

ACAP TYPE Extension

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months. and may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress".

To learn the current status of any Internet-Draft, please check the `1id-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ftp.is.co.za` (Africa), `ftp.nordu.net` (Europe), `munni.oz.au` (Pacific Rim), `ds.internic.net` (US East Coast), or `ftp.isi.edu` (US West Coast).

This document suggests a proposed protocol for the Internet community, and requests discussion and suggestions for improvements. Distribution of this draft is unlimited.

The protocol discussed in this document is experimental and subject to change. Persons planning on either implementing or using this protocol are STRONGLY URGED to get in touch with the author before embarking on such a project.

1. Abstract

The Application Configuration Access Protocol [[ACAP](#)] defines rough typing information in the form of an attribute naming convention. This extension to ACAP allows a MIME content-type/subtype with parameters to be associated with a given piece of data, providing knowledgeable clients with useful information in a way which maintains compatibility with innocent clients and servers.

2. Conventions Used in this Document

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 "Key words for use in RFCs to Indicate Requirement Levels" [[KEYWORDS](#)].

3. Specification

The TYPE extension may be used with any ACAP server which returns "TYPE" as one of the supported capabilities in the initial untagged ACAP response.

Servers that support the TYPE extension define a new type of metadata on attributes, called "type". This metadatum is Read-Write, is protected by the same ACL which protects the rest of the attribute, and contains the MIME content-type/subtype of the "value" metadata associated with the attribute, along with any associated parameters.

The "type" metadatum for an attribute MUST only be stored if the "value" metadata are being set as well, in the same STORE command (this is to deal effectively with inheritance; for example, it would be bad if the "type" could be set in an inheriting attribute for an inherited value, when the inherited value could change at a later date. The value and the type of the value need to be considered together, as a single unit).

Note that, in the case of a multivalued attribute, all of the values are described by a single "type" metadatum, and thus MUST have the same MIME content-type.

3.1. Client Issues

A client MAY request the "type" metadatum from any server which supports the TYPE extension. This MUST NOT be taken as authoritative; the associated "value" metadata might not in fact be syntactically legal for the given type. Nevertheless, the type MAY be used as a hint to indicate how the data should be treated or displayed.

When a client stores a single or multi-value into the "value" metadata of an attribute, it MAY, in addition, store a value into the "type" metadatum of the attribute to indicate the type of the associated "value" metadata. The type stored MUST be a legal MIME type, and the "value" metadata MUST be legal values for that type.

If the client stores a NIL to the "value" metadata, it MUST either store a NIL to the "type" metadatum, or omit the type information from the STORE.

If the client stores a DEFAULT to the "value" metadata, it MUST either store a DEFAULT to the "type" metadatum, or omit the type information from the STORE.

If a server does not implement the TYPE extension, clients MUST NOT assume anything about the type of the value associated with a given attribute, or attempt to STORE to the "type" metadatum.

3.2. Server Issues

Servers implementing this extension SHOULD announce it via the initial ACAP greeting, with the capability "TYPE".

Servers receiving a STORE command for a "type" metadatum MUST ensure that the type is a legally formatted MIME type; if it is not, servers MUST return a tagged BAD response.

If a client performs a STORE to the "type" metadatum of an attribute, without simultaneously storing into the "value" metadata, the server MUST return a tagged BAD response.

Servers MAY parse the "value" metadata and ensure that they conform to the specified type; if they do not, a server MAY issue a tagged NO response.

Servers MUST NOT reject STOREs to "type" metadatum merely because they lack knowledge of the specific type, as long as the type is correctly formatted.

If a server receives a request to STORE a non-NIL or DEFAULT into the "value" metadata of an attribute without an accompanying value for the "type" metadatum, the server MUST behave as though the "type" metadatum were being set to "application/octet-stream".

Note that this implies that the server MUST NOT reject a STORE into a value that would be a legal store if this extension were not in place -- a STORE without a supplied type MUST cause the type to change to the most general type available given the restrictions imposed by the base protocol on the types of data that a given attribute may assume.

If a server receives a request to STORE a NIL to an attribute's "value" metadata, the "type" MUST revert to NIL as well. If the client STORES NIL to the "value" metadata, and explicitly specifies a non-NIL value for the "type" metadata, the server MUST issue a tagged BAD response.

If a server receives a request to STORE a DEFAULT to an attribute's "value" metadata, the "type" MUST revert to DEFAULT as well. If the client STORES DEFAULT to the "value" metadata, and explicitly specifies a non-DEFAULT value for the "type" metadata, the server MUST issue a tagged BAD response.

3.3. Examples

```
Example: C: A001 Store ("/user/rob/people/kelly"
      "people.name" "Joe Kelly"
      "people.description" ("value" "<bold>richtext</bold>"
                           "type" "text/enriched")
      "people.icon.bin" ("value" {1024+}
                        <icon data> "type" "image/png")
S: A001 OK "Store completed"
```

(where <icon data> stands for the 1024 bytes of data that make up the image/png object being stored).

4. Formal Syntax

The following syntax specification uses the augmented Backus-Naur Form (BNF) notation as specified in [\[ABNF\]](#). This uses the ABNF core rules as specified in [Appendix A](#) of the ABNF specification [\[ABNF\]](#).

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

"metadata-type" describes the format of the data which may be stored as "type" metadata.

```
metadata-type ::= type "/" subtype *(";" SPACE parameter)
               ;; type, subtype, and parameter as defined in \[MIME-IMB\]
               ;; free insertion of linear-white-space is not permitted.
```


5. Security Considerations

Clients SHOULD NOT automatically launch potentially unsafe helper applications to view data.

6. Copyright

Copyright (C) The Internet Society 1998. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

7. References

[ABNF] Crocker, D., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", [RFC 2234](#), November 1997.

<url:ftp://ds.internic.net/rfc/rfc2234.txt>

[ACAP] Newman, C., and Myers, J., "Application Configuration Access Protocol", [RFC 2244](#), November 1997.

<url:ftp://ds.internic.net/rfc/rfc2244.txt>

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

<url:ftp://ds.internic.net/rfc/rfc2119.txt>

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

<url:ftp://ds.internic.net/rfc/rfc2045.txt>

8. Author's Address

Robert H. Earhart
Carnegie Mellon
5000 Forbes Ave.
Pittsburgh PA, 15213-3890

Email: earhart+@cmu.edu

Expires September 1998

