Iceland, Italy, Norway, Sweden, The British Library (UK), The Library of Congress (USA), and the Internet Archive (IA). The California Digital Library and the Los Alamos National Laboratory also provided input on extending and generalizing the format.

The WARC format offers a standard way to structure, manage and store billions of resources collected from the web and elsewhere. It is used to build applications for harvesting, managing, accessing, mining and exchanging content. While it represents the unique standard format for web archives, it has been adopted beyond the web archiving community to store born-digital or digitized materials. The way WARC files will be created and resources stored and rendered will depend on software and applications implementations.

Besides the primary content recorded in ARCs, the extended WARC format accommodates related secondary content, such as assigned metadata, abbreviated duplicate detection events, later-date transformations, and segmentation of large resources. The extension may also be useful for more general applications than web archiving. To aid the development of tools that are backwards compatible, WARC content is clearly distinguishable from pre-revision ARC content.

The WARC file format is made sufficiently different from the legacy ARC format files so that software tools can unambiguously detect and correctly process both WARC and ARC records; given the large amount of existing archival data in the previous ARC format, it is important that access and use of this legacy not be interrupted when transitioning to the WARC format.

Information and documentation — WARC file format

1 Scope

This document specifies the WARC file format:

- to store both the payload content and control information from mainstream Internet application layer protocols, such as the HTTP, DNS, and FTP;
- to store arbitrary metadata linked to other stored data (e.g. subject classifier, discovered language, encoding);
- to support data compression and maintain data record integrity;
- to store all control information from the harvesting protocol (e.g. request headers), not just response information;
- to store the results of data transformations linked to other stored data;
- to store a duplicate detection event linked to other stored data (to reduce storage in the presence of identical or substantially similar resources);
- to be extended without disruption to existing functionality;
- to support handling of overly long records by truncation or segmentation, where desired.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

RFC1035¹⁾ Mockapetris, P. *Domain names — Implementation and specification*, STD 13, November 1987

RFC2045²⁾ Freed, N. and Borenstein, N. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, November 1996

RFC2540³⁾ Eastlake, D. *Detached Domain Name System (DNS) Information*, March 1999

RFC2616⁴⁾ Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Berners-Lee, T. *Hypertext Transfer Protocol — HTTP/1.1*. June 1999 (TXT, PS, PDF, HTML, XML)

RFC3629⁵⁾ Yergeau, F. *UTF-8, a transformation format of ISO 10646*. STD 63, November 2003

RFC3986⁶⁾ Berners-Lee, T., Fielding, R., Masinter, L. *Uniform Resource Identifier (URI): Generic Syntax*. STD 66, January 2005 (TXT, HTML, XML)

1) Available at: <https://www.ietf.org/rfc/rfc1035.txt>.

2) Available at: <https://www.ietf.org/rfc/rfc2045.txt>.

3) Available at: <https://tools.ietf.org/html/rfc2540>.

4) Available at: <https://www.ietf.org/rfc/rfc2616.txt>.

5) Available at: <https://tools.ietf.org/html/rfc3629>.

6) Available at: <https://www.ietf.org/rfc/rfc3986.txt>.

RFC4027⁷⁾ Josefsson, S. *Domain Name System Media Types*, April 2005

RFC4291⁸⁾ Hinden, R. and Deering, S. *IP Version 6 Addressing Architecture*, February 2006

RFC5322⁹⁾ Resnick, P. (ed.) *Internet Message Format*, October 2008

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <http://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

3.1.1

WARC record

basic constituent of a WARC file, consisting of a sequence of WARC records

3.1.2

WARC record content block

part (zero or more octets) of a WARC record that follows the header and that forms the main body of a WARC record

3.1.3

WARC record payload

data object referred to, or contained by a WARC record as a meaningful subset of the content block

3.1.4

WARC record header

beginning of a WARC record, consisting of one first line declaring the record to be in the WARC format with a given version number, followed by lines of named fields up to a blank line

3.1.5

WARC named fields

set of elements consisting of a name, a colon, and a value, with long values continued on indented lines

3.1.6

WARC logical record

<segmentation> record composed of multiple segments, each represented by a WARC record

3.2 Abbreviated terms

ABNF	augmented Backus-Naur form
ARC	archive
CRLF	carriage return line feed
DNS	domain name system
FTP	file transfer protocol

7) Available at: <https://tools.ietf.org/html/rfc4027>.

8) Available at: <https://tools.ietf.org/html/rfc4291>.

9) Available at: <https://tools.ietf.org/html/rfc5322>.

HTTP	hypertext transport protocol
IANA	Internet Assigned Numbers Authority
IESG	Internet Engineering Steering Group
RFC	request for comments
UR (I/L/N)	uniform resource (identifier/locator/name)
WARC	web archive

4 File and record model

A WARC format file is the simple concatenation of one or more WARC records. The first record usually describes the records to follow. In general, record content is either the direct result of a retrieval attempt (web pages, inline images, URL redirection information, DNS hostname lookup results, stand-alone files, etc.) or is synthesized material (e.g. metadata, transformed content) that provides additional information about archived content.

A WARC record shall consist of a record header followed by a record content block and two new lines. The WARC record header shall consist of one first line declaring the record to be in the WARC format with a given version number, then a variable number of line-oriented named fields terminated by a blank line. The WARC record header format shall follow the general rules of HTTP/1.1 [RFC2616] and [RFC5322] headers with one major exception: it shall also allow UTF-8 characters, as specified in [RFC3629].

The top-level view of a WARC file can be expressed in an ABNF grammar, reusing the augmented constructs defined in section 2.1 of HTTP/1.1 [RFC2616]. (In particular, note that to avoid the risk of confusion, where any WARC rule has the same name as an [RFC2616] rule, the definition here has been made the same, except in the case of the CHAR rule, which in WARC includes multibyte UTF-8 characters.)

```
warc-file      = 1*warc-record
warc-record   = header CRLF
                block CRLF CRLF
header        = version warc-fields
version       = "WARC/1.1".CRLF
warc-fields   = *named-field CRLF
block         = *OCTET
```

The record version shall appear first in every record and hence shall also begin the WARC file itself.

