

SML  
Internet-Draft  
Intended status: Standards Track  
Expires: 20 January 2026

H.-J. Happel  
audriga GmbH  
19 July 2025

Structured Email  
draft-ietf-sml-structured-email-04

## Abstract

This document specifies how a machine-readable version of the content of email messages can be added to those messages.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 20 January 2026.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Conventions Used in This Document . . . . .	3
3. Representing structured data . . . . .	3
3.1. Knowledge representation language . . . . .	4
3.2. Vocabularies . . . . .	4
4. Structured data in email messages . . . . .	5
4.1. Designation . . . . .	5
4.2. Placement . . . . .	5
4.2.1. Full representation . . . . .	6
4.2.2. Partial representation . . . . .	6
4.2.3. Non-representation . . . . .	7
4.3. Identifiers . . . . .	7
4.3.1. Using identifiers in structured data . . . . .	7
4.3.2. Using structured data identifiers in text/html . . . . .	8
5. Structured data across email messages . . . . .	8
5.1. Forwarding . . . . .	9
5.2. Replies . . . . .	9
5.3. Error replies . . . . .	11
5.4. Updates . . . . .	13
6. Header fields and message flags . . . . .	13
6.1. Presence of structured data . . . . .	13
7. Examples . . . . .	14
7.1. Full representation . . . . .	14
7.2. Full representation (machine-readable only) . . . . .	14
7.3. Full representation with inline image . . . . .	14
7.4. Partial representation . . . . .	14
7.5. Non-representation . . . . .	15
8. Appendix (Structured Email Signature) . . . . .	15
9. Security and trust . . . . .	16
10. Implementation status . . . . .	16
10.1. Structured Email plugin for Roundcube Webmail . . . . .	16
11. Security considerations . . . . .	16
12. Privacy considerations . . . . .	17
13. IANA Considerations . . . . .	17
14. Informative References . . . . .	17
Author's Address . . . . .	19

## 1. Introduction

Information on websites and in email messages mostly addresses human readers. However, various attempts have been made to make such information - fully or in part - machine-readable, so that tools can assist users in dealing with that information more efficiently.

One widespread approach is the usage of [SchemaOrg] vocabulary, which can be embedded in the HTML markup of websites. It is then crawled by web search engines and used to improve the quality of search result snippets (e.g., by showing ratings, opening hours, or contact information).

Similarly, a number of web shops, hotels, and airlines include Schema.org vocabulary in order receipt email messages sent to customers. This information is extracted and used by some ISPs and open source tools ([SchemaOrgEmail]). However, these implementations differ in many details.

The goal of this specification is to provide a clear and comprehensive specification for this practice and to provide ground for potential future extensions.

## 2. Conventions Used in This Document

The terms "message" and "email message" refer to "electronic mail messages" or "emails" as specified in [RFC5322]. The term "Message User Agent" (MUA) denotes an email client application as per [RFC5598].

The terms "machine-readable data" and "structured data" are used in contrast to "human-readable" messages and denote information expressed "in a structured format (..) which can be consumed by another program using consistent processing logic" [MachineReadable].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 3. Representing structured data

In order to exchange structured data, one needs to chose a formal language and a serialization format. Based on this, vocabularies can be used to establish a shared understanding of structured data among different parties, such as email senders and receivers.

### 3.1. Knowledge representation language

The Resource Description Framework ([RDF]) is a formal language for knowledge representation standardized by the W3C. It is underlying [SchemaOrg] and thus already used for annotating websites and emails. Among the various serializations for RDF, JSON-LD ([JSONLD]) has become the most commonly used serialization used on websites ([WDCStats]).

Hence, structured data in email messages SHOULD be expressed in the JSON-LD serialization of RDF.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/1>

### 3.2. Vocabularies

Using RDF/JSON-LD, users are free to express any kind of information in structured data. For reuse and reference however, it is common to agree upon core concepts/entities and properties for a certain domain. Those are typically collected and shared in so-called vocabularies.

