CAQH CORE Connectivity Rule
Version C3.1.0
May 2020
## Revision History for CAQH CORE Connectivity Rule vC3.1.0

<table>
<thead>
<tr>
<th>Version</th>
<th>Revision</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0.0</td>
<td>Major</td>
<td>Phase IV CAQH CORE 470 Connectivity Rule balloted and approved via CAQH CORE Voting Process.</td>
<td>September 2015</td>
</tr>
</tbody>
</table>
| C3.1.0  | Minor    | • Non-substantive adjustments to support re-organization of operating rules into rule sets organized by business transaction (e.g., Eligibility & Benefits, Claim Status, etc.) rather than phase (e.g., Phase I, II, etc.) as approved by the CAQH CORE Board in 2019.  
• Operating rule naming, versioning and numbering methodologies updated to align with business transaction-based rule sets. | May 2020   |
CAQH Committee on Operating Rules for Information Exchange (CORE)
Connectivity Rule vC3.1.0

Table of Contents

1. Background ............................................................................................................................................ 5
   1.1. Affordable Care Act Mandates ................................................................................................. 6
   1.2. Industry Neutral Standards Addressed in this Rule ............................................................ 7

2. Issues to be Addressed and Business Justification ........................................................................... 7
   2.1. Problem Space ...................................................................................................................... 7
   2.2. CAQH CORE Process in Addressing the Problem Space .................................................... 7
   2.3. CAQH CORE Connectivity Rule vC3.1.0 Builds on Foundation Established by Previous CAQH
        CORE Connectivity Rules ..................................................................................................... 9
   2.3.1. Base Minimum Requirements Specified in the CAQH CORE Connectivity Rule vC1 .......... 9
   2.3.2. CAQH CORE Connectivity Rule v2 Specified Robust, Prescriptive Requirements .......... 10
   2.4. CAQH CORE Connectivity Rule v3 - Key Enhancements relative to CAQH CORE Connectivity
        Rule v2 ................................................................................................................................. 11
       2.4.1. Convergence on a Single Message Envelope Standard .................................................. 11
       2.4.2. Convergence on a Single Submitter Authentication Method ............................................ 11
       2.4.3. Enhancements to Message Interactions ........................................................................... 12
       2.4.4. Improved Support for Security Compliance and Stronger Security ................................. 12
       2.4.5. CAQH CORE Process for Maintaining Processing Mode & Payload Type Specifications. 13
       2.4.6. Backward Compatibility with CAQH CORE Connectivity Rules vC1 and vC2 ............. 13

3. Scope..................................................................................................................................................... 14
   3.1. What the Rule Applies To............................................................................................................. 14
   3.2. Standards Used in this Rule ....................................................................................................... 15
   3.3. When the Rule Applies .............................................................................................................. 15
   3.4. When the Rule Does Not Apply .............................................................................................. 16
   3.5. What the Rule Does Not Require ............................................................................................. 16
   3.6. Outside the Scope of this Rule .................................................................................................. 17
   3.7. CAQH CORE-required Processing Mode and Payload Type Tables ....................................... 17
       3.7.1. CAQH CORE-required Processing Mode Table .............................................................. 17
       3.7.2. CAQH CORE-required Payload Type Table ................................................................. 17
       3.7.3. Maintenance of the CAQH CORE-required Processing Mode and Payload Type Tables 17
   3.8. How This Rule Relates to Previous CAQH CORE Operating Rules ........................................ 18
   3.9. Assumptions ............................................................................................................................. 18

4. Rule........................................................................................................................................................ 18
   4.1. CAQH CORE Message Envelope and Submitter Authentication Requirements ..................... 18
       4.1.1. Message Envelope Requirement ...................................................................................... 19
       4.1.2. Submitter Authentication Requirement ............................................................................. 19
       4.1.3. Specifications for SOAP+WSDL Envelope Standard (normative) .................................. 19
       4.1.4. Real Time and Batch Payload Attachment Handling ....................................................... 36
   4.2. General Specifications Applicable to the SOAP Envelope Method ........................................... 36
       4.2.1. Required Transport Method ............................................................................................ 36
       4.2.2. Request and Response Handling ..................................................................................... 36
       4.2.3. Real Time Requests ........................................................................................................... 36
       4.2.4. Batch Submission ............................................................................................................. 37
       4.2.5. Batch Response Pickup .................................................................................................... 37
       4.2.6. Error Handling ................................................................................................................ 37
       4.2.7. Audit Handling ................................................................................................................. 41
       4.2.8. Tracking of Date and Time and Payload ID .................................................................... 41
       4.2.9. Capacity Plan .................................................................................................................. 41
       4.2.10. Real Time Response, Timeout and Retransmission Requirements .................................... 42
   4.3. Publication of Entity-Specific Connectivity Companion Document ........................................ 42
   4.4. Envelope Metadata Fields, Descriptions, Intended Use and Syntax/Value-Sets ........................ 43
       4.4.1. Message Envelope ........................................................................................................... 43
       4.4.2. Table of CAQH CORE Envelope Metadata .................................................................. 44
       4.4.3. Specification of Processing Mode and Enumeration Payload Type Fields ....................... 48
CAQH Committee on Operating Rules for Information Exchange (CORE)
Connectivity Rule vC3.1.0

5. CAQH CORE Safe Harbor.................................................................................................................... 49
6. Conformance Requirements ............................................................................................................... 49
7. Appendix............................................................................................................................................... 51
   7.1. References ........................................................................................................................................................................... 51
   7.2. Abbreviations and Definitions Used in this Rule ............................................................................. 52
   7.3. Sequence Diagrams ................................................................................................................................................................. 60
   7.3.1. Real Time Interaction ........................................................................................................................................................... 60
   7.3.2. Batch Interactions .............................................................................................................................................................. 64
   7.3.3. Generic Batch Interactions .............................................................................................................................................. 73
1. Background

Each of the CAQH CORE Operating Rules complement each other to encourage feasible industry progress. Continuing to build on the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules, the Affordable Care Act (ACA) Section 1104 has mandated that CAQH CORE Operating Rules should be adopted that include rules around the health care claims and encounter reporting, health care services request for review and response, health plan premium payment, benefit enrollment and maintenance transactions, and attachments to allow the industry to leverage its investment in the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules and apply them to exchanging the following HIPAA mandated transactions:

- ASC X12N 005010X223 Health Care Claim Institutional (837) ASC X12N 005010X222 Health Care Claim Professional (837) and ASC X12N 005010X224 Health Care Claim Dental (837) and their respective errata (collectively hereafter referenced as ASC X12N 837 v5010 Claim)
- ASC X12N 005010X217 Health Care Services Review – Request for Review and Response (278) and associated errata (hereafter referenced as ASC X12N v5010 278 Request and Response and referred to as prior authorization in general)
- ASC X12N 005010X218 Payroll Deducted and Other Group Premium Payment for Insurance Products (820) and associated errata (hereafter referenced as ASC X12N v5010 820)
- ASC X12N 005010X220 Benefit Enrollment and Maintenance (834) and associated errata (hereafter referenced as ASC X12N v5010 834)

The use of the ASC X12N v5010 820 and ASC X12N v5010 834 transactions by the Insurance Exchanges\(^1\) is out of scope for this CAQH CORE Connectivity Rule vC3.1.0.

Note: HHS has not adopted a standard for health claims attachments or indicated what standard(s) it might consider for the transaction, and an effective date for these operating rules is not included in the ACA. Thus, the immediate focus of this CAQH CORE Connectivity Rule vC3.1.0 will not include attachments.

The CAQH CORE Connectivity Rule vC3\(^2\) was developed using a consensus-based approach among industry stakeholders, and is designed to facilitate interoperability, improve utilization of administrative transactions, enhance efficiency and lower the cost of information exchange in healthcare. Therefore, a key goal of this CAQH CORE Connectivity Rule vC3.1.0 is to continue to facilitate the industry’s momentum to increase access to the HIPAA-mandated administrative transactions and to enable all HIPAA-covered entities or their agents\(^3\), business associates, intermediaries, and vendors to build on and extend the connectivity and infrastructure capabilities established in the CAQH CORE Eligibility & Benefits and Claim Status Operating Rules, which were then applied to the CAQH CORE Operating Payment & Remittance Rules.

\(^{1}\) 45 CFR §155.20 Definitions. Exchange means a governmental agency or non-profit entity that meets the applicable standards of this part and makes QHPs available to qualified individuals and/or qualified employers. Unless otherwise identified, this term includes an Exchange serving the individual market for qualified individuals and a SHOP serving the small group market for qualified employers, regardless of whether the Exchange is established and operated by a State (including a regional Exchange or subsidiary Exchange) or by HHS.

\(^{2}\) Formerly the Phase IV CAQH CORE Connectivity Rule.

\(^{3}\) One who agrees and is authorized to act on behalf of another, a principal, to legally bind an individual in particular business transactions with third parties pursuant to an agency relationship. Source: West’s Encyclopedia of American Law, edition 2. Copyright 2008 The Gale Group, Inc. All rights reserved.
An important component of this goal is to further facilitate interoperability by moving the healthcare industry to a single message envelope\(^4\) standard along with a single submitter authentication\(^5\) method as set forth in Section 2.2.2 of the ACA-mandated CAQH CORE Connectivity Rule vC2.2.0.

An ancillary goal of this CAQH CORE Connectivity Rule vC3.1.0 is to reinforce and clarify the “Safe Harbor”\(^6\) established in CAQH CORE Connectivity Rule vC1.1.0 and CAQH CORE Connectivity Rule vC2.2.0 that application vendors, providers and health plans, business associates or other intermediaries can be assured will be supported by any HIPAA-covered entity or its agent. Essentially, all HIPAA-covered entities or their agents must support the connectivity requirements as specified in this rule. Clarification of the “safe harbor” addresses the requirement that when a HIPAA-covered entity or its agent are exchanging the transactions addressed by this rule using any other connectivity method as permitted by the CAQH CORE Safe Harbor, the Processing Mode requirements specified in the CAQH CORE-required Processing Mode Table also apply. (See §5.) However, this rule is not intended to require trading partners to remove existing connections that do not match the rule, nor is it intended to require that all trading partners must use this method for all new connections. CAQH CORE expects that in some technical circumstances, trading partners may agree to use different communication mechanism(s) and/or security requirements than that described by this rule.

### 1.1. Affordable Care Act Mandates

This CAQH CORE Rule is part of a set of rules that addresses requirements in Section 1104 of the Affordable Care Act (ACA). Section 1104 contains an industry mandate for the use of operating rules to support implementation of the HIPAA standards. Using successful, yet voluntary, national industry efforts as a guide, Section 1104 defines operating rules as “the necessary business rules and guidelines for the electronic exchange of information that are not defined by a standard or its implementation specifications.” As such, operating rules build upon existing healthcare transaction standards. The ACA outlines three sets of healthcare industry operating rules to be approved by the Department of Health and Human Services (HHS) and then implemented by the industry.

The third set of ACA-mandated operating rules address the health care claims or equivalent encounter information transactions, enrollment and disenrollment in a health plan, health plan premium payments, claims attachments, and referral certification and authorization.\(^7\) The ACA requires HHS to adopt a set of operating rules for these five transactions by July 2014\(^8\). In a letter dated 09/12/12 to the Chairperson of the National Committee on Vital and Health Statistics (NCVHS),\(^9\) the Secretary of HHS designated CAQH CORE as the operating rule authoring entity for the remaining five HIPAA-mandated electronic transactions.

Section 1104 of the ACA also adds the health claims attachment transaction to the list of electronic healthcare transactions for which the HHS Secretary must adopt a standard under HIPAA. The ACA requires the health claims attachment transaction standard to be adopted by 01/01/14, in a manner ensuring that it is effective by 01/01/16.\(^10\)

---

\(^4\) See §7.2 Abbreviations and Definitions Used in this Rule.

\(^5\) Ibid.

\(^6\) See §5 Safe Harbor and §7.2 Abbreviations and Definitions Used in this Rule.

\(^7\) The first set of operating rules under ACA Section 1104 applies to eligibility and claim status transactions; these operating rules were effective 01/01/13. The second set of operating rules applies to EFT and ERA; these operating rules were effective 01/01/14.

\(^8\) This date is statutory language and statutory language can be changed only by Congress.

\(^9\) 09/12/12 HHS [Letter from the Secretary](#) to the Chairperson of NCVHS.

\(^10\) This date is statutory language and statutory language can be changed only by Congress.
1.2. Industry Neutral Standards Addressed in this Rule

This CAQH CORE Connectivity Rule vC3.1.0 addresses industry neutral transport, transport security, message envelope, and submittor authentication standards as well as CAQH CORE specified message envelope metadata,\textsuperscript{11, 12} for both Real Time and Batch Processing Modes of transmitted transactions, and communications-level errors and acknowledgements. These standards include the public Internet, Hypertext Transport Protocol (HTTP), Secure Sockets Layer (SSL), Transport Layer Security (TLS), SOAP, MTOM, XSD, WSDL, and the X.509 Digital Certificate for submittor authentication.

2. Issues to be Addressed and Business Justification

2.1. Problem Space

Recognizing that the healthcare industry uses multiple connectivity methods for electronic administrative transactions – some based on open standards, others on proprietary approaches – the CAQH Committee on Operating Rules for Information Exchange (CORE\textsuperscript{®}) aimed to fill that gap by formulating connectivity and security rules to support healthcare industry specific transactions. Requirements related to connectivity, infrastructure, e.g., response times, companion guides, system availability, etc., were addressed in multiple transaction-specific operating rules. The CAQH CORE Connectivity Rules vC1\textsuperscript{13} and vC2\textsuperscript{14} specifically addressed the message envelope, corresponding envelope metadata, vocabularies and semantics needed, Real Time and Batch Processing Modes, and the industry’s developing use of the public Internet. However, there were challenges experienced by the industry when implementing CAQH CORE Connectivity Operating Rules vC1 and vC2, which this CAQH CORE Connectivity Rule vC3 addresses, e.g.:

- **Complexity**: Provides a simpler and more prescriptive rule with fewer options (e.g., single envelope standard, and single authentication standard)
- **Transaction Support**: Provides more robust and uniform support for handling transaction payload by requiring MTOM for SOAP (both Real Time and Batch Processing Mode); provides better support for the new set of transactions relative to the previous rules, e.g., by supporting additional message interactions
- **Security**: Improves security by removing Username+Password which is a weak form of B2B authentication, and by requiring the use of only X.509 Client Certificate-based authentication over SSL/TLS, which is a stronger form of authentication. Improves support for FIPS 140-2 compliance for entities requiring such compliance, in terms of transport security and message envelope security

2.2. CAQH CORE Process in Addressing the Problem Space

As part of the development of the CAQH CORE Connectivity Rule vC3 environmental scans as well as extensive business and market analysis were conducted to gain insights into the current industry landscape regarding legislative, market movements and national initiatives. The results of these efforts identified several potential opportunity areas as a focus for the CAQH CORE Connectivity Rule vC3.

\textsuperscript{11} See §7.2 Abbreviations and Definitions Used in this Rule.

\textsuperscript{12} See §4.4 Envelope Metadata Fields, Descriptions, Intended Use and Syntax/Value Sets and §7.2 Abbreviations and Definitions Used in this Rule.

\textsuperscript{13} Formerly the Phase I CAQH CORE Connectivity Rule.

