Network Working Group
Internet-Draft
Updates: 2045, 6838 (if approved)
Intended status: Informational
Expires: November 7, 2015

S. Leonard
Penango, Inc.
M. Kerwin

May 6, 2015

The Archive Top-Level Media Type for File Archives
draft-seantek-kerwin-arcmedia-type-01

Abstract

This document defines a new top-level media type to be known as "archive", which defines a fundamental type of media with unique presentational, hardware, and processing aspects.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The purpose of this memo is to update [RFC2045] and [RFC6838] to include a new top-level media type to be known as "archive". [RFC6838] describes mechanisms for specifying and describing the format of Internet Message Bodies via media type/subtype pairs. "archive" defines a fundamental type of media with unique presentational, hardware, and processing aspects. Various subtypes of this top-level type are immediately anticipated, and will be covered under separate documents.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Definition of an archive

The archive top-level media type identifies a container of one or more data objects and metadata about them. Archives are used to collect multiple data objects together into a single object for easier portability and storage. Archive formats can provide many optional services, including:

1. compression

2. encryption

3. authentication

4. backup and restoration

5. filesystem imaging

6. software packaging and distribution

7. volume-splitting (archive split into multiple objects)

8. block storage

Formats and techniques that support one or more of these services already exist under separate registrations. For example, the Content-Encoding header can be used to signal compressed Internet message content. The distinguishing feature of the archive top-level type is that these services are integrated into the format itself, along with the inclusion of object-specific metadata.

Formats contemplated under this top-level type are designed to concatenate multiple objects into a single data stream, along with names and other metadata. When an Internet-facing application handles content labeled with this type, it SHOULD treat the archive as a discrete data item. For example, an Internet mail user agent might display an archive-labeled type with an archive icon, possibly with a preview of the objects contained therein, as opposed to automatically extracting its contents.

3. Encoding and Transport

Unrecognized subtypes of archive SHOULD be treated as "archive/file". Like "application/octet-stream", the purpose of the "archive/file" type is to provide default handling; it does not represent a particular archive format. Implementations SHOULD defer handling of unrecognized subtypes of archive to a robust general-purpose archive processing application, if such an application is available.

If default archive handling is not supported, the archive MAY be treated as if it were "application/octet-stream".

Unless noted in the subtype registration, subtypes of archive MUST be assumed to contain binary data, implying the use of base64 content encoding for email and binary transfer for ftp and http.

4. Registration Template

The formal syntax for the subtypes of the archive top-level type SHOULD look like this:

Type name:
archive

Subtype name:
xxxxxxxx

Required parameters:
none

Optional parameters:
TBD

Encoding considerations:
base64 encoding is recommended when transmitting archive/* documents through MIME electronic mail.

Security considerations:
see Section 9 below

Published specification:
TBD

Applications that use this media type:
TBD

Fragment identifier considerations:
The considerations of this document, plus any extra syntaxes not inconsistent with this document.

Additional information:

Deprecated alias names for this type: (Include non-archive alias names, such as those in application.)

Magic number(s): TBD

File extension(s): TBD

Macintosh file type code(s): TBD

See Appendix A for references to some of the expected subtypes.

Person and email address to contact for further information:
TBD

Intended usage:

TBD (COMMON will be the most common)

Restrictions on usage:
TBD

Author:
TBD

Change controller:
TBD

Provisional registration? (standards tree only):
(Yes/No)

(Any other information that the author deems interesting may be added below this line.)

The optional parameters consist of starting conditions and variable values used as part of the subtypes.

5. Common Required and Optional Parameters

Archive formats usually include parametric meta-data within the format. Consequently, subtypes of archive SHOULD NOT specify the same information as parameters to the type.

Some archive formats are very old, or are designed to be backwards-compatible with older formats, and as such might not have been designed with transport across the Internet in mind. For example, modern versions of the ZIP file format [ZIP] include support for the Universal Character Set [ISO10646], however the default encoding of filenames within a ZIP archive has always been Code Page 437 [CP437]. Due to the historical nature of archives, and to support interoperability with older implementations, sometimes it is preferable to communicate the archive as-is, rather than updating it to a more modern or universal format.

Implementations that are archive-type aware MUST support the following parameters for maximum compatibility. At the same time, new archive types SHOULD NOT rely on these parameters for disambiguation; new archive types SHOULD be designed in such a way that "universal" interoperability is achieved using information contained within the archive format itself.

