ISO TC 46/SC 4 N

Date: 2008-XX-XX

ISO 28500

ISO TC 46/SC 4/WG 12

Secretariat: Standards New Zealand

Information and documentation — The WARC File Format

Élément introductif — Élément central — Élément complémentaire

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: International Standard
Document subtype:
Document stage: DIS
Document language: E

Contents

Page

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

ISO/DIS 28500 was prepared by Technical Committee ISO/TC 46, *Information and documentation*, Subcommittee SC 4, *Technical interoperability*.

Introduction

Web sites and web pages emerge and disappear from the world wide web every day. For the past ten years, memory 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) must 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 is 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 is expected to be a standard way to structure, manage and store billions of resources collected from the web and elsewhere. It will be used to build applications for harvesting (such as the open source Heritrix web crawler), managing, accessing, and exchanging content. The way WARC files will be created and resources will be 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 — The WARC File Format

1 Scope

This international standard specifies the WARC file format:

- to store both the payload content and control information from mainstream Internet application layer protocols, such as 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 referenced documents are indispensable for the application 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.

[ARC] Burner, Mike; Kahle, Brewster. - ARC File Format, 15 September 1996; http://www.archive.org/web/researcher/ArcFileFormat.php.

[W3CDTF] Date and Time Formats: note submitted to the W3C 15 September 1997 (W3C profile of ISO8601). http://www.w3.org/TR/NOTE-datetime

[DCMI] DCMI Metadata Terms. http://dublincore.org/documents/dcmi-terms/

[RFC1035] Mockapetris, P., “Domain names - implementation and specification,” STD 13, RFC 1035, November 1987. http://www.faqs.org/rfcs/rfc1035.html

[RFC1884] Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” RFC 1884, December 1995. http://www.faqs.org/rfcs/rfc1884.html

[RFC1950] Deutsch, L. and J-L. Gailly, “ZLIB Compressed Data Format Specification version 3.3,” RFC 1950, May 1996 (TXT, PS, PDF). http://www.faqs.org/rfcs/rfc1950.html

[RFC1951] Deutsch, P., “DEFLATE Compressed Data Format Specification version 1.3,” RFC 1951, May 1996 (TXT, PS, PDF). http://www.faqs.org/rfcs/rfc1951.html

[RFC1952] Deutsch, P... “GZIP file format specification version 4.3,” RFC 1952, May 1996 (TXT, PS, PDF). http://www.faqs.org/rfcs/rfc1952.html

[RFC2045] Freed, N. ; Borenstein, N. “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies,” RFC 2045, November 1996. http://www.faqs.org/rfcs/rfc2045

[RFC2047] Moore, K.. “MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text,” RFC 2047, November 1996 (TXT, HTML, XML). http://www.faqs.org/rfcs/rfc2047

[RFC2048] Freed, N.; Klensin, J.,; Postel, J. “Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures,” BCP 13, RFC 2048, November 1996 (TXT, HTML, XML). http://www.faqs.org/rfcs/rfc2048

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML). http://www.faqs.org/rfcs/rfc2119.html

[RFC2540] Eastlake, D., “Detached Domain Name System (DNS) Information,” RFC 2540, March 1999. http://www.faqs.org/rfcs/rfc2540.html

[RFC2616] Fielding, R.; Gettys, J.; Mogul, J.; Frystyk, H.; Masinter, L.; Leach, P.; Berners-Lee, T. “Hypertext Transfer Protocol -- HTTP/1.1,” RFC 2616, June 1999 (TXT, PS, PDF, HTML, XML). http://www.faqs.org/rfcs/rfc2540.html

[RFC2822] Resnick, P., “Internet Message Format,” RFC 2822, April 2001. http://www.faqs.org/rfcs/rfc2822

[RFC3548] Josefsson, S., “The Base16, Base32, and Base64 Data Encodings,” RFC 3548, July 2003. http://www.faqs.org/rfcs/rfc3548.html

[RFC3629] Yergeau, F., “UTF-8, a transformation format of ISO 10646”, STD 63, RFC 3629, November 2003. http://www.faqs.org/rfcs/rfc3629.html