\textsuperscript{14} Formerly the Phase II CAQH CORE Connectivity Rule.
Table 2.2-1 Potential CAQH CORE Connectivity Rule vC3 Opportunities

<table>
<thead>
<tr>
<th>Opportunity Area A: Improving Rule Language/Clarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Opportunity #1: Improve clarity around Real Time and Batch requirements, error handling</td>
</tr>
<tr>
<td>Rule Opportunity #2: Address CAQH CORE Connectivity Rule vC2.2.0 implemener feedback specific to technical issues</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunity Area B: Enhancing Envelope Standards and Metadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Opportunity #3: Expand ongoing payload-agnostic approach for explicitly enumerating Payload Types for transactions newly mandated by ACA</td>
</tr>
<tr>
<td>Rule Opportunity #4: Explore convergence of Envelope Standards</td>
</tr>
<tr>
<td>Rule Opportunity #4A: Explore Suitability of other envelope approaches (e.g., JavaScript Object Notation (JSON))</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunity Area C: Enhancing Reliability and Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Opportunity #5: Reliable and secure handling of attachments</td>
</tr>
<tr>
<td>Rule Opportunity #6: Explore convergence of Authentication Standards</td>
</tr>
<tr>
<td>Rule Opportunity #7: Explore industry-wide policy for uniform use of digital certificates</td>
</tr>
<tr>
<td>Rule Opportunity #8: Explore TLS 1.X as part of base requirement for transport security</td>
</tr>
<tr>
<td>Rule Opportunity #9: Explore enhanced envelope level security (e.g., Signature, SAML Authorization), determining B2B nature of transactions and that some signatures may be applied at the document (payload) level.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunity Area D: Exploring Additional Transport Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Opportunity #10: Explore support for ONC DIRECT as an additional transport option</td>
</tr>
<tr>
<td>Rule Opportunity #11: Explore support for Representational State Transfer (REST) as an additional transport option</td>
</tr>
<tr>
<td>Rule Opportunity #12: Explore support for Secure File Transfer Protocol (SFTP) as an additional transport option</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Opportunity Area E: Specificity Around Message Interaction Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule Opportunity #13: Defining Transaction Specific Message Interaction (e.g., Real Time, Batch) Requirements</td>
</tr>
</tbody>
</table>

To select the opportunities that would provide the best value to the industry CAQH CORE developed an objective approach using a set of 44 business and technical criteria to evaluate and compare the potential rule opportunities identified for CAQH CORE Connectivity Rule vC3, recognizing that all of the CAQH CORE Connectivity Rules are expected to evolve. Some key business and technical criteria among them are that the CAQH CORE Connectivity Rule vC3 will:

- Not create or promote proprietary approaches to electronic interactions/transactions
- Not be based on the least common denominator but rather will encourage feasible progress, promote cost savings, and efficiency
- Address both Batch and Real Time Processing Modes, with a movement towards Real Time (where/when appropriate)
- Be developed using a consensus-based, multi-stakeholder approach
- Builds upon existing standards
- Be focused on Business to Business (B2B) transactions
- Create a base and not a “ceiling”
- Be vendor neutral
- Be built upon HIPAA, and align with other key industry bodies in order to promote interoperability
- Address interest in XML, or other evolving standards where appropriate
- Support the Guiding Principles of HHS’ Nationwide Health Information Network (now the eHealth Information Exchange\(^{15}\))

\(^{15}\) See §7.2 Abbreviations and Definitions Used in this Rule.
Table 2.2-2 shows the results of applying the business and technical criteria to the potential rule opportunities.

| Rule Opportunity #1: Improve clarity around Real Time and Batch requirements, error handling | Rule Opportunity #5: Reliable and secure handling of attachments | Rule Opportunity #4A: Explore suitability of other envelope approaches (e.g., JavaScript Object Notation (JSON)) |
| Rule Opportunity #2: Address CAQH CORE Connectivity Rule vC2.2.0 implementer feedback specific to technical issues | Rule Opportunity #7: Explore industry-wide policy for uniform use of digital certificates | Rule Opportunity #11: Explore support for Representational State Transfer (REST) as an additional transport option |
| Rule Opportunity #3: Expand ongoing payload-agnostic approach for explicitly enumerating Payload Types for transactions newly mandated by ACA | Rule Opportunity #9: Explore enhanced envelope level security (e.g., Signature, SAML Authorization), determining B2B nature of transactions and that some signatures may be applied at the document (payload) level. | Rule Opportunity #12: Explore support for Secure File Transfer Protocol (SFTP) as an additional transport option |
| Rule Opportunity #4: Explore convergence of Envelope Standards | Rule Opportunity #10: Explore support for ONC DIRECT as an additional transport option |
| Rule Opportunity #6: Explore convergence of Authentication Standards |
| Rule Opportunity #8: Explore TLS 1.X as part of base requirement for transport security |
| Rule Opportunity #13: Defining Transaction Specific Message Interaction (e.g., Real Time, Batch) Requirements |

2.3. CAQH CORE Connectivity Rule vC3.1.0 Builds on Foundation Established by Previous CAQH CORE Connectivity Rules

2.3.1. Base Minimum Requirements Specified in the CAQH CORE Connectivity Rule vC1

The CAQH CORE Connectivity Rule vC1.1.0 established the requirement to use the HTTP/S secure transport protocol over the public Internet. It also specified a minimum set of metadata that must be outside the ASC X12N payload (e.g., date/time, payload ID, and other elements), and aspects of connectivity/security such as connectivity response times, acknowledgements and errors. The CAQH CORE Connectivity Rule vC1.1.0 also established the CAQH CORE Connectivity “Safe Harbor” which allows HIPAA-covered entities or their agents to implement other connectivity/security methods in addition to the requirement to support the CAQH CORE Connectivity Rule.
2.3.2. CAQH CORE Connectivity Rule v2 Specified Robust, Prescriptive Requirements

CAQH CORE was aware the CAQH CORE Connectivity Rule vC1.1.0 did not provide the optimum level of specificity for implementations as it was developed as a first step. CORE-certified implementations were based on many types of message enveloping methods: e.g., HTTP POST with name/value pairs, HTTP MIME Multipart, W3C XML Schema and SOAP+WSDL among others. Further, within each of these envelope method implementations, significant variations existed in field names and locations of CAQH CORE Connectivity Rule vC1.1.0 metadata, message envelope structure, submitter authentication methods, routing approaches and security-related information. As a result, such variations among enveloping methods and metadata posed a major challenge for interoperability. Therefore, the CAQH CORE Connectivity Rule vC2.2.0 specified more prescriptive requirements for message envelopes, message envelope metadata, and submitter authentication methods.

2.3.2.1. Two Message Envelope Standards Specified in CAQH CORE Connectivity Rule vC2

Just as paper documents need to be placed in an envelope (container), electronic documents (e.g., eligibility inquiries, electronic claims, etc.) must be placed into a container for electronic transmission from the sender to the receiver. These electronic containers, called message envelopes, must also include the critical information needed to identify the sender, receiver, and other information essential for ensuring the electronic documents in the message envelope are delivered to the intended recipient securely and reliably. For message envelopes the terms for the various pieces of information required are called Message Envelope Metadata specifying the fields and their corresponding values within the message envelope that describe the documents (message payload). A message envelope consists of a well-defined structure for organizing and formatting the message envelope metadata, which also includes other information, such as date, time, unique identifiers for each message envelope to enable reliable tracking and auditing.

The CAQH CORE Connectivity Rule vC2.2.0 further facilitated interoperability by requiring the use of two message envelope standards that were shown to meet the agreed upon CAQH CORE Connectivity Rule vC2 criteria, have significant installed base in the healthcare industry, and perform well under real world transaction loads. These two envelope standards were HTTP MIME Multipart and SOAP + WSDL.

Since both these standards have significant merits, the advantages and challenges of having a single envelope standard versus both of these envelope standards as part of the CAQH CORE Connectivity Rule vC2 was debated. The major advantage of a rule based on a single envelope standard is that it would be more definitive and facilitate better interoperability. However, having just one standard would require implementers of the other envelope standard (i.e., the one that was not chosen) to modify their implementations to be compliant with the CAQH CORE Connectivity Rule vC2.2.0. Since both standards met the criteria and had large installed bases, CAQH CORE determined that convergence on a single standard in CAQH CORE Connectivity Rule vC2 would create a barrier to adoption of the CAQH CORE Connectivity Rule vC2.2.0 by a large segment of the industry.

2.3.2.2. CAQH CORE Connectivity Rule vC2 Specified Two Submitter Authentication Methods

HIPAA Security regulations\footnote{68 FR 8376, Feb. 20, 2003, as amended at 74 FR 42767, Aug. 24, 2009; 78 FR 5693, Jan. 25, 2013.} at 45 CFR §164.304 Definitions define “authentication as the corroboration that a person [entity] is the one claimed” and further identifies that the “Technical safeguards are the technology and the policy and procedures for its use that protect electronic protected health information and control access to it.” O’Reilly\footnote{D. Russell and G.T. Gangemi Sr., "Computer Security Basics", O’Reilly & Associates, Inc., 1992.} goes on to further describe authentication as “The process of proving that a subject (e.g., a user or a system) is what the subject claims to be. Authentication is a measure used to verify the eligibility of a subject and the ability of that subject to access certain information. It protects against the fraudulent use of a system or the fraudulent transmission of information. There are
three classic ways to authenticate oneself: something you know, something you have, and something you are.”

Thus, it is essential to validate a particular entity’s identity for granting access to sensitive data or functionalities contained within the system. One of the most common authentication methods used in general today is a Username+Password. Digital certificates are another commonly used method and are considered to “provide the most secure means of authenticating identities.”\(^{18}\) Each authentication method has advantages and disadvantages in terms of security, usability, and breadth of support. Password-based authentication methods, however, do not provide strong security.

Organizations that receive and process (or relay) requests (i.e., as a server) generally enforce a specific authentication method to control access to their resources. Supporting this authentication method is a credential issuance and management scheme defined by an organizational policy. The complexity of supporting two such policies and credential management mechanisms is high at the entity where submitter authentication is enforced (server), but is relatively low at the submitter (client). For this reason, the CAQH CORE Connectivity Rule vC2.2.0 required only server-side implementations to support one of two submitter authentication methods:

- Username+Password
- X.509 Digital Certificate

2.4. CAQH CORE Connectivity Rule v3 - Key Enhancements relative to CAQH CORE Connectivity Rule v2

To address the problems described in §2.1 and to advance the vision for future CAQH CORE Connectivity that was identified in CAQH CORE Connectivity Rules vC1 and vC2, this CAQH CORE Connectivity Rule vC3 has key enhancements by converging on single envelope and authentication standard, improving transaction support for the third set of ACA-mandated transactions, and by improving robustness and security. These enhancements are described below.

2.4.1. Convergence on a Single Message Envelope Standard

The CAQH CORE Connectivity Rule vC2 identified convergence to a single envelope standard as a vision for future connectivity rule based on greater industry experience with implementing the two message envelope standards specified in the rule.

After extensive analysis CAQH CORE determined that converging on the use of SOAP+WSDL as the single message envelope standard in this CAQH CORE Connectivity Rule vC3 includes these benefits:

- Limits variations in use of SOAP for real time and batch processing modes by requiring the use of MTOM for both processing modes
- Relatively simple rule change
- Significant ROI through improvements in interoperability
- Simplicity of rule requirements
- Reduction of implementation cost and complexity by having fewer options
- XML based and therefore extensible
- Good tooling support for SOAP in most platforms
- Alignment with clinical initiatives and industry trends

2.4.2. Convergence on a Single Submitter Authentication Method

The CAQH CORE Connectivity Rule vC2 identified convergence to a single authentication standard as a vision for future a connectivity rule based on greater industry experience with implementing the two message authentication standards specified in the rule.

After extensive analysis CAQH CORE determined that converging on the use of the X.509 digital certificate as the single authentication standard in this CAQH CORE Connectivity Rule vC3 includes these benefits:

- Relatively simple rule change
- Significant ROI through improvements in interoperability
- Simplicity of rule requirements
- Reduction of implementation cost and complexity by having fewer options
- X.509 client certificate-based authentication over SSL/TLS is significantly more secure than Username+Password
- Alignment with clinical initiatives and industry trends

2.4.3. Enhancements to Message Interactions

The CAQH CORE Connectivity Rule vC2 defined message interactions for conducting Real Time and Batch interactions. CAQH CORE Connectivity Rule vC3 preserves the Real Time and Batch interactions while adding some message interactions that could be used as generic building blocks for supporting current or future transactions. The message interactions for the third set of ACA-mandated transactions are illustrated in Section 7 using Uniform Markup Language (UML) sequence diagrams, also known simply as sequence diagrams.

A sequence diagram is an interaction diagram used to visualize how a client (submitter) and a server (receiver) operate with one another and in what order for the transactions addressed by this CAQH CORE Connectivity Rule vC3.1.0. Some interactions are scenarios in which the business transaction (message payload) is to be processed in Real Time by the server while other interactions are scenarios in which the business transaction(s) (message payload) are to be processed as a batch after the server has successfully received the batch and the communication session has ended. When an interaction includes multiple client requests and server responses, e.g., a batch of health care claims, each pair of interactions and its corresponding (synchronous) response is shown in the sequence diagram. The UML sequence diagrams in this CAQH CORE Connectivity Rule vC3.1.0 are specific to the HIPAA mandated transactions to which this rule applies.

2.4.4. Improved Support for Security Compliance and Stronger Security

The CAQH CORE Connectivity Rule v2.2.0 requires the implementation of the Secure Sockets Layer (SSL) v3.0 as a minimum while optionally allowing entities to implement the Transport Layer Security (TLS) v1.0 or higher when an entity is required to comply with the National Institute of Standards and Technology (NIST) Federal Information Processing Standards (FIPS 140).

Analysis conducted by CAQH CORE indicated that while SSL v3.0 is commonly used in the industry, some HIPAA-covered entities or their agents (e.g., Federal government trading partners, eHealth Exchange) are required to also comply with the FIPS 140-2, which essentially prohibits the use of SSL v3.0 and TLS v1.0. As per NIST 800-52r1, federal government entities are required to implement TLS 1.1 or higher. Relative to SSL v3.0, TLS 1.1 and to a larger extent, TLS 1.2 has improvements in security for data in transit (e.g., in message integrity, encryption algorithms, and key generation). However, platform and programming support for, and industry experience in implementing TLS 1.1 and TLS 1.2 is limited at this time. Considering this, this CAQH CORE Connectivity Rule vC3 (See §3.2) strikes a balance between the need to accommodate HIPAA-covered entities or their agents that must also comply with FIPS 140-2, while allowing non-government entities to continue using non-FIPS compliant security at the transport security layer as well as at the message envelope security layer.

Further, by allowing the use of TLS 1.1 or higher in lieu of SSL v3.0 for both FIPS 140-2 compliance and for the sake of stronger security, this CAQH CORE Connectivity Rule vC3 (See §3.2) enables a transition path from SSL v3.0 to TLS 1.1 or higher.
2.4.5. CAQH CORE Process for Maintaining Processing Mode & Payload Type Specifications

Processing modes or computing modes are classifications of different types of computer processing, e.g., Batch, Real Time. In the context of CAQH CORE Operating Rules, the concept of processing mode applies to the timeframe within which a receiver of a payload of transactions processes those transactions and returns to the sender a payload of appropriate acknowledgements.

A message payload is the essential data that is being routed between a sender and a receiver during a connectivity session. In the context of this CAQH CORE Operating Rule, a payload could be one or more healthcare claims or referral requests, etc. In order to enable efficient and effective handling of the various kinds of payloads that could be exchanged, a unique “payload type identifier” is assigned to each kind of payload.