[[TODO: write this list]]

- Code Page - like charset but only applies to certain strings in the archive, when the archive format is ambiguous. Do not attempt to apply this parameter as one would apply charset to text/*

- Endianness?

- Time/Y2K representation issues?

- Anything else?

6. Split Archives

Several archive formats (notably RAR and ZIP) support split archives. A "split archive" can be stored in multiple files, or more generally, across multiple storage media.

For example, the ZIP format supports two types of splits: "split archive" and "spanned archive". A "split archive" is a standard ZIP archive split over multiple files using file extensions .z01, .z02, etc.; the final file in the sequence uses the .zip file extension. The "spanned archive" was designed for use on floppy disks with restrictive space limitations; all archive files have the same filename, and volume labels (presumably on floppy disks) are used to store sequence information. Neither sub-format is merely a naive division of the octet stream: each ZIP file is parseable in its own right, and contains its own offset values.

The TAR format (or family of formats, including cpio and ustar) was originally designed for streaming to and from tape devices, so splitting is accomplished differently.

[[TODO: Consider how to label this content. archive/zip^01? archive/zip; split=01? Something else? How shall 01 be associated with 02, 03, etc., when the Content-Disposition: ; filename="" parameter is "presentation-information" and may be separated from the Content-Type header information?]]

7. Fragment Identifier Syntax

As archives usually store objects in hierarchical structures similar to filesystems, archives can serve as virtual filesystems. Respondents have noted that the objects stored in an archive can be addressed by a fragment syntax that resembles a filesystem path. At the same time, archives can store objects in different ways (along with different types of metadata), suggesting that a common baseline with flexible extension points is more appropriate than a fixed universal syntax. [[TODO: This will be explored in future drafts. Note the similarities with this and the file: URI...]]

[[TODO: consider how to provide a fragment for content in the archive. NB: most archives do NOT provide Content-Type/media type information! So /foo.html being an HTML file is just an _assumption_, and possibly a very wrong one at that. There is no IETF registry for file extensions.]]

8. Piped-Composite Type Suffix Syntax

[[TODO: discuss tar piped through bzip2, gzip, etc. as a distinct file format, rather than an application of the Content-Encoding: header. Suggest common suffix like archive/tar|bzip2, where | is some useful character but not + since + is for structured syntaxes.]]

9. Security Considerations

Archives can store files, file metadata, and even entire filesystems; thus, security issues loom large because archives can contain just about anything. These concerns are magnified by the arbitrary transport of such data across the Internet. [[TODO: complete.]]

10. References

10.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

10.2. Informative References

[CP437] Microsoft Developer Network, "Code Page 437 MS-DOS Latin US", April 2015, <https://msdn.microsoft.com/en-us/library/cc195060.aspx>.

[ISO10646] International Organization for Standardization, "Information Technology - Universal Multiple-Octet Coded Character Set (UCS)", ISO/IEC 10646:2003, December 2003.

[ZIP] Lindner, P., "application/zip registration at IANA", June 1993, <http://www.iana.org/assignments/media-types/application/zip>.

Appendix A. Expected Subtypes

The following archive formats will be explored for registration as subtypes along with this effort:

Archiving Only TAR

Multipurpose (archiving, compression, encryption) ZIP, ACE, RAR, 7-Zip, StuffIt, FreeArc

Software Packaging MSI, RPM, JAR, XPI, CAB, CRX, APK

Disk Imaging ISO, NRG, BIN/CUE, VMDK, WIM, PartImage, IMG/IMA/IMZ, DMG

Appendix B. Change Log

Changes since -00

- retool to use XML2RFC - lots of layout changes
- remove large sections of text, as suggested by Ned Freed and Dave Crocker
- replace "primary" with "top-level", and "content-type" with "media type" throughout
- add reference to RFC 6838 (BCP 13) - Media Type Specifications and Registration Procedures
- lots of editorial changes

Authors' Addresses

Sean Leonard
Penango, Inc.
5900 Wilshire Boulevard
21st Floor
Los Angeles, CA 90036
USA

Email: dev+ietf@seantek.com
URI: http://www.penango.com/

Matthew Kerwin

Email: matthew@kerwin.net.au
URI: http://matthew.kerwin.net.au/