The WARC record relies heavily on named fields. Each named field consists of a name followed by a colon (":") and the field value. Field names are not case-sensitive. The field value may be preceded by any amount of linear white space (LWS), though a single space is preferred. Header fields can be extended over multiple lines by preceding each extra line with at least one space or tab character.

Named fields may appear in any order and field values may contain any UTF-8 character. Both defined-fields and extension-fields follow the generic named-field format. Extension-fields may be used in extensions of the core format.

```
named-field   = field-name ":" [ field-value ]
field-name    = token
field-value   = *(field-content | LWS)          ; further qualified
                                                    ; by field
                                                    ; definitions
field-content = <the OCTETs making up the field-value
                and consisting of either *TEXT or combinations
                of token, separators, and quoted-string>
OCTET        = <any 8-bit sequence of data>
token        = 1*<any US-ASCII character>
                except CTLs or separators>
separators   = "(" | ")" | "<" | ">" | "@"
```

```

        | "," | ";" | ":" | "\" | "<">
        | "/" | "[" | "]" | "?" | "="
        | "{" | "}" | SP | HT
TEXT      = <any OCTET except CTLs,
           but including LWS>
CHAR      = <UTF-8 characters; RFC 3629> ; (0-191, 194-244)
DIGIT     = <any US-ASCII digit "0".."9">
CTL       = <any US-ASCII control character
           (octets 0 - 31) and DEL (127)>
CR        = <ASCII CR, carriage return> ; (13)
LF        = <ASCII LF, linefeed> ; (10)
SP        = <ASCII SP, space> ; (32)
HT        = <ASCII HT, horizontal-tab> ; (9)
CRLF     = CR LF
LWS       = [CRLF] 1*(SP | HT) ; semantics same as
           ; single SP
quoted-string = (<"> *(qdtex | quoted-pair) <"> )
qdtex      = <any TEXT except <">>
quoted-pair = "\" CHAR ; single-character quoting
uri        = <'URI' per RFC3986>

```

Although UTF-8 characters are allowed, the ‘encoded-word’ mechanism of [RFC2047] may also be used when writing WARC fields and shall also be understood by WARC reading software.

NOTE In WARC 1.0 standard (ISO 28500:2009), uri was defined as “<” <'URI' per RFC3986> “>”. This rule has been changed to meet requests from implementers.

The rest of the WARC record grammar concerns defined-field parameters such as record identifier, record type, creation time, content length, and content type.

```

defined-field = WARC-Type
              | WARC-Record-ID
              | WARC-Date
              | Content-Length
              | Content-Type
              | WARC-Concurrent-To
              | WARC-Block-Digest
              | WARC-Payload-Digest
              | WARC-IP-Address
              | WARC-Refers-To
              | WARC-Refers-To-Target-URI
              | WARC-Refers-To-Date
              | WARC-Target-URI
              | WARC-Truncated
              | WARC-Warcinfo-ID
              | WARC-Filename ; warcinfo only
              | WARC-Profile ; revisit only
              | WARC-Identified-Payload-Type
              | WARC-Segment-Origin-ID ; continuation only
              | WARC-Segment-Number
              | WARC-Segment-Total-Length ; continuation only

```

Every WARC record shall have a type, reported in the WARC-Type field. Eight WARC record types are defined in this document as follows:

- ‘warcinfo’;
- ‘response’;
- ‘resource’;
- ‘request’;
- ‘metadata’;
- ‘revisit’;
- ‘conversion’;
- ‘continuation’.

The relevant fields for each record type are described in detail in [Clause 6](#). Each field's meaning and legal value format are described in [Clause 5](#).

The record block shall contain octet content, interpreted based on the record type and other header values. All records shall include a Content-Length field to specify the length of the block.

Some record types (and possibly future record types) also define a payload, such as a meaningful subset of the block or content from a predecessor record. Some headers pertain to the payload of a record rather than the block directly.

For example, in a 'response' record with a content block consisting of HTTP headers and a data object, the payload would be the data object. All 'response', 'resource', 'request', 'revisit', 'conversion' and 'continuation' records may have a payload. All 'warcinfo' and 'metadata' records shall not have a payload.

Content matching the warc-file rule shall have the MIME content-type "application/warc", as specified in [Clause 8](#).

Content matching only the warc-fields rule is useful as a simple descriptive format, and has MIME content-type "application/warc-fields", as specified in [Clause 8](#).

New named fields and new records types may be defined in extensions of the core format. However, it is strongly recommended to discuss any addition to verify that a suitable field or type does not already exist to avoid collision. Discussion should notably be held within the IIPC. See Reference [11] for more information.

5 Named fields

5.1 General

Named fields within a WARC record provide information about the current record. WARC both reuses appropriate headers from other standards and defines new headers, all beginning "WARC-", for WARC-specific purposes.

WARC named fields of the same type shall not be repeated in the same WARC record (for example, a WARC record shall not have several WARC-Date or several WARC-Target-URI), except as noted (e.g. WARC-Concurrent-To).

Because new fields may be defined in extensions to the core WARC format, WARC processing software shall ignore fields with unrecognized names.

5.2 WARC-Record-ID (mandatory)

A WARC-Record-ID is an identifier assigned to the current record that is globally unique for its period of intended use. No identifier scheme is mandated by this specification, but each WARC-Record-ID shall be a legal URI and clearly indicate a documented and registered scheme to which it conforms (e.g. via a URI scheme prefix such as "http:" or "urn:"). Care should be taken to ensure that this value is written with no internal white space.

```
WARC-Record-ID = "WARC-Record-ID" ":" "<" uri ">"
```

All records shall have a WARC-Record-ID field.

5.3 Content-Length (mandatory)

The Content-Length is the number of octets in the block, similar to [RFC2616]. If no block is present, a value of "0" (zero) shall be used.

```
Content-Length = "Content-Length" ":" 1*DIGIT
```

All records shall have a Content-Length field.

