ISO TC 46/SC 4 N 595

Date: 2006-02-6

ISO/WD XXXXX

ISO TC 46/SC 4/WG

Secretariat: ANSI

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: (20) Preparatory
Document language: E

Contents

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/WD XXXXX was prepared by Technical Committee ISO/TC 46, *Information and documentation*, Subcommittee SC 4, *Technical interoperability*. It is derived from a working specification created in the context of an open-source software project and previously published in a series of drafts to prepare for publication as an Internet RFC.

Introduction

BACKGROUND INFORMATION ON WEB ARCHIVING (PROPOSAL) 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 Web ARChive (WARC) 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] format 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 format arose from the discussion and experiences of the International Internet Preservation Consortium (IIPC) [IIPC], whose members included 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, 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 [HERITRIX] web crawler), managing, accessing, and exchanging content.

Besides the primary content currently recorded, the extension of the 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.

Information and documentation — The WARC File Format

1 Scope [Goals]

This international standards specifies the Web ARChive file format, which provides the following:

— ability to store both the payload content and control information from mainstream Internet application layer protocols, such as HTTP, FTP, NNTP, and SMTP ;

— ability to store arbitrary metadata linked to other stored data (e.g. subject classifier, discovered language, encoding) ;

— support for data compression and maintenance of data record integrity ;

— ability to store all control information from the harvesting protocol (e.g. request headers), not just response information ;

— ability to store the results of data transformations linked to other stored data ;

— ability to store a duplicate detection event linked to other stored data (to reduce storage in the presence of identical or substantially similar resources) ;

— ability to be extended without disruption to existing functionality ;

— ability to store globally unique record identifiers ;

— support for deterministic handling of long records (e.g. truncation, segmentation).

The WARC file format is sufficiently different from the legacy ARC format files 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.

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.

[RDF] “Resource Description Framework (RDF)” (HTML). <http://www.w3.org/RDF/>

[RFC0822] Crocker, D., “Standard for the format of ARPA Internet text messages,” STD 11, RFC 822, August 1982. <ftp://ftp.isi.edu/in-notes/rfc822.txt>

[RFC1035] Mockapetris, P., “Domain names - implementation and specification,” STD 13, RFC 1035, November 1987. <ftp://ftp.isi.edu/in-notes/rfc1035.txt>

[RFC1884] Hinden, R. and S. Deering, “IP Version 6 Addressing Architecture,” RFC 1884, December 1995. <ftp://ftp.isi.edu/in-notes/rfc1884.txt>

[RFC1950] Deutsch, L. and J-L. Gailly, “ZLIB Compressed Data Format Specification version 3.3,” RFC 1950, May 1996. (TXT) <ftp://ftp.isi.edu/in-notes/rfc1950.txt>; (PS) <ftp://ftp.isi.edu/in-notes/rfc1950.ps>; (PDF) <ftp://ftp.isi.edu/in-notes/rfc1950.pdf>

[RFC1951] Deutsch, P., “DEFLATE Compressed Data Format Specification version 1.3,” RFC 1951, May 1996. (TXT) <ftp://ftp.isi.edu/in-notes/rfc1951.txt>; (PS) <ftp://ftp.isi.edu/in-notes/rfc1951.ps>; (PDF) <ftp://ftp.isi.edu/in-notes/rfc1951.pdf>

[RFC1952] Deutsch, P., Gailly, J-L., Adler, M., Deutsch, L., and G. Randers-Pehrson, “GZIP file format specification version 4.3,” RFC 1952, May 1996. (TXT) <ftp://ftp.isi.edu/in-notes/rfc1952.txt>; (PS) <ftp://ftp.isi.edu/in-notes/rfc1952.ps>; (PDF) <ftp://ftp.isi.edu/in-notes/rfc1952.pdf>

[RFC2045] Freed, N. and N. Borenstein, “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies,” RFC 2045, November 1996. <ftp://ftp.isi.edu/in-notes/rfc2045.txt>

[RFC2048] Freed, N., Klensin, J., and J. Postel, “Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures,” BCP 13, RFC 2048, November 1996. (TXT) <ftp://ftp.isi.edu/in-notes/rfc2048.txt;> (HTML) <http://xml.resource.org/public/rfc/html/rfc2048.html>; (XML) <http://xml.resource.org/public/rfc/xml/rfc2048.xml>