[RFC3986] Berners-Lee, T.; Fielding, R.; Masinter, L. “Uniform Resource Identifier (URI): Generic Syntax,” STD 66, RFC 3986, January 2005 (TXT, HTML, XML). http://www.faqs.org/rfcs/rfc3986.html

[RFC4027] Josefsson, S., “Domain Name System Media Types,” RFC 4027, April 2005. http://www.faqs.org/rfcs/rfc4027.html

[RFC4501] Josefsson, S., “Domain Name System Uniform Resource Identifiers,” RFC 4501, May 2006. http://www.rfc-archive.org/getrfc.php?rfc=4501

3 Terms, definitions and acronyms

3.1 Terms and definitions

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

In the context of segmentation, a logical record may be composed of multiple segments, each represented by a WARC record.

3.2 Acronyms

ABNF	Augmented Backus-Naur Form
ARC	ARChive
CRLF	Carriage Return Line Feed
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, standalone 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 newlines. 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. With one major exception, allowing UTF-8 [RFC3629], the WARC record header format largely follows the tradition of HTTP/1.1 [RFC2616] and [RFC2822] headers.

The top-level view of a WARC file can be expressed in an augmented Backus-Naur Form (BNF) 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/1.0" CRLF
        warc-fields  = *named-field CRLF
        block        = *OCTET
```

The record version shall appear first in every record and hence also shall 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 case-insensitive. The field value may be preceded by any amount of linear whitespace (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; RFC3629>  ; (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   = ( <"> *(qdtext | quoted-pair ) <"> )
qdtext          = <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.

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-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. There are eight WARC record types: 'warcinfo', 'response', 'resource', 'request', 'metadata', 'revisit', 'conversion', and 'continuation'. The relevant fields for each record type are further described in WARC Record Types. Each field's meaning and legal value format are described in named fields.

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’, ‘conversion’ and ‘continuation’ records may have a payload. All ‘warcinfo’, ‘metadata’ and ‘revisit’ records shall not have a payload.

Content matching the warc-file rule shall have the MIME content-type "application/warc", registered below in section 8.2.

Content matching only the warc-fields rule is useful as a simple descriptive format, and has MIME content-type "application/warc-fields", registered below in section 8.3.

5 Named fields

5.1 General

Named fields within a WARC record provide information about the current record, and allow additional per-record information. 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)

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 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 whitespace.

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

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

5.3 Content-Length (mandatory)

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)

A 14-digit UTC timestamp formatted according to YYYY-MM-DDThh:mm:ssZ, described in the W3C profile of ISO8601 [W3CDTF]. The timestamp shall represent the instant that data capture for record creation began. Multiple records written as part of a single capture event (see section 5.7) shall use the same WARC-Date, even though the times of their writing will not be exactly synchronized.

```
WARC-Date   = "WARC-Date" ":" w3c-iso8601
w3c-iso8601 = <YYYY-MM-DDThh:mm:ssZ>
```

All records shall have a WARC-Date field.

5.5 WARC-Type (mandatory)

The type of WARC record: one of 'warcinfo', 'response', 'resource', 'request', 'metadata', 'revisit', 'conversion', or 'continuation'. Other types of WARC records may be defined in extensions of the core format. Types are further described in WARC Record Types.

A WARC file needs not 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" |  future-type
future-type = token
```

All records shall have a WARC-Type field.

WARC processing software shall ignore records of unrecognized type.

5.6 Content-Type

The MIME type [RFC2045] of the information contained in the record's block. For example, in HTTP request and response records, this would be 'application/http' as per section 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 an 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-Record-IDs 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 target-URI; for example, it might 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, it is allowed to repeat several WARC-Concurrent-To fields within the same WARC record.

5.8 WARC-Block-Digest

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 ([RFC3548]) value:

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

This document recommends no particular algorithm.

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

5.9 WARC-Payload-Digest

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 ([RFC3548]) value:

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

This document recommends no particular algorithm.