5.4 WARC-Date (mandatory)

The WARC-Date is a UTC timestamp as described in ISO 8601^[1], for example YYYY-MM-DDThh:mm:ssZ. The timestamp shall represent the instant that data capture for record creation began. Multiple records written as part of a single capture event (see 5.7) shall use the same WARC-Date, even though the times of their writing will not be exactly synchronized.

WARC-Date may be specified at any of the levels of granularity described in^[1]. If WARC-Date includes a decimal fraction of a second, the decimal fraction of a second shall have a minimum of 1 digit and a maximum of 9 digits. WARC-Date should be specified with as much precision as is accurately known. This document recommends no particular algorithm for access software to choose a record by date when an exact match is not available.

```
WARC-Date = "WARC-Date" ":" iso8601
iso8601 = <a UTC timestamp formatted according to [1]>
```

All records shall have a WARC-Date field.

See [Annex A](#) for examples on usage of WARC-Date fields.

5.5 WARC-Type (mandatory)

WARC-Type is the type of WARC record. Record types defined in this document are

- 'warcinfo',
- 'response',
- 'resource',
- 'request',
- 'metadata',
- 'revisit',
- 'conversion', and
- 'continuation'.

Other types of WARC records may be defined in extensions of the core format. Types are further described in [Clause 6](#).

A WARC file needs not to contain any particular record types, though starting all WARC files with a 'warcinfo' record is recommended.

```
WARC-Type = "WARC-Type" ":" record-type
record-type = "warcinfo" | "response" | "resource"
| "request" | "metadata" | "revisit"
| "conversion" | "continuation"
```

All records shall have a WARC-Type field.

WARC processing software shall ignore records of unrecognized type.

See [Annex A](#) for examples on usage of WARC-Type fields.

5.6 Content-Type

The Content-Type field is the MIME type (as defined in [RFC2045]) of information contained in the record's block. For example, in HTTP request and response records, this would be 'application/http' as specified in 19.1 of [RFC2616] (or 'application/http; msgtype=request' and 'application/http; msgtype = response' respectively). In particular, the content-type is not the value of the HTTP Content-Type header in a HTTP response but a MIME type to describe the full archived HTTP message (hence 'application/http' if the block contains request or response headers).

```

Content-Type = "Content-Type" ":" media-type
media-type  = type "/" subtype *("; parameter)
type        = token
subtype     = token
parameter  = attribute "=" value
attribute   = token
value      = token | quoted-string

```

All records with a non-empty block (non-zero Content-Length), except 'continuation' records, should have a Content-Type field. Only if the media type is not given by a Content-Type field, a reader may attempt to guess the media type via inspection of its content and/or the name extension(s) of the URI used to identify the resource. If the media type remains unknown, the reader should treat it as type "application/octet-stream".

5.7 WARC-Concurrent-To

The WARC-Concurrent-To field (or fields) contains the WARC-Record-ID of any records created as part of the same capture event as the current record. A capture event comprises the information automatically gathered by a retrieval against a single WARC-Target-URI; for example, it may be represented by a 'response' or 'revisit' record plus its associated 'request' record.

```
WARC-Concurrent-To = "WARC-Concurrent-To" ":" "<" uri ">"
```

This field may be used to associate records of types 'request', 'response', 'resource', 'metadata', and 'revisit' with one another when they arise from a single capture event. (When so used, any WARC-Concurrent-To association shall be considered bidirectional even if the header only appears on one record.) The WARC-Concurrent-To field shall not be used in 'warcinfo', 'conversion', and 'continuation' records.

As an exception to the general rule, several WARC-Concurrent-To fields may be repeated within the same WARC record.

See [Annex A](#) for examples on usage of WARC-Concurrent-To fields.

5.8 WARC-Block-Digest

The WARC-Block-Digest is an optional parameter indicating the algorithm name and calculated value of a digest applied to the full block of the record.

```

WARC-Block-Digest = "WARC-Block-Digest" ":" labelled-digest
labelled-digest  = algorithm ":" digest-value
algorithm        = token
digest-value     = token

```

An example is a SHA-1 labelled Base32 ([RFC4648]) value:

```
WARC-Block-Digest: sha1:AB2CD3EF4GH5IJ6KL7MN8OPQ
```

No particular algorithm is recommended.

Any record may have a WARC-Block-Digest field.

5.9 WARC-Payload-Digest

A WARC-Payload-Digest is an optional parameter indicating the algorithm name and calculated value of a digest applied to the payload referred to or contained by the record, which is not necessarily equivalent to the record block.

```
WARC-Payload-Digest = "WARC-Payload-Digest" ":" labelled-digest
```

An example is a SHA-1 labelled Base32 ([RFC4648]) value:

```
WARC-Payload-Digest: sha1:3EF4GH5IJ6KL7MN8OPQAB2CD
```

No particular algorithm is recommended.

The payload of an application/http block is its 'entity-body' (specified in [RFC2616]). In contrast to WARC-Block-Digest, the WARC-Payload-Digest field may also be used for data not actually present in the current record block, for example when a block is left off in accordance with a 'revisit' profile (see [6.7](#)),

or when a record is segmented (the WARC-Payload-Digest recorded in the segments of a segmented record shall be the digest of the payload of the logical record).

The WARC-Payload-Digest field may be used on WARC records with a well-defined payload and shall not be used on records without a well-defined payload.

5.10 WARC-IP-Address

The WARC-IP-Address is the numeric Internet address contacted to retrieve any included content. An IPv4 address shall be written as a "dotted quad"; an IPv6 address shall be written as specified in [RFC4291]. For a HTTP retrieval, this will be the IP address used at retrieval time corresponding to the hostname in the record's target-URI.

```
WARC-IP-Address = "WARC-IP-Address" ":" (ipv4 | ipv6)
ipv4             = <"dotted quad">
ipv6            = <per section 2.2 of RFC4291>
```

The WARC-IP-Address field may be used on 'response', 'resource', 'request', 'metadata', and 'revisit' records, but shall not be used on 'warcinfo', 'conversion' or 'continuation' records.