[SchemaOrg] is a widespread vocabulary, which was design for annotating content on websites. A small subset of its concepts is already used by email senders and processed by email providers.

Users that want to add structured data into email message SHOULD use concepts from [SchemaOrg], if they fit their use case. They MAY however use any valid JSON-LD.

There might also be certain vocabularies for email-specific use cases (such as [I-D.happel-sml-structured-vacation-notices-00]), that will be specifically endorsed by the IETF or by respective RFCs.

MUAs may choose freely if and how to use structured data extracted from messages. If they do not explicitly support a certain vocabulary, MUAs may also rely on extensions or passing data to outside applications, similar to the case of "email attachments" (i.e., MIME body parts with content-disposition type attachment [RFC2183]).

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/2>

## 4. Structured data in email messages

This section defines aspects of adding structured data to a MIME message and its interrelation with other body parts.

### 4.1. Designation

This document targets structured data describing the content of an email message itself. Since users may add other arbitrary structured data (e.g., as MIME body parts of type `application/ld+json`) to an email message, we need to define which kinds of structured data are supposed to be representative of the email message content.

For this reason, senders **MUST** set a header field `Content-Purpose` to the value `Machine-readable` on an `application/ld+json` body part, which is meant to provide a machine-readable description of the message content. A MUA **SHOULD NOT** show such body parts as a file attachment in the list of email attachments.

The `Content-Purpose: Machine-readable` header field **MAY** also be set for body parts with other media types than `application/ld+json`. A system **SHOULD** treat such body parts as if their media type would be `application/ld+json` according to this specification, if they can extract JSON-LD data (e.g.: `application/jose`; [RFC7515])

### 4.2. Placement

When aiming to describe human-readable content in a machine-readable way, there may exist three general relations between both types of content in which the machine-readable version of the content may be:

- \* fully representative of the human-readable content
- \* describing only parts of the human-readable content
- \* describing none of the human-readable-content

From the perspective of the machine-readable content, we call those cases "Full representation", "Partial representation" and "Non-representation". Those distinctions matter for MUAs, as they can make choices for the autoproccessing or presentation of messsages and their body parts.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/3>

#### 4.2.1. Full representation

Full representation denotes the case in which structured data describes the entire content of an email message or of a certain body part, in the sense of providing an "\_alternative\_" version of the same information" as in the informal definition of multipart-alternative in [RFC2046].

If a message is sent to a definitely non-human recipient (e.g., an API), application/ld+json should be used as Content-Type in the message header.

If a message is sent to a human recipient, a sender MUST use a multipart/alternative for each body part that is \_fully\_ described by structured data. In this case, the multipart/alternative should contain a text/plain and a text/html version of the content for backwards compatibility, plus the application/ld+json body part containing the structured data representation.

For automated processing, it is important if a receiving MUA can determine if a message is entirely described by structured data. In practice, the majority of messages will contain just one multipart/alternative body part, for which such conclusion is easy to derive.

In this case, if the single multipart/alternative body part contains an application/ld+json representation for which the MUA is able to process the vocabulary or is able to process the structured data otherwise, the MUA SHOULD prefer the application/ld+json representation, unless instructed otherwise by the user.

In case of more complex MIME structures, it is up to the discretion of the MUA how to process or render the message.

Some countries require senders to include legal disclaimers in email messages. In the case of "full representation", a sender MAY include a "structured email signature" as shown in the Appendix either in the "full representation" structure data or in an additional "non-representation" body part.

#### 4.2.2. Partial representation

If structured data is intended to describe only a \_subset\_ of a certain human-readable body part, it MUST be added as a multipart/related entity with the content type application/ld+json.

This multipart/related entity MUST also contain the human textual content of the body part (e.g., text/plain and text/html). Also, any MIME body part referenced from the structured data in the application/ld+json body part, MUST be enclosed in this multipart/related entity.

MUAs SHOULD render such messages as if no application/ld+json would be included. MUAs MAY process the application/ld+json data for providing an enhanced user experience of their resp. the user's choice.