[RFC2141] Moats, R., “URN Syntax,” RFC 2141, May 1997. (TXT) <ftp://ftp.isi.edu/in-notes/rfc2141.txt>; (HTML) <http://xml.resource.org/public/rfc/html/rfc2141.html>; (XML) <http://xml.resource.org/public/rfc/xml/rfc2141.xml>

[RFC2234] Crocker, D., Ed. and P. Overell, “Augmented BNF for Syntax Specifications: ABNF,” RFC 2234, November 1997. (TXT) <ftp://ftp.isi.edu/in-notes/rfc2234.txt>; (HTML) <http://xml.resource.org/public/rfc/html/rfc2234.html>; (XML) <http://xml.resource.org/public/rfc/xml/rfc2234.xml>

[RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, “Uniform Resource Identifiers (URI): Generic Syntax,” RFC 2396, August 1998. (TXT) <ftp://ftp.isi.edu/in-notes/rfc2396.txt>; (HTML) <http://xml.resource.org/public/rfc/html/rfc2396.html>; (XML) <http://xml.resource.org/public/rfc/xml/rfc2396.xml>

[RFC2540] Eastlake, D., “Detached Domain Name System (DNS) Information,” RFC 2540, March 1999. <ftp://ftp.isi.edu/in-notes/rfc2540.txt>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, “Hypertext Transfer Protocol -- HTTP/1.1,” RFC 2616, June 1999. (TXT) <ftp://ftp.isi.edu/in-notes/rfc2616.txt>; (HTML) <http://xml.resource.org/public/rfc/html/rfc2616.html>; (XML) <http://xml.resource.org/public/rfc/xml/rfc2616.xml>

[RFC4027] Josefsson, S., “Domain Name System Media Types,” RFC 4027, April 2005. <ftp://ftp.isi.edu/in-notes/rfc4027.txt>

[RFC4501] Josefsson, S., “Domain Name System Uniform Resource Identifiers,” RFC 4501, May 2006. <ftp://ftp.isi.edu/in-notes/rfc4501.txt>

3 Terms, definitions and acronyms

3.1 Terms and definitions

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

WARC record – The basic constituent of a WARC file, consisting of a sequence of WARC records.

WARC record content block – The part (zero or more octets) of a WARC record that follows the header and that forms the main body of a WARC record.

WARC record header – The beginning of a WARC record, consisting of a header-line followed by lines of named fields up to a blank line.

WARC record header-line – A line of whitespace-separated text tokens that begins each WARC record.

WARC named fields – A set of elements consisting of a name, a colon, and a value, with long values continued on indented lines.

WARC positional fields – A set of elements consisting of text tokens, identified by their relative position, that appear on a header-line or in the value of a named field.

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
IETF	Internet Engineering Task Force
RFC	Request For Comments
UR(I/L/N)	Uniform Resource (Identifier/Locator/Name)
WARC	Web ARChive

4 The WARC record model

4.1 General

A WARC format file is the simple concatenation of one or more WARC records. A record consists of a record header followed by a record content block and two new lines. The WARC record header declares baseline identifying information about the current record, and allows additional per-record information. It consists of one first line of required positional fields, then a variable number of lines of named fields. Each record's content block contains zero or more bytes of data, interpreted according to the record type and any preceding headers.

Newlines are represented by CRLF. The WARC reccord can be expressed in the following IETF Augmented Backus-Naur Form (ABNF) grammar specified in [RFC2234]. (All-caps "core" elements are as defined in RFC2234.)

```
warc-file   = 1*warc-record
warc-record = header block CRLF CRLF
header      = header-line CRLF *named-field CRLF
block       = *OCTET
```

Elements of the ABNF grammar are further specified and explained in clauses 4 and 6.

4.2 Header-line

The record *header-line* is a newline-terminated sequence of whitespace-delimited text tokens representing parameters common to every record (whether or not captured from the web). Token order is significant.

```
header-line = warc-id vwsp data-length vwsp creation-date vwsp
               record-id vwsp segment-status vwsp
vwsp        = 1*SPACE
```

The amount of whitespace between *header-line* tokens is variable. This gives archive builders the flexibility to add padding and later adjust pre-written header parameters when final values are only completely known after the record content *block* has been written.

warc-id

A fixed pattern, "warc/JJ.NN", that appears first in every record and hence begins the WARC file itself. The pattern specifies the major (JJ) and minor (NN) version numbers of the WARC specification to which the WARC record conforms; JJ and NN are fixed two-digit strings, zero-padded on the left. This document specifies version 00.13, meaning a fixed pattern of,

```
warc/00.13
```

starts each record. The major and minor version numbers, both given as exactly two digits. They serve to identify the file format and version to outside inspection, and to assist error recovery when a process reading a WARC file fails to find the next record boundary where expected.