5.11 WARC-Refers-To

The WARC-Refers-To field contains the WARC-Record-ID of a single record for which the present record holds additional content.

```
WARC-Refers-To = "WARC-Refers-To" ":" "<" uri ">"
```

The WARC-Refers-To field may be used to associate a 'metadata' record to another record it describes. The WARC-Refers-To field may also be used to associate a record of type 'revisit' or 'conversion' with the preceding record which helped determine the present record content. The WARC-Refers-To field shall not be used in 'warcinfo', 'response', 'resource', 'request', and 'continuation' records.

See [Annex A](#) for examples on usage of WARC-Refers-To fields.

5.12 WARC-Refers-To-Target-URI

The WARC-Refers-To-Target-URI field contains the WARC-Target-URI of a record for which the present record is considered a revisit of.

```
WARC-Refers-To-Target-URI = "WARC-Refers-To-Target-URI" ":" uri
```

The WARC-Refers-To-Target-URI field may be used to associate a record of type 'revisit' with another record which contains the resource which has not been archived.

The WARC-Refers-To-Target-URI field may be used in 'revisit' records and shall not be used in 'warcinfo', 'response', 'metadata', 'conversion', 'resource', 'request', and 'continuation' records.

5.13 WARC-Refers-To-Date

The WARC-Refers-To-Date field contains the WARC-Date of a record for which the present record is considered a revisit of.

```
WARC-Refers-To-Date = "WARC-Refers-To-Date" ":" iso8601
iso8601 = <a UTC timestamp formatted according to [1]>
```

The WARC-Refers-To-Date field may be used to associate a record of type 'revisit' with another record which contains the resource which has not been archived.

The WARC-Refers-To-Date field may be used in 'revisit' records and shall not be used in 'warcinfo', 'response', 'metadata', 'conversion', 'resource', 'request', and 'continuation' records.

5.14 WARC-Target-URI

The WARC-Target-URI is the original URI whose capture gave rise to the information content in this record. In the context of web harvesting, this is the URI that was the target of a crawler's retrieval request. For a 'revisit' record, it is the URI that was the target of a retrieval request. Indirectly, such as for a 'metadata', or 'conversion' record, it is a copy of the WARC-Target-URI appearing in the original record to which the newer record pertains. The URI in this value shall be written as specified in [RFC3986].

```
WARC-Target-URI = "WARC-Target-URI" ":" uri
```

All 'response', 'resource', 'request', 'revisit', 'conversion' and 'continuation' records shall have a WARC-Target-URI field. A 'metadata' record may have a WARC-Target-URI field. A 'warcinfo' record shall not have a WARC-Target-URI field.

5.15 WARC-Truncated

For practical reasons, writers of the WARC format may place limits on the time or storage allocated to archiving a single resource. As a result, only a truncated portion of the original resource may be available for saving into a WARC record.

Any record may indicate that truncation of its content block has occurred and give the reason with a WARC-Truncated field.

```
WARC-Truncated = "WARC-Truncated" ":" reason-token
reason-token   = "length"           ; exceeds configured max
                ; length
                | "time"            ; exceeds configured max time
                | "disconnect"      ; network disconnect
                | "unspecified"     ; other/unknown reason
```

Other reasons may be defined in extensions of the core format.

For example, if the capture of what appeared to be a multi-gigabyte resource was cut short after a transfer time limit was reached, the partial resource could be saved to a WARC record with this field.

The WARC-Truncated field may be used on any WARC record. The WARC Content-Length field shall still report the actual truncated size of the record block.

5.16 WARC-Warcinfo-ID

When present, the WARC-Warcinfo-ID indicates the WARC-Record-ID of the associated 'warcinfo' record for this record. Typically, the WARC-Warcinfo-ID parameter is used when the context of the applicable 'warcinfo' record is unavailable, such as after distributing single records into separate WARC files. WARC writing applications (such as web crawlers) may choose to always record this parameter.

```
WARC-Warcinfo-ID = "WARC-Warcinfo-ID" ":" "<" uri ">"
```

The WARC-Warcinfo-ID field value overrides any association with a previously occurring (in the WARC) 'warcinfo' record, thus providing a way to protect the true association when records are combined from different WARC files.

The WARC-Warcinfo-ID field may be used in any record type except 'warcinfo'.

5.17 WARC-Filename

The WARC-Filename is the filename containing the current 'warcinfo' record.

```
WARC-Filename = "WARC-Filename" ":" (TEXT | quoted-string)
```

The WARC-Filename field may be used in 'warcinfo' type records and shall not be used for other record types.

5.18 WARC-Profile

The WARC-Profile is a URI signifying the kind of analysis and handling applied in a 'revisit' record. (Like an XML namespace, the URI may, but needs not, return human-readable or machine-readable documentation.) If reading software does not recognize the given URI as a supported kind of handling, it shall not attempt to interpret the associated record block.

```
WARC-Profile = "WARC-Profile" ":" uri
```

The section 'revisit' defines two initial profile options for the WARC-Profile header for 'revisit' records.

The WARC-Profile field is mandatory on 'revisit' type records and undefined for other record types.

5.19 WARC-Identified-Payload-Type

The WARC-Identified-Payload-Type is the content-type of the record's payload as determined by an independent check. This string shall not be arrived at by blindly promoting a HTTP Content-Type value up from a record block into the WARC header without direct analysis of the payload, as such values may often be unreliable.

```
WARC-Identified-Payload-Type = "WARC-Identified-Payload-Type" ":"  
                               media-type
```

The WARC-Identified-Payload-Type field may be used on WARC records with a well-defined payload and shall not be used on records without a well-defined payload.

5.20 WARC-Segment-Number

The WARC-Segment-Number reports the current record's relative ordering in a sequence of segmented records.

```
WARC-Segment-Number = "WARC-Segment-Number" ":" 1-DIGIT
```

In the first segment of any record that is completed in one or more later 'continuation' WARC records, this parameter is mandatory. Its value there is "1". In a 'continuation' record, this parameter is also mandatory. Its value is the sequence number of the current segment in the logical whole record, increasing by 1 in each next segment.

See [Clause 7](#) on record segmentation for full details on the use of WARC record segmentation.