#### 4.2.3. Non-representation

In the case of non-representation, there is no relation between structured data and the human readable content.

This may be useful for special scenarios, such as embedding "preemptive" structured vacation notices as described in [I-D.happel-sml-structured-vacation-notices-00] into email messages.

As in the case of partial representation, MUAs receiving such messages may take according action based on the structured data extracted.

#### 4.3. Identifiers

There are existing use cases for cross-referencing between different parts of a MIME message, for which [RFC2392] defines the cid: and mid: URI schemes.

In a similar fashion, cross-referencing might occur between structured data and other message parts.

##### 4.3.1. Using identifiers in structured data

Most nodes and properties in JSON-LD are identified using IRIs [RFC3987]. Since any [RFC2392] (cid:/mid:) reference forms a valid IRI, those references can be directly used in JSON-LD.

There are two main cases for which cid:-identifiers SHOULD be used in structured data.

First, if structured data references binary content such as images or other files, which already exist as MIME body parts within the same message.

Second, if a `cid:` value is used in a JSON-LD `@id` property, the corresponding JSON-LD node can be considered to describe the MIME body part identified by that `cid:`. This MAY be used to denote that certain structured data is explicitly describing that MIME body part. This MUST NOT be used for the main text/plain or text/html body parts, though.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/4>

#### 4.3.2. Using structured data identifiers in text/html

In the case of "partial representation", a MUA will still primarily display the human readable part of a message (e.g., text/plain or text/html).

It might however be helpful if the MUA is able to determine which parts of human readable text refer to certain structured data - e.g., to offer actions based on structured data directly in the context of the corresponding human-readable content.

For this purpose, the sender may add a HTML "data-id" property ([HTMLData]) to any HTML entity in the text/html body, which references the `@id` property of a JSON-LD node in the structured data.

Besides referencing the corresponding JSON-LD node, a sender might also want to denote if the underlying data is "extensively" described or just mentioned in the human readable representation. For example the New York Times cooking newsletter typically `_features_` few recipes, while mentioning a larger number of recipes, laos referencing their web URL.

For providing an adequate user experience, the MUA should be able to understand which recipies are featured in an email and which are just mentioned.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/5>

#### 5. Structured data across email messages

This sections deals with aspects that go beyond the scope of an individual MIME message.



### 5.1. Forwarding

Forwarding messages including structured data needs to be considered from a privacy perspective, particularly in cases of "non-representation", when the user has no way to determine structured data from the human readable part of the message.

A MUA MUST strip non-representative structured data when a user is forwarding messages to somebody else in her MUA. Note that this does not apply to automated forwarding of messages.

Beyond that, privacy issues also apply to forwarding regular email messages. Improvements of the status quo might hence be considered beyond the specific context of structured email.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/6>

### 5.2. Replies

In order to allow responses to structured email messages, the [SchemaOrg] vocabulary specifies a property called `potentialAction` ([PotentialAction]).

When using the `mailto:` URI ([RFC6068]) in its target property, this indicates that the sender allows to receive a `_structured` email `reply_` under the mentioned address:

```
{
  "@context": "http://schema.org",
  "@type": "EmailMessage",
  "potentialAction": [
    {
      "@type": "ConfirmAction",
      "identifier": "111",
      "name": "Approve Expense",
      "target": "mailto:jane@example.org"
    },
    {
      "@type": "CancelAction",
      "identifier": "222",
      "name": "Disapprove Expense",
      "target": "mailto:jane@example.org"
    }
  ],
  "name": "(Dis)approve with mailto",
  "description": "Approval request for John's $10.13 expense for office supplies"
}
```

In this case, an SML-capable MUA SHOULD offer the option to answer the message with a `_structured email reply_`. Such `_structured email replies_` can be considered as a form of pre-defined response templates suggested by the sender. As shown in the example, multiple options for a structured email reply might exist.