The payload of an application/http block is its 'entity-body' (per [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 'revisit'), or when a record is segmented (the WARC-Payload-Digest recorded in the first segment 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 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 per [RFC1884]. For an 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 RFC1884>
```

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-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.

5.12 WARC-Target-URI

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 properly escaped according to [RFC3986] and written with no internal whitespace.

```
        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.

5.13 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
                          | future-reason
        future-reason     = token
```

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 field Content-Length shall still report the actual truncated size of the record block.

5.14 WARC-Warcinfo-ID

When present, indicates the WARC-Record-ID of the associated 'warcinfo' record for this record. Typically, the 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 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 WARCs.

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

5.15 WARC-Filename

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.

5.16 WARC-Profile

A URI signifying the kind of analysis and handling applied in a 'revisit' record. (Like an XML namespace, the URI may, but need 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.

5.17 WARC-Identified-Payload-Type

The content-type of the record's payload as determined by an independent check. This string shall not be arrived at by blindly promoting an 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.

5.18 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 the section below, Record Segmentation, for full details on the use of WARC record segmentation.

5.19 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 the section below, Record segmentation, for full details on the use of WARC record segmentation.

5.20 WARC-Segment-Total-Length

In the final record of a segmented series, 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 the section below, 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 below.

Because new record types that extend the WARC format may be defined in future standards, 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.

'operator'

Contact information for the operator who created this WARC resource. A name or name and email address is recommended.

'software'

The software and software version used to create this WARC resource. For example, "heritrix/1.12.0".

'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.

'hostname'

The hostname of the machine that created this WARC resource, such as "crawling17.archive.org".

'ip'

The IP address of the machine that created this WARC resource, such as "123.2.3.4".

'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.)

'http-header-from'

The HTTP 'From' header usually sent by the harvester along with each request. (The same considerations as for 'user-agent' apply.)

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 in the middle of a WARC file.

See annex C.1 below 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 below.

See annex C.2 below for an example of a ‘response’ record.

6.3.2 for '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. That is, 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 the 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 an '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' (per [RFC2616]), with any transfer-encoding removed. If a truncated 'response' record block contains less than the full entity-body, the payload is considered truncated at the same position.

This document does not specify conventions for recording information about the 'https' secure socket transaction, such as certificates exchanged, consulted, or verified.

6.3.3 for other URI schemes

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

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 below.

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 annex C.3 below for an example of a ‘resource’ record.

6.4.2 for 'http' and 'https' schemes

For a target-URI of the 'http' or 'https' schemes, a 'resource' record block shall contain the returned 'entity-body' (per [RFC2616], with any transfer-encodings removed), possibly truncated.

6.4.3 for '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 for '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 for other URI schemes

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

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.

See annex C.4 below for an example of a ‘request’ record.

6.5.2 for '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 the 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' (per [RFC2616]), with any transfer-encoding removed. If a truncated 'request' record block contains less than the full entity-body, the payload is considered truncated at the same position.

This document does not specify conventions for recording information about the 'https' secure socket transaction, such as certificates exchanged, consulted, or verified.

6.5.3 for 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 one specific other record.

The format of the metadata record block may vary. The "application/warc-fields" format, defined earlier, 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 annex C.5 below 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 sections. 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 annex C.6 below 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 recorded version.

To indicate this profile, use the URI:

```
http://netpreserve.org/warc/1.0/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 must 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.

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.

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 an HTTP "304 Not Modified" response.