5.21 WARC-Segment-Origin-ID

The WARC-Segment-Origin-ID identifies the starting record in a series of segmented records whose content blocks are reassembled to obtain a logically complete content block.

```
WARC-Segment-Origin-ID = "WARC-Segment-Origin-ID" ":" "<" uri ">"
```

This field is mandatory on all 'continuation' records, and shall not be used in other records. See [Clause 7](#) on record segmentation for full details on the use of WARC record segmentation.

5.22 WARC-Segment-Total-Length

In the final record of a segmented series, the WARC-Segment-Total-Length reports the total length of all segment content blocks when concatenated together.

```
WARC-Segment-Total-Length = "WARC-Segment-Total-Length" ":"  
                             1*DIGIT
```

This field is mandatory on the last 'continuation' record of a series, and shall not be used elsewhere.

See [Clause 7](#) on record segmentation for full details on the use of WARC record segmentation.

6 WARC record types

6.1 General

The purpose and use of each defined record type is described in [6.2](#) to [6.9](#).

Because new record types that extend the WARC format may be defined in future versions of the WARC standard, WARC processing software shall skip records of unknown type.

6.2 'warcinfo'

A 'warcinfo' record describes the records that follow it, up through end of file, end of input, or until next 'warcinfo' record. Typically, this appears once and at the beginning of a WARC file. For a web archive, it often contains information about the web crawl which generated the following records.

The format of this descriptive record block may vary, though the use of the "application/warc-fields" content-type is recommended. Allowable fields include, but are not limited to, all [DCMI] plus the following field definitions. All fields are optional.

- a) **'operator'**: contact information for the operator who created this WARC resource. A name or name and email address is recommended.
- b) **'software'**: the software and software version used to create this WARC resource. For example, "heritrix/1.12.0".
- c) **'robots'**: the robots policy followed by the harvester creating this WARC resource. The string 'classic' indicates the 1994 web robots exclusion standard rules are being obeyed.
- d) **'hostname'**: the hostname of the machine that created this WARC resource, such as "[crawling17.archive.org](#)".
- e) **'ip'**: the IP address of the machine that created this WARC resource, such as "123.2.3.4".
- f) **'http-header-user-agent'**: the HTTP 'user-agent' header usually sent by the harvester along with each request. Note that if 'request' records are used to save verbatim requests, this information is redundant. (If a 'request' or 'metadata' record reports a different 'user-agent' for a specific request, the more specific information should be considered more reliable.)
- g) **'http-header-from'**: the HTTP 'from' header usually sent by the harvester along with each request. (The same considerations as for 'user-agent' apply.)

It may also contain technical information such as base encoding of the digests used in named fields.

So that multiple record excerpts from inside WARC files are also valid WARC files, it is optional that the first record of a legal WARC be a 'warcinfo' description. Also, to allow the concatenation of WARC files into a larger valid WARC file, it is allowable for 'warcinfo' records to appear anywhere in a WARC file.

See [B.1](#) for an example of a 'warcinfo' record.

6.3 'response'

6.3.1 General

A 'response' record should contain a complete scheme-specific response, including network protocol information, where possible. The exact contents of a 'response' record are determined not just by the record type but also by the URI scheme of the record's target-URI, as described in [6.3.2](#) to [6.3.3](#).

See [B.2](#) for an example of a 'response' record.

6.3.2 'http' and 'https' schemes

For a target-URI of the 'http' or 'https' schemes, a 'response' record block should contain the full HTTP response received over the network, including headers, i.e. it contains the 'response' message defined by section 6 of HTTP/1.1 ([RFC2616]), or by any previous or subsequent version of HTTP compatible with section 6 of HTTP/1.1 ([RFC2616]).

The WARC record's Content-Type field should contain the value defined by HTTP/1.1, "application/http;msgtype=response". When software bugs, network issues, or implementation limits cause response-like material to be collected that is not perfectly compliant with HTTP specifications, WARC writing software may record the problematic content using its best effort determination of the interesting material boundaries. That is, neither the use of the 'response' record with a 'http' target-URI nor the 'application/http' content-type serves as an absolute guarantee that the contained material is a legal HTTP response.

A WARC-IP-Address field should be used to record the network IP address from which the response material was received.

When a 'response' is known to have been truncated, this shall be noted using the WARC-Truncated field.

A WARC-Concurrent-To field (or fields) may be used to associate the 'response' to a matching 'request' record or concurrently-created 'metadata' record.

The payload of a 'response' record with a target-URI of scheme 'http' or 'https' is defined as its 'entity-body' (as specified in ([RFC2616])). If a truncated 'response' record block contains less than the full entity-body, the payload is considered truncated at the same position.

Conventions for recording information about the 'https' secure socket transaction, such as certificates exchanged, consulted, or verified, are not in the scope of this document.

6.3.3 Other URI schemes

The contents of the 'response' record for other URI schemes are not specified in this document.

6.4 'resource'

6.4.1 General

A 'resource' record contains a resource, without full protocol response information. For example, a file directly retrieved from a locally accessible repository or the result of a networked retrieval where the protocol information has been discarded. The exact contents of a 'resource' record are determined not just by the record type but also by the URI scheme of the record's target-URI, as described in 6.4.2 to 6.4.5.

For all 'resource' records, the payload is defined as the record block.

A 'resource' record, with a synthesized target-URI, may also be used to archive other artefacts of a harvesting process inside WARC files.

See [B.3](#) for an example of a 'resource' record.

6.4.2 'http' and 'https' schemes

For a target-URI of the 'http' or 'https' schemes, a 'resource' record block shall contain the returned 'entity-body' (as specified in [RFC2616]), possibly truncated.

6.4.3 'ftp' scheme

For a target-URI of the 'ftp' scheme, a 'resource' record block shall contain the complete file returned by an FTP operation, possibly truncated.

6.4.4 'dns' scheme

For a target-URI of the 'dns' scheme ([RFC4501]), a 'resource' record shall contain material of content-type 'text/dns' (registered by [RFC4027] and defined by [RFC2540] and [RFC1035]) representing the results of a single DNS lookup as described by the target-URI.

