CoRE
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

C. Amsüss

M. Richardson
Sandelman Software Works

20 October 2025

# URI-Path abbreviation in CoAP
# draft-ietf-core-uri-path-abbrev-02

## Abstract

Applications built on CoAP face a conflict between the technical need for short message sizes and the interoperability requirement of following BCP190 and thus registering (relatively verbose) well-known URI paths. This document introduces an option that allows expressing well-known paths in as little as two bytes.

## About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-ietf-core-uri-path-abbrev/.

Discussion of this document takes place on the Constrained RESTful Environments Working Group mailing list (mailto:core@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/core/. Subscribe at https://www.ietf.org/mailman/listinfo/core/.

Source for this draft and an issue tracker can be found at https://github.com/core-wg/uri-path-abbrev.

## Status of This Memo

This Internet-Draft will expire on 23 April 2026.



Table of Contents



## 1. Introduction

When building application components on CoAP ([RFC7252]), sending small messages is a general goal in the ecosystem (i.e., constrained environments, where data rates are limited and large packets can lead to packet loss, see [RFC7228]). While CoAP can operate with a wide range of URIs, short path names are therefore favored.

Those short path names need to be discovered, and [RFC7252] and [RFC6690] provide mechanisms for that. Applications that can not discover such paths because they precede a discovery step, such as the discovery itself, setting up a security context ([RFC9528]) or establishing an initial identity ([RFC9148]) can not rely on discovered short paths, and need to use well-known paths. The best practice established in [BCP190] requires applications to use the prefix ".well-known" for their paths, making the combined paths easily longer than the rest of the CoAP message.

This document establishes a CoAP option that allows abbreviating the path component of the request URI through a numeric registry.

### 1.1. Motivating example

The design criteria for [RFC9528] described in Section 2.11 of [I-D.ietf-lake-reqs-04] give a fragmentation limit of 47 bytes CoAP message payload for 6TiSCH and 51 bytes for some parameters (and implementations) of LoRaWAN, and high performance penalties of not fitting in those frames. An EDHOC message 1 on its own carries a minimum of 37 bytes. The 18 bytes of an encoded "/.well-known/edhoc" path push the size over either limit, whereas an equivalent Uri-Path-Abbrev stays well below the limit.

### 1.2. Conventions and Definitions

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.

This document assumes basic familiarity with CoAP ([RFC7252]), in particular its Uri-* options.

## 2. The Uri-Path-Abbrev option

The Uri-Path-Abbrev option (short for "URI path, abbreviated") expresses a request’s URI path in a more compact form.

The Uri-Path-Abbrev value represents a particular path, and is thus equivalent to any number of Uri-Path options. Those paths are typically in a "/.well-known" location as described in [RFC8615]. The numeric option values are coordinated by IANA in the Uri-Path-Abbrev registry established in this document in Section 6.2.

| No. | C | U | N | R | Name | Format | Len. | Default |
|---|---|---|---|---|---|---|---|---|
| CPA13 | x | | | | Uri-Path-Abbrev | uint | 0-4 | (none) |

Table 1: Uri-Path-Abbrev Option Summary (C = Critical, U = Unsafe, N = NoCacheKey, R = Repeatable)

// RFC-Editor: This document uses the CPA (code point allocation)
// convention described in [I-D.bormann-cbor-draft-numbers]. For
// each usage of the term "CPA", please remove the prefix "CPA" from
// the indicated value and replace the residue with the value
// assigned by IANA; perform an analogous substitution for all other
// occurrences of the prefix "CPA" in the document. Finally, please
// remove this note. There is one occurrence of the value 13 encoded
// in a1270d, which, if the number were to change, would need
// updating.

The option is a critical, safe-to-forward, and part of the cache key, and used in CoAP requests. Table 1 summarizes these, extending Table 4 of [RFC7252]). Its OSCORE treatment is as Class E ([RFC8613]).

The option has an integer value from the registry established in Section 6.2.