A structured email reply MUST be a structured email message entirely consisting of "full representation" body parts. Structured data MUST contain the corresponding action including all properties. An additional property `actionStatus` MUST be set to `CompletedActionStatus`. The action SHOULD also contain additional properties describing the agent executing the property and the `startTime/endTime`:

```
{
  "@context": "https://schema.org",
  "@type": "ConfirmAction",
  "identifier": "111",
  "startTime": "2025-06-22T00:00",
  "endTime": "2025-06-22T00:00",
  "agent": {
    "@context": "https://schema.org",
    "@type": "Person",
    "name": "Jane Doe",
  },
  "name": "Approve Expense",
  "target": "mailto:jane@example.org",
  "actionStatus": "CompletedActionStatus"
}
```

Accordingly, there can be two different ways of replying to a structured email: `_regular email replies_` such as supported by many MUAs, and machine-readable `_structured email replies_`. MUAs should ensure that both types of reply can be clearly distinguished by end users.

For discussion, see also:

reply: <https://github.com/hhappel/draft-happel-structured-email/issues/7>

MUAs may also need to ensure that certain actions are not triggered multiple times - either within the same MUA or across multiple concurrent MUAs. For this purpose, the `\Answered` flag ([RFC9051]) is not appropriate, as it has an established meaning and implementations for regular, manually authored responses. Therefore, a MUA MUST set a flag `$structuredDataActionSent` if a `potentialAction` has been responded to - either by the user or some other mechanism on behalf of the user.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/11>

### 5.3. Error replies

In general, an original sender may not assume that a structured email has been processed by a recipient. Hence, there will typically be no response or error message returned, if the receiving MUA cannot make sense of a structured email for whatever reason.

This may be slightly different when sending a `_structured email reply_` in response to an initial structured email. In this case, the original sender MAY want to signal an issue with a response received, such as if a contradicting response has already been received, or if a response is formally inconsistent in another way.

In this case, an error replay MAY be returned to the sender of the erroneous response. Such error reply MUST be a structured email message entirely consisting of "full representation" body parts.

This structured data MUST contain

- \* The action which caused an issue with `actionStatus` set to `FailedActionStatus`
- \* In addition, the `identifier` property MUST be present, while all the other initial properties MAY be omitted
- \* Additionally, an error object SHOULD be included which describes the issue in more detail

```
{
  "@context": "https://schema.org",
  "@type": "ConfirmAction",
  "identifier": "111",
  "startTime": "2025-06-22T00:00",
  "endTime": "2025-06-22T00:00",
  "agent": {
    "@context": "https://schema.org",
    "@type": "Person",
    "name": "Jane Doe",
  },
  "name": "Approve Expense",
  "target": "mailto:jane@example.org",
  "actionStatus": "FailedActionStatus",
  "error": {
    "@context": "https://sml.draft.iana.org",
    "@type": "ActionProcessingError",
    "identifier": "333",
    "description": "Parsing error in line 5",
    "startTime": "2025-07-22T00:00",
    "errorType": "SyntaxError"
  }
}
```

The following values are allowed for the `errorType` property:

- \* `SyntaxError` (e.g., invalid JSON)
- \* `VocabularyError` (e.g., unknown action type or missing required property)
- \* `UnknownProcess` (e.g., invalid identifier)
- \* `InvalidState` (e.g., action is already in `CompletedActionStatus`)
- \* `Unauthorized` (e.g., the sender is not permitted to execute this action)
- \* `InternalServerError` (e.g., the recipient is unable to process the action)
- \* `General`: any other error

A recipient **SHOULD** not send an error reply if there is reason to believe that a sender is trying to act maliciously (e.g., trying to brute-force action identifiers).

An original sender of an action who is receiving an error reply **MUST** never send an automated reply to the error reply message to avoid message loops.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/8>

#### 5.4. Updates

In human-readable messages, human language can be used to update or recall information that was conveyed in prior messages. Accordingly, there needs to be a machine-readable mechanism that allows to express the update or recall of structured data.

