

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 27, 2008

A. Melnikov
Isode Ltd
P. Coates
Sun Microsystems
January 24, 2008

Discovery of CONVERT parameters
draft-melnikov-lemonade-convert-discovery-00.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

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 inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on July 27, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).

Abstract

This is a companion document to the Lemonade CONVERT ([draft-ietf-lemonade-convert-XX.txt](#)) extension. It summarizes various proposals for CONVERT MIME type and conversion parameter discovery.

Internet-Draft

CONVERT discovery

January 2008

Table of Contents

1.	Requirements notation	3
2.	Discovery of available conversions and controlling default conversions	3
2.1.	Client preferences regarding default conversions:	
	MEDIACAPS Command	3
2.2.	Discovery of available conversions	4
2.2.1.	GETMETADATA	4
2.2.2.	CONVERSIONS command	6
3.	IANA Considerations	7
4.	Security Considerations	7
5.	Acknowledgments	7
6.	Normative References	8
	Authors' Addresses	8
	Intellectual Property and Copyright Statements	9

Internet-Draft

CONVERT discovery

January 2008

1. Requirements notation

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](#)].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange. The five characters [...] means that something has been elided.

[[anchor2: Editorial comments and questions are marked like this.]]

2. Discovery of available conversions and controlling default conversions

2.1. Client preferences regarding default conversions: MEDIACAPS Command

Arguments: list of supported MIME types and corresponding conversion parameters

Responses: none

Result: OK - MEDIACAPS command completed
BAD - unrecognized syntax of an argument

The client list MIME types and corresponding conversion parameters in the order of preference, starting with the most preferred MIME media type(s).

Servers MUST ignore conversion parameters and MIME types that they don't recognize.

If a MEDIACAPS command was issued on a connection and the client has requested to perform the default conversion (see [section 5](#) for more details), the server MUST use one of the MIME types specified by the client in this command as the target MIME type. The server SHOULD use the first MIME type from the ordered list that it supports.

Example:

```
C: A01 MEDIACAPS ("TEXT" "HTML") ("TEXT" "PLAIN" "FORMAT" ("FLOWED"
"FIXED")) ("TEXT" "*" "CHARSET" ("UTF-8" "US-ASCII")) ("IMAGE"
("JPEG" "PNG" "GIF") "PIX-X" "240" "PIX-Y" "320")
```

[[anchor5: The FORMAT conversion parameter is not registered with IANA]]

With such command the client is saying (each parenthesized list converted to a sentence): "I do text/html. I will also do text/plain, preferably with format=flowed, but I can handle format=fixed too. For all text media types I do, I can handle a charset of either UTF-8 or US-ASCII. I can handle image/jpeg, image/png, and, least preferred, image/gif, and my ideal resolution is 240x320."

ABNF for this command is as follows:

```
mediacaps-cmd = "MEDIACAPS" 1*(SP mediacap)
```

```
mediacap = "(" media-type SP media-subtype
          *( SP media-param SP media-param-values ) ")"
```

```
media-type = astring
```

```
media-subtype = DQUOTE "*" DQUOTE /
               astring /
               "(" astring *( SP astring ) ")"
;; "*" means all subtypes for the media-type specified
;; in the command.
;; Otherwise, either the specific subtype or a list of them.
```

```
media-param = astring
```

```
media-param-values = astring / "(" astring *(SP astring) ")"
```

;; Either a single acceptable value or a list of
;; acceptable values.

[2.2.](#) Discovery of available conversions

[[anchor7: Note that only one of the proposals specified in subsections of this section will be standardized.]]

[2.2.1.](#) GETMETADATA

[[anchor9: Proposal # 1]]

To determine which conversions are supported, server annotations are used. For each MIME format (`<type>/<subtype>` [\[MIME-INT\]](#)) that can be converted, an annotation with the name `"/convert/<type>/<subtype>/types"` SHOULD exist. The `"value.shared"` attribute of this annotation contains a semicolon separated list of type/subtype output formats.

The selection of available conversions MAY be adjustable by the

server administrator, and MAY be sensitive to the current user. The selection of available conversions MAY also depend on information about the client obtained through a different mechanism outside the scope of CONVERT (e.g. dynamically through device description mechanisms or when the device was associated to the account).

For each source MIME type that the client is interested in, it SHOULD determine which target conversions are supported by reading the `"value.shared"` attribute.

In addition to the subtype-specific annotations, a special "wildcard" annotation named `"/convert/<type>/@/types"` MAY be used to reference any subtype of `<type>` media type. A client that doesn't find an `"/convert/<type>/<subtype>/types"` annotation SHOULD check the value of the `"/convert/<type>/@/types"` annotation.

Note that names of server annotations are case-sensitive (see [\[METADATA\]](#)). In order to guaranty interoperability, clients and servers MUST use the lowercased version of `<type>` and `<subtype>` when constructing an annotation name described above.