Occurrences of this string are not definitively the same as record boundaries, since the string may by chance occur inside a record. However, its fixed length and form may make it useful when attempting to recover from file corruption that has rendered one or more data-length parameters unreliable.

data-length

The combined length of the header and block sections of this record, in octets, starting with the first letter ("w") of the first token, through to the end of the content block — but not including the two record-ending CRLF newlines. After proceeding this many octets from that first character of the record header, there should be two CRLF newlines and either the beginning of a new record or the end of the file. (WARC reading implementations may choose to tolerate more or fewer CRLF newlines at the end of a record.) The data-length is the most important header parameter for efficient bulk processing, which permits, for example, entire records to be skipped without parsing their contents.

If the first next token does not match the first token of a WARC record (“warc/JJ.NN”), then the previous data-length should be considered in error; corrective action might include searching for a nearby occurrence of "warc/JJ.NN and other character patterns indicative of a legal record beginning.

creation-date

A 14-digit timestamp in the format YYYYMMDDhhmmss representing the GMT time when record creation began. Multiple records written as part of a single collection action may share the same creation-date, even though the times of their writing will not be exactly synchronized.

record-id

An identifier assigned to the record that is globally unique for its period of intended use. No identifier scheme is mandated by this specification, but each record-id should 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:"). The record-id is a strong feature of the WARC in that it allows unique record reference (e.g., from other WARC records and from search indexes); on those occasions when unique reference is not important, the record-id may be specified as a hyphen (“-“). The record-id should always be written with no internal whitespace.

segment-status

A token of the form *CM*, where *C* is a letter representing a segment code and *M* is a positive integer representing a segment number. Segment numbering starts with 1, and every logical record is considered to have at least one segment. A record is considered to end when its final segment is encountered with a segment code *C* different from “p”. Defined values for *C* are:

p partial – this segment (numbered *M*) is part of a still incomplete record
w whole -- this segment (*M*) ends the record, which is complete
t truncated – this segment (*M*) ends the record, truncated due to a time constraint
z truncated – this segment (*M*) ends the record, truncated due to a size constraint
x truncated – this segment (*M*) ends the record, truncated for an unspecified reason

The most common segment-status is “w1”, meaning the logical record is wholly contained in its first and only segment (“whole in one”); those applications that construct a record as one long string may wish to write “w1” in the header as an optimistic default, and later change the “w” to a “p” in the unusual case that the record will not fit in one segment. A complete logical record spanning 7 parts will have segments with this series of segment-status codes:

p1, p2, p3, p4, p5, p6, w7

To keep segments grouped with the appropriate logical record, it is a requirement that every non-initial segment contain the named field Segment-Origin-ID. To indicate truncation, the last segment’s number should be preceded by the code “t”, “z”, or “x”; for example, a 3 part record whose last segment was the result of a web capture that had insufficient time to finish will have the series “p1, p2, t3”.

4.3 Named fields following the header-line

Zero or more named fields, all of them optional, follow the *header-line.* These fields have a line-oriented syntax very similar to that of email headers [RFC0822] but with unrestricted "text" values (none of its 13 reserved special characters). Essentially, an element consists of a name, a colon, and a value, where long values may be continued on indented lines after the name. Here, this format is called the “named field syntax” and is precisely specified by:

```
named-field = field-name ":" [field-body] CRLF
field-name  = 1*<any CHAR, excluding control-chars and ":">
field-body  = text [CRLF 1*WSP field-body]
text        = 1*<any UTF-8 character, including bare
                 CR and bare LF, but NOT including CRLF>
                                       ;  (Octal, Decimal.)
CHAR        = <any ASCII/UTF-8 character> ;  (0-177,  0.-127.)
CR          = <ASCII CR, carriage return> ;  (  15,      13.)
LF          = <ASCII LF, linefeed>        ;  (  12,      10.)
SPACE       = <ASCII SP, space>           ;  (  40,      32.)
HTAB        = <ASCII HT, horizontal-tab>  ;  (  11,       9.)
CRLF        = CR LF
WSP         = SPACE / HTAB                ; semantics = SPACE
```

This standard defines a number of parameters that may appear as *named-fields*. If there are no named fields present, the entire WARC record header is the line of positional parameters followed by one blank line (two consecutive CRLF newlines).