To indicate this profile, use the URI:

```
http://netpreserve.org/warc/1.0/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.

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.

Metadata records may be used to further describe transformation records. Wherever practical, a 'conversion' record should contain a 'WARC-Refers-To' field to identify the prior material converted.

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

See annex C.7 below for an example of a ‘conversion’ record.

6.9 'continuation'

Record blocks from 'continuation' records must 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 the below section Record Segmentation. See also annex C.8 below 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 Registration of MIME media types application/warc and application/warc-fields

8.1 General

This section describes, as per [RFC2048], the MIME types associated with the WARC format.

8.2 application/warc

MIME media type name: application

MIME subtype names: warc

Required parameters: None

Optional parameters: None

Encoding considerations:

Content of this type is in 'binary' format.

Security considerations:

The WARC record syntax poses no direct risk to computers and networks. Implementers need to be aware of source authority and trustworthiness of information structured in WARC. Readers and writers subject themselves to all the risks that accompany normal operation of data processing services (e.g., message length errors, buffer overflow attacks).

Interoperability considerations: None

Published specification: TBD

Applications which use this media type: Large- and small-scale archiving

Additional information: None

Person and email address to contact for further information:

Gordon Mohr gojomo@archive.org, John Kunze jak@ucop.edu

Intended usage: COMMON

Author/Change controller: IESG

8.3 application/warc-fields

MIME media type name: application

MIME subtype names: warc-fields

Required parameters: None

Optional parameters: None

Encoding considerations:

Content of this type is in 'binary' format.

Security considerations:

The WARC field syntax poses no direct risk to computers and networks. Implementers need to be aware of source authority and trustworthiness of information structured in WARC. Readers and writers subject themselves to all the risks that accompany normal operation of data processing services (e.g., message length errors, buffer overflow attacks).

Interoperability considerations: None

Published specification: TBD

Applications which use this media type: Large- and small-scale archiving

Additional information: None

Person and email address to contact for further information:

Gordon Mohr gojomo@archive.org, John Kunze jak@ucop.edu

Intended usage: COMMON

Author/Change controller: IESG

9 IANA considerations

After IESG approval, IANA is expected to register the WARC type "application/warc" using the application provided in this document.

Annex A
(informative)

Compression recommendations

A.1 General

The WARC format defines no internal compression. Whether and how WARC files should be compressed is an external decision.

However, experience with the precursor ARC format at the Internet Archive has demonstrated that applying simple standard compression can result in significant storage savings, while preserving random access to individual records.

For this purpose, the GZIP format with customary "deflate" compression is recommended, as defined in [RFC1950], [RFC1951], and [RFC1952]. Freely available source code implementing this format is available, and the technique is free of patent encumbrances. The GZIP format is also widely used and supported across many free and commercial software packages and operating systems.

This section documents recommended, but optional, practices for compressing WARC files with GZIP.

A.2 Record-at-time compression

Per section 2.2 of the GZIP specification, a valid GZIP file consists of any number of gzip "members", each independently compressed.

Where possible, this property should be exploited to compress each record of a WARC file independently. This results in a valid GZIP file whose per-record subranges also stand alone as valid GZIP files.

External indexes of WARC file content may then be used to record each record's starting position in the GZIP file, allowing for random access of individual records without requiring decompression of all preceding records.

Note that the application of this convention causes no change to the uncompressed contents of an individual WARC record.

A.3 GZIP WARC file name suffix

A gzip compressed WARC file should have the customary ".gz" appended to it, making the complete suffix, ".warc.gz".

Annex B
(informative)

WARC file size and name recommendations

1GB (10^9 bytes) is recommended as a practical target size for WARC files, when record sizes allow. Oversized records may be truncated, segmented, or placed in oversized WARC files, at a project's discretion.

It is helpful to use practices within an institution that make it unlikely or impossible to duplicate aggregate WARC file names. The convention used inside the Internet Archive with ARC files is to name files according to the following pattern:

Prefix-Timestamp-Serial-Crawlhost.warc.gz

Prefix is an abbreviation usually reflective of the project or crawl that created this file. Timestamp is a 14-digit GMT timestamp indicating the time the file was initially begun. Serial is an increasing serial-number within the process creating the files, often (but not necessarily) unique with regard to the Prefix. Crawlhost is the domain name or IP address of the machine creating the file.

IIPC member institutions have expressed an interest in adopting a common naming strategy, with per-institution unique identifiers to assist in marking WARC files with their institution of origin. It is proposed that all such WARC file names adhering to this future convention begin "iipc".

This specification does not require any particular WARC file naming practice, but conventions similar to the above are recommended within WARC-creating institutions. The file name prefix "iipc" should not be used unless participating in a future IIPC naming registry.

Annex C
(informative)

Examples of WARC records

C.1 Example of 'warcinfo' record