As this rule becomes widely adopted and implemented in health care the experience and learning gained from implementers CAQH CORE recognizes there may be a need to modify either the CAQH CORE Processing Mode requirements or the CAQH CORE Payload Types or both in order to be agile and flexible in meeting emerging or new industry needs. To meet this anticipated need to enable review and maintenance of the processing modes for the administrative transactions addressed by this rule and payload type identifiers are specified in a separate companion document to this rule. A process and policy to address the review and maintenance will be developed by CAQH CORE. (See §3.7.3)

The CAQH CORE-required Payload Types Table includes payload type values for all HIPAA mandated ASC X12N v5010 transactions, including those transactions that are addressed in the CAQH CORE Operating Rules for eligibility, claim status, and ERA. While HIPAA-covered entities or their agents are required to use this CAQH CORE Connectivity Rule vC3.1.0 for the exchange of claims, prior authorization, benefit enrollment and maintenance, and health plan premium payment transactions, subject to the Safe Harbor provisions of the CAQH CORE Connectivity Rule vC3.1.0 HIPAA-covered entities or their agents may also use this CAQH CORE Connectivity Rule vC3.1.0 for the exchange of eligibility, claim status and ERA transactions in accordance with the Safe Harbor provision of the CAQH CORE Connectivity Rule vC2.2.0. However, this does not permit any HIPAA-covered entity or its agent to discontinue support for the exchange of the eligibility, claim status and ERA transactions as required in the CAQH CORE Connectivity Rule vC2.2.0. HIPAA-covered entities or their agents may also use this CAQH CORE Connectivity Rule vC3.1.0 for the exchange of ASC X12N transactions not mandated by HIPAA.

2.4.6. Backward Compatibility with CAQH CORE Connectivity Rules vC1 and vC2

CAQH CORE thoroughly examined maintaining backward compatibility with CAQH CORE Connectivity Rules vC1 and vC2 while also evolving the CAQH CORE Connectivity Rule vC3, which is applicable to a different set of administrative transactions. These ACA-mandated CAQH CORE Operating Rules currently remain in effect and cannot be modified by CAQH CORE Connectivity Rule vC3. In general, the concept of backward compatibility in relationship to technical specifications means that implementers of a newer version of a specification will be able to interact and interoperate with implementers of a previous version easily and without major modifications to either version.

In the context of this CAQH CORE Connectivity Rule vC3 backward compatibility means that key requirements to support two message envelope standards and two submitter authentication methods specified in previous versions of CAQH CORE Connectivity Rules would become an impediment to realizing some of the CAQH CORE Connectivity Rule vC3 high priority rule opportunities agreed to by CAQH CORE. (See §2.2 Table 2.2-2.) Since this CAQH CORE Connectivity Rule vC3 is intended to be independent of the current ACA-mandated CAQH CORE Operating Rules and must stand alone on its

---

19 See §7.2 Abbreviations and Definitions Used in this Rule.

20 The first set of operating rules under ACA Section 1104 applies to eligibility and claim status transactions; these operating rules were effective 01/01/13. The second set of operating rules applies to EFT and ERA; these operating rules were effective 01/01/14.
own merits, implementers of those rules are not required to de-implement or otherwise discontinue support for any of those rules.

Further, as HIPAA-covered entities or their agents may also use this CAQH CORE Connectivity Rule vC3 for the exchange of transactions addressed by the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules in accordance with the Safe Harbor provision of the CAQH CORE Connectivity Rule vC2, the improvements made in this CAQH CORE Connectivity Rule vC3 can also benefit those transactions. However, this does not permit any HIPAA-covered entity or its agent to discontinue support for the exchange of transactions addressed in the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules as required in the CAQH CORE Connectivity Rule vC2.

3. Scope

3.1. What the Rule Applies To

The technical scope of this CAQH CORE Connectivity Rule vC3.1.0 can be described in terms of the specific network layers within the Open Systems Interconnection Basic Reference Model\textsuperscript{21} (OSI model). As shown in the diagram below, the scope of this CAQH CORE Connectivity Rule vC3.1.0 is OSI Layers 3 and 4 (Transport and Network layers) and OSI Layers 5 and 6 (Session and Presentation layers, also called Message Encapsulation layers).

As shown in the Figure 3.1.1 above, typically an application file (or Payload) such as ASC X12 or HL7 is created or processed by an application that resides in the Application Layer (Layer 7 in the OSI Model). The Message Encapsulation layer (Layers 5 and 6 in the OSI Model) create a Message Envelope, and handles connectivity and security. The underlying layers (Layers 1 through 4) provide the necessary message transport and the network infrastructure (e.g., TCP/IP is provided at Layer 3).

As shown in Figure 3.1.2 below, the Message Envelope is outside the Message Payload (content), and inside the Transport Protocol envelope. Here, the Transport Protocol Envelope corresponds to OSI Model Layers 3 and 4, Message Envelope corresponds to OSI Model Layers 5 and 6, and Message Payload (content) corresponds to OSI Model Layer 7. The CAQH CORE Connectivity Rule vC1 established the CAQH CORE foundational use of HTTP/S as the transport protocol over the public Internet, hence the transport protocol envelope consists of HTTP headers. Examples of message payload include HIPAA administrative transactions (ASC X12N), HL7 clinical messages, zipped files, etc.

**Figure 3.1.2**

- Public Internet (TCP/IP) - established in CAQH CORE Connectivity Rule vC1
- HTTP over SSL 3.0, or optionally TLS 1.1 or higher. (HIPAA covered entities that must also be FIPS 140-2 compliant or that require stronger transport security may implement TLS 1.1 or higher in lieu of SSL 3.0)
- Message Envelope & Message Metadata (SOAP + MTOM) – CAQH CORE Connectivity Rule vC3
- HIPAA Administrative Transactions (X12), HL7 Clinical Messages, Zipped Files, Personal Health Record, Other Content

### 3.2. Standards Used in this Rule

The following is a list of standards and their versions on which this Rule is based:

- HTTP Version 1.1
- SSL Version 3.0.
  - This does not preclude the optional use of TLS 1.1 (or a higher version) for connectivity with trading partners that require FIPS 140-2 compliance or whose security policies require the enhanced security afforded by TLS 1.1 or higher. Entities that must also be FIPS 140-2 compliant or whose security policies require enhanced security may implement TLS 1.1 or higher in lieu of SSL 3.0.
- SOAP Version 1.2
- WSDL Version 1.1

### 3.3. When the Rule Applies

The CAQH CORE Connectivity Rule vC3.1.0 applies when trading partners are exchanging any transaction specified in the third set of the Affordable Care Act (ACA) §1104 administrative transactions, i.e.:

- ASC X12N v5010 837 Claim

---

22 Hereafter the combination of HTTP and SSL/TLS is referenced as HTTP/S.
CAQH Committee on Operating Rules for Information Exchange (CORE)
Connectivity Rule vC3.1.0

- ASC X12N v5010 278 Request and Response
- ASC X12N v5010 820
- ASC X12N v5010 834

The CAQH CORE Connectivity Rule vC3.1.0 may also be applied to other payload types. Note: some entities may also apply this rule to other ASC X12N administrative transactions. This CAQH CORE Connectivity Rule vC3.1.0 is a Safe Harbor (See §5), and therefore only needs to be used if mutually agreed to by the trading partners. It is expected that in some instances, other or existing mechanisms may be more appropriate methods of connectivity. Further, HIPAA-covered entities or their agents may also use this CAQH CORE Connectivity Rule vC3.1.0 for the exchange of eligibility, claim status and ERA transactions in accordance with the Safe Harbor provision of the CAQH CORE Connectivity Rule vC2.2.0. However, this does not permit any HIPAA-covered entity or its agent to discontinue support for the exchange of transactions addressed in the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules as required in the CAQH CORE Connectivity Rule vC2.2.0.

3.4. When the Rule Does Not Apply

The CAQH CORE Connectivity Rule vC3.1.0 DOES NOT apply in the following scenarios:

- When HIPAA-covered entities or their agents exchange payloads other than
  - ASC X12N v5010 837 Claim
  - ASC X12N v5010 278 Request and Response
  - ASC X12N v5010 820
  - ASC X12N v5010 834

This rule does not address requirements for the use of the ASC X12N v5010 820 and the ASC X12N v5010 834 transactions by the ACA Federal or state Health Information Exchanges (HIX).

This rule is designed to be payload agnostic, and as such it is expected that HIPAA-covered entities or their agents will use this methodology for other payloads as described in §3.3; however, the rule does not require this.

3.5. What the Rule Does Not Require

The CAQH CORE Connectivity Rule vC3.1.0 (See §5):

- DOES NOT require trading partners to discontinue existing connections that do not match the rule.
- DOES NOT require that trading partners must use a CAQH CORE-compliant method for all new connections.
- DOES NOT require that all CAQH CORE trading partners use only one method for all connections.
- DOES NOT require any HIPAA-covered entity or its agent to do business with any trading partner or other HIPAA-covered entity or its agent.

Further, the CAQH CORE Connectivity Rule vC3.1.0 DOES NOT require the following:

- Additional centralized services other than those that are already provided in the Internet (e.g., Domain name and TCP/IP routing services).
- Additional directories or data repositories.
- Additional centralized Public Key Infrastructure (PKI) Certificate Authorities, identity management or authentication servers.
- Use of specific hardware platforms, software or programming languages.
3.6. **Outside the Scope of this Rule**

The following items are outside the scope of this rule:

- The use of the message envelope and metadata defined in this rule for those messages that are sent over TCP/IP connections that are private (e.g., Intranet, leased lines, or VPN).
- Non-TCP/IP protocols such as packet switching (e.g., X.25, SNA, and Frame Relay).
- Submitter Authorization is a local decision at the site that receives a request.
- The list of trusted Certificate Authorities is a decision between trading partners.
- The maximum size of a batch file that is accepted by a Server. The Server implementer may publish its file size limit, if any, in its Connectivity Companion Guide. (See §4.2.6.2)

3.7. **CAQH CORE-required Processing Mode and Payload Type Tables**

This CAQH CORE Connectivity Rule vC3.1.0 is comprised of the complete rule itself, which specifies all rule requirements; and a companion document to the rule, which specifies additional rule requirements addressing CAQH CORE-required Processing Modes and Payload Type Tables. This enables the necessary flexibility to review and maintain the processing modes and payload types based on Federal regulation or Federal notices to the industry impacting the transactions addressed by this rule.

3.7.1. **CAQH CORE-required Processing Mode Table**

The CAQH CORE-required Processing Mode Table (see §4.4.3) specifies the comprehensive and normative processing mode requirements (i.e., Real Time and/or Batch) for the transactions addressed by this rule.

3.7.2. **CAQH CORE-required Payload Type Table**

The CAQH CORE-required Payload Type Table (see §4.4.3) specifies the comprehensive and normative identifiers for the CORE Envelope Metadata Payload Type Element as defined in the Table of CORE Envelope Metadata. (See §4.4.2.)

The Payload Type identifiers specified in the CAQH CORE-required Payload Type Table apply when an entity is exchanging the transactions addressed by this rule in conformance with the requirements specified in §4 and subsections.

3.7.3. **Maintenance of the CAQH CORE-required Processing Mode and Payload Type Tables**

CAQH CORE recognizes that as this rule becomes widely adopted and implemented in healthcare, the experience and learning gained from implementers may indicate a need to modify either the CAQH CORE-required Processing Mode Table or the CAQH CORE-required Payload Type Table or both to meet emerging or new industry needs. Given this anticipated need a process and policy to enable the review and maintenance of these tables specified in the companion document to this rule, COREProcessingModePayloadTypeTables.docx, will be developed by CAQH CORE.

Such review and maintenance of either the CAQH CORE-required Processing Mode Table or the CAQH CORE-required Payload Type Table or both will follow standard CAQH CORE processes for rule revisions. CAQH CORE will develop such a process and policy for the first review of potential revisions of these tables in accordance with CAQH CORE Guiding Principles following the approval of the CAQH CORE Connectivity vC3. The first review may commence

- One year after the passage of a Federal regulation requiring implementation of this CAQH CORE Operating Rule
- Or
- When Federal regulation or Federal notices to the industry impacting the transactions addressed by this rule are published.
Substantive changes necessary to the tables will be reviewed and approved by CAQH CORE as necessary to ensure accurate and timely revision. The impact of any such changes to any CAQH CORE Infrastructure Rules will be considered during the review of potential revisions. CAQH CORE Infrastructure Rules address other requirements for conducting the transactions addressed by this rule, such as response times for Real Time and/or Batch, System Availability, Companion Document flow and format, etc.

3.8. How This Rule Relates to Previous CAQH CORE Operating Rules

The CAQH CORE Connectivity Rule vC1 established the required use of the public Internet. The CAQH CORE Connectivity Rule vC2 extended the CAQH CORE Connectivity Rule vC1 by establishing a Safe Harbor and specifying the connectivity that all HIPAA-covered entities or their agents must implement and support. (See §5) Each of the previous CAQH CORE Connectivity rule requirements has been incorporated into this CAQH CORE Connectivity Rule v3.1.0 except that the MIME Multipart envelope and Username+Password submitter authentication requirements are not retained in this CAQH CORE Connectivity Rule vC3.1.0. The use of MTOM for SOAP Real Time in this rule implies that use of CDATA tags for SOAP Real Time inline payload, or use of Base64 encoding for payloads with non-printable characters are not a requirement in this rule. Further, relative to CAQH CORE Connectivity vC2, the SSL/TLS requirements in this rule have been updated (§3.2) to improve support for FIPS 140-2 compliance.

Since this CAQH CORE Connectivity Rule vC3.1.0 is intended to be independent of both the CAQH CORE Connectivity Rule vC1 and CAQH CORE Connectivity Rule vC2 and must stand alone on its own merits; implementers of those rules are not required to de-implement or otherwise discontinue support for any of these CAQH CORE rules requirements.

While this CAQH CORE Connectivity Rule vC3.1.0 is mandated for the exchange of transactions addressed in the CAQH CORE Prior Authorization, Healthcare Claims, Premium Payment, and Benefit Enrollment Operating Rules it may also be used for the exchange of transactions addressed by the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules in accordance with the Safe Harbor provision of the CAQH CORE Connectivity Rule vC2.2.0. However, this does not permit any HIPAA-covered entity or its agent to discontinue support for the exchange of transactions addressed in the CAQH CORE Eligibility & Benefits, Claim Status, and Payment & Remittance Operating Rules as required in the CAQH CORE Connectivity Rule vC2.2.0.

3.9. Assumptions

The following assumptions apply to this rule:

- Interoperability, utilization and efficiency will improve by having fewer connectivity/security variations and uniform enveloping standards and metadata.
- This rule is based upon a specific set of open standards and the versions of these standards specified in §3.1. As open standards and versions evolve, appropriate version control practices may need to be applied to keep the Rule consistent with industry best practices with regards to standard versions.
- This rule is a component of the larger set of CAQH CORE Operating Rules; as such, all the CAQH CORE Guiding Principles apply to this rule and all other rules.

4. Rule

This section specifies the requirements for transport, message envelope, submitter authentication, envelope metadata and the specifications for SOAP+WSDL. The rationale and business justification for these conformance requirements are described in §2.

4.1. CAQH CORE Message Envelope and Submitter Authentication Requirements

This rule requires HIPAA-covered entities or their agents to support only one set of requirements for message enveloping and one method for submitter authentication in order to reduce variations and enable greater interoperability in the market.)
4.1.1. Message Envelope Requirement

This rule requires the use of SOAP+WSDL (See §4.1.3).