Apart from the format and repeatability, the option’s properties only deviate from the Uri-Path (for which it stands in) in that this option is safe to forward. This has consequences for the interactions with the Proxy-URI option, but is generally desirable: It allows the option to be used with proxies that do not implement the option.

## 2.1. Server processing

A server receiving this option process it like the equivalent sequence of Uri-Path options.

A server that supports a Uri-Path-Abbrev value MUST also support the equivalent request composed of Uri-Path components.

A server receiving the option with an unknown value MUST treat it as an unprocessable critical option, returning 4.02 Bad Option (or reject the message
// This option may go away if https://github.com/core-wg/corrclar/
// issues/52 (https://github.com/core-wg/corrclar/issues/52) is

// resolved before this document is published.) and MUST NOT return a 4.04 Not Found response, because the equivalent path may be present on the server.

Since clients that use the option are usually aware of the possibility of failure, there is no need to provide a diagnostic payload in the error (as is generally recommended in Section 5.4.1 of [RFC7252]). A machine readable (and, albeit beyond the scope of this document, actionable) response is described in Section 3.1.1 of [RFC9290]: the server can set Content-Format 257 in the response and send the payload a1270d, which is the CBOR encoding for the CoAP problem detail "Unprocessed CoAP option" with the value CPA13.

## 2.2. Client processing

A client can use the option instead of the Uri-Path option if there is a suitable value that can express the requested path.

The option SHOULD only be sent when it is known that the server has support for the concrete value. This knowledge is typically not learned, but follows from other specifications mandating support for it.

A client can also send a value if it is merely likely that the server supports it. (The decision threshold varies by application, but generally it’s closer to 95% than a mere more-likely-than-not). This is called "tentative use".

In that case, the client needs to reliably detect failure of the option, and to fall back to repeating the request with the Uri-Path spelled out, to operate reliably.

There are four possible indications of the option not being supported:

1. A 4.02 Bad Option response.

2. A 5.02 Bad Gateway caused by a proxy that received a RST or lack of response.

3. A RST caused by handling of a Non-confirmable message.

4. Not receiving a response to a Non-confirmable message.

// There is ongoing discussion about whether that behavior is
// desirable at https://github.com/core-wg/corrclar/issues/52
// (https://github.com/core-wg/corrclar/issues/52). In the unlikely

// case that discussion concludes before this document, the 4.02
// outcome might be shown as preferred in here and in server
// processing.

Some of the complexity of detecting lack of server-side support (items 3 and 4) can be avoided by not using the option with Non-confirmable requests in tentative use.

As multicast requests generally do not result in errors being returned, tentative use is not available for multicast requests.

A generic client implementation SHOULD NOT use this option without explicit instructions from a higher layer or the known specification of the numeric value: Conditions for tentative use are generally not met.

### 2.3. Proxy processing

A proxy MAY expand or introduce a Uri-Path-Abbrev before consulting its cache.

It MAY expand a Uri-Path-Abbrev option before forwarding, in particular if it has reason to assume that the option is not understood. Like a generic client, it SHOULD NOT introduce an abbreviation without good reason; and then, it MUST fall back to the expanded form, as to not introduce unexpected errors to the client.

A proxy that knows Uri-Path-Abbrev but not the concrete value SHOULD forward it unmodified, which is the behavior it would apply if it did not know the option. A reason to reject the request instead is when the proxy is tasked with enforcing access control (see Section 5).

When cross-proxying to protocols that can not transport this option (such as HTTP), the proxy needs to expand the path.

### 2.4. Interaction with other options

The option is mutually exclusive with the Uri-Path option. Receiving both options in a single request MUST be treated like the presence of a critical request option that could not be processed (that option being either the Uri-Path-Abbrev option or the conflicting option).