```
WARC/1.0
WARC-Type: warcinfo
WARC-Date: 2006-09-19T17:20:14Z
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 1.0
conformsTo:
 http://www.archive.org/documents/WarcFileFormat-1.0.html
```

C.2 Example of 'request' record

```
WARC/1.0
WARC-Type: request
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Warcinfo-ID: <urn:uuid:d7ae5c10-e6b3-4d27-967d-34780c58ba39>
WARC-Date: 2006-09-19T17:20:24Z
Content-Length: 236
WARC-Record-ID: <urn:uuid:4885803b-eebd-4b27-a090-144450c11594>
Content-Type: application/http;msgtype=request
WARC-Concurrent-To: <urn:uuid:92283950-ef2f-4d72-b224-f54c6ec90bb0>

GET /images/logoc.jpg HTTP/1.0
User-Agent: Mozilla/5.0 (compatible; heritrix/1.10.0)
From: stack@example.org
Connection: close
Referer: http://www.archive.org/
Host: www.archive.org
Cookie: PHPSESSID=009d7bb11022f80605aa87e18224d824
```

C.3 Example of 'response' record

```
WARC/1.0
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: 2006-09-19T17:20:24Z
WARC-Block-Digest: sha1:UZY6ND6CCHXETFVJD2MSS7ZENMWF7KQ2
WARC-Payload-Digest: sha1:CCHXETFVJD2MUZY6ND6SS7ZENMWF7KQ2
WARC-IP-Address: 207.241.233.58
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 2006 17:18:40 GMT
Server: Apache/2.0.54 (Ubuntu)
Last-Modified: Mon, 16 Jun 2003 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]
```

C.4 Example of 'resource' record

```
WARC/1.0
WARC-Type: resource
WARC-Target-URI: file://var/www/htdoc/images/logoc.jpg
WARC-Date: 2006-09-30T16:40:32Z
WARC-Record-ID: <urn:uuid:23200706-de3e-3c61-a131-g65d7fd80cc1>
Content-Type: image/jpeg
WARC-Payload-Digest: sha1:DBXHDRBXLF4OMUZ5DN4JJ2KFUAOB6VK8
WARC-Block-Digest: sha1:DBXHDRBXLF4OMUZ5DN4JJ2KFUAOB6VK8
Content-Length: 1662

[image/jpeg binary data here]
```

C.5 Example of 'metadata' record

```
WARC/1.0
WARC-Type: metadata
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Date: 2006-09-19T17:20:24Z
WARC-Record-ID: <urn:uuid:16da6da0-bcdc-49c3-927e-57494593b943>
WARC-Concurrent-To: <urn:uuid:92283950-ef2f-4d72-b224-f54c6ec90bb0>
Content-Type: application/warc-fields
WARC-Block-Digest: sha1:VXT4AF5BBZVHDYKNC2CSM8TEAWDB6CH8


Content-Length: 59
```

```
via: http://www.archive.org/
hopsFromSeed: E
fetchTimeMs: 565
```

C.6 Example of 'revisit' record

```
WARC/1.0
WARC-Type: revisit
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Date: 2007-03-06T00:43:35Z
WARC-Profile: http://netpreserve.org/warc/1.0/server-not-modified
WARC-Record-ID: <urn:uuid:16da6da0-bcdc-49c3-927e-57494593bbbb>
WARC-Refers-To: <urn:uuid:92283950-ef2f-4d72-b224-f54c6ec90bb0>
Content-Type: message/http
Content-Length: 226

HTTP/1.x 304 Not Modified
Date: Tue, 06 Mar 2007 00:43:35 GMT
Server: Apache/2.0.54 (Ubuntu) PHP/5.0.5-2ubuntu1.4
Connection: Keep-Alive
Keep-Alive: timeout=15, max=100
Etag: "3e45-67e-2ed02ec0"
```

C.7 Example of 'conversion' record

```
WARC/1.0
WARC-Type: conversion
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Date: 2016-09-19T19:00:40Z
WARC-Record-ID: <urn:uuid:16da6da0-bcdc-49c3-927e-57494593dddd>
WARC-Refers-To: <urn:uuid:92283950-ef2f-4d72-b224-f54c6ec90bb0>
WARC-Block-Digest: sha1:XQMRY75YY42ZWC6JAT6KNXKD37F7MOEK
Content-Type: image/neoimg
Content-Length: 934