4.1.2. Submitter Authentication Requirement

This rule requires the use of X.509 Client Authentication (mutual authentication) over SSL 3.0 (TLS 1.1 or higher may be used as per the specifications in §3.2).

4.1.3. Specifications for SOAP+WSDL Envelope Standard (normative23)

This section defines the SOAP+WSDL envelope method for CAQH CORE Connectivity Rule vC3.1.0. The XML Schema that is defined below is used within the Web Services Definition Language (WSDL) specification.

Note: The terms SOAP, WSDL, MTOM, Normative and Non-normative are defined in Appendix §7.2: Abbreviations and Definitions used in this Rule.

4.1.3.1. CAQH CORE Connectivity Rule vC3.1.0 XML Schema Specification (normative)

The CAQH CORE Connectivity Rule vC3.1.0 compliant XML Schema Specification file name below is called CORERule4.0.0.xsd, and is available at http://www.caqh.org/sites/default/files/core/wsdl/CORERule4.0.0.xsd. This schema has ten elements, each representing a type of request or response message envelope:

- Real Time Request Schema (Element name is COREEnvelopeRealTimeRequest)
- Real Time Response (Element name is COREEnvelopeRealTimeResponse)
- Batch Submission (Element name is COREEnvelopeBatchSubmission)
- Batch Submission Response (Element name is COREEnvelopeBatchSubmissionResponse)
- Batch Submission Acknowledgement Retrieval Request (Element name is COREEnvelopeBatchSubmissionAckRetrievalRequest)
- Batch Submission Acknowledgement Retrieval Response (Element name is COREEnvelopeBatchSubmissionAckRetrievalResponse)
- Batch Results Retrieval Request (Element name is COREEnvelopeBatchResultsRetrievalRequest)
- Batch Results Retrieval Response (Element name is COREEnvelopeBatchResultsRetrievalResponse)
- Batch Results Acknowledgement Submission (Element name is COREEnvelopeBatchResultsAckSubmission)
- Batch Results Acknowledgement Submission Response (Element name is COREEnvelopeBatchResultsAckSubmissionResponse)

A consequence of the CAQH CORE XML Schema Specification being normative is that any changes to the structure and syntax of the SOAP Body make the implementation non-compliant. Any such implementations must be done under the CAQH CORE Safe Harbor provision.

23 See §7.2 Abbreviations and Definitions used in this Rule for a definition of Normative.
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="COREEnvelopeRealTimeRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ProcessingMode" type="RealTimeMode" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="TimeStamp" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SenderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ReceiverID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CORERuleVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="Payload" type="xs:base64Binary" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeRealTimeResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ProcessingMode" type="RealTimeMode" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="TimeStamp" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SenderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ReceiverID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CORERuleVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="Payload" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xs:element name="ErrorCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ErrorMessage" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeBatchSubmission">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ProcessingMode" type="BatchMode" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadLength" type="xs:int" minOccurs="1" maxOccurs="1"/>
        <xs:element name="TimeStamp" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SenderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ReceiverID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CORERuleVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CheckSum" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="Payload" type="xs:base64Binary" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeBatchSubmissionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ProcessingMode" type="BatchMode" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadLength" type="xs:int" minOccurs="0" maxOccurs="1"/>
        <xs:element name="TimeStamp" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SenderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ReceiverID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CORERuleVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CheckSum" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="Payload" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xs:element name="ErrorCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ErrorMessage" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeBatchSubmissionAckRetrievalRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ProcessingMode" type="BatchMode" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="PayloadLength" type="xs:int" minOccurs="0" maxOccurs="1"/>
        <xs:element name="TimeStamp" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="SenderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ReceiverID" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CORERuleVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="CheckSum" type="xs:string" minOccurs="0" maxOccurs="1"/>
        <xs:element name="Payload" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xs:element name="ErrorCode" type="xs:string" minOccurs="1" maxOccurs="1"/>
        <xs:element name="ErrorMessage" type="xs:string" minOccurs="1" maxOccurs="1"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
<xs:complexType>
  <xs:sequence>
    <xs:element name="PayloadType" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="ProcessingMode" type="BatchMode" minOccurs="1" maxOccurs="1"/>
    <xs:element name="PayloadID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="PayloadLength" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="TimeStamp" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="SenderID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="ReceiverID" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="CORERuleVersion" type="xs:string" minOccurs="1" maxOccurs="1"/>
    <xs:element name="CheckSum" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Payload" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
The CAQH CORE Connectivity Rule vC3.1.0 Web Services Definition Language (WSDL) file below is called CORERule4.0.0.wsdl, and is available at http://www.caqh.org/sites/default/files/core/wsdl/CORERule4.0.0.wsdl. The WSDL below makes use of the XML Schema (CORERule4.0.0.xsd) as specified in §4.1.3.1. Within this WSDL the following types of messages are defined:

- Real Time Request Message (Message name is RealTimeRequestMessage)
- Real Time Response Message (Message name is RealTimeResponseMessage)
- Batch Submission Request Message (Message name is BatchSubmissionMessage)
- Batch Submission Response Message (Message name is BatchSubmissionResponseMessage)
- Batch Submission Acknowledgement Retrieval Request (Message name is BatchSubmissionAckRetrievalRequestMessage)
- Batch Submission Acknowledgement Retrieval Response (Message name is BatchSubmissionAckRetrievalResponseMessage)
- Batch Results Retrieval Request Message (Message name is BatchResultsRetrievalRequestMessage)
• Batch Results Retrieval Response Message (Message name is BatchResultsRetrievalResponseMessage)
• Batch Results Acknowledgement Submission Message (Message name is BatchResultsAckSubmissionMessage)
• Batch Results Acknowledgement Submission Response Message (Message name is BatchResultsAckSubmissionResponseMessage)

Using the above message definitions, the following types of transactions are defined:

• Real Time Transaction (Operation name is RealTimeTransaction)
• Batch Submit Transaction (Operation name is BatchSubmitTransaction)
• Batch Submit Acknowledgement Retrieval Transaction (Operation name is BatchSubmitAckRetrievalTransaction)
• Batch Results Retrieval Transaction (Operation name is BatchResultsRetrievalTransaction)
• Batch Results Acknowledgement Transaction (Operation name is BatchResultsAckSubmitTransaction)
• Generic Batch Submission Transaction (Operation name is GenericBatchSubmissionTransaction)
• Generic Batch Submission Acknowledgment Retrieval Transaction (Operation name is GenericBatchSubmissionAckRetrievalTransaction)
• Generic Batch Retrieval Transaction (Operation name is GenericBatchRetrievalTransaction)
• Generic Batch Receipt Confirmation Transaction (Operation name is GenericBatchReceiptConfirmationTransaction)

The CAQH CORE Connectivity WSDL uses an implicit style of specification, which allows the optional use of additional elements within the SOAP Header. Server entities that require the use of SOAP Header elements must define their use in the entity’s Connectivity Companion Document. Client or Server entities that do not use these SOAP Header elements must ignore them.
The following sections show Request and Response messages using the SOAP envelope, based on the WSDL schemas defined above. The SOAP Real Time Request/Response examples below are non-normative. They are based on the real-world examples provided by CAQH CORE Participating Organizations, but have been updated to use the CAQH CORE-required metadata that is part of CAQH CORE Connectivity Rule vC3.1.0.

4.1.3.3. **Real Time Request Message Structure (non-normative)**

The Real Time Request message structure shown below specifies SOAP 1.2. SOAP Version 1.2 must be implemented by all Servers.

---

24 A non-normative description is informational only. See §7.2 Abbreviations and Definitions Used in this Rule.
This shows the following components:

1. The HTTP Headers are shown colored in blue.

2. The portion of the SOAP envelope colored in green has the remaining metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)

3. The Real Time Payload file (MTOM attachment) is shown colored in orange.

```
POST /CORE/PriorAuthRealTime HTTP/1.1
Host: server_host:server_port
Content-Type: multipart/related; boundary= MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614;
type="application/xop+xml"; start="0.urn:uuid:5117AAE1116EA8B87A1200060184615"; start-info="application/soap+xml";
action="http://www.caqh.org/soap/wSDL/coretransactions/RealTimeTransaction"

--MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:5117AAE1116EA8B87A1200060184615>

<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeRealTimeRequest
      xmlns:ns1="http://www.caqh.org/soap/WSDL/CORERule4.0.0.xsd">
      <PayloadType>X12_278_Request_005010X217E1_2</PayloadType>
      <ProcessingMode>RealTime</ProcessingMode>
      <PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</PayloadID>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>HospitalA</SenderID>
      <ReceiverID>PayerB</ReceiverID>
      <CORERuleVersion>4.0.0</CORERuleVersion>

      <Payload>
        <xop:Include href="cid:1.urn:uuid:5117AAE1116EA8B87A1200060184692" xmlns:xop="http://www.w3.org/2004/08/xop/include" />
      </Payload>
    </ns1:COREEnvelopeRealTimeRequest>
  </soapenv:Body>
</soapenv:Envelope>

--MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <1.urn:uuid:5117AAE1116EA8B87A1200060184692>

<Real Time Request Payload (e.g., a payload of type X12_278_Request_005010X217E1_2) goes here>

--MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614--
```

### 4.1.3.4. Real Time Response Message Structure (non-normative)

The Real Time Response message structure shown below specifies SOAP 1.2. The HTTP Header is shown in blue. The remainder of the request is the SOAP Envelope. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0 (See §4.4)
4.1.3.5. **Batch Submission Message (non-normative)**

The Batch Submission message structure shown below specifies SOAP 1.2, and also uses MTOM (See §7.1) to send the payload file. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)
3. The Batch file (MTOM attachment) is shown colored in orange.

The Batch Payload Submission in a Generic Push interaction (i.e., Step 1 in the sequence diagram shown in §7.3.3.1) uses the same request message as the Batch Submission Request message structure depicted below, with *PayloadType* values based on what is being submitted.

---

25 The Batch Payload Submission in a Generic Push interaction (i.e., Step 1 in the sequence diagram shown in §7.3.3.1) uses the same request message as the Batch Submission Request message structure depicted below, with *PayloadType* values based on what is being submitted.
4.1.3.6. **Batch Submission Response Message (non-normative)**

The Batch Submission Response message structure shown below specifies SOAP 1.2 and MTOM. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)

---

26 The response to Batch Payload submission in a Generic Push interaction (i.e., Step 2 in the sequence diagram in §7.3.3.1) uses the same response message as the Batch Submission Response message structure depicted below, with PayloadType values based on the response to what is being submitted.
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_0B72121B1FEFA9BBDD31200060195339;
type="application/xop+xml"; start=0urn:uuid:0B72121B1FEFA9BBDD31200060195340@apache.org;
start-info="application/soap+xml"; action="http://www.caqh.org/SOAP/WSDL/CORETransactions/BatchSubmitTransactionResponse"

Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:0B72121B1FEFA9BBDD31200060195340@apache.org>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeBatchSubmissionResponse
      xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule4.0.0.xsd">
      <PayloadType>X12_BatchReceiptConfirmation</PayloadType>
      <ProcessingMode>Batch</ProcessingMode>
      <PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</PayloadID>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>PayerB</SenderID>
      <ReceiverID>HospitalA</ReceiverID>
      <CORERuleVersion>4.0.0</CORERuleVersion>
      <ErrorCode>Success</ErrorCode>
      <ErrorMessage></ErrorMessage>
    </ns1:COREEnvelopeBatchSubmissionResponse>
  </soapenv:Body>
</soapenv:Envelope>
--MIMEBoundaryurn_uuid_0B72121B1FEFA9BBDD31200060195339--

**4.1.3.7. Batch Submission Acknowledgement Retrieval Request Message (non-normative)**

The Batch Submission Acknowledgement Retrieval Request message structure shown below specifies SOAP 1.2. The use of MTOM in Batch mode request/response creates multipart MIME even though there is no payload. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)

POST /CORE/PriorAuthBatch HTTP/1.1
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614;
type="application/xop+xml"; start=0urn:uuid:5117AAE1116EA8B87A1200060184615@apache.org;
start-info="application/soap+xml"; action="BatchSubmitAckRetrieveTransaction"

--MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:5117AAE1116EA8B87A1200060184615@apache.org>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeBatchSubmissionAckRetrieveRequest
      xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule4.0.0.xsd">
      <PayloadType>X12_999_RetrieveRequest_005010X231A1</PayloadType>
      <ProcessingMode>Batch</ProcessingMode>
      <PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</PayloadID>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>HospitalA</SenderID>
      <ReceiverID>PayerB</ReceiverID>
      <CORERuleVersion>4.0.0</CORERuleVersion>
      <ErrorID>Success</ErrorID>
      <ErrorMessage></ErrorMessage>
    </ns1:COREEnvelopeBatchSubmissionAckRetrieveRequest>
  </soapenv:Body>
</soapenv:Envelope>
--MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614--
4.1.3.8. Batch Submission Acknowledgement Retrieval Response Message (non-normative)\(^{27}\)

The Batch Submission Acknowledgement Retrieval Response message structure shown below specifies SOAP 1.2 and MTOM. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)
3. The MTOM Attachment is colored in orange.

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: multipart/related; boundary=MIMEBoundaryurn_0B72121B1FEFA9BDD31200060195339

--MIMEBoundaryurn_0B72121B1FEFA9BDD31200060195339
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:0B72121B1FEFA9BDD31200060195340@apache.org>

<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeBatchSubmissionAckRetrievalResponse xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule4.0.0.xsd">
      <PayloadType>X12_999_Response_005010X231A1</PayloadType>
      <ProcessingMode>Batch</ProcessingMode>
      <PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</PayloadID>
      <PayloadLength>1551254</PayloadLength>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>PayerB</SenderID>
      <ReceiverID>HospitalA</ReceiverID>
      <CORERuleVersion>4.0.0</CORERuleVersion>
      <CheckSum>43B8485AB5</CheckSum>
      <Payload>
        <xop:Include
          href="cid:1.urn:uuid:5117AAE1116EA8B87A1200060184692@apache.org"
          xmlns:xop="http://www.w3.org/2004/08/xop/include" />
      </Payload>
    </ns1:COREEnvelopeBatchSubmissionAckRetrievalResponse>
  </soapenv:Body>
</soapenv:Envelope>

--MIMEBoundaryurn_0B72121B1FEFA9BDD31200060195339
Content-Type: image/jpeg
Content-Transfer-Encoding: binary
Content-ID: <1.urn:uuid:5117AAE1116EA8B87A1200060184692@apache.org>

<999 file>
```

\(^{27}\) Although this example shows an ASC X12 v5010 999 payload type being sent as a response from a server to the client, this could also include an ASC X12 v5010 TA1. Alternatively, the server may elect to send only an ASC X12 v5010 TA1 without any functional group.
4.1.3.9. **Batch Results Retrieval Request Message (non-normative)**

The Batch Results Retrieval Request message structure shown below specifies SOAP 1.2. The use of MTOM in Batch Mode request/response creates multipart MIME even though there is no payload (which may be the case for a Batch Retrieval Request). This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)

```xml
POST /CORE/PriorAuthBatch HTTP/1.1
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614;
type="application/xop+xml"; start="0.ururn:uuid:5117AAE1116EA8B87A12000601846158apache.org";
start-info="application/soap+xml"; action="BatchResultsRetrievalTransaction"

--MIMEBoundaryurn_uuid_5117AAE1116EA8B87A1200060184614
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.ururn:uuid:5117AAE1116EA8B87A12000601846158apache.org>

<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeBatchResultsRetrievalRequest
      xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule4.0.0.xsd">
      <PayloadType>X12_278_Request_Batch_Results_005010X217E1_2</PayloadType>
      <ProcessingMode>Batch</ProcessingMode>
      <PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</PayloadID>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>HospitalA</SenderID>
      <ReceiverID>PayerB</ReceiverID>
      <CORERuleVersion>4.0.0</CORERuleVersion>
    </ns1:COREEnvelopeBatchResultsRetrievalRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

4.1.3.10. **Batch Results Retrieval Response Message (non-normative)**

The Batch Results Retrieval Response message structure shown below specifies SOAP 1.2 and MTOM. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)
3. The MTOM Attachment is colored in orange.

