

MMUSIC Working Group  
Internet-Draft  
Expires: April 23, 2006

J. Ott  
Helsinki University of Technology  
K. Loos  
Uni Bremen TZI  
October 20, 2005

**Using HTTP for IMG Transport**  
**draft-ott-mmusic-img-http-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/1id-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 April 23, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

Abstract

The IMG framework document defines a set of abstract operations to distribute and retrieve Internet Media Guides (IMG). This document specifies the mapping of the abstract operations QUERY and RESOLVE to HTTP which includes mapping IMG URIs are parameters to HTTP URIs. IMG envelopes are used to encapsulate IMG contents.

This is an initial strawman intended to provoke discussion.

Table of Contents

- [1. Introduction . . . . .](#) [3](#)
- [2. Terminology . . . . .](#) [3](#)
- [3. Mapping of IMG Operations . . . . .](#) [4](#)
  - [3.1. IMG to HTTP URI mapping . . . . .](#) [4](#)
  - [3.2. Requests, Responses, and Errors . . . . .](#) [5](#)
  - [3.3. Contents . . . . .](#) [6](#)
- [4. Processing Rules . . . . .](#) [7](#)
  - [4.1. Forming Requests . . . . .](#) [7](#)
  - [4.2. Processing Requests and Forming Responses . . . . .](#) [8](#)
  - [4.3. Processing Responses . . . . .](#) [9](#)
- [5. HTTP Intermediaries . . . . .](#) [10](#)
- [6. Examples . . . . .](#) [10](#)
- [7. Security Considerations . . . . .](#) [10](#)
- [8. Open Issues . . . . .](#) [10](#)
- [9. References . . . . .](#) [11](#)
  - [9.1. Normative References . . . . .](#) [11](#)
  - [9.2. Informational References . . . . .](#) [11](#)
- [Authors' Addresses . . . . .](#) [12](#)
- [Intellectual Property and Copyright Statements . . . . .](#) [13](#)



## **1. Introduction**

Internet Media Guides (IMGs) are used to encapsulate and communicate metadata about (ephemeral) contents available via arbitrary networks, e.g., TV and multimedia streams, downloadable contents, or other services. The IMG framework [4] defines a set of abstract operations for large-scale multicast distribution ("IMG ANNOUNCE"), individual subscription and asynchronous (change) notification ("IMG SUBSCRIBE" and "IMG NOTIFY"), and interactive retrieval ("IMG QUERY" and "IMG RESOLVE"). Furthermore, [3] specifies a common encapsulation format for metadata in support of the IMG operations. Finally, an IMG URI scheme is being defined in a companion draft.

For mapping the abstract client-server-style operations "IMG QUERY" and "IMG RESOLVE", HTTP is the natural choice. This document specifies how an IMG QUERY is mapped to an HTTP GET and the IMG RESOLVE to the corresponding HTTP response. Furthermore, some assumptions for a mapping from IMG URI to HTTP URIs are provided, and the encapsulation of IMG contents -- as HTTP resources -- is defined. Rules for client and server side processing are given and guidance is given on how to work with HTTP intermediaries (such as proxies and caches). Finally, an example of an IMG-based interaction is included.

## **2. Terminology**

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [BCP 14](#), [RFC 2119](#) [1] and indicate requirement levels for compliant implementations.

The definitions from [4] and [2] apply. In addition, the following terms are defined:

**IMG resource:** A common name for a full or delta IMG, for an IMG fragment, or an IMG pointer encoded in some specific metadata format. The IMG resource does not include a possible IMG envelope used for encapsulation. An IMG resource is identified by an IMG URI (including the version number!) and may be stored in one or more locations.

**IMG resource instance:** an IMG resource stored in a particular location. An IMG resource instance may be identified by means of an HTTP URI. All instances of the same IMG resource are have equivalent contents.



### 3. Mapping of IMG Operations

The abstract IMG operations QUERY and RESOLVE are used to retrieve (parts of) an IMG identified by an IMG URI that qualifies the IMG resource and its version number, indicates whether the full or a delta IMG is to be retrieved, and may specify some further conditions and parameters. A QUERY operation may be invoked to follow a previously received IMG pointer (e.g., in an ANNOUNCE or a NOTIFY operation or found on a web page), to refresh (selected parts of) a full IMG, or to bootstrap retrieval of metadata, among others.

This section specifies the details of how IMG QUERY and IMG RESOLVE are mapped to HTTP/1.1 [2] that SHOULD be used for carrying IMGs; but HTTP/1.0 is acceptable as well.

