

# CoRE: Corr-Clar

2025-06-18 CoRE Interim 10

Plan: Work on issues via PRs

[https://github.com/core-wg/corrclar/issues?  
q=is%3Aissue+](https://github.com/core-wg/corrclar/issues?q=is%3Aissue+)

2024-12-18: [draft-ietf-core-corr-clar-01](#)

# ✓✓👁️👁️ pull/47: discussing query parameters

Status: Two small text fixes.

PR #47 updated with a conclusion paragraph.

*In the definitions of URIs and of HTTP, the URI query parameters are not much different from path components.*

*In practice, implementations tend to provide ways to handle query parameters more in terms of additional parameters to a single resource handler that handles a number of related URIs, differing in query parameters only.*

*There appears to be little to be gained by discussing this more in the specifications (outside the present document); however, the duality between these two perspectives on query parameters needs to be kept in mind in discussions.*

→ PR #47 reviews required. Merge. Done.

# 👁️👁️ pull/45: Enable the use of BERT

BERT: Blocks > 1KiB

Defined for RFC8323's reliable transports (TCP, TLS, WS(S))

Clarify:

- no reason to forbid BERT for non-reliable transports
- but want to be able to negotiate its use (how?)

Add to process: "GAP"

- fully specified
- but still cannot be used for undefined contexts
- open for use once context defined

Instead of context: Use between **consenting adults**?  
Additional information sources (discovery, negotiation)?

## Conclusion

- Add GAP category, use in PR #45.
- PR #45 reviews required. Merge. Done.
- Any ideas about negotiation? BERT? Max-Message-Size?

# 👁️ pull/43 Treat Proxy-Scheme as having a default

(Explain how this simplifies forward proxy)

Change!

Handle this with Proxy-Scheme-Number in href?

Otherwise, need to introduce FEATURE:

- Incubate here
- Standardize elsewhere

# 👁️👁️ issues/48: 4.05 vs. NON

Cluster with: #14 Separation message / request-response layers in options processing

## 4.2. Messages Transmitted Reliably [CON/ACK]

[...] A recipient MUST either (a) **acknowledge a Confirmable message with an Acknowledgement message** or (b) reject the message if the recipient lacks context to process the message properly, including situations where the message is Empty, uses a code with a reserved class (1, 6, or 7), or has a message format error.

Rejecting a Confirmable message is effected by sending a matching Reset message and otherwise ignoring it.

## 4.3. Messages Transmitted without Reliability [NON]

[...] ]A recipient MUST reject the message if it lacks context to process the message properly, including the case where the message is Empty, uses a code with a reserved class (1, 6, or 7), or has a message format error. Rejecting a Non-confirmable message MAY involve sending a matching Reset message, and apart from the Reset message the rejected message MUST be silently ignored.

### **5.4.1. Critical/Elective**

**Unrecognized options of class "critical" that occur in a [NON] MUST cause the message to be **rejected** (Section 4.3).**

### **5.8. Method Definitions**

**A request with an unrecognized or unsupported Method Code MUST generate a 4.05 (Method Not Allowed) piggybacked **response**.**

(There are no piggybacked responses for NON, so the need for a piggybacked response can only apply to CON transactions.)

[Besides: bad layer violation...]

Clarify/**correct**: besides rejecting a message, it can be **consistently** responded to with 4.05 or 4.02 independent of whether it is CON or NON.