No named-field is required except for Segment-Origin-ID, which must occur in every non-initial segment of a multi-segment logical record, The ‘type’ and ‘content-type’ fields are strongly recommended at the beginning of every record (meaning the initial segment of a multi-segment logical record).

In principle, more than one instance of a named-field (bearing the same name) may occur in one record, but in practice this really only makes sense for the ‘Related-Resource’ field.

The rest of this section describes the currently defined named-fields.

type: ***type-specific-parameters***

The type of WARC record. If this parameter is absent, the record type defaults to 'data'. Although starting a WARC file with a 'warcinfo' record is recommended, any combination of record types may appear inside a WARC file. If no type is specified, it defaults to 'data'. Record types are defined in clause 5.

content-type: ***string***

The MIME type [RFC2045] of the information contained in the record's content block. This may be the fully structured MIME type with embedded spaces. For content in an HTTP request or response record, the content-type should be "application/http" as per Section 19.1 of [RFC2616] (or 'application/http; msgtype=request' or 'application/http; msgtype=response' respectively). In particular, it is not the value of the HTTP Content-Type header in an HTTP response but a MIME type to describe the content body (hence 'application/http' if the content body contains response headers and the response itself).. If no content-type is specified it defaults to "application/octet-stream". An example of this field is:

```
content-type: application/http; msgtype=request
```

revisit: ***ref-uri comparison***

An indication that the content block holds an empty or partial representation of the resource referenced by *ref-uri*, which usually identifies another WARC record (which may itself hold a partial representation) but which may be specified as "-" if a resource identifier is unavailable. Typically, this field is used when the content visited was either a complete or substantial duplicate of material previously archived. An example of this field is:

```
revisit: urn:uuid:d7ae5c10-e6b3-4d27-967d-34780c58ba39 same
```

The *comparison* parameter may be one of "same" (resource was identical), "different" (resource was different and the content block, if non-empty, describes the differences in free text), or "patch" (resource was different and the content block represents the differences). The content block for this record is empty or contains a patch (if not specified, a content-type of "text/patch" is assumed), which is a set of machine-readable instructions that can be used (on Unix systems) automatically to construct a complete representation when applied to the referenced resource

The purpose of this field is to permit reduction in redundant storage 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. A 'revisit' field may not make sense with some record types and should only be used when interpreting the record requires consulting a previous resource. It is not required that any revisit of a previously-visited URI use 'revisit'.

note: ***text***

This field can be used to enter any free text comment or observation about the WARC record.

IP-Address: ***ip-address***

If the content block was obtained directly from an Internet host, this field can be used to hold the numeric Internet address of that host. An IPv4 address should be written as a "dotted quad"; an IPv6 address as per [RFC1884]. For example, in the case of an HTTP retrieval, this field would hold the IP address corresponding to the hostname in the record's target URI. An example of this field is:

```
IP-Address: 137.227.232.150
```

Checksum: ***algorithm:value***

This field can be used to indicate the name of a digest algorithm and the string representing the resulting value of the computation of the algorithm on the content block. An example is:

```
Checksum: sha1:AB2CD3EF4GH5IJ6KL7MN8OPQ
```

No particular algorithm is recommended, FUTURE though a future recommendation is possible.

Related-Resource: ***relationship uri***

A specified *relationship* to a resource referenced by *uri*, such as another WARC record. This field permits WARC records to relate to resources in ways other than specified elsewhere in this document (eg, see the 'http-request' and 'conversion' record types). The *relationship* string should be taken from a vocabulary such as Dublin Core terms [DCMI] or given as "-" (undefined). An example of this field is:

```
Related-Resource: isRequiredBy http://example.org/film_archive/rb983
```

A potential strategy, after choosing one record to be primary, is to extend its record-id as described in Annex A about record-id considerations. This creates satellite record-ids for related records that contain the primary record-id as an initial substring, which greatly optimizes the detection (and in some cases derivation) of related records.

Segment-Origin-ID: ***record-id***

The identifier of the first segment of a multi-segment logical record. This field is required of every non-initial segment of a multi-segment logical record. An example of this field is:

```
Segment-Origin-ID: http://nt2.info/ark:/12148/jv46637b2w
```

Warcinfo-ID: ***record-id***

When present, indicates the 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 record this parameter routinely (e.g., before computing checksums). The Warcinfo-ID parameter 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. FUTURE Use of this parameter in a record of type 'warcinfo' is undefined and reserved for possible future extension.