HTTP is used "as is". This particularly implies the following: all variants underlying transport protocols (e.g., TCP, TCP/TLS) may be used; transport connections are newly established as needed, they may be torn down or kept open; different authentication schemes (e.g., HTTP Digest) may be applied; etc. This also implies that intermediaries (such as proxies or caches) may be used; some considerations on this case are discussed in [Section 5](#) below.

When mapping the IMG operations to HTTP, the IMG querier becomes the HTTP client and the IMG resolver is equivalent to the HTTP server.

#### 3.1. IMG to HTTP URI mapping

An IMG resource MAY be stored in multiple locations (IMG resource instance), each identified by means of an HTTP URI (in addition, a IMG resource instance may be cached in multiple locations but still be identified by the same HTTP URI). There is no explicit algorithmic mapping rule specified to transform an IMG URI into an HTTP URI. Instead, several procedural approaches may be followed: 1) One or more HTTP URIs may be included in an IMG envelope containing an IMG resource, thereby providing the explicit mapping. 2) Directories/databases may be used to obtain the HTTP URI(s) corresponding to an IMG URI. 3) The HTTP URI may be explicitly available from some other resource (e.g., a web page). For the following discussion, it is assumed that one of these methods succeeds in delivering the sought HTTP URI.

An HTTP URI to identify an IMG instance follows the regular HTTP URI format and defines numerous parameters to reflect specifics of IMG QUERY operations. Generally, the HTTP URI for IMGs has the following format:

```
http[s]://hostport/resource-name?param1&param2&...
```



The schemes signifies whether regular HTTP transport or secure HTTP shall be used. "hostport" defines the transport address of the HTTP server to be contacted to retrieve the IMG resource. The resource-name is local to the HTTP server (and may, e.g., represent a filename). The parameters define how the operation shall be performed. The following parameters are defined:

type: Defines the type of IMG resource requested: "full" requests a full IMG for the indicated version number; "delta" the delta encoding between the "version" and "diffVersion" as defined below; pointer requests an IMG pointer of the IMG resource instance. If the "type" parameter is not present, "full" is assumed as default.  
version: Indicates the version number of the IMG to be retrieved. If the parameter is not present, the latest available version is requested.  
diffVersion: Only valid for if type=="delta". Defines the reference against which the delta shall be generated. If this parameter is absent, the diffVersion is assumed to be the initial version 1.

### **3.2. Requests, Responses, and Errors**

An IMG QUERY operation is represented as an HTTP GET request. The Accept: header SHOULD be included and indicate support for the MIME type "application/img-envelope+xml".

The IMG RESOLVE response to an IMG QUERY is mapped onto an HTTP responses as follows.

200 (OK) response in case the full requested IMG resource is included in the response body.  
206 (Partial Content) in case only parts of the requested IMG resource is included in the response body. In this case, entity headers MUST specify which parts are delivered.

In any case, the HTTP message body MUST contain response entity of an appropriate content type (e.g., "application/img-envelope+xml").

Depending on further request parameters and on the operation of the HTTP server, any other appropriate HTTP response code may be returned. Examples are "304 Not Modified" if an If-Modified-Since: header was present, "401" or "407" if further authorization (e.g., by means of digest authentication is required, and "301" or "302" for a redirection to a different HTTP URI.

HTTP error codes are returned in case of failures. All classes and types of error responses are applicable as appropriate. For example, "404 Not Found" is used to indicate the non-availability of an IMG resource instance on the inquired server, "403 Forbidden" to indicate authorization failure, and "400 Bad Request" to indicate a generally





malformed QUERY.

While HTTP header fields such as "Accept" and "Content-Type" allow for basic content negotiation, this only negotiates the IMG envelope content type but not the type of the encapsulated metadata. It is assumed that an IMG URI will imply a certain metadata format, thereby making a two-stage content negotiation unnecessary.

Editor's note: This simplification seems appropriate and not too harmful. Or should we allow for and explicitly specify that an IMG may have multiple representations?

### **3.3. Contents**

IMGs are intended for processing by machines and therefore responses are usually returned in machine-readable format encoded in the respective metadata format and encapsulated in an IMG envelope. The QUERY request MUST NOT contain a body but the Accept: header MUST list acceptable content types and this list SHOULD include the "application/img-envelope+xml" and, if so, this MUST also be returned by the server (unless an error occurred and no content is returned at all).

In addition, HTTP server processing IMG QUERY requests may be capable of generating a human readable format of an IMG. If the IMG querier seeks to exploit this potential capability it SHOULD indicate a corresponding human-readable format, e.g., text/plain or text/html, in the Accept: header. In this case, the server MAY respond with the human readable format.