---

28 The Batch Payload retrieval within a Generic Pull interaction (i.e., step 1 in the sequence diagram in §7.3.3.2) uses the same request message as the Batch Results Retrieval Request message structure depicted below, with different PayloadType values based on what is being retrieved.

29 The Batch Payload retrieval within a Generic Pull interaction (i.e., step 1 in the sequence diagram in §7.3.3.2) uses the same request message as the Batch Results Retrieval Request message structure depicted below, with different PayloadType values based on what is being retrieved.
The Batch Results Acknowledgement Submission message structure shown below specifies SOAP 1.2, and also uses MTOM (See §7.2) to send the payload file. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)
3. The Batch file (MTOM attachment) is shown colored in orange.

The acknowledgment submission within a Generic Pull interaction (i.e., step 3 in the sequence diagram in §7.3.3.2) uses the same request message as the Batch Results Acknowledgement Submission message structure depicted below, with different PayloadType values as appropriate.
4.1.3.12. Batch Results Acknowledgement Submission Response Message (non-normative)\textsuperscript{31}

The Batch Results Acknowledgement Submission Response message structure shown below specifies SOAP 1.2 and MTOM. This shows the following:

1. The HTTP Headers are shown colored in blue.
2. The portion of the SOAP envelope colored in green has the metadata that is defined as part of the CAQH CORE Connectivity Rule vC3.1.0. (See §4.4)

\textsuperscript{31} The response to the acknowledgment submission within a Generic Pull interaction (i.e., step 4 in the sequence diagram in §7.3.3.2) uses the same response message as the Batch Results Acknowledgement Submission Response message structure depicted below, with different \textit{PayloadType} values as appropriate.
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: multipart/related; boundary=MIMEBoundaryurn_uuid_0B72121B1FEFA9BDD31200060195339;
type="application/xop+xml"; start="0.urn:uuid:0B72121B1FEFA9BDD31200060195340@apache.org";
start-info="application/soap+xml"; action="BatchResultsAckTransaction"
--MIMEBoundaryurn_uuid_0B72121B1FEFA9BDD31200060195339
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <0.urn:uuid:0B72121B1FEFA9BDD31200060195340@apache.org>

<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeBatchResultsAckSubmissionResponse
      xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule4.0.0.xsd">
      <PayloadType>X12_Response_ConfirmReceiptReceived</PayloadType>
      <ProcessingMode>Batch</ProcessingMode>
      <PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</PayloadID>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>PayerB</SenderID>
      <ReceiverID>HospitalA</ReceiverID>
      <CORERuleVersion>4.0.0</CORERuleVersion>
      <ErrorCode>Success</ErrorCode>
      <ErrorMessage></ErrorMessage>
    </ns1:COREEnvelopeBatchResultsAckSubmissionResponse>
  </soapenv:Body>
</soapenv:Envelope>

Error Message Structure (non-normative)
The Error message structure shown below uses the SOAP Fault specifications within SOAP 1.2. As described in §4.2.4, SOAP Faults must be used to send errors at the SOAP level. The HTTP Headers are shown colored in blue. The remainder of the request is the SOAP Envelope.

HTTP/1.1 500
Content-Length: 2408
Content-Type: application/soap+xml

  <soapenv:Header>
  </soapenv:Header>
  <soapenv:Body>
    <soapenv:Fault>
      <soapenv:Reason>
        <soapenv:Text xml:lang="en">There was an error in the incoming SOAP request</soapenv:Text>
      </soapenv:Reason>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>

Envelope Processing Error Message (non-normative)
The Error message structure shown below illustrates a SOAP-based message that indicates an error has occurred within processing the envelope. The HTTP Headers are shown colored in blue. The remainder of the request is the SOAP Envelope. The envelope structure and metadata that is defined within CAQH CORE Connectivity Rule vC3.1.0 is colored in green.
4.1.4. Real Time and Batch Payload Attachment Handling

Payload must be sent as an MTOM\(^{32}\) encapsulated object.

4.2. General Specifications Applicable to the SOAP Envelope Method

4.2.1. Required Transport Method

HIPAA-covered entities or their agents must be able to implement HTTP/S Version 1.1 over the public Internet as a transport method for the third set of the ACA Section 1104 required transactions as specified in §1 of this rule. Receivers (servers) must be able to perform the role of an HTTP/S server, while senders (clients) must be able to perform the role of an HTTP/S client. By using the HTTP/S protocol, all information exchanged between the sender (client) and receiver (server) is encrypted by a session-level private key negotiated at connection time.

4.2.2. Request and Response Handling

HTTP/S supports a request-response message pattern, meaning that the sender (client) submits a message and then waits for a response from the message receiver (server). This works well for the submission of ASC X12N messages in both Batch and Real Time Processing Modes, but the response message from the receiver (server) is different depending on whether the sender’s (client’s) message is a Real Time request, Batch submission, or Batch request pickup.

4.2.3. Real Time Requests

Real Time requests must include a single inquiry or submission as specified in the transaction’s corresponding CAQH CORE Infrastructure Rule. In this processing mode the response from the message receiver (server) is either

- A transport or message envelope error response (See §4.2.6)

  Or

- The corresponding ASC X12 message response (e.g. ASC X12C 005010X231A1 Implementation Acknowledgement for Health Care Insurance (999) [hereafter ASC X12C v5010 999]

  Or

\(^{32}\) MTOM is defined in Appendix §7.2: Definitions and Abbreviations used in this Rule.
4.2.4. Batch Submission

Batch requests are sent in the same way as Real Time requests. In this Processing Mode the response will differ because message receivers (servers) are not required to provide a corresponding ASC X12 response in the timeframes specified in the transaction’s corresponding CAQH CORE Infrastructure Rule for Real Time.

For Batch submissions, the response must be only the standard SOAP message indicating whether the request was accepted or rejected. Message receivers (servers) must not respond to a batch submission with an ASC X12 response, such as an ASC X12C v5010 999 in the HTTP response to the batch request, even if their systems’ capabilities allow such a response. All ASC X12 responses must be available for pick up by the message sender (client) in accordance with the respective CAQH CORE Infrastructure Rule for the transaction.

4.2.5. Batch Response Pickup

Batch responses must be picked up after the message receiver (server) has had a chance to process a Batch submission corresponding ASC X12 response in the timeframes specified in the transaction’s corresponding CAQH CORE Infrastructure Rule.

Under this usage pattern, the message sender (client) connects to the message receiver (server) using HTTP/S and sends a SOAP message requesting available files, and the responder then sends back the file(s) in the HTTP/S SOAP response message (payload).

4.2.6. Error Handling

As shown in Figure 4.2.6 below, a submitted request goes through at least four logical layers that process the request. Errors relative to OSI Layers 3 and below are not addressed.

- Processing of HTTP headers (typically handled by a web-server)
- Validating the SOAP Envelope (can be handled by messaging middle-ware or integration brokers)
- Processing the CORE specific metadata located in the SOAP Envelope
- Processing the Payload (e.g., ASC X12, typically handled by application business logic)

Once a request (e.g., ASC X12N v5010 278 Request) is submitted it goes through these four logical layers. At each of these layers, some part of the request is processed. At each layer there can be errors (indicated by the dotted arrows being returned to the request submitter), which may be returned to the request submitter. If there is an error in processing the message at any logical layer, the request does not get passed to the next layer. If no errors are encountered at that layer, the request is passed to the next processing layer. The last logical layer that processes the request is the Payload Processing Layer. Once this layer processes the payload, it returns a response or error (e.g., ASC X12N v5010 278 Response or ASC X12C v5010 999 or ASC X12C TA1).
Figure 4.2.6

Note: In Figure 4.2.6 above, the dotted line arrows indicate error messages being returned to the Submitter if there is a processing error at the corresponding logical processing layer. The straight line arrows indicate the request and response messages.

**4.2.6.1. HTTP Status and Error Codes (Normative, Not Comprehensive)**

The processing and error codes for the HTTP Layer are defined as part of the HTTP specifications [http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html]. The intended use of these status and error codes in processing the requests are specified in Table 4.2.6.1 and are consistent with the HTTP status codes from CAQH CORE Connectivity Rule vC1.1.0.

The status and error codes included in Table 4.2.6.1 only represent a short list of several commonly used status codes in the standard. An exhaustive list of HTTP Status Codes and descriptions are included in the HTTP specification [http://tools.ietf.org/html/rfc2616#section-6.1.1]. This rule requires the use of the appropriate HTTP error or status codes as applicable to the error/status situation. The list of status/error codes below is not intended to constrain the use of standard HTTP status/error codes relative to their original specification. The descriptions below are not intended to override the original definitions but to provide contextual information based on the use of these HTTP Status and Error Codes for CAQH CORE Connectivity error handling.

**Table 4.2.6.1**

<table>
<thead>
<tr>
<th>HTTP Status/Error Codes (Normative, Not Comprehensive)</th>
<th>CAQH CORE Rule Specific Description 34 (Intended Use)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 OK</td>
<td>Success</td>
</tr>
<tr>
<td>202 Accepted</td>
<td>Real Time or Batch file submission has been accepted (but not necessarily processed)</td>
</tr>
<tr>
<td>400 Bad Request</td>
<td>Incorrectly formatted HTTP headers</td>
</tr>
<tr>
<td>403 Forbidden</td>
<td>Access denied</td>
</tr>
</tbody>
</table>

33 An exhaustive list of HTTP Status Codes and definitions are included in section 6.1.1 of the HTTP specification [http://tools.ietf.org/html/rfc2616#section-6.1.1]

4.2.6.2. **SOAP Envelope Validation – SOAP Faults (Normative)**

Errors at the SOAP Envelope validation layer are returned as SOAP faults [http://www.w3.org/TR/soap12-part1/#soapfault]. The full list of enumerated SOAP Faults may be found in the SOAP 1.2 specification. Table 4.2.6.2 provides perspective on two of the errors that are commonly used in relation to the CAQH CORE Rule.

The set of SOAP Faults below is not comprehensive – additional SOAP Faults that comply with the SOAP 1.2 specifications can be used. The descriptions below are not intended to override the original definitions but to provide contextual information based on the use of these SOAP Faults for CAQH CORE Connectivity error handling.

<table>
<thead>
<tr>
<th>SOAP Faults (Normative; Not Comprehensive)</th>
<th>CAQH CORE Rule Specific Description (Intended Use)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>The envelope sent by the sender (client) did not conform to the expected format. In the case of SOAP, this error should be sent as a SOAP fault with “Sender” fault code.</td>
</tr>
<tr>
<td>Receiver</td>
<td>The message could not be processed for reasons attributable to the receiver (server) (e.g., upstream process is not reachable). In the case of SOAP, this error should be sent as a SOAP fault with “Receiver” fault code.</td>
</tr>
</tbody>
</table>

4.2.6.3. **CAQH CORE Connectivity Envelope Metadata Processing Status and Error Codes (Normative, Comprehensive)**

To handle CAQH CORE-compliant envelope processing status and error codes, two fields called `ErrorCode` and `ErrorMessage` are included in the CORE-compliant Envelope. (See §4.4.2) `ErrorMessage` is a free form text field that describes the error (for the purpose of troubleshooting/logging). When an error occurs, `PayloadType` is set to `CoreEnvelopeError`. The set of `ErrorCodes` in this table is normative and comprehensive, which means the use of other error codes is not permitted.

<table>
<thead>
<tr>
<th>CORE-compliant Envelope Processing Status/Error Codes (Normative, Comprehensive)</th>
<th>CAQH CORE Status Code Description (Intended Use)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>Envelope was processed successfully.</td>
</tr>
<tr>
<td><code>&lt;FieldName&gt;Illegal</code></td>
<td>Illegal value provided for <code>&lt;FieldName&gt;</code>. Value provided is not valid based on the metadata constraints defined in the CAQH CORE Connectivity Rule.</td>
</tr>
</tbody>
</table>

---

35 An exhaustive list of HTTP Status Codes and definitions are included in section 6.1.1 of the HTTP specification [http://tools.ietf.org/html/rfc2616#section-6.1.1].
Table 4.2.6.3

<table>
<thead>
<tr>
<th>CORE-compliant Envelope Processing Status/Error Codes (Normative, Comprehensive)</th>
<th>CAQH CORE Status Code Description(^{35}) (Intended Use)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;FieldName&gt;UnSupported</td>
<td>Value is a legal value, but is not supported by the end point receiving the request. Server Connectivity Guide should indicate where to find specific SOAP Operations if multiple URLs are used to support CAQH CORE Connectivity vC3.</td>
</tr>
<tr>
<td>VersionMismatch</td>
<td>The CAQH CORE Rule Version sent is not valid at the receiver (server).</td>
</tr>
<tr>
<td>UnAuthorized</td>
<td>The sender could not be authorized (e.g., using the fields in the metadata, or using the client certificate information).</td>
</tr>
<tr>
<td>NotSupported</td>
<td>A request was received at this server with a valid PayloadType or ProcessingMode but is currently not implemented by this server (e.g., it may be implemented at a different server within this organization)</td>
</tr>
<tr>
<td>ChecksumMismatched</td>
<td>The checksum value computed on the recipient did not match the value that was sent in the envelope.</td>
</tr>
</tbody>
</table>

4.2.6.4. Examples of HTTP Status and Error Codes (non-normative)

The following illustrates the status and error codes that may be returned:

- A SOAP request that has illegal HTTP headers gets a response with HTTP Error Code: “400 Bad Request.”
- A SOAP request with an unauthenticated submitter’s client certificate gets a response with HTTP Error Code: “403 Forbidden.”
- A SOAP request with HTTP headers properly formatted but using the wrong SOAP Version (1.1 instead of 1.2) gets HTTP Status 500.

4.2.6.5. Examples of SOAP Faults (non-normative)

The following illustrates some situations where “Sender” SOAP Faults may be returned:

- Invalid version of SOAP (e.g., SOAP 1.1)
- SOAP envelope does not have a SOAP Body
- SOAP Body does not contain the CAQH CORE Connectivity Elements

The following illustrates some situations where “Receiver” SOAP Faults may be returned:

- Failure to connect to a backend system for processing of the message

4.2.6.6. Examples of CAQH CORE Connectivity Envelope Metadata Processing Error Messages (non-normative)

ErrorMessage field is intended to provide a descriptive text of the error message in free form text, to aid in logging and troubleshooting. It is the responsibility of the implementer to keep this message consistent with the semantics of the ErrorCode, and not in conflict with it. The ErrorMessage must be related to the ErrorCode as defined in the table above. The following illustrates ErrorMessage fields that may be returned:

- For ErrorCode=VersionMismatch, the ErrorMessage could be “Expecting CORERuleVersion=X, Received CORERuleVersion=Y”
• For ErrorCode=SenderIdIllegal, the ErrorMessage could be “SenderID length exceeds maximum allowed length”
• For ErrorCode=TimeStampIllegal, the ErrorMessage could be “Timestamp is missing the time-zone information”
• For ErrorCode=CheckSumIllegal, the ErrorMessage could be “Unknown algorithm”, or “Unknown encoding type”
• For ErrorCode=UnAuthorized, the ErrorMessage could be “Unauthorized Sender – please contact XXX to get proper credentials”.
• For ErrorCode=NotSupported, the ErrorMessage could be “The requested PayloadType is supported at a different URL, please review Connectivity Companion Guide”

4.2.7. Audit Handling
Auditing is a local decision by each trading partner. The CAQH CORE recommended best practice is for each trading partner to audit all the envelope metadata and payload for each transaction.