6.4.5 Other URI schemes

The contents of the 'resource' record for other URI schemes are not specified in this document.

6.5 'request'

6.5.1 General

A 'request' record holds the details of a complete scheme-specific request, including network protocol information, where possible. The exact contents of a 'request' record are determined not just by the record type but also by the URI scheme of the record's target-URI, as described below (from 6.5.2 to 6.5.3).

See [B.4](#) for an example of a 'request' record.

6.5.2 'http' and 'https' schemes

For a target-URI of the 'http' or 'https' schemes, a 'request' record block should contain the full HTTP request sent over the network, including headers. That is, it contains the 'request' message defined by section 5 of HTTP/1.1 [RFC2616], or by any previous or subsequent version of HTTP compatible with section 5 of HTTP/1.1 [RFC2616].

The WARC record's Content-Type field should contain the value defined by HTTP/1.1, "application/http;msgtype=request".

A WARC-IP-Address field should be used to record the network IP address to which the request material was directed.

A WARC-Concurrent-To field (or fields) may be used to associate the 'request' to a matching 'response' record or concurrently-created 'metadata' record.

The payload of a 'request' record with a target-URI of scheme 'http' or 'https' is defined as its 'entity-body' (as specified in [RFC2616]). If a truncated 'request' record block contains less than the full entity-body, the payload is considered truncated at the same position.

Conventions for recording information about the 'https' secure socket transaction, such as certificates exchanged, consulted, or verified, are not in the scope of this document.

6.5.3 Other URI schemes

This document does not specify the contents of the 'request' record for other URI schemes.

6.6 'metadata'

A 'metadata' record contains content created in order to further describe, explain, or accompany a harvested resource, in ways not covered by other record types. A 'metadata' record will almost always refer to another record of another type, with that other record holding original harvested or transformed content. (However, it is allowable for a 'metadata' record to refer to any record type, including other 'metadata' records.) Any number of metadata records may reference another specific record.

The format of the metadata record block may vary. The "application/warc-fields" format may be used. Allowable fields include all [DCMI] plus the following field definitions. All fields are optional.

— 'via': the referring URI from which the archived URI was discovered.

- 'hopsFromSeed': a symbolic string describing the type of each hop from a starting 'seed' URI to the current URI.
- 'fetchTimeMs': time in milliseconds that it took to collect the archived URI, starting from the initiation of network traffic.

A 'metadata' record may be associated with other records derived from the same capture event using the WARC-Concurrent-To header. A 'metadata' record may be associated to another record which it describes, using the WARC-Refers-To header.

See [B.5](#) for an example of a 'metadata' record.

6.7 'revisit'

6.7.1 General

A 'revisit' record describes the revisitation of content already archived, and might include only an abbreviated content body which has to be interpreted relative to a previous record. Most typically, a 'revisit' record is used instead of a 'response' or 'resource' record to indicate that the content visited was either a complete or substantial duplicate of material previously archived.

Using a 'revisit' record instead of another type is optional, for when benefits of reduced storage size or improved cross-referencing of material are desired.

A 'revisit' record shall contain a WARC-Profile field which determines the interpretation of the record's fields and record block. Two initial values and their interpretation are described in the following subclauses. A reader which does not recognize the profile URI shall not attempt to interpret the enclosing record or associated content body.

The purpose of this record type is to reduce storage redundancy when repeatedly retrieving identical or little-changed content, while still recording that a revisit occurred, plus details about the current state of the visited content relative to the archived version.

See [B.6](#) for an example of a 'revisit' record.

6.7.2 Profile: Identical Payload Digest

This 'revisit' profile may be used whenever a subsequent consideration of a URI provides payload content which a strong digest function, such as SHA-1, indicates is identical to a previously archived version. It is not necessary that the URI of the original capture and the revisit be identical.

To indicate this profile use the URI:

<http://netpreserve.org/warc/1.1/revisit/identical-payload-digest>

To report the payload digest used for comparison, a 'revisit' record using this profile shall include a WARC-Payload-Digest field, with a value of the digest that was calculated on the payload.

A 'revisit' record using this profile may have no record block, in which case a Content-Length of zero shall be written. If a record block is present, it shall be interpreted the same as a 'response' record type for the same URI, but truncated to avoid storing the duplicate content. A WARC-Truncated header with reason 'length' shall be used for any identical-digest truncation. It is recommended that server response headers be preserved in the revisit record in this manner.

For records using this profile, the payload is defined as the original payload content whose digest value was unchanged.

Using a WARC-Refers-To header to identify a specific prior record from which the matching content can be retrieved is recommended, to minimize the risk of misinterpreting the 'revisit' record. The following

two optional fields can also be used to associate the revisit record with the original record. Their use is recommended.

- WARC-Refers-To-Target-URI. Its value should be equal to the WARC-Target-URI in the WARC record that the current record is considered a duplicate of.
- WARC-Refers-To-Date. Its value should be equal to the WARC-Date in the WARC record that the current record is considered a duplicate of.

6.7.3 Profile: Server Not Modified

This 'revisit' profile may be used whenever a subsequent consideration of a URI encounters an assertion from the providing server that the content has not changed, such as a HTTP "304 Not Modified" response.

To indicate this profile, use the URI:

<http://netpreserve.org/warc/1.1/revisit/server-not-modified>

A 'revisit' record using this profile may have no content body, in which case a Content-Length of zero shall be written. If a content body is present, it should be interpreted the same as a 'response' record type for the same URI, truncated if desired.

For records using this profile, the payload is defined as the original payload content from which a 'Last-Modified' and/or 'ETag' value was taken.

Using a WARC-Refers-To header to identify a specific prior record from which the unmodified content can be retrieved is recommended, to minimize the risk of misinterpreting the 'revisit' record. WARC-Refers-To-Date may be optionally used to help resolve the original capture. Its value should be equal to the WARC-Date in the WARC record that the current record is considered a duplicate of. Its use is recommended.