If the the IMG querier requests both a machine and a human readable format, the IMG resolver MUST respond with the machine-readable format (application/img-envelope+xml) and MAY decide to also include the human-readable format. If it so decides, the IMG resolver MUST use the MIME type "multipart/alternative" for the message body.

Partial content MAY be requested by the IMG querier using the Range: header. The use of the Range: header is independent of and subordinate to the query type specification using the "type" parameter in the HTTP URI. That is, if a full IMG is requested and the Range: header is given, the range specifies which part of the full IMG resource is to be returned. If a delta IMG is requested, first the delta IMG is formed by the IMG resolver and then the part identified by the Range: header is returned. In both cases, the partial IMG resource MUST be returned using a 206 response and the corresponding rules (section 10.2.7 of [2] apply. The Range: header SHOULD NOT be used with IMG pointers.



Further entity headers, such as Expires: and Last-Modified: SHOULD be used to describe the freshness and the lifespan properties of the returned IMG resource.

#### **4. Processing Rules**

This section specifies the precise processing rules for IMG QUERY and RESOLVE operations via HTTP.

##### **4.1. Forming Requests**

For a QUERY operation, the IMG querier MUST first derive an HTTP URI for the sought IMG resource. If multiple HTTP URIs are found (leading to different IMG resource instances) the IMG querier MAY select one of these according to local preferences. If different weights (for load balancing) and/or priorities are associated with the multiple HTTP URIs the IMG querier SHOULD make its choice taking into these weights and/or the priorities. A specification defining how to obtain HTTP URI from an IMG URI MUST specify whether weights and/or priorities are supported and how these must be interpreted.

The IMG querier SHOULD add the appropriate URI parameters defined in [Section 3.1](#) to fully specify the type of query, the version inquired, etc.

Having obtained the HTTP URI, the IMG querier MUST create an HTTP GET message according to [2].

The IMG querier MUST include an Accept: header and SHOULD add "application/img-envelope+xml" as one of the values. It MAY also add "text/html" or "text/plain". If the IMG querier includes more than a single value in the Accept: header it MUST be prepared to receive content in the form of "multipart/alternative".

The IMG querier MAY include Accept-Encoding: and related headers to further specify how the IMG resource shall be conveyed by the IMG resolver. The IMG querier MAY specify a Range: header if only a subset of the IMG resource is to be requested. The Range: header MUST NOT be used if the query type is "pointer". Also, if the Range: header is used, the IMG querier MUST NOT indicate more than a single content type in the Accept: header.

The IMG querier MAY include further HTTP headers as appropriate.

The IMG querier MUST NOT include a body in the HTTP GET request message.



Once the HTTP GET message is complete, the querier follows regular HTTP procedures to contact the IMG resolver -- optionally via a proxy or cache -- using a transport protocol defined for use with HTTP, following the HTTP rules for connection setup, re-use, and teardown, etc.

#### **4.2. Processing Requests and Forming Responses**

An IMG resolver receiving an IMG query in the form of an HTTP GET message MUST validate the request as per [2]. It MUST perform any authentication/authorization checks and return appropriate error responses (e.g., "401" along with a challenge). Basic authentication SHOULD NOT be used.

Once the IMG QUERY has been authorized, the IMG resolver MUST validate whether it has the requested IMG resource instance available in the proper version(s). If the resource is not available the IMG QUERY MUST be refused: This MAY be done by means of a "404 Not Found" response (if the IMG resolver has no knowledge of the IMG resource) but MAY also lead to a redirection (e.g., "301" or "302" pointing the IMG querier to a different location). In the latter case, the HTTP URI of new location MUST be included in the Location: header of the response.

If the IMG resolver cannot fulfill the request, it MUST generate an appropriate error response: If a delta encoding is requested but the IMG resolver does not support delta encoding, an "501(?)" response MUST be returned. If none of the requested content type (e.g., only text/plain) is supported, a "406" response MUST be generated. If the requested Range: cannot be satisfied, a "416" MUST be returned. Further response codes MAY be used as appropriate and in line with HTTP.

If the request can be fulfilled, the IMG querier SHOULD generate a response entity based upon the requested query type: return the full IMG resource, calculate a delta encoding, or return all known IMG pointers for the requested resource.

The IMG resolver MAY generate one or more response entities depending on the Accept: header:

For the MIME type application/img-envelope+xml, the IMG resolver MUST create an IMG envelop to embed the requested IMG resource. It MUST include the versioning, validity, and (if applicable) delta encoding information in the IMG envelope.