The Uri-Path-Abbrev option MUST NOT be used in combination with the Proxy-Uri option (or the similar Proxy-CRI option (of [I-D.ietf-core-href])) by clients. Proxies that understand Uri-Path-Abbrev and convert Uri-* options into Proxy-Uri MUST expand any Uri-Path-Abbrev if they know the value.

By the (de)composition rules around Proxy-Uri, and because Uri-Path-Abbrev is safe-to-forward, a proxy (being generally unaware of this specification) is allowed to combine the option with Proxy-Uri (or Proxy-CRI) when it combines the Uri-* options. In such a combined message, the Uri-Path segments to which the Uri-Path-Abbrev corresponds are appended to the path as if all components were present as individual options in the request without conflicting. Servers that support both Uri-Path-Abbrev and Proxy-URI/-CRI SHOULD process requests accordingly. (This is not a strict requirement, as there are no known implementations of proxies that actually compose a Proxy-URI/-CRI from individual options, nor is there a reason known why they should).

### 2.5. Choice of the option number

// RFC-Editor: Please remove this section during publication. It is
// valuable only to the working group and designated experts who
// mange the limited resource of option numbers, and stays available
// for future documents that may want to apply similar rationale
// throughout the draft versions.

The option’s desired properties (critical, safe-to-forward, part of the cache key) limit the available number space to patterns ending with 0bXXX01 where XXX is not 111. All options encodable with a short (1+0-byte) delta, i.e. <= 12, are already assigned. Usually, we’d then look at other options this is typically combined with, and if there are none (as is the case here), go for a large value in the next more efficient (1+1-byte) space so that the small-but-quite-not-1+0-byte numbers stay usable as 1+0-byte options when combined with their "favorite pairs". (This was done with the EDHOC option which is usually paired with OSCORE).

However, in this case, there is an extra concern: The option number should also be smaller than Uri-Query, so that processing the options in a linear fashion can happen as it would happen with Uri-Path: The path gets processed, setting up a state machine to process the query. As Uri-Query has option number 15, the value 13 is the only good choice for this option.

## 3. Initial Uri-Path-Abbrev values

This document registers values for the following well-known URIs:

* /.well-known/core

* /.well-known/rd (see [RFC9175])

* /.well-known/edhoc (see [RFC9528])

* For EST ([RFC9148]):

  - /.well-known/est/crts

  - /.well-known/est/sen

  - /.well-known/est/sren

  - /.well-known/est/skg

  - /.well-known/est/skc

  - /.well-known/est/att

  EST does allow using other paths, such as different root resources or arbitrary labels; for those, no abbreviations are supported in this document.

* For [I-D.ietf-anima-constrained-voucher]:

  - /.well-known/brski/es

  - /.well-known/brski/rv

  - /.well-known/brski/vs

Note that the core and rd paths are commonly used with Uri-Query options.

## 4. Implementation Status

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".

* aiocoap https://christian.amsuess.com/tools/aiocoap/ (https://christian.amsuess.com/tools/aiocoap/)

  A general-purpose implementation of CoAP for unconstrained sytems, published under MIT License.

  In its current main branch, the implementation covers the server side of this specification, applying expansion automatically before looking up which resource to serve. For client, all it provides is the option field where to place a number if the application decides it is suitable, relying on the client application to perform the fallback.

  It implements version ietf-core-uri-path-abbrev-01. Implementation experience: Generally straightforward unless one tries to preserve the information whether Uri-Path-Abbrev was used for the server application (but that was probably just a bad idea in the first place).

  Contact information: Christian Ams端ss (author), updated 2025-09-26

* libcoap (preliminary)

  A report on a yet unpublished implementaiton and related tests of ietf-core-uri-path-abbrev-01 in libcoap has been submitted at https://github.com/core-wg/uri-path-abbrev/issues/25 (https://github.com/core-wg/uri-path-abbrev/issues/25) on 2025-10-01.