To update or recall structured data, senders MUST set the SUPERSEDES header field ([RFC4021]) of the "update" message with the message id of the original email message. An "update" message with empty structured data can be used to signal a full recall of previously send structured data.

The processing of an "update" message by the receiving MUA is up to its own discretion, as meaningful action may depend an multiple factos.

MUAs MAY consider:

- \* An update might be triggered by a previous action of the user
- \* Adding the original message id as an identifier property to the structured data to preserve its origin

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/9>

#### 6. Header fields and message flags

This sections presents header fields and IMAP flags which are supposed to support MUAs in dealing with structured email.

##### 6.1. Presence of structured data

In some use cases, MUAs might benefit from information about message details without having to evaluate the full message body.

For example, the \$hasAttachment IMAP flag ([HasAttachment]) was proposed to signal the existence of MIME attachments in a message which otherwise would need to be redetermined based on complex MIME parsing.

The following procedures should apply to structured email.

A sending MUA SHOULD add a header field Structured data if a message contains structured data. The value for this field MUST include only one of the following values (case-insensitive):

- \* Full for full representation

- \* Partial for partial representation
- \* Other for non-representation
- \* Mixed for any combination of the previous cases

The Structured data fields SHOULD additionally include (case-insensitive, comma-separated) the value Action, if a message contains a "potentialAction" a MUA might want to investigate.

Similarly, the IMAP flags \$hasStructuredData and \$hasStructuredDataAction MAY be used, if an inbound message is found to contain structured data, but neither of the aforementioned header fields.

For discussion, see also:

<https://github.com/hhappel/draft-happel-structured-email/issues/10>

## 7. Examples

The following section shows some example MIME hierarchies of email messages containing structured data.

### 7.1. Full representation

```
multipart/alternative/  
├─ text/plain  
├─ text/html  
└─ application/ld+json
```

### 7.2. Full representation (machine-readable only)

```
application/ld+json
```

### 7.3. Full representation with inline image

```
multipart/alternative/  
├─ text/plain  
└─ multipart/related/  
    ├─ multipart/alternative/  
    │   ├─ text/html  
    │   └─ application/ld+json  
    └─ image/png
```

### 7.4. Partial representation

```

multipart/related/
├── multipart/alternative/
│   ├── text/plain
│   └── text/html
└── application/ld+json

```

#### 7.5. Non-representation

```

multipart/mixed/
├── multipart/alternative/
│   ├── text/plain
│   └── text/html
└── application/ld+json

```

### 8. Appendix (Structured Email Signature)

The following snippet of structured data uses the Schema.org publisher property of an EmailMessage.

```

{
  "@context": "https://schema.org/",
  "@type": "EmailMessage",
  "publisher": {
    "@type": "Organization",
    "legalName": "MUSEO NACIONAL DEL PRADO DIFUSIN, S.A.U., S.M.E.",
    "legalAddress": {
      "@type": "PostalAddress",
      "addressLocality": "Madrid, Spain",
      "postalCode": "28014",
      "streetAddress": "Casado del Alisal, 10, bajo B"
    },
    "legalRepresentative" : {
      "@context": "https://schema.org",
      "@type": "Person",
      "name": "Jane Doe",
    },
    "identifier": {
      "@type": "PropertyValue",
      "name": "Registration data in the Company Register",
      "value": "Volume 23578, Entry 1, Section 8, Sheet M-423094, 74 Folio
74"
    },
    "vatID": "A84888056"
  }
}

```

## 9. Security and trust

Email user agents that want to support structured email should follow guidance to ensure trust and security standards. These will be elaborated in a separate specification (see [I-D.draft-happel-structured-email-trust-03]).

## 10. Implementation status

< RFC Editor: before publication please remove this section and the reference to [RFC7942] >

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC7942]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC7942], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

### 10.1. Structured Email plugin for Roundcube Webmail

An open source plugin for the Roundcube Webmail software is developed to serve as an example implementation for this specification ([RC-SML]).