4.4 Content block following the header

A *content block* of zero or more octets follows the header. The block may contain arbitrary binary data, up through the remaining number of octets as specified in the previously-given *data-length* parameter. After the content block, two CRLF newlines should be given, although they are never counted in the declared record *data-length*.

5 Record types

5.1 General

New record types that extend WARC may be defined in the future. WARC processing software is encouraged to tolerate (e.g., skip over) records of unknown type. If no record type is given, a default type of 'data' is assumed. Some record types take positional parameters, for example, a "uri" is required with an HTTP response:

```
type: http-response uri
```

Several record types have a content block that is expected to be structured. If no content-type parameter accompanies such a record, the content block is assumed to be structured according to the named field syntax following the WARC header-line.

5.2 type: warcinfo

A 'warcinfo' record describes the records that follow it until the end of file (end of input), or another 'warcinfo' record is reached. Typically, this appears once in a WARC file, usually at the beginning. For a web archive, it often contains a description of a web crawl (e.g., seeds, depth, timeout, purpose, maximum file size). If no content-type parameter accompanies this record, the content block is assumed to contain a set of named fields in the same format as those following the WARC header-line. An example record is:

```
warc/00.13 390 20070214235805 urn:uuid:d7ae5c10-e6b3-4d27-967d-34780c58ba39 w1
type: warcinfo

software: Heritrix 1.4.0 http://crawler.archive.org
hostname: crawler017.archive.org
ip: 207.241.227.234
isPartOf: testcrawl-20050708
title: testcrawl with WARC output
creator: IA_Admin
http-header-user-agent: Mozilla/5.0
     (compatible; heritrix/1.4.0 +http://crawler.archive.org)
```

The precise specification of the content block for this record type is outside the scope of this document.

5.3 type: http-response *uri*

An 'http-response' record contains an entire HTTP protocol response, including headers and content-body, from an Internet retrieval. The *uri* parameter, which names the target of the HTTP request, is required; it is the original URI whose capture gave rise to the information content in this record. The URI in this value should be properly escaped according to [RFC2396] and written with no internal whitespace.

Often the payload of such a response reflects the main collection objective of the archiving service, whose responsibility it is to distinguish payload from protocol headers during subsequent processing. A response record may come with the named parameters 'IP-Address' and 'Related-Resource'. An example record is:

```
warc/00.13 7425 20050708010101 http://nt2.info/ark:/13030/km19rs47q w1
IP-Address: 207.241.224.241
Checksum: sha1:2ZWC6JAT6KNXKD37F7MOEKXQMRY75YY4

HTTP/1.x 200 OK
Date: Fri, 08 Jul 2005 01:01:01 GMT
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/5.0.4-0.3
Last-Modified: Sun, 12 Jun 2005 00:31:01 GMT
Etag: "914480-1b2e-42ab8245"
Accept-Ranges: bytes
Content-Length: 6958
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: image/jpeg

[6958 bytes of binary data here ]
```

5.4 type: http-request *response-uri*

An 'http-request' record holds the protocol request headers (e.g., GET or POST) associated with a particular HTTP request. The *response-uri* parameter, which references the HTTP response, is required, but may be given as “-“ to leave it undefined. The URI in this value should be properly escaped according to [RFC2396] and written with no internal whitespace. A request record may often come with the named parameter 'Related-Resource'. An example record is:

```
warc/00.13 289 20050708010101 uuid:f569983a-ef8c-4e62-b347-295b227c3e51 w1
```

```
type: http-request http://nt2.info/ark:/13030/km19rs47q
IP-Address: 207.241.224.241

GET /images/logo.jpg HTTP/1.0
Host: www.archive.org
User-Agent: Mozilla/5.0 (compatible; crawler/1.4 +http://example.com)
```

5.5 type: dns-response *dns-uri*

A ‘dns-response’ record is used to hold the results of an Internet Domain Name System “A” record lookup on a given *dns-uri*. Records of this type are often used in a web archiving context to hold the IP address of a hostname that may be the target of repeated HTTP requests. A request for DNS information can be summarized in a URI in accordance with an IETF Network Working Group draft proposal [RFC4501]. DNS information as retrieved can be represented in the formats specified by [RFC1035], [RFC2540], and [RFC4027]. An example record is:

```
warc/00.13 252 20060909004930 - w1
type: dns-response dns:ca.water.usgs.gov
content-type: text/dns

20060909004930
ca.water.usgs.gov.      60      IN      A       137.227.232.150
ca.water.usgs.gov.      60      IN      A       137.227.232.151
ca.water.usgs.gov.      60      IN      A       137.227.232.152
```

If present, the IP-Address named field should be the address of the DNS server that provided the DNS record.