## 5. Security Considerations

Having alternative expressions for information that is input to policy decisions can be problematic when the mechanism performing the check has a different interpretation of the presented data than the mechanism at time of use. That concern is not new to this document: Both the Proxy-Uri of [RFC7252] and the Proxy-Cri option of [I-D.ietf-core-href] have the same properties in that regard. The appropriate behavior is for policy checkers to reject any request that contains critical options that is not understood; the application protected by the checker may provide the checker with an allow-list of options that it will treat as unchecked input.

# 6. IANA Considerations

## 6.1. CoAP option: Uri-Path-Abbrev

IANA is requested to enter an one option into the CoAP Option Numbers registry in the CoRE Parameters group:

* Number: CPA13
* Name: Uri-Path-Abbrev
* Reference: this document

## 6.2. Uri-Path-Abbrev registry

IANA is requested to establish a new registry in the CoRE parameters group: Values of the first Uri-Path-Abbrev option in a CoAP request correspond to a URI path according to this registry.

The policy for adding any value is IETF Review (as described in [RFC8126]). Change control for the registry follows this document's publication stream. Initial values are given in Table 2.

Entry fields are:

* Option value.

  An non-negative integer that can be expressed in 32 bits, unique within this registry.

  All positive values whose most significant bit of the most significant byte is 1 are reserved.

  The Python invocation python3 -c 'print("reserved" if (250).bit_length() % 8 == 0 else "unreserved")' can be used to quickly test this property for any positive number (250 in this example).

* Expanded path.

  This text is the URI path (starting with /) that the option is expanded to.

* Reference.

  A document that requested the allocation.

Reviewer instructions:

The reviewer is instructed to be frugal with the 128 values that correspond to a single-byte value, focusing on applications that are expected to be useful in different constrained ecosystems.

The expanded path is expected to be well-known paths at the time of writing, but it is up to the reviewers to exceptionally also admit paths that are not well-known.

| Option value | Expanded path | Reference |
|---|---|---|
| 0 | /.well-known/core | Section 3 of this document |
| 1 | /.well-known/rd | Section 3 of this document, and [RFC9176] |
| 2 | /.well-known/edhoc | Section 3 of this document, and [RFC9528] |
| 301 | /.well-known/est/crts | Section 3 of this document, and [RFC9148] |
| 302 | /.well-known/est/sen | Section 3 of this document, and [RFC9148] |
| 303 | /.well-known/est/sren | Section 3 of this document, and [RFC9148] |
| 304 | /.well-known/est/skg | Section 3 of this document, and [RFC9148] |
| 305 | /.well-known/est/skc | Section 3 of this document, and [RFC9148] |
| 306 | /.well-known/est/att | Section 3 of this document, and [RFC9148] |
| 401 | /.well-known/brski/es | Section 3 of this document, and [I-D.ietf-anima-constrained-voucher] |
| 402 | /.well-known/brski/rv | Section 3 of this document, and [I-D.ietf-anima-constrained-voucher] |
| 403 | /.well-known/brski/vs | Section 3 of this document, and [I-D.ietf-anima-constrained-voucher] |

Table 2: Initial values for the Uri-Path-Abbrev registry

# 7. References

## 7.1. Normative References

[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/rfc/rfc2119>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, DOI 10.17487/RFC7252, June 2014, <https://www.rfc-editor.org/rfc/rfc7252>.

[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/rfc/rfc8174>.

## 7.2. Informative References

[BCP190] Best Current Practice 190, <https://www.rfc-editor.org/info/bcp190>. At the time of writing, this BCP comprises the following:

Nottingham, M., "URI Design and Ownership", BCP 190, RFC 8820, DOI 10.17487/RFC8820, June 2020, <https://www.rfc-editor.org/info/rfc8820>.

[I-D.bormann-cbor-draft-numbers] Bormann, C., "Managing CBOR codepoints in Internet-Drafts", Work in Progress, Internet-Draft, draft-bormann-cbor-draft-numbers-06, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-bormann-cbor-draft-numbers-06>.