Example: Discover all image conversions

```
C: a GETMETADATA "/convert/image/@/types" value.shared
S: * METADATA "/convert/image/@/types"
    (value.shared "image/jpeg;image/png;image/gif")
S: a OK GETMETADATA complete
```

The above example shows that the server supports one kind of input image transcoding, from image/jpeg to three different outputs: JPEG, PNG, and GIF.

For a given conversion, optional transcoding parameters MAY be present. These are mapped into the "value.shared" attribute in the "/convert/<srctype>/<srcsubtype>/<desttype>/<destsubtype>/params" annotation. A client wishing to use a conversion parameter SHOULD check if the server will accept it by reading the "value.shared" attribute.

Example: Discover optional parameters for image/jpeg -> image/gif.

```
C: a GETMETADATA /convert/image/jpeg/image/gif/params
    "value.shared"
S: * METADATA /convert/image/jpeg/image/gif/params
    ("value.shared" "pix-x;pix-y")
S: a OK GETMETADATA complete
```

The above example shows that to convert from image/jpeg to image/gif,

the transcoding supports the following types of optional parameters: pix-x (width), pix-y (height).

As with conversion types, some "wildcarding" is permitted. Thus if the same parameters are allowed for all conversions to image/gif, then the server can store the one metadata value "/convert/@/@/image/gif/parameters".

A client MAY use these values to check whether or not a desired conversion is possible, or it might, for example, present the parameters as a GUI preferences pane for the user to customize.

If the client is going to check which conversion parameters are available, it MUST read the "value.shared" attribute from the following annotations in the following order:

```
"/convert/<srctype>/<srcsubtype>/<desttype>/<destsubtype>/params"  
"/convert/<srctype>/@/<desttype>/<destsubtype>/params"  
"/convert/@/@/<desttype>/<destsubtype>/params"
```

The client MUST use the "value.shared" attribute value from the first existing annotation in the list specified above.

[2.2.2.](#) CONVERSIONS command

```
[[anchor11: Proposal # 2]]
```

Arguments: source MIME type
 target MIME type

Responses: untagged responses: CONVERSION

Result: OK - CONVERSIONS command completed
 BAD - unrecognized syntax of an argument,
 unexpected extra argument, missing argument, etc.

The first parameter to the CONVERSIONS command is a source MIME type, the second parameter is the target MIME type. Both parameters are partially (e.g. "text/*") or completely ("*") wildcardable.

Conversions matching the source/target pair and their associated conversion parameters are returned in untagged CONVERSIONS responses. If source/target doesn't match any conversion supported by the server, no CONVERSIONS response is returned.

Examples:

For conversion info from GIF to JPEG (no untagged CONVERT would be

returned if no conversion was possible):

```
C: a CONVERSIONS "image/gif" "image/jpeg"  
S: * CONVERSION "image/gif" "image/jpeg" ("width" "height"  
      "depth" "interlaced")  
S: a OK CONVERSIONS completed
```

For conversion info from GIF to anything:

```
C: b CONVERSIONS "image/gif" *
S: * CONVERSION "image/gif" "image/jpeg" ("width" "height"
    "depth" "interlaced")
S: * CONVERSION "image/gif" "image/png" (...)
[...]
S: b OK CONVERSIONS completed
```

For conversion of anything to JPEG:

```
C: c CONVERSIONS * "image/jpeg"
S: * CONVERSION "image/gif" "image/jpeg" ("width" "height"
    "depth" "interlaced")
S: * CONVERSION "image/png" "image/jpeg" (...)
[...]
S: c OK CONVERSIONS completed
```

For conversions from all image formats to all text formats (maybe via OCR?):

```
C: d CONVERSIONS "image/*" "text/*"
S: d OK CONVERSIONS completed
```

[[anchor12: ABNF is missing for this proposal.]]

[3.](#) IANA Considerations

TBD if needed.

[4.](#) Security Considerations

[[anchor15: TBD]]

[5.](#) Acknowledgments

Authors would also like to thank Dave Cridland for the MEDIACAPS command proposal and Dan Karp for the CONVERSIONS command proposal.

[6.](#) Normative References

- [ABNF] Crocker, D., Ed. and P. Overell, Ed., "Augmented BNF for Syntax Specifications: ABNF", [RFC 4234](#), October 2005.
- [METADATA] Daboo, C., "IMAP METADATA Extension", [draft-daboo-imap-annotatemore](#) (work in progress), December 2007.
- [MIME-IMT] Freed, N. and N. Borenstein, "MIME (Multipurpose Internet Mail Extensions) Part Two: Media Types", [RFC 2046](#), November 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", [RFC 3501](#), March 2003.

Authors' Addresses

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

Email: Alexey.Melnikov@isode.com

Peter Coates
Sun Microsystems
185 Falcon Drive
Whitehorse, YT Y1A 6T2
Canada

Email: peter.coates@Sun.COM

Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM 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.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