6.7.4 Other profiles

Other documents may define additional profiles to accomplish other goals, such as recording the apparent magnitude of difference from the previous visit, or to encode the visited content as a "diff" (where "diff" is the file comparison utility that outputs the differences between two files) of the content previously stored.

6.8 'conversion'

A 'conversion' record shall contain an alternative version of another record's content that was created as the result of an archival process. Typically, this is used to hold content transformations that maintain viability of content after widely available rendering tools for the originally stored format disappear. As needed, the original content may be migrated (transformed) to a more viable format in order to keep the information usable with current tools while minimizing loss of information (intellectual content, look and feel, etc.). Any number of 'conversion' records may be created that reference a specific source record, which may itself contain transformed content. Each transformation should result in a freestanding, complete record, with no dependency on survival of the original record.

Wherever practical, a 'conversion' record should contain a WARC-Refers-To field so as to identify the prior material converted.

Metadata records may be used to further describe transformation records.

For 'conversion' records, the payload is defined as the record block.

See [B.7](#) for an example of a 'conversion' record.

6.9 'continuation'

Record blocks from 'continuation' records shall be appended to corresponding prior record block(s) (e.g. from other WARC files) to create the logically complete full-sized original record. That is, 'continuation' records are used when a record that would otherwise cause a WARC file size to exceed a desired limit is broken into segments. A continuation record shall contain the named fields WARC-Segment-Origin-ID and WARC-Segment-Number, and the last 'continuation' record of a series shall contain a WARC-Segment-Total-Length field. The full details of WARC record segmentation are described in [Clause 7](#).

See [B.8](#) for an example of a 'continuation' record.

7 Record segmentation

A record that will not fit into a single WARC file of desired maximum size may be broken into a number of separate records, called segments.

The first segment of a segmented series shall carry the original record type (not 'continuation'), and a WARC-Segment-Number field with a value of "1".

All subsequent segments shall have a record type of 'continuation', with an incremented WARC-Segment-Number field. They shall also include a WARC-Segment-Origin-ID field with a value of the WARC-Record-ID of the record containing the first segment of the set. All segments of a set shall have identical target-URI values. Segments may have individual WARC-Block-Digest fields.

The last segment shall contain a WARC-Segment-Total-Length field specifying the total length, in bytes, of all segment content blocks, if reassembled. The last segment may also contain a WARC-Truncated field, if appropriate.

The WARC-Payload-Digest recorded in the first segment of a segmented record is the digest of the payload of the logical record.

Segments other than the first should not contain other optional fields, as segments merely serve to continue the record data block of the first record.

To reassemble all segments into the intended complete logical record, the content blocks of all records with the same 'WARC-Segment-Origin-ID' value are collected and appended, in 'WARC-Segment-Number' order, to the origin record's content block. The resulting assembled record adopts as its Content-Length the 'WARC-Segment-Total-Length' value. It also adopts any 'WARC-Truncated' reason of the final segment.

Segmentation shall not be used if there is another way to store the record within the desired WARC file target size. Specifically, if a record could be stored without segmentation by starting a new WARC file, segmentation shall not be used. Further, when segmentation is used, the size of the first segment shall be maximized. Specifically, the origin segment shall be placed in a new WARC file, preceded only by a 'warcinfo' record (if any).

Segmentation may be applied to any original record type other than 'continuation', but its use on 'warcinfo', 'request', 'metadata' and 'revisit' records is not recommended.

8 WARC file name, size and compression

The MIME type of a WARC file shall be application/warc.

The MIME type of warcinfo records, WARC metadata records, and potentially other records types in the future, shall be application/warc-fields.

A WARC file shall have a name. See [Annex C](#) for recommendations on WARC files names.

It is recommended to limit the size of a WARC file. See [Annex C](#) for a recommended size limit according to experience.

WARC files may be compressed. See [Annex D](#) for recommendations on WARC file compression.

STANDARDSISO.COM : Click to view the full PDF of ISO 28500:2017

Annex A (informative)

Use cases for writing WARC records

Different use cases are listed in [Tables A.1 to A.5](#), which develop some situations where WARC files and WARC records may be generated. These use cases correspond to the needs of the web archiving community.

NOTE In a web harvesting context, the files constituting the websites are stored as WARC records in WARC files. Depending on the web harvesting process configuration, the different pieces of a website might not be contained in a single WARC file or in a set of WARC files but might be spread out and stored along pieces of other harvested websites. Thus, to render the archive of a website to users, access software might have to extract files contained in WARC records from different WARC files. External indexes might be used for a quicker access.

Other use cases might be conceived by other users to satisfy their own needs. Moreover, solutions adopted for each use case are not the only solutions that may be used. These are presented as examples.

The first column in each table describes the use case and its different steps. The second column in each table indicates what type of record is generated. Only the most complex named fields are specified in order to clarify the use of these fields: WARC-Type (mandatory field), WARC-Date (mandatory field), WARC-Concurrent-To (optional field), WARC-Refers-To (optional field). The other mandatory or useful named fields are not presented in this document.

It has been assumed that these WARC records are written in an already opened WARC file, containing a 'warcinfo' record.

Table A.1 – Use case one:
An archiving crawler fetches <http://netpreserve.org/reports/iipc2007conference.pdf> from the World Wide Web and writes it in a WARC file

Description of use case	Record generated
Date: 2007-10-24 at 10:14:22 GMT	
A request is sent by the crawler to the server hosting http://netpreserve.org/reports/iipc2007conference.pdf	WARC record created: WARC-Type: 'request' WARC-Date: 2007-10-24T10:14:22Z WARC-Concurrent-To: WARC-Record-ID of the following 'response' record
A response is received by the crawler from the server	WARC record created: WARC-Type: 'response' WARC-Date: 2007-10-24T10:14:22Z
Metadata further describing the harvesting process/ the harvested record are added (e.g. information coming from the log files)	WARC record created: WARC-Type: 'metadata' WARC-Date: 2007-10-24T10:14:22Z WARC-Concurrent-To: WARC-Record-ID of the previous 'response' record
If the file harvested on the web is too big to be contained in a single WARC file (e.g. 1,5 GB), the WARC record is segmented and a second record is created	Second WARC record created: WARC-Type: 'continuation' WARC-Date: 2007-10-24T10:14:22Z