4.2.8. Tracking of Date and Time and Payload ID
In order to comply with the corresponding transaction’s CAQH CORE Infrastructure Rule, message receivers (servers) will be required to track the times of any received inbound messages, and respond with the outbound message for that Payload ID. In addition, as specified in the CAQH CORE Envelope Metadata Table 4.4.2, message senders (clients) must include the date and time the message was sent in the CORE metadata element Time Stamp.

4.2.9. Capacity Plan

4.2.9.1. Real Time Transactions
A HIPAA-covered entity or its agent must have a capacity plan such that it can receive and process a large number of single concurrent Real Time transactions via an equivalent number of concurrent connections. These single transactions must be received, processed and the appropriate response provided back to the sender (client) within response time requirements specified in the transaction’s corresponding CAQH CORE Infrastructure Rule.

Three major factors affect the specific number of Large Volume of Single Real Time Transactions (See §7.2) capable of being transported and processed within a given CAQH CORE response time frame. They are:

1. The amount of message metadata and message encapsulation structure which is required for each transaction;
2. The characteristics of the message handling software and how concise its design and coding are; and,
3. The architecture of the intervening hardware, software and communication platform.

HIPAA-covered entities or their agents must attest that their capacity planning addresses the above three factors that affect large volume single Real Time processing. HIPAA-covered entities or their agents must also attest that they have the ability to track, on a calendar week basis, any change to their agreed upon volume capacity.

36 See Appendix 7.2: Abbreviations and Definitions used in this Rule for a definition of Large Volume of Single Real time Transactions (Synchronous).
In the circumstances where the transaction volume throughput is exceeded by one of the trading partners, the receiving organization may declare a denial of service event and request a temporary waiver of the applicable CAQH CORE response time rule's performance criteria, and/or other appropriate action.

### 4.2.9.2. Batch Transactions

The HIPAA-covered entity or its agent’s messaging system must have the capability to receive and process large Batch transaction files if the entity supports Batch transactions. These transactions must be received, processed and the appropriate response provided back to the sender (client) within the time specified in the applicable CAQH CORE Rule.

Three major factors that affect the specific number of Large Batch payloads capable of being transported and processed within a given time frame are:

1. The availability and use of capabilities in the messaging protocol which support in-line files, file attachments, and automated integrity assurance routines, etc., together with the quality and characteristics of their implementation;
2. The characteristics of the message handling software and its conciseness of design and coding; and,
3. The architecture of the intervening hardware, software and communication platform.

HIPAA-covered entities or their agents must attest that their capacity planning addresses the above three factors that affect large Batch processing. The maximum number of transaction sets to be included in a large Batch file is determined between trading partners.

### 4.2.10. Real Time Response, Timeout and Retransmission Requirements

Real Time response time must conform to the transaction’s corresponding CAQH CORE Infrastructure Rule requirements.

- If a Real Time response message is not received within the 60 second response period, the submitter’s (client) system should send a duplicate transaction no sooner than 90 seconds after the original attempt was sent.
- If no Real Time response is received after the second attempt, the submitter’s (client) system should submit no more than 5 duplicate transactions within the next 15 minutes.
- If additional attempts result in the same timeout termination, the submitter’s (client) system must notify the submitter to contact the receiver directly to determine if system availability problems exist or if there are known Internet traffic constraints causing the delay.

### 4.3. Publication of Entity-Specific Connectivity Companion Document

Servers must publish detailed specifications in a Connectivity Companion Document on the entity’s public web site. CAQH CORE recommends specifying the following. This list of recommendations is not intended to be either exhaustive or prohibitive as the specific details of a trading partner relationship are outside the scope of the CAQH CORE rules.

- CAQH CORE Rule Version for Connectivity.
- Details on the message format and the supported transactions (e.g., Real Time, Batch transactions).
- Details about the entity’s ASC X12 Interchange; e.g., will an interchange contain multiple functional groups; will the TA1 be in its own interchange without any functional group(s).
- Value of ReceiverID for that site.
- Production and Testing URLs for Real Time and Batch transactions.
• Maximum number of Real Time and Batch transactions that can be sent per minute by a single trading partner (client).
• Maximum size of payload for Batch Processing Mode that can be received by a Server.
• Authentication/Authorization policies using X.509 Client Certificates (e.g., how to enroll and obtain a Client Certificate to connect to that receiver (server)).
• Information on obtaining the receiver’s (server’s) Root Certificate Authority and/or Intermediate Certificate Authority public key certificate.
• System Availability as required by the corresponding transaction’s CAQH CORE Infrastructure Rule.
• Business/Technical points of contact.
• Rules of behavior for programs that connect to this site (e.g., must not deliberately submit Batch files that contain Viruses).
• If the Server only accepts FIPS 140-2 compliant connections, or if the Server organization security policy requires a stronger transport security than SSL v3.0, the version of TLS (1.1 or higher), and the algorithm (e.g., SHA-2) that is expected for Checksum element.

4.4. Envelope Metadata Fields, Descriptions, Intended Use and Syntax/Value-Sets

The Envelope Metadata specified in Table 4.4.2 below pertains to the Message Envelope SOAP+WSDL. With the exception of Error Code and ErrorMessage fields, which are only sent in the response, the CAQH CORE required envelope metadata for the request and response are required to be identical.

4.4.1. Message Envelope

As shown in Figure 4.4.1 below, the Message Envelope is outside the Message Payload (content), and inside the transport protocol envelope. The CAQH CORE Connectivity Rule vC1.1.0 established the use of HTTP/S as the transport protocol over the public Internet, hence the transport protocol envelope consists of HTTP headers. Examples of message payload include HIPAA administrative transactions (ASC X12), HL7 clinical messages, zipped files, etc.

Figure 4.4.1
### 4.4.2. Table of CAQH CORE Envelope Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Expected Use</th>
<th>Field Name</th>
<th>Requirement Indicator for Real Time and Batch</th>
<th>DataType</th>
<th>Field Constraints or Value-sets (Not Comprehensive)</th>
</tr>
</thead>
</table>
| Payload Type  | Payload Type specifies the type of payload included within a request, (e.g., HIPAA ASC X12N transaction set 837, 820, 278, etc.). | • Message routing  
• Efficient processing  
• Auditing | PayloadType | Required for both | Coded Set | Please see CAQH CORE-required Payload Type Table document for enumeration of PayloadType field. |
| Processing Mode | Processing Mode indicates Batch or Real Time\(^{38}\)  
Processing Mode (as defined by CAQH CORE) | • Messaging routing  
• Resource allocation  
• Transaction scheduling  
• Message or transaction auditing | ProcessingMode | Required for both | Coded Set | RealTime, Batch |
| Payload Length | Defines the length of the actual payload in bytes. | • Efficient processing and resource allocation.  
• Auditing  
• Trouble-shooting | PayloadLength | Required for Batch interactions except under certain conditions\(^{39}\)  
Shall not be used for Real time. | Integer (Base 10) | |

\(^{37}\) Mixed case or Camel Case (e.g., `PayloadType`) capitalization is used for the field names to provide readability within the messages [http://en.wikipedia.org/wiki/CamelCase](http://en.wikipedia.org/wiki/CamelCase).

\(^{38}\) See Appendix 7.2: *Abbreviations and Definitions used in this Rule* for a definition of Batch and Real Time.

\(^{39}\) Some requests or responses within Batch interactions may not have a payload. This could occur when requesting a payload or when there is no payload in the response.
### Table 4.4.2 CAQH CORE Envelope Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Expected Use</th>
<th>Field Name</th>
<th>Requirement Indicator for Real Time and Batch</th>
<th>Field Constraints or Value-sets (Not Comprehensive)</th>
</tr>
</thead>
</table>
| Payload ID  | Payload ID (unique within the domain of the party that sets this value) is a payload identifier assigned by the Sender in both Batch and Real Time Processing Modes. If the payload is being resent in the absence of confirmation of receipt to persistent storage, the same PayloadID may be re-used. | • Auditing  
• Trouble-shooting | PayloadID | Required for both Real Time and Batch. | String  
*PayloadID will conform to ISO UUID standards (described at ftp://ftp.rfc-editor.org/in-notes/rfc4122.txt), with hexadecimal notation, generated using a combination of local timestamp (in milliseconds) as well as the hardware (MAC) address*40, to ensure uniqueness. |
| Time Stamp  | The Sender (request) or Receiver (response) Time Stamp. This does not require a shared time server for consistent time. | • Auditing  
• Trouble-shooting | TimeStamp | Required for both | `dateTime` (http://www.w3.org/TR/xmilschema11-2/#dateTime) |

---

40 In multithreaded environments, in addition to the hardware (MAC) address and timestamp, the Process-ID or Thread-ID may also be used as additional parameters to ensure `PayloadID` uniqueness across multiple processes and/or threads. However, the use of MAC address is not a requirement of this rule.
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Expected Use</th>
<th>Field Name</th>
<th>Requirement Indicator for Real Time and Batch</th>
<th>DataType</th>
<th>Field Constraints or Value-sets (Not Comprehensive)</th>
</tr>
</thead>
</table>
| Sender Identifier| A unique business entity identifier representing the message envelope creator. Sender Identifier is better suited for identifying business entities and trading partners than User Name because:  
• User Name is usually anonymized for security reasons and to protect privacy.  
• User Name attribute does not exist if another authentication method is used.  
• Authentication and messaging may happen on different layers and therefore may be handled by disparate applications and processes. | Message routing and processing by a receiver  
• Transaction auditing.  
• As a reference to a business agreement. | SenderID | Required | String | Maximum length 50 characters  
The use of OIDs (e.g., HL7 or IANA) is recommended, but not required. |

---

41 Unique within the Sender’s (client’s) domain.

42 §2 shows the layers in the OSI model.
## Table 4.4.2 CAQH CORE Envelope Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Expected Use</th>
<th>Field Name</th>
<th>Requirement Indicator for Real Time and Batch</th>
<th>DataType</th>
<th>Field Constraints or Value-sets (Not Comprehensive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receiver Identifier</td>
<td>A unique(^{43}) business entity identifier representing the next-hop receiver.</td>
<td>• Transaction auditing. • As a reference to a business agreement. • Message routing by the receiver.</td>
<td>ReceiverID</td>
<td>Required</td>
<td>String</td>
<td>Maximum length 50 characters The use of OIDs (e.g., HL7 or IANA) is recommended, but not required.</td>
</tr>
<tr>
<td>CORE Rule Version</td>
<td>The CORE Rule version that this envelope is using. For response messages returned by a Server, this is the version of the Server implementation.</td>
<td>• Message routing and processing. • Auditing</td>
<td>CORERuleVersion</td>
<td>Required for both</td>
<td>Coded Set</td>
<td>4.0.0</td>
</tr>
<tr>
<td>Checksum</td>
<td>An element used to allow receiving site to verify the integrity of the message that is sent.</td>
<td>Message Integrity verification</td>
<td>CheckSum</td>
<td>Required for Batch interactions except under certain conditions(^{44}) Not used for Real Time</td>
<td>String</td>
<td>Algorithm is SHA-1(^{45}) Encoding is Hex. Checksum must be computed only on the payload and not on the metadata.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error code to indicate the error when processing the envelope.</td>
<td>• Error handling • Troubleshooting</td>
<td>ErrorCode</td>
<td>Required in Response (for both Real Time and Batch) Not used in Request.</td>
<td>Coded Set</td>
<td>Please see Section on Error Handling for a definition of error codes.</td>
</tr>
</tbody>
</table>

\(^{43}\) Unique within a Receiver’s (server's) domain.

\(^{44}\) Some requests or responses within Batch interactions may not have a payload. This could occur when requesting a payload or when there is no payload in the response.

\(^{45}\) Entities requiring FIPS 140-2 compliance may use SHA-2 instead of SHA-1. If SHA-2 is used, then the entity’s Connectivity Companion Document will specify that SHA-2 is expected in incoming messages from trading partners.
Table 4.4.2 CAQH CORE Envelope Metadata

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
<th>Expected Use</th>
<th>Field Name</th>
<th>Requirement Indicator for Real Time and Batch</th>
<th>DataType</th>
<th>Field Constraints or Value-sets (Not Comprehensive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Message</td>
<td>Text error message that describes the condition that caused the error. The text of the ErrorMessage must provide additional information describing how the Error can be resolved, and must not provide conflicting information from that provided in the ErrorCode.</td>
<td>• Logging • Troubleshooting</td>
<td>ErrorMessage</td>
<td>Required in Response (for both Real Time and Batch) Not used in Request</td>
<td>String</td>
<td>Maximum length of 1024 characters. Please see Section on Error Handling for examples of Error Messages.</td>
</tr>
</tbody>
</table>

4.4.3. Specification of Processing Mode and Enumeration Payload Type Fields

4.4.3.1. Processing Mode Table (Normative)

A HIPAA-covered entity or its agent must support the transaction processing mode requirements (i.e., Real Time and/or Batch) as specified in the COREProcessingModePayloadTypeTables.docx companion document to this CAQH CORE Connectivity Rule vC3.1.0 when exchanging transactions in conformance with this CAQH CORE Connectivity Rule vC3.1.0.

The Processing Mode requirements specified in the CAQH CORE-required Processing Mode Table also apply when a HIPAA-covered entity or its agent are exchanging the transactions addressed by this rule using any other connectivity method as permitted by the CAQH CORE Safe Harbor. (See §5.)

4.4.3.2. Enumeration of Payload Types When Handling ASC X12 Payloads (Normative)

A HIPAA-covered entity or its agent must support the requirements for identifying the payload (PayloadType), which is the essential data being carried within the content of the Message Envelope as specified in the COREProcessingModePayloadTypeTables.docx companion document to this CAQH CORE Connectivity Rule vC3.1.0. (See Figure 4.4.1, Table 4.4.2, and §6.1.). (See §3.7.3 for maintenance requirements.)

4.4.3.3. Enumeration Convention for PayloadType when Handling Non-ASC X12 Payloads (Non-normative)

The Envelope metadata specification in §4.4.3 includes a PayloadType field that is enumerated for ASC X12 payload types. This envelope may also be used to transport other types of payloads. In such cases, the convention for the PayloadType field is as follows:

<SDO>_<PayloadType>_<Version>_<Sub-version>

Note: SDO stands for Standards Development Organization.
For example, an HL7 based ADT04 Version 2.3.1 payload may specify the PayloadType as HL7_ADT04_2_3_1.

5. CAQH CORE Safe Harbor

This rule specifies a “Safe Harbor” that any stakeholder can be assured will be supported by any HIPAA-covered entity or its agent. This rule further specifies the connectivity method that all HIPAA-covered entities or their agents and all voluntarily CORE-certified organizations must implement and with which conformance must be demonstrated.

As such, this rule:

- **DOES NOT** require trading partners (e.g., a provider or a health plan) to discontinue using existing connections that do not match the rule.
- **DOES NOT** require trading partners to use a CAQH CORE-compliant method for all new connections.
- **DOES NOT** require all trading partners to use only one method for any connections.
- **DOES NOT** require any entity to do business with any trading partner or other entity.