5.6 type: metadata *ref-uri*

A 'metadata' record contains content created in order to further describe, explain, or accompany another resource in ways not covered by other record types. The other resource is referenced by the *ref-uri* parameter, often identifying another WARC record or an Internet-accessible resource (however, it may be given as “-“). Any number of metadata records may be created that reference the same resource (e.g., another WARC record). A metadata record may also come with the named parameter 'Related-Resource' to specify other related records.

The format of the metadata is outside the scope of this document, but potential formats include the named field syntax used earlier and [RDF] or other XML-based formats. If no content-type parameter accompanies this record, the content block is assumed to contain a set of named fields in the same format as those following the WARC header-line. An example record is:

```
warc/00.13 282 20070214235805 http://nt2.info/ark:/13030/xt35jw94m w1
type: metadata http://nt2.info/ark:/13030/zd4852997p

erc:
who:    Lederberg, Joshua
what:   Studies of Human Families for Genetic Linkage
when:   1974
where:  http://profiles.nlm.nih.gov/BB/AA/TT/tt.pdf
```

5.7 type: conversion *ref-uri flag*

A 'conversion' record contains an alternative version of another resource’s content. The other resource is referenced by the *ref-uri* parameter, often identifying another WARC record or an Internet-accessible resource. The *flag* parameter may be given as either “noenvelope” (to indicate that any protocol control information stored in the original resource was stripped off before conversion) or “-“ (no flag specified). The flag is a place to signal the common expected case in which an http-response record was not actually converted, but only the content inside the protocol envelope (the "payload") was converted.

This record may also come with the named parameter 'Related-Resource' to specify other related records. An example record is:

```
warc/00.13 15153 20060909004930 http://nt2.info/ark:/13030/br41q9831h4 w1
type: conversion http://nt2.info/ark:/13030/km19rs47q noenvelope
content-type: image/jp2k

[ 14,984 bytes of binary image data here ]
```

Typically, a 'conversion' record 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 transformation 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.

5.8 type: data

A 'data' record contains digital content of unspecified type. This type is the default if no record type is given. Examples include a file directly retrieved from a locally accessible repository, or the result of a networked retrieval where the protocol information has been discarded.

6 Truncated and segmented records

6.1 General

For practical reasons, users 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.

Additionally, users will often want to keep individual WARC files near or below some target size, such as 500MB or 1GB. If some records would be too large to be contained by a single WARC file of desired maximum size, those records will have to be split between multiple WARC files.

This clause defines mechanisms for indicating that a WARC record has been truncated or split into multiple records, called segments,. This is based on the concept of a logical record that may span multiple WARC records, perhaps in held in different WARC files. Each segment is represented by a separate WARC record.

6.2 Record truncation

Any record may indicate that truncation has occurred and give the reason by using the segment-status code on the WARC header-line. A code of "t" means that truncation occurred because of a time constraint and a code of "z" means that truncation occurred because of a size constraint. The reason may be left unspecified by using a code of "x".

6.3 Record segmentation

A record that will not fit into a single WARC file of desired maximum size may be broken into any number of separate records, called segments. Together these segments comprise the logical record. As much as possible, segmentation should be avoided. When segmentation is needed, segments other than the first shall come with the named field, Segment-Origin-ID, to tie the logical record segments together.

The first segment shall carry the record-type that the record would have had were it not broken into segments, and a segment number in the header-line segment-status of "1".

All subsequent segments shall come with a Segment-Origin-ID field and an incremented segment number. They shall also include a 'Segment-Origin-ID' field with a value of the Record-ID of the record containing the first segment of the set. Segments other than the first should contain no other named fields, as they merely serve to continue the record data block of the first record.

The last segment may also contain an indication of truncation, if appropriate. For example, a 4-segment logical record that is truncated due to excessive size, and whose first segment's record-id is 54321, would have the series of segment-status and Segment-Origin-ID fields according to this table:

Segment-Status	Segment-Origin-ID
p1	*undefined*
p2	54321
p3	54321
z4	54321

Example of series of parameters in a 4-segment logical record.

To reassemble all segments into the intended complete logical record, all records with the same 'Segment-Origin-ID' value shall be collected and appended, in segment number order, to the origin record.