For the MIME types text/plain and text/html, the IMG resolver SHOULD generate a human-readable response from the IMG resource as far as possible. If both are included, the IMG resolver MAY decide whether to generate none, one (and which one), or two



response entities. An IMG envelope MUST NOT be used for this content type.

If only a single entity is generated in response to the IMG QUERY, this entity MUST be placed in the message body of the HTTP response and the Content-Type: header MUST be set accordingly. If both machine- and human-readable responses are generated, the IMG resolver MUST combine the two entities into a single one by means of the multipart/alternative MIME type which then MUST be placed in the message body of the HTTP response. The IMG resolver MUST include a Content-Length: header.

The HTTP response message -- that constitutes the IMG RESOLVE -- MUST then be returned to the IMG querier.

#### **4.3. Processing Responses**

An IMG querier receiving an IMG response in the form of an HTTP GET message MUST validate the response as per [2]. If necessary (e.g., for authentication purposes), the IMG querier SHOULD re-issue the QUERY request with appropriate additions (e.g., authentication credentials filled in). Similarly, if a request was rejected because of other parameter settings (e.g., requested content type, range, etc.), the request SHOULD be corrected and re-issued if possible.

If the request was redirected to a different location (indicated by a 3xx response), the IMG querier SHOULD re-issue the (modified) request to the new location.

If the request was successful (indicated by a 2xx response), the IMG querier SHOULD process the message body containing the response entities as discussed in the following. If only a single entity is included in the message body, it SHOULD extract the entity and proceed as defined below. If multiple entities are returned (indicated by a Content-Type: of multipart/alternative) each entity is extracted and processed separately according to its respective content type.

An IMG resource is returned in an IMG envelope and is indicated by the Content-Type: application/img-envelope+xml. The IMG envelope contains the versioning, validity, optional delta encoding, and further information. If no range is indicated in the HTTP response, the received envelope constitutes the entirety of the full or delta IMG resource. In this case, the IMG resource received MUST be forwarded for further (local) IMG processing. Otherwise, the corresponding entity header defines which part of the full or delta IMG resource is included. In this latter case, the IMG querier MUST first reassemble all response parts and, if necessary, issue





additional requests to obtain missing pieces. Only the fully reassembled response MUST be forwarded for further (local) IMG processing. If an IMG querier issues additional requests obtain a complete IMG resource, it MUST do so using the same URI header parameters as in previous requests.

A human-readable variant of an IMG resource -- indicate by the Content-Type: text/plain or text/html -- is returned without an IMG envelope and is not meant for machine processing. The result MAY be rendered by the IMG querier or by some other entity. If both text/plain and text/html entities are received it is up to the IMG querier which one to give preference for rendering.

## **5. HTTP Intermediaries**

TBD.

## **6. Examples**

TBD.

## **7. Security Considerations**

This document specifies access to IMG resource via HTTP and HTTPS. Security considerations concerning HTTP and HTTPS similarly apply. Furthermore, the security considerations of IMGs apply.

## **8. Open Issues**

Presently, this document assumes an XML-based (rather than a MIME-based) envelope. It needs to be extended to cover MIME as well (which appears straightforward).

Need to discuss if (and where) to indicate preferences for metadata format in case multiple representations exist for the same IMG resource (or will this be a different resource then?).

Need to discuss how to deal with Content-Encoding (such as gzip).

Need to define how to indicate which delta encodings are acceptable.

Intermediaries are ffs.

Examples are tbd.



## **9. References**

### **9.1. Normative References**

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] 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.
- [3] Walsh, R., "The IMG Envelope", [draft-walsh-mmusic-img-envelope-03](#) (work in progress), June 2005.

### **9.2. Informational References**

- [4] Walsh, R., Luoma, J., Asaeda, H., Schulzrinne, H., and Y. Nomura, "A Framework for the Usage of Internet Media Guides", [draft-ietf-mmusic-img-framework-08](#) (work in progress), July 2004.



Authors' Addresses

Joerg Ott  
Helsinki University of Technology  
Otakaari 5A  
Espoo 02150  
Finland

Email: [jo@netlab.hut.fi](mailto:jo@netlab.hut.fi)

Kevin Loos  
Uni Bremen TZI  
Bibliothekstrasse 1  
Bremen 28359  
Germany

Email: [logic@tzi.org](mailto:logic@tzi.org)



## Intellectual Property Statement

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](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

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 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.

## Copyright Statement

Copyright (C) The Internet Society (2005). 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.

## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.