CAQH CORE expects that in some circumstances, trading partners may agree to use different communication method(s) and/or security requirements than those described in this rule to achieve the technical goals of the specific connection. Examples of potential different communication methods that could be implemented under this CAQH CORE Safe Harbor provision include a VPN (virtual private network) or SFTP (secure file transfer protocol.) Such connectivity gateways are not considered compliant with this CAQH CORE Connectivity Rule vC3.1.0. When a HIPAA-covered entity or its agent implement a different communication method(s) as permitted by this CAQH CORE Safe Harbor all payload processing modes specified for the transactions addressed by this rule must be supported in each connectivity gateway implemented which does not comply with this CAQH CORE Connectivity Rule vC3.1.0 requirements. (See §4.4.3.1)

This CAQH CORE Connectivity Rule vC3.1.0 is the CAQH CORE Safe Harbor connectivity method that a HIPAA-covered entity or its agent MUST use if requested by a trading partner. If the HIPAA-covered entity or its agent do not believe that this CAQH CORE Safe Harbor is the best connectivity method for that particular trading partner, it may work with its trading partner to implement a different, mutually agreeable connectivity method. However, if the trading partner insists on using this CAQH CORE Safe Harbor, the HIPAA-covered entity or its agent must accommodate that request. This clarification is not intended in any way to modify entities’ obligations to exchange electronic transactions as specified by HIPAA or other federal and state regulations.

6. Conformance Requirements

**Conformance** with this CAQH CORE Operating Rule can be voluntarily demonstrated and certified through successful completion of the approved CAQH CORE Certification Test Suite with a third party CAQH CORE-authorized Testing Vendor, followed by the entity’s successful application for a CORE Certification. A CORE Certification demonstrates that a HIPAA-covered entity has successfully tested for conformity with the CAQH CORE Operating Rules, and the entity or its product has fulfilled all relevant conformance requirements.

Only the Department of Health and Human Services (HHS) can decide whether a particular entity’s system is compliant or noncompliant with HIPAA Administrative Simplification requirements (which include HIPAA-adopted CAQH CORE Operating Rules). HHS may adjudicate on an entity’s compliance and assess civil money penalties or penalty fees for noncompliance under the following HIPAA Administrative Simplification mandates:

- HIPAA regulations mandate that the Secretary “will impose a civil money penalty upon a covered entity or business associate if the Secretary determines that the covered entity or business associate has violated an administrative simplification provision.” ([47 CFR 160.402](https://www.gpo.gov/fdsys/pkg/CFR-2015-title45-vol2/pdf/CFR-2015-title45-vol2.pdf))
- Under the ACA, HIPAA mandates a certification process for HIPAA-covered health plans only, under which HIPAA-covered health plans are required to file a statement with HHS certifying that their data
and information systems are in compliance with applicable standards and associated operating rules. (**Social Security Act, Title XI, Section 1173(h)**). HIPAA also mandates that a HIPAA-covered health plan must “ensure that any entities that provide services pursuant to a contact with such health plan shall comply with any applicable certification and compliance requirements” (**Social Security Act, Title XI, Section 1173(h)(3)**).

- HIPAA also mandates that HHS is to "conduct periodic audits to ensure that health plans... are in compliance with any standards and operating rules.” (**Social Security Act, Title XI, Section 1173(h)**)
7. Appendix

7.1. References

Note: These were used for rule creation as well as to create the analysis artifacts as part of CAQH CORE Connectivity vC3.

<table>
<thead>
<tr>
<th>Table 7.1</th>
<th>Author</th>
<th>Document Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>HL7 (Health Level 7)</td>
<td>HL7 Object Identifier (OID) Registry</td>
<td><a href="http://www.hl7.org/oid/index.cfm">http://www.hl7.org/oid/index.cfm</a></td>
<td></td>
</tr>
<tr>
<td>Internet Assigned Numbers Authority (IANA)</td>
<td>IANA Private Enterprise Number (PEN) aka “OID” Registration Page</td>
<td><a href="http://www.iana.org/cgi-bin/enterprise.pl">http://www.iana.org/cgi-bin/enterprise.pl</a></td>
<td></td>
</tr>
<tr>
<td>Internet Engineering Task Force (IETF)</td>
<td>Key Words for use in RFCs to Indicate Requirement Levels</td>
<td><a href="http://www.ietf.org/rfc/rfc2119.txt">http://www.ietf.org/rfc/rfc2119.txt</a></td>
<td></td>
</tr>
<tr>
<td>Internet Engineering Task Force (IETF)</td>
<td>The MIME Multipart/Form-Data (RFC 2388)</td>
<td><a href="http://www.ietf.org/rfc/rfc2388.txt">http://www.ietf.org/rfc/rfc2388.txt</a></td>
<td></td>
</tr>
<tr>
<td>OASIS</td>
<td>Web Services Reliable Messaging Protocol 1.1 (WS-RM)</td>
<td><a href="http://docs.oasis-open.org/ws-rx/wsrmi/1.1/wsrm.html">http://docs.oasis-open.org/ws-rx/wsrmi/1.1/wsrm.html</a></td>
<td></td>
</tr>
<tr>
<td>OASIS</td>
<td>Web Service Secure Conversation 1.3</td>
<td><a href="http://docs.oasis-open.org/wss/ws-secureconversation/200512/wss-secureconversation-1.3-os.html">http://docs.oasis-open.org/wss/ws-secureconversation/200512/wss-secureconversation-1.3-os.html</a></td>
<td></td>
</tr>
<tr>
<td>OASIS</td>
<td>Universal Description, Discovery and Integration (UDDI) 1.0</td>
<td><a href="http://www.oasis-open.org/committees/uddi-spec/doc/contribs.htm#uddiv1">http://www.oasis-open.org/committees/uddi-spec/doc/contribs.htm#uddiv1</a></td>
<td></td>
</tr>
</tbody>
</table>
Table 7.1

<table>
<thead>
<tr>
<th>Author</th>
<th>Document Name</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3C (World Wide Web Consortium)</td>
<td>Simple Object Access Protocol (SOAP) 1.2</td>
<td><a href="http://www.w3.org/TR/soap12-part1/">http://www.w3.org/TR/soap12-part1/</a></td>
</tr>
<tr>
<td>W3C (World Wide Web Consortium)</td>
<td>Web Services Description Language (WSDL) 1.1</td>
<td><a href="http://www.w3.org/TR/2001/NOTE-wsdl-20010315">http://www.w3.org/TR/2001/NOTE-wsdl-20010315</a></td>
</tr>
</tbody>
</table>

7.2. Abbreviations and Definitions Used in this Rule

Table 7.2

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASC X12 Interchange</td>
<td>An ASC X12 Interchange is a graphic character string structured using delimited, tagged data concepts. An ASC X12 Interchange begins with an Interchange Control Header segment: Segment ID = ISA and ends with an Interchange Control Trailer segment: Segment ID – IEA. An ASC X12 Interchange may be composed of one or more Functional Groups (GS/GE Control Segments). An ASC X12 Functional Group is composed of one or more Transaction Sets (ST/SE Control Segments). An ASC X12 Interchange may be a Logical file or a physical file as determined by the originator of the Interchange. As such, a physical file may consist of one or more ASC X12 Interchanges. The ISA Interchange Control Header segment does not identify the content of any included Functional Groups. The Functional Group Control Header segment identifies the transaction set(s) in the Functional Group: GS08-480 Version/Release/Industry Indicator Code.</td>
</tr>
<tr>
<td>Asynchronous</td>
<td>A message exchange interaction is said to be asynchronous when the associated messages are chronologically and procedurally decoupled, e.g., in a request-response interaction, the client agent can process the response at some indeterminate point in the future when its existence is discovered. Mechanisms to do this include polling, notification by receipt of another message, etc. [WS Glossary, 2004]</td>
</tr>
<tr>
<td>Term or Concept</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
</tr>
</tbody>
</table>
| **Batch (Batch Mode, Batch Processing Mode)** | Batch Mode is when the initial (first) communications session is established and maintained open and active only for the time required to transfer a batch file of one or more transactions. A separate (second) communications session is later established and maintained open and active for the time required to acknowledge that the initial file was successfully received and/or to retrieve transaction responses.  
Batch Processing Mode is also considered to be an asynchronous processing mode, whereby the associated messages are chronologically and procedurally decoupled. In a request-response interaction, the client agent can process the response at some indeterminate point in the future when its existence is discovered. Mechanisms to implement this capability may include: polling; notification by receipt of another message; or receipt of related responses (as when the request receiver "pushes" the corresponding responses back to the requestor), etc.  
Batch (asynchronous) Processing Mode is from the perspective of both the requester and responder. If a Batch (asynchronous) request is sent via intermediaries, then such intermediaries may, or may not, use Batch Processing Mode to further process the request. |
| **Batch Files (Payload)** | A single submission of a message payload that contains one ASC X12 Interchange containing one Functional Group containing one ASC X12 transaction set consisting of more than one business transaction. |
| **Client** | An entity that sends/relays a message to a Server. |
| **CAQH CORE Safe Harbor** | The connectivity requirements that application vendors, providers, and health plans (or other information sources) are required to support in order to provide assurance that these requirements are supported by any HIPAA-covered entities or their agents. |
| **Extensibility** | Extensibility is a property of a system, format, or standard that allows evolution in performance or format within a common framework, while retaining partial or complete compatibility among systems that belong to the common framework.  
Extensibility is a system design principle where the implementation takes into consideration future growth. It is a systematic measure of the ability to extend a system and the level of effort required to implement the extension. Extensions can be through the addition of new functionality or through modification of existing functionality. The central theme is to provide for change while minimizing the impact to existing system functions. |

---

Table 7.2

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Federal Information Processing Standards Security Requirements for Cryptographic Modules (FIPS 140-2)</td>
<td>The Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST) is the official series of publications relating to standards and guidelines adopted and promulgated under the provisions of Section 5131 of the Information Technology Management Reform Act of 1996 (Public Law 104-106) and the Computer Security Act of 1987 (Public Law 100-235). These mandates have given the Secretary of Commerce and NIST important responsibilities for improving the utilization and management of computer and related telecommunications systems in the Federal government. The NIST, through its Information Technology Laboratory, provides leadership, technical guidance, and coordination of government efforts in the development of standards and guidelines in these areas (<a href="http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf">http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf</a>).</td>
</tr>
</tbody>
</table>
| Interoperability                                                                | Interoperability is the capability of different information technology systems, software applications and networks to communicate, execute programs, exchange data accurately, effectively and consistently, among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units and to use the information that has been exchanged.  
Interoperability also requires no specific architecture and is independent of vendors and their various operating systems, programming languages, hardware, and network infrastructure.  
Functional interoperability is the capability to reliably exchange information without errors. Semantic interoperability allows systems to interpret and make effective use of the information exchanged among systems. |

---


Table 7.2

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
</table>
| Interoperability Specification<sup>50</sup> | An Interoperability Specification focuses on a set of constrained standards for information interchange that address the core requirements of the Use Cases. It does not define all functions, constructs and standards necessary to implement a conforming system in the real world environment.  
An Interoperability Specification defines how two or more systems exchange standard data content in a standard manner.  
Interoperability Specifications define the necessary business and technical actors, the transactions between them including the message, content and terminology standards for the actual information exchange.  
Interoperability Specifications do not specify the functional requirements or behaviors of the systems or applications.  
Interoperability Specifications, unless otherwise noted, are not intended to define or prescribe any system architecture or implementation. At the most basic level, the Interoperability Specifications define specific information exchange standards that are to be used by any two systems. Information exchange must be placed within the context of a transaction between defined technical actors which fulfill higher level business requirements derived from the use cases. In some cases the necessary technical actors may require some architectural structure or make some assumptions involving synchronous or asynchronous data exchanges, or require specific type of exchange, such as a message or document. These requirements may constrain to some degree the total range of choices regarding system architectures. When constraints are necessary to meet the use case requirements, the Interoperability Specification will note this and will retain as much architectural neutrality as possible. When appropriate, Interoperability Specifications may provide architectural examples and discuss considerations of such examples.  
HITSP and ONC do not define "Interoperability," but, do define "Interoperability Specification." |
| Large Batch Files (Payload)             | A single submission of a message payload that contains more than one ASC X12 Interchange, each of which may contain one or more Functional Groups, each of which may contain one or more ASC X12 transaction sets. |
| Large Volume of Single Real time Transactions (Synchronous) | A high number of Real Time transactions arriving at the receiving system concurrently.  
CORE defines large volume as "X"% of an organization’s average daily received transaction volume (based on all trading partners) within one minute. "X" is defined by organization. |
| Media Access Control (MAC) Address      | A media access control address (MAC address) is a unique identifier assigned to network interfaces for communications on the physical network segment. |
| Message Encapsulation Layer             | This refers to the Open Systems Interconnect (OSI) layers 5 and 6. |
| Message Envelope Standard               | SOAP+WSDL, described in Section “Specifications for SOAP + WSDL”. |
| Metadata                                | Data about data. In the context of CAQH CORE Connectivity, metadata is the information in the message envelope that describes the payload. |

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTOM</td>
<td>W3C Message Transmission Optimization Mechanism (<a href="http://www.w3.org/TR/soap12-mtom/">http://www.w3.org/TR/soap12-mtom/</a>).</td>
</tr>
<tr>
<td>Normative</td>
<td>In standards terminology, &quot;normative&quot; means &quot;considered to be a prescriptive part of the standard&quot; [Wikipedia].</td>
</tr>
<tr>
<td>Non-normative</td>
<td>Informational, not intended to be part of the specification.</td>
</tr>
<tr>
<td>OSI</td>
<td>Open Systems Interconnection Basic Reference Model (OSI Reference Model, or OSI Model for short) is a layered, abstract description for communications and computer network protocol design. From top to bottom, the OSI Model consists of the Application, Presentation, Session, Transport, Network, Data Link and Physical Layers [Wikipedia].</td>
</tr>
<tr>
<td>Open Standard</td>
<td>&quot;Open Standards&quot; are those standards made available to the general public and are developed (or approved) and maintained via a collaborative and consensus driven process. Open Standards facilitate interoperability and data exchange among different products or services and are intended for widespread adoption.</td>
</tr>
<tr>
<td>Payload</td>
<td>The essential data that is being carried within a packet or other transmission unit. The payload does not include the &quot;overhead&quot; data required to get the packet to its destination. A packet is the unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network. When any file (e-mail message, HTML file, Graphics Interchange Format file, Uniform Resource Locator request, and so forth) is sent from one place to another on the Internet, the Transmission Control Protocol (TCP) layer of TCP/IP divides the file into &quot;chunks&quot; of an efficient size for routing. Each of these packets is separately numbered and includes the Internet address of the destination. The individual packets for a given file may travel different routes through the Internet. When they have all arrived, they are reassembled into the original file (by the TCP layer at the receiving end).</td>
</tr>
</tbody>
</table>

---


52 SearchSecurity.com. [http://searchsecurity.techtarget.com/sDefinition/0,sid14,gci214475,00.html](http://searchsecurity.techtarget.com/sDefinition/0,sid14,gci214475,00.html).
Table 7.2