Annex A
(informative)

Considerations in choice of record ID

The WARC format differs significantly from the ARC format in requiring the record-id parameter. The record-id should be globally unique for its period of intended use. If that period is indefinite, the record-id should be maintained to a level appropriate for any persistent identifier, in which case identifier opaqueness is usually desirable.

There is no reason why the archiving institution may not choose record-ids that are also "actionable" (submittable as retrieval requests to widely available tools such as web browsers) as long as there are providers to service them. This specification does not dictate what identifier scheme to use; suitable schemes include URN [RFC2141], [ARK], [GUID], etc.

Also worth considering is the establishment of lexical conventions for record-ids that reveal or suggest relationships among content blocks. Although some record types are already required to reference certain related resource and the 'Related-Resource' parameter may also be used, great optimization can be had when relatedness can be inferred by third parties through identifier comparison rather than by lookup in a database or examination of the relevant WARC files.

These conventions are suggested by [RFC2396], formalized by the [ARK] scheme, and are applicable to such things as the summarizing of large search results from Internet-wide indexing engines. As an example of a convention that could be adopted by users of any identifier scheme, the "/" character could be reserved as a separator used to introduce an extension string that is appended to a primary record-id. If the record-id of a primary block of captured content were,

```
http://abc.org/12026/987654321
```

The convention could also reserve the extension strings "_s", "_d", and "_t" to indicate record- ids for secondary, duplicate, and transform blocks, respectively. Over time this might result in the assignment of record-ids such as,

```
http://abc.org/12026/987654321/_s1
http://abc.org/12026/987654321/_s2
http://abc.org/12026/987654321/_d9
http://abc.org/12026/987654321/_d10
http://abc.org/12026/987654321/_t
```

Annex B
(informative)

WARC application to specific protocols

B.1 HTTP and HTTPS

A full HTTP or HTTPS response, with protocol information and content-body (if any), can be saved verbatim into a WARC file as an 'http-response' type record, with a MIME content-type of "application/http" (or "application/http; msgtype=response").

A full HTTP or HTTPS request, including all request headers and content-body (if any), can similarly be saved verbatim into a WARC file as an "http-request" type record, with a MIME content-type of "application/http" (or "application/http; msgtype=request").

For either a request or response, an 'IP-Address' field may be used to record the network IP address to which the request was directed, using the best available DNS information at the time.

Additional metadata about the HTTP or HTTPS transaction may be stored in a 'metadata' type record, OUTSIDE THE SCOPE in a format to be specified elsewhere. In particular, information about the secure session in which an HTTPS transaction occurs, such as certificates presented or consulted and authentication information exchanged, may be stored in one or more 'metadata' type records.

The multiple records which pertain to a single HTTP or HTTPS logical group of records will all have unique record-id values.

As any mixture of record types may appear for a single collection event, and in any order, there is no specific record type which is automatically considered primary. Generally, all may refer back to the one record which appeared first, but this is not required. (A request record may refer to a response record or vice-versa; either could refer to a 'metadata' record or a 'metadata' record could refer to either.) Multiple and bidirectional 'Related-Resource' fields may appear.

In the case where resources from a website have been harvested or otherwise received without performing normal HTTP operations, or where HTTP protocol information has been lost, it may be appropriate to store the plain content in WARC 'data' type records, but using the content MIME type in place of "application/http".

B.2 Other resources with URIs, and other protocols

Any resource, even if it is not retrieved via an Internet operation, may be archived in a WARC file under a 'data' type record. This includes files retrieved from a locally-accessible file system or other repository.

OUTSIDE THE SCOPE Specific conventions for other protocols and media types are expected to be defined as necessary. In general, the WARC format should be capable of archiving any digital resource which has a specific time of collection and a discrete length.

The 'http-request' and 'http-response' record types should be used for verbatim or lossless transcripts of collection activity, including protocol information. The 'data' record type should be used for content without any protocol-specific enveloping. Additional information about a resource or transaction can be supplied in a protocol- or media-appropriate manner with 'metadata' type records.

Annex C
(informative)

Compression recommendations

C.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 **[RFC1952]**, **[RFC1950]**, and **[RFC1951]**. 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 clause documents recommended, but optional, practices for compressing WARC files with GZIP.

C.2 Record-at-a-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. In particular, the declared record length remains the length of the uncompressed record.