Table A.2 — Use case two: The XML version of the French Gazette of 2007-11-01 has been transferred to the National Library of France (via FTP or email) and is written in a WARC file

Description of use case Date: 2007-11-02 at 15:20:44 GMT	Record generated
The resource is archived	WARC record created: WARC-Type: 'resource' WARC-Date: 2007-11-02T15:20:44Z
Metadata further describing the archiving process/the archived record are added (e.g. information about the transfer)	WARC record created: WARC-Type: 'metadata' WARC-Date: 2007-11-02T15:20:44Z WARC-Concurrent-To: WARC-Record-ID of the previous 'resource' record

Table A.3 — Use case three: An archiving crawler fetches <http://netpreserve.org/reports/iipc2007conference.pdf> from the World Wide Web that has not changed since the latest harvest

Description of use case Date: 2007-11-24 at 18:28:24 GMT	Record generated
A request is sent by the crawler to the server hosting http://netpreserve.org/reports/iipc2007conference.pdf	WARC record created: WARC-Type: 'request' WARC-Date: 2007-11-24T18:28:24Z WARC-Concurrent-To: WARC-Record-ID of the following 'revisit' record
The crawler detects that the file is the same as previously archived and that it has not changed. The entire file is not recorded to avoid duplicates and reduce storage redundancy	WARC record created: WARC-Type: 'revisit' WARC-Date: 2007-11-24T18:28:24Z WARC-Refers-To: WARC-Record-ID of the already written record WARC-Refers-To-Target-URI: target-URI of the already written record WARC-Refers-To-Date: date of the already written record

Table A.4 — Use case four: After the end of the harvest, Jhove is used to validate the format of <http://netpreserve.org/reports/iipc2007conference.pdf> and the validation results are stored in a WARC file and linked to the corresponding record

Description of use case Date: 2007-11-01 at 20:54:02 GMT	Record generated
Results of the validation process are added in another WARC file	WARC record created: WARC-Type: 'metadata' Date: 2007-11-01T20:54:02Z WARC-Refers-To: WARC-Record-ID of the described WARC record

Table A.5 — Use case five: <http://netpreserve.org/reports/iipc2007conference.pdf> file format has become obsolete as it cannot be read anymore by the existing rendering tools

Description of use case Date: 2020-01-23 at 16:14:32 GMT	Record generated
A file in the new format is generated	WARC record created: WARC-Type: 'conversion' WARC-Date: 2020-01-23T16:14:32Z WARC-Refers-To: WARC-Record-ID of the WARC record whose payload has been migrated
Metadata describing the migration process are added (e.g. tool used)	WARC record created: WARC-Type: 'metadata' WARC-Date: 2020-01-23T16:14:32Z WARC-Refers-To: WARC-Record-ID of the previous conversion record

Table A.6 — Different use cases for record timestamps, featuring different levels of precision

Description of use case	Record generated
A web page has been captured by a WARC 1.1 compliant crawler	WARC record created: WARC-Date: 2016-01-11T23:24:25.412030Z
A web page has been captured by a legacy crawler	WARC record created: WARC-Date: 2016-01-11T23:24:25Z
A video has been donated by a website owner and converted to the WARC format	WARC record created: WARC-Date: 2016-01

Annex B (informative)

Examples of WARC records

B.1 Example of 'warcinfo' record

```
WARC/1.1
WARC-Type: warcinfo
WARC-Date: 2016-09-19T17:20:14Z
WARC-Record-ID: <urn:uuid:d7ae5c10-e6b3-4d27-967d-34780c58ba39>
Content-Type: application/warc-fields
Content-Length: 381
software: Heritrix 1.12.0 http://crawler.archive.org
hostname: crawling017.archive.org
ip: 207.241.227.234
isPartOf: testcrawl-20050708
description: testcrawl with WARC output
operator: IA_Admin
http-header-user-agent:
  Mozilla/5.0 (compatible; heritrix/1.4.0 +http://crawler.archive.org)
format: WARC file version 1.1
conformsTo:
  http://bibnum.bnf.fr/warc/WARC_ISO_28500_version0-1_latestdraft.pdf
```

B.2 Example of 'response' record

```
WARC/1.1
WARC-Type: response
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Warcinfo-ID: <urn:uuid:d7ae5c10-e6b3-4d27-967d-34780c58ba39>
WARC-Date: 2016-09-19T17:20:24Z
WARC-Block-Digest: sha1:UZY6ND6CCHXETFVJD2MSS7ZENMWF7KQ2
WARC-Payload-Digest: sha1:CCHXETFVJD2MUZY6ND6SS7ZENMWF7KQ2
WARC-IP-Address: 207.241.213.58
WARC-Record-ID: <urn:uuid:92283950-ef2f-4d72-b224-f54c6ec90bb0>
Content-Type: application/http;msgtype=response
WARC-Identified-Payload-Type: image/jpeg
Content-Length: 1902
HTTP/1.1 200 OK
Date: Tue, 19 Sep 2016 17:18:40 GMT
Server: Apache/2.0.54 (Ubuntu)
Last-Modified: Mon, 16 Jun 2013 22:28:51 GMT
ETag: "3e45-67e-2ed02ec0"
Accept-Ranges: bytes
Content-Length: 1662
Connection: close
Content-Type: image/jpeg
  [image/jpeg binary data here]
```

B.3 Example of 'resource' record

```
WARC/1.1
WARC-Type: resource
WARC-Target-URI: file:///var/www/htdocs/images/logoc.jpg
WARC-Date: 2016-09-30T16:40:32Z
WARC-Record-ID: <urn:uuid:23200706-de3e-3c61-a131-g65d7fd80cc1>
Content-Type: image/jpeg
WARC-Payload-Digest: sha1:DBXHDRBXL40MUZ5DN4JJ2KFUAOB6VK8
WARC-Block-Digest: sha1:DBXHDRBXL40MUZ5DN4JJ2KFUAOB6VK8
Content-Length: 1662
  [image/jpeg binary data here]
```