[I-D.ietf-anima-constrained-voucher] Richardson, M., Van der Stok, P., Kampanakis, P., and E. Dijk, "Constrained Bootstrapping Remote Secure Key Infrastructure (cBRSKI)", Work in Progress, Internet-Draft, draft-ietf-anima-constrained-voucher-29, 18 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-anima-constrained-voucher-29>.

[I-D.ietf-core-href] Bormann, C. and H. Birkholz, "Constrained Resource Identifiers", Work in Progress, Internet-Draft, draft-ietf-core-href-26, 16 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-core-href-26>.

[I-D.ietf-lake-reqs-04]
Vuini, M., Selander, G., Mattsson, J. P., and D. Garcia-Carillo, "Requirements for a Lightweight AKE for OSCORE", Work in Progress, Internet-Draft, draft-ietf-lake-reqs-04, 8 June 2020, <https://datatracker.ietf.org/doc/html/draft-ietf-lake-reqs-04>.

[RFC6690] Shelby, Z., "Constrained RESTful Environments (CoRE) Link Format", RFC 6690, DOI 10.17487/RFC6690, August 2012, <https://www.rfc-editor.org/rfc/rfc6690>.

[RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <https://www.rfc-editor.org/rfc/rfc7228>.

[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/rfc/rfc7942>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz, "Object Security for Constrained RESTful Environments (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019, <https://www.rfc-editor.org/rfc/rfc8613>.

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", RFC 8615, DOI 10.17487/RFC8615, May 2019, <https://www.rfc-editor.org/rfc/rfc8615>.

[RFC8790] Kernen, A. and M. Mohajer, "FETCH and PATCH with Sensor Measurement Lists (SenML)", RFC 8790, DOI 10.17487/RFC8790, June 2020, <https://www.rfc-editor.org/rfc/rfc8790>.

[RFC9148] van der Stok, P., Kampanakis, P., Richardson, M., and S. Raza, "EST-coaps: Enrollment over Secure Transport with the Secure Constrained Application Protocol", RFC 9148, DOI 10.17487/RFC9148, April 2022, <https://www.rfc-editor.org/rfc/rfc9148>.

[RFC9175] Amsss, C., Preu Mattsson, J., and G. Selander, "Constrained Application Protocol (CoAP): Echo, Request-Tag, and Token Processing", RFC 9175, DOI 10.17487/RFC9175, February 2022, <https://www.rfc-editor.org/rfc/rfc9175>.

[RFC9176] Amsss, C., Ed., Shelby, Z., Koster, M., Bormann, C., and P. van der Stok, "Constrained RESTful Environments (CoRE) Resource Directory", RFC 9176, DOI 10.17487/RFC9176, April 2022, <https://www.rfc-editor.org/rfc/rfc9176>.

[RFC9290] Fossati, T. and C. Bormann, "Concise Problem Details for Constrained Application Protocol (CoAP) APIs", RFC 9290, DOI 10.17487/RFC9290, October 2022, <https://www.rfc-editor.org/rfc/rfc9290>.

[RFC9528] Selander, G., Preu Mattsson, J., and F. Palombini, "Ephemeral Diffie-Hellman Over COSE (EDHOC)", RFC 9528, DOI 10.17487/RFC9528, March 2024, <https://www.rfc-editor.org/rfc/rfc9528>.

## Appendix A. Further development

Several possible further directions are anticipated in this document, but not specified at this point in time; they are left for further documents that update and thus compatibly extend this document.

In any case, server implementations that treat unknown option values or repeated Uri-Path-Abbrev options like any other critical unprocessable option (i.e., by responding with 4.02 Bad Option) support the transition to such an extension.

* Some values of the option might admit repetition of the option. A document that updates this document and the Uri-Path-Abbrev registry will need to specify how later occurrences of the option extend the series of equivalent Uri-Path options from the first value, or defer some of that decision to that first value’s entry.

* The mechanism of expanding one option into another option might be expressed using the terminology of SCHC.