C.3 GZIP WARC file name suffix

The name of a gzip-compressed WARC file should have the customary ".gz" appended to it, making the complete suffix, ".warc.gz".

Annex D
(informative)

Collected ABNF for WARC

```
warc-file      = 1*warc-record
warc-record    = header block CRLF CRLF
header         = header-line CRLF *named-field CRLF
block          = *OCTET
header-line    = warc-id vwsp data-length vwsp creation-date vwsp
                 record-id vwsp segment-status vwsp
vwsp           = 1*SPACE
warc-id        = "warc/" DIGIT DIGIT "." DIGIT DIGIT
data-length    = 1*DIGIT
record-id      = uri
uri            = <'URI' per RFC3986>
creation-date  = <YYYYMMDDhhmmss>               ; Greenwich Mean Time
segment-status= SegCode SegNum
SegCode        = "p" / "w" / "t" / "z"
SegNum         = 1*DIGIT
named-field    = defined-fields / field-name ":" [ field-body ] CRLF
defined-fields =     "type: warcinfo" CRLF
          / "type:" vwsp "http-response" vwsp uri CRLF
          / "type:" vwsp "http-request" vwsp response-uri CRLF
          / "type:" vwsp "dns-request" vwsp dns-uri CRLF
          / "type:" vwsp "metadata" vwsp ( ref-uri / "-" ) CRLF
          / "type:" vwsp "conversion" vwsp ref-uri vwsp flag CRLF
          / "type:" vwsp "http-request" vwsp response-uri CRLF
          / "type:" vwsp "data" CRLF
          / "content-type:" vwsp <type/subtype> CRLF ; per RFC 2045
          / "revisit:" vwsp ref-uri vwsp
                         ("same" / "different" / "patch" ) CRLF
          / "note:" vwsp field-body CRLF
          / "IP-Address:" vwsp <ip-address> CRLF     ; per RFC 1884
          / "Checksum:" vwsp "sha1:" field-body CRLF
          / "Related-Resource:" vwsp relationship vwsp uri CRLF
          / "Segment-Origin-ID:" vwsp warc-record-id CRLF
          / "Warcinfo-ID:" vwsp warc-record-id CRLF
response-uri   = uri
dns-uri        = uri
ref-uri        = uri
warc-record-id = uri
flag           = "noenvelope" / "-"
relationship   = "-" / <string from DCMI terms>
field-name     = 1*<any CHAR, excluding control-chars and ":">
field-body     = text [CRLF 1*WSP field-body]
text           = 1*<any UTF-8 character, including bare
                 CR and bare LF, but NOT including CRLF>
                                              ; (Octal,  Decimal.)
CHAR           = <any ASCII/UTF-8 character>  ; (0-177,  0.-127.)
CR             = <ASCII CR, carriage return>  ; (   15,      13.)
LF             = <ASCII LF, linefeed>         ; (   12,      10.)
SPACE          = <ASCII SP, space>            ; (   40,      32.)
HTAB           = <ASCII HT, horizontal-tab>   ; (   11,       9.)
CRLF           = CR LF
WSP            = SPACE / HTAB                 ; semantics = SPACE
```

Annex E
(informative)

WARC file name and size recommendations

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 unique identifiers attributed to institutions 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".

The WARC File Format specification does not require any particular WARC file naming practice, but recommends conventions similar to the above be adopted within WARC-creating institutions. The file name prefix "iipc" should be avoided unless participating in the IIPC naming registry.

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

Annex F
(informative)

Registration of MIME media type application/warc

This Annex describes, as defined in [RFC2048], the MIME types associated with the WARC format.

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. Implementors 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

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

Bibliography

[ARC] Burner, M. and B. Kahle, “The ARC File Format,” September 1996 (HTML). <http://www.archive.org/web/researcher/ArcFileFormat.php>

[ARK] Kunze, J. and R. Rodgers, “The ARK Persistent Identifier Scheme,” August 2005 (PDF). <http://www.cdlib.org/inside/diglib/ark/arkspec.pdf>

[DCMI] “DCMI Metadata Terms,” December 2006 (HTML). <http://dublincore.org/documents/2006/12/18/dcmi-terms/>

[GUID] “Wikipedia: Globally Unique Identifiers” (HTML). <http://en.wikipedia.org/wiki/GUID>

[HERITRIX] “Heritrix Open Source Archival Web Crawler” (HTML). <http://crawler.archive.org>

[IIPC] “International Internet Preservation Consortium (IIPC)” (HTML).< http://www.netpreserve.org>