Beyond that, some ISPs and open source tools provide implementation partly compliant with this specification ([SchemaOrgEmail]).

## 11. Security considerations

See section "security and trust".



## 12. Privacy considerations

See section "security and trust".

## 13. IANA Considerations

This document has no IANA actions at this time.

(TBD IMAP flags)

## 14. Informative References

- [HTMLData] WHATWG, "HTML Living Standard: Embedding custom non-visible data with the data-\* attributes",  
<[https://html.spec.whatwg.org/multipage/dom.html#attr-data-\\*](https://html.spec.whatwg.org/multipage/dom.html#attr-data-*)>.
- [HasAttachment]  
IETF imapext WG mailing list, "Registering \$hasAttachment & \$hasNoAttachment",  
<<https://mailarchive.ietf.org/arch/msg/imapext/MVE5eNH0aNIVGUvN1RKtBL8b278/>>.
- [JSONLD] W3C JSON-LD Working Group, "JSON-LD 1.1",  
<<https://www.w3.org/TR/json-ld/>>.
- [MachineReadable]  
NIST, "NIST IR 7511 Rev. 4",  
<[https://csrc.nist.gov/glossary/term/Machine\\_Readable](https://csrc.nist.gov/glossary/term/Machine_Readable)>.
- [PotentialAction]  
W3C Schema.org Community Group, "Schema.org: potentialAction", <<https://schema.org/potentialAction>>.
- [RC-SML] audriga GmbH, "Structured Email plugin for Roundcube Webmail",  
<<https://github.com/audriga/roundcube-structured-email/>>.
- [RDF] W3C RDF Working Group), "RDF 1.1 Concepts and Abstract Syntax", <<https://www.w3.org/TR/rdf11-concepts/>>.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996,  
<<https://www.rfc-editor.org/info/rfc2046>>.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC2183] Troost, R., Dorner, S., and K. Moore, Ed., "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, DOI 10.17487/RFC2183, August 1997, <<https://www.rfc-editor.org/info/rfc2183>>.
- [RFC2392] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators", RFC 2392, DOI 10.17487/RFC2392, August 1998, <<https://www.rfc-editor.org/info/rfc2392>>.
- [RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, DOI 10.17487/RFC3987, January 2005, <<https://www.rfc-editor.org/info/rfc3987>>.
- [RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, DOI 10.17487/RFC4021, March 2005, <<https://www.rfc-editor.org/info/rfc4021>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [RFC5598] Crocker, D., "Internet Mail Architecture", RFC 5598, DOI 10.17487/RFC5598, July 2009, <<https://www.rfc-editor.org/info/rfc5598>>.
- [RFC6068] Duerst, M., Masinter, L., and J. Zawinski, "The 'mailto' URI Scheme", RFC 6068, DOI 10.17487/RFC6068, October 2010, <<https://www.rfc-editor.org/info/rfc6068>>.
- [RFC7515] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <<https://www.rfc-editor.org/info/rfc7515>>.
- [RFC7942] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", BCP 205, RFC 7942, DOI 10.17487/RFC7942, July 2016, <<https://www.rfc-editor.org/info/rfc7942>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[RFC9051] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <<https://www.rfc-editor.org/info/rfc9051>>.

[SchemaOrg] W3C Schema.org Community Group, "Schema.org", <<https://schema.org/>>.

[SchemaOrgEmail] Structured Email, "Schema.org for email", <[https://structured.email/content/related\\_work/frameworks/schema\\_org\\_for\\_email.html](https://structured.email/content/related_work/frameworks/schema_org_for_email.html)>.

[WDCStats] Web Data Commons Project, "Web Data Commons - Microdata, RDFa, JSON-LD, and Microformat Data Sets", <<http://webdatacommons.org/structureddata/#toc3>>.

#### Author's Address

Hans-Joerg Happel  
audriga GmbH  
Email: [happel@audriga.com](mailto:happel@audriga.com)  
URI: <https://www.audriga.com>