[image/neoimg binary data here]
```

C.8 Example of segmentation ('continuation' record)

Let us take the example of the 'response' record given earlier, and segment it to fit the within a WARC file no larger than 2K. The first WARC file would contain the first segment, a record of type 'response' with a WARC-Segment-Number of 1. Note that the block-digest has changed -- as the block is no longer the same as the standalone 'response' record -- but the payload-digest has not changed, as the reassembled record will have the same internal payload.

```
WARC/1.0
WARC-Type: response
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Date: 2006-09-19T17:20:24Z
WARC-Block-Digest: sha1:2ASS7ZUZY6ND6CCHXETFVJDENAWF7KQ2
WARC-Payload-Digest: sha1:CCHXETFVJD2MUZY6ND6SS7ZENMWF7KQ2
WARC-IP-Address: 207.241.233.58
WARC-Record-ID: <urn:uuid:39509228-ae2f-11b2-763a-aa4c6ec90bb0>
WARC-Segment-Number: 1
```

```
Content-Type: application/http;msgtype=response
Content-Length: 1600

HTTP/1.1 200 OK
Date: Tue, 19 Sep 2006 17:18:40 GMT
Server: Apache/2.0.54 (Ubuntu)
Last-Modified: Mon, 16 Jun 2003 22:28:51 GMT
ETag: "3e45-67e-2ed02ec0"
Accept-Ranges: bytes
Content-Length: 1662
Connection: close
Content-Type: image/jpeg

[first 1360 bytes of image/jpeg binary data here]
```

The next file would contain the 'continuation' record, with fields to identify the start of the segmentation series (WARC-Segment-Origin-ID), to indicate this record's place in the series (WARC-Segment-Number), and to report that this the last record and what the total size is (WARC-Segment-Total-Length).

```
WARC/1.0
WARC-Type: continuation
WARC-Target-URI: http://www.archive.org/images/logoc.jpg
WARC-Date: 2006-09-19T17:20:24Z
WARC-Block-Digest: sha1:T7HXETFVA92MSS7ZENMFZY6ND6WF7KB7
WARC-Record-ID: <urn:uuid:70653950-a77f-b212-e434-7a7c6ec909ef>
WARC-Segment-Origin-ID: <urn:uuid:39509228-ae2f-11b2-763a-
aa4c6ec90bb0>
WARC-Segment-Number: 2
WARC-Segment-Total-Length: 1902
WARC-Identified-Payload-Type: image/jpeg
Content-Length: 302

[last 302 bytes of image/jpeg binary data here]
```

Annex D
(informative)

Use cases for writing WARC records

Below are listed different use cases developing some situations where WARC files and WARC records may be generated. These use cases correspond to the needs of the web archiving community.

> N.B.: 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 may not be contained in a single WARC file or in a set of WARC files but may be spread out and stored along pieces of other harvested websites. Thus, to render the archive of a website to users, access software may have to extract files contained in WARC records from different WARC files. External indexes may be used for a quicker access.

Other users may imagine other use cases to answer 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 describes the use case and its different steps.

The second column indicates what type of record is generated. Only the most complex named field 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 the document.

Note: we suppose these WARC records are written in an already opened WARC file, containing a 'warcinfo' record.

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. **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

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). This file is archived in a WARC file. **Date: 2007-11-02 at 15:20:44 GMT**	
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

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 **Date: 2007-11-24 at 18:28:24 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: 2007-11-24T18:28:24Z 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

Use case four: After the end of the harvest, Jhove is used to validate the format of http://netpreserve.org/reports/iipc2007conference.pdf. It produces validation results that have to be stored in a WARC file and linked to the corresponding record. **Date: 2007-11-01 at 20:54:02 GMT**	
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

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. It is necessary to migrate this file from the obsolete format to a new format. **Date: 2020-01-23 at 16:14:32 GMT**	
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