  Such a generalization is not aimed for in this document; authors of any future document providing such a framework are encouraged to provide an equivalent but machine-readable explanation of the mechanism specified here.

* The registry for Uri-Path-Abbrev values is set up such that first values can not have the most significant bit of the first byte set.

  This allows future documents to reuse the option for any CBOR expressions, e.g. the path component of a CRI [I-D.ietf-core-href]. Note that those CBOR structures can only use the major types 4 to 7 for the top-level item, but that includes all containers (arrays, maps and tags).

  Senders and recipients of this option do not need to concern themselves with that extension mechanism unless they implement it: As the first value is compared to known registry entries, any CBOR item contained in it will simply not match any known value. Should the working group decide not to use that extension point, the registry's policy can be relaxed to also allow values with that leading bit set.

* A future document may update this document to allow registering values that are allowed to use together with Uri-Path values (but at the time of writing, no examples are known by which such a design could be properly vetted). In particular, that update weakens the "MUST" in Section 2.4.

* This option is designed to stand in for the Uri-Path option alone, not for any other option; this simplifies its interaction rules.

  In particular, application authors who seek to express Uri-Query options in a more concise or easier to process way are advised to avail themselves of the FETCH method introduced in [RFC8790].

## Appendix B. Change log

Since ietf-core-uri-path-abbrev-01: Processing WGA review input and cleanup.

* Using Uri-Path-Abbrev without knowing it will be supported now has a term ("tentative use") and was reworded.

* The "SHOULD" to support fallback mode was lifted to "SHOULD only be [used when the server is known to support it]", with the recovery being a factual necessity for reliability in tentative use.

* The fallback detection was extended to account for possible reactions to NONs (namely, RST, not replying at all, and 5.02).

* Use with multicast was limited to non-tentative use.

* Added reference to libcoap work.

* Added pointer to RFC9290 (Problem Details) error messages, discouraging diagnostic payloads.

* Sections on future work were unified in the appendix.

* Appendix on open questions was moved into the issue tracker at https://github.com/core-wg/uri-path-abbrev/issues/ (https://github.com/core-wg/uri-path-abbrev/issues/).

* Cleanup of IANA considerations where -01 previous changes were not accounted for.

* "Choice of option number" spelled out and set up for removal before publication.

* Editorial changes.

Since ietf-core-uri-path-abbrev-00: Processing previous two interims.

* Rename option to Uri-Path-Abbrev.

* Allocate per-resource codes for EST and cBRSKI.

* Allocate code for EDHOC.

* Defer repeated use to future extensions.

* Rearrange content to have dedicated server, client and proxy subsections for option processing.

* Establish that generic clients SHOULD NOT use this without reason.

* More explicit language for proxies, including cross-proxies.

* Add introduction and motivating example.

* Add RFC7942 Implementation Status section.

Since draft-amsuess-core-shopinc-02:

Adopted into WG unmodified as I-D.ietf-core-uri-path-abbrev

Since draft-amsuess-core-shopinc-01: Processing 2025-08-27 interim.

* Document is standards track.

* Change name of the option from Short-Uri-Path to Uri-Path-Abbr.

* Close question of whether use of option 13 is justified (it is).

* Minor editorials.

Since draft-amsuess-core-shopinc-00:

* Switched option type from opaque to uint (retaining the lockout for values that look like CBOR arrays/maps).

* MCR joined as author.

* Added initial values for BRSKI and EST.

* Allow 4.04 responses.

* Add guidance for choosing prefixes and rules.

* Large editorial changes.

## Acknowledgments

This document was created out of discussion with Esko Dijk and Michael Richardson. Carsten Bormann provided useful input on shaping the registry. Jon Shallow provided much input, in particular around gaps in the fallback process.

## Authors' Addresses

Christian Amsss
Austria
Email: christian@amsuess.com

Michael Richardson
Sandelman Software Works
Email: mcr+ietf@sandelman.ca