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>According to the CAQH CORE Connectivity Rule vC1.1.0, performance is defined in only two components:</td>
</tr>
<tr>
<td></td>
<td>Response Time – the time required to receive a Request, process it completely and send an appropriate response, as specified in the CAQH</td>
</tr>
<tr>
<td></td>
<td>CORE Eligibility and Benefits (270/271) Infrastructure Rule for Real time and Batch exchanges.</td>
</tr>
<tr>
<td></td>
<td>System Availability – the time an information source's (health plan, clearinghouse/switch or other intermediary system) processing system is</td>
</tr>
<tr>
<td></td>
<td>capable of properly processing Request/Response transactions, as specified in the CAQH CORE Eligibility and Benefits (270/271) Infrastructure</td>
</tr>
<tr>
<td></td>
<td>Rule for system availability.</td>
</tr>
</tbody>
</table>

53 CAQH CORE Eligibility & Benefits (270/271) Infrastructure Rule Section 4: Real Time Response Time Requirements.

54 CAQH CORE Eligibility & Benefits (270/271) Infrastructure Rule Section 5: Batch Response Time Requirements.

55 CAQH CORE Eligibility & Benefits (270/271) Infrastructure Rule Section 6: System Availability Requirements.
### Table 7.2

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Evaluation Criteria</td>
<td>For the purpose of evaluating the measurable performance dimensions of potential messaging methodologies to be used in Real time healthcare transactions, Performance Evaluation Criteria may include: Response Time – the time required to receive a Request, process it completely, and send an appropriate response. 56 Maximum Arrival Rate Before Saturation – the maximum number of properly formed arriving Request transactions per time period (usually seconds or minutes), above which the ability for increased acceptance for further processing stops. 57 Overhead Information – Digital information transferred across the functional interface between a user and a telecommunications system, or between functional units within a telecommunications system, for the purpose of directing or controlling the transfer of user information or the detection and correction of errors. Note: Overhead information originated by the user is not considered to be system overhead information. Overhead information generated within the communications system and not delivered to the user is system overhead information. Thus, the user throughput is reduced by both overheads while system throughput is reduced only by system overhead. 58 Capacity – the maximum number of completed Request/Response transaction sets per specific time period. Quality of Service – the number of properly and accurately completed Request/Response transaction sets divided by the number of properly submitted transactions (Requests). When making such performance measurements and evaluations, it is important to consider the architecture of networks and systems to assure their similarity, and/or to assess the relevance and impact of any differences. 57</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Processing modes or computing modes are classifications of different types of computer processing, e.g., batch, real time. In the context of CAQH CORE Operating Rules, the concept of processing mode applies to the timeframe within which a receiver of a payload of transactions processes those transactions and returns to the sender of the payload appropriate acknowledgements. See Batch and Real Time for CAQH CORE definitions.</td>
</tr>
</tbody>
</table>

---

56 CAQH CORE Eligibility & Benefits (270/271) Infrastructure Rule Section 4: Real Time Response Time Requirements and Section 5: Batch Response Time Requirements.


<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real time (Real time Mode, Real time Processing Mode)</td>
<td>Real Time Mode(^{59}) is when an entity is required to send a transaction and receive a related response within a single communications session, which is established and maintained open and active until the required response is received by the entity initiating that session. Communication is complete when the session is closed. Real Time Mode &amp; Real Time Processing Mode are also considered to be a synchronous processing mode. (See Synchronous). Real Time, or synchronous, Processing Mode is from the perspective of both the requester and responder.</td>
</tr>
<tr>
<td>Safe Harbor</td>
<td>A “Safe Harbor” is generally defined as a statutory or regulatory provision that provides protection from a penalty or liability.(^{60}) In many IT-related initiatives, a safe harbor describes a set of standards/guidelines that allow for an “adequate” level of assurance when business partners are transacting business electronically.</td>
</tr>
<tr>
<td>Server</td>
<td>An entity that receives a message from a Client, which it may process, or relay to another Server.</td>
</tr>
<tr>
<td>SOAP</td>
<td>W3C Simple Object Access Protocol Version 1.2. (<a href="http://www.w3.org/TR/soap12-part1/">http://www.w3.org/TR/soap12-part1/</a>)</td>
</tr>
<tr>
<td>Standard</td>
<td>A standard is a document, established by consensus and approved by a recognized body that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context.(^{61})</td>
</tr>
<tr>
<td>Standard Development Organization</td>
<td>Standards Development Organizations (SDOs) are organizations whose processes are accredited by ANSI. A SDO may also include non-ANSI accredited organizations such as W3C, OASIS, ISO, UN/CEFACT and IETF.</td>
</tr>
<tr>
<td>Support [Supported]</td>
<td>Means that the entity must have the capability as specified and required.</td>
</tr>
<tr>
<td>Submitter Authentication</td>
<td>X.509 Certificate based Authentication over SSL or TLS, described in Subsection “Submitter Authentication Handling.”</td>
</tr>
<tr>
<td>Synchronous</td>
<td>The application sending the request message waits for the response, which is returned on the same communications connection (i.e., synchronous request/reply). This message exchange pattern is used for most real time transactions.</td>
</tr>
</tbody>
</table>

---

\(^{59}\) Ibid.


Table 7.2

<table>
<thead>
<tr>
<th>Term or Concept</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport Layer Security (TLS)</td>
<td>Transport Layer Security (TLS)(^{62}) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that &quot;provide communications security over the Internet&quot;. TLS and SSL encrypt the segments of network connections above the Transport Layer, using symmetric cryptography for privacy and a keyed message authentication code for message reliability. Several versions of the protocols are in widespread use in applications like web browsing, electronic mail, Internet faxing, instant messaging and voice-over-IP (VoIP). TLS is an IETF standards track protocol, last updated in RFC 5246, and is based on the earlier SSL specifications developed by Netscape Corporation (<a href="http://tools.ietf.org/html/rfc5246">http://tools.ietf.org/html/rfc5246</a>). Future enhancements and development by the IETF will occur within the TLS specification.</td>
</tr>
<tr>
<td>WSDL</td>
<td>W3C Web Services Definition Language Version 1.1 (<a href="http://www.w3.org/TR/2001/NOTE-wsdl-20010315">http://www.w3.org/TR/2001/NOTE-wsdl-20010315</a>).</td>
</tr>
</tbody>
</table>

7.3. Sequence Diagrams

The UML sequence diagrams below show interactions between a client and a server. When the interactions include multiple requests/responses, each pair of requests and its corresponding (synchronous) response is shown encapsulated in a white rectangle. Each request followed by synchronous response (shown in a single white rectangle) is in a client-server connection that can be expected to be opened for a request and closed after the corresponding synchronous response is received. Subsequent requests/responses occur in new client-server connections. Servers are stateless and are not assumed to keep session information between connections, unless such information is sent as part of the requests (e.g., using ASC X12C 999 or ASC X12C TA1 payloads).

7.3.1. Real Time Interaction

This section describes Real Time interactions that include the following steps:

- Submission of Real Time Payload (step 1 in the diagrams)
- Real Time (Synchronous) response (step 2 in the diagrams)

Example 1: Health Care Services Review – Request for Review and Response (ASC X12N v5010 278)

The UML sequence diagram below shows a Health Care Services Review – Request for Review and Response Real Time transaction between a HIPAA-covered Healthcare Provider and a HIPAA-covered Health Plan. The interactions are described in the diagram below.

![Health Care Services Review – Request for Review and Response Real Time Interaction Diagram](image)

The requester of a Real Time response expects one and only one response on the payload; for example, in the above message interaction, the response payload can only be an ASC X12N v5010 278, or an

---

ASC X12C v5010 999 or an ASC X12C TA1. The following describes the Real Time interaction as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Healthcare Provider submits a Real Time request to a Health Plan, using PayloadType=X12_278_Request_005010X217E1_2.</td>
<td>Health Care Services Review - Request for Review &amp; Response</td>
</tr>
<tr>
<td>2</td>
<td>Health Plan responds (synchronously to request message 1) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_278_Response_005010X217E1_2 or X12_999_Response_005010X231A1 or X12_TA1_Response_005010X231A1)</td>
<td>Health Care Services Review - Request for Review &amp; Response</td>
</tr>
</tbody>
</table>

Example 2: Payroll Deducted and Other Group Premium Payment for Insurance Products (ASC X12N v5010 820)

The UML sequence diagram below shows a Payroll Deducted and Other Group Premium Payment for Insurance Products Real Time transaction between a HIPAA-covered Health Plan Sponsor (Client) and a HIPAA-covered Health Plan (Server). The interactions are described in the diagram below.

The requester of a Real Time response expects one and only one response on the payload; for example, in the above message interaction, the response payload can only be an ASC X12C v5010 999, or an ASC X12C TA1. The following describes the Real Time interaction as shown in the above diagram.
**Example 3: Benefit Enrollment and Maintenance (ASC X12N v5010 834)**

The UML sequence diagram below shows a *Benefit Enrollment and Maintenance* Real Time transaction between a Health Plan Sponsor (Client) and a Health Plan (Server). The interactions are described in the diagram below.

The requester of a Real Time response expects one and only one response on the payload; for example, in the above message interaction, the response payload can only be only an ASC X12C v5010 999, or an ASC X12C TA1. The following describes the Real Time interaction as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Health Plan Sponsor submits a Real Time request to a Health Plan, using PayloadType=X12_834_Request_005010X218A1.</td>
<td>Benefit Enrollment and Maintenance</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan responds (synchronously to request message 1) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_999_Response_005010X231A1 or X12_TA1_Response_005010X231A1)</td>
<td>Benefits Enrollment and Maintenance</td>
</tr>
</tbody>
</table>
**Example 4: Healthcare Claim (ASC X12N v5010 837 Claim)**

The UML sequence diagram below shows an Institutional Healthcare Claim Real Time transaction between a HIPAA-covered Healthcare Provider (Client) and a HIPAA-covered Health Plan (Server). The interactions are described in the diagram below.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Health Plan Sponsor submits a Real Time request to a Health Plan, using PayloadType=X12_837_Request_005010X223A1_2.</td>
<td>Healthcare Claim: Institutional</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan responds (synchronously to request message 1) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_999_Response_005010X231A1 or X12_TA1_Response_005010X231A1)</td>
<td>Benefit Enrollment and Maintenance</td>
</tr>
</tbody>
</table>

The requester of a Real Time response expects one and only one response on the payload; for example, in the above message interaction, the response payload can only be an ASC X12N v5010 277CA, or an ASC X12C v5010 999, or an ASC X12C TA1. The following describes the Real Time interaction as shown in the above diagram.
### 7.3.2. Batch Interactions

This section describes Batch interactions that include the following steps:

- Submission of Batch Payload (steps 1 and 2 in the diagrams)
- Retrieval of Acknowledgment for the submission (steps 3 and 4 in the diagrams)
- Retrieval of Batch Processing Results (steps 5 and 6 in the diagrams)
- Submission of Acknowledgment for the results retrieved (steps 7 and 8 in the diagrams)

The Batch interactions can be conducted using specific payload types as shown in 7.3.2.1 or with Mixed Payload types as show in 7.3.2.2.

#### 7.3.2.1. Batch Interaction for Specific Payload Types

Within the Batch Interaction for Specific Payload Types, the Batch Payload consists of a single type of transaction set.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
</table>
| 2                | A Health Plan responds (synchronously to request message 1) to the request either with an:  
HTTP Error  
or HTTP Successful Response with the CORE Envelope Error  
or HTTP Successful Response with the CORE Envelope Response  
(PayloadType=\textit{X12\_277CA\_Response\_005010X214E1\_2}  
\textit{or X12\_999\_Response\_005010X231A1}  
\textit{or X12\_TA1\_Response\_005010X231A1}) | Healthcare Claim: Institutional |
**Example 1: Health Care Claim (ASC X12N v5010 837 Claim):**
The UML sequence diagram below shows a typical Batch Interaction between a HIPAA-covered Healthcare Provider, and a HIPAA-covered Health Plan specifically for ASC X12N v5010 837 batch payloads.
The following describes the Batch interaction as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Healthcare Provider submits a Batch of requests to a Health Plan, using PayloadType = (X12_837_Request_005010X223A1_2) for an Institutional claim, or (X12_837_Request_005010X222A1) for a Professional claim, or (X12_837_Request_005010X224A1_2) for a Dental Claim.</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan responds (synchronously to request message 1) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=(X12_BatchReceiptConfirmation))</td>
</tr>
<tr>
<td>3</td>
<td>A Healthcare Provider sends a Request to a Health Plan to solicit the acknowledgement ((X12_999_RetrievalRequest_005010X231A1)) for the Batch file that was just submitted.</td>
</tr>
<tr>
<td>4</td>
<td>A Health Plan responds (synchronously to request message 3) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Retrieval Response) (PayloadType= (X12_999_Response_005010X231A1) or (X12_TA1_Response_00501X231A1) or (X12_005010_Response_NoBatchAckFile))</td>
</tr>
<tr>
<td>5</td>
<td>A Healthcare Provider sends a Request to a Health Plan to solicit the Health Care Claim Acknowledgement for the batch of claims that was submitted in message sequence 1 using PayloadType=(X12_277CA_Request_005010X214E1_2).</td>
</tr>
<tr>
<td>6</td>
<td>A Health Plan responds (synchronously to request message 5) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType= (X12_277CA_Response_005010X214E1_2) or (X12_005010_Response_NoBatchResultsFile))</td>
</tr>
</tbody>
</table>

Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document

- Health Care Claim: Institutional
- Batch Receipt Confirmation Response
- Implementation Acknowledgement Retrieval
- Implementation Acknowledgement Retrieval
- Health Care Claim Acknowledgement
- Health Care Claim Acknowledgement
<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>A Healthcare Provider submits the acknowledgement Batch Results Acknowledgement Submission (PayloadType= X12_999_SubmissionRequest_005010X231A1 or X12_TA1_SubmissionRequest_00501X231A1) to a Health Plan. This acknowledgment submission is required by the CAQH CORE Infrastructure Rule corresponding to the specific transaction.</td>
<td>Implementation Acknowledgement Submission (Request)</td>
</tr>
<tr>
<td>8</td>
<td>A Health Plan responds (synchronously to request message 7) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_Response_ConfirmReceiptReceived)</td>
<td>Implementation Acknowledgement Submission (Response)</td>
</tr>
</tbody>
</table>

The UML sequence diagram below shows a typical Batch Interaction between a HIPAA-covered Healthcare Provider and a HIPAA-covered Health Plan for ASC X12N v5010 278 batch payloads.

The following describes the Batch interaction as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Healthcare Provider submits a Batch of requests to a Health Plan, using PayloadType as X12_278_Request_005010X217E1_2.</td>
<td>Health Care Services Review – Request for Review &amp; Response</td>
</tr>
<tr>
<td>Message Sequence</td>
<td>Description</td>
<td>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>----------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>A Health Plan responds (synchronously to request message 1) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType= X12_BatchReceiptConfirmation)</td>
<td>Batch Receipt Confirmation Response</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>A Healthcare Provider sends a Request to a Health Plan to solicit the acknowledgement (X12_999_RetrievalRequest_005010X231A1) for the Batch file that was just submitted.</td>
<td>Implementation Acknowledgement Retrieval</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>A Health Plan responds (synchronously to request message 3) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Retrieval Response) (PayloadType= X12_999_Response_005010X231A1 or X12_TA1_Response_00501X231A1 or X12_005010_Response_NoBatchAckFile)</td>
<td>Implementation Acknowledgement Retrieval</td>
</tr>
<tr>
<td><strong>5</strong></td>
<td>A Healthcare Provider sends a Request to a Health Plan to solicit the results of processing the batch that was submitted in message sequence 1, using Payload Type: X12_278_Request_005010X217E1_2</td>
<td>Health Care Services Review – Request for Review &amp; Response</td>
</tr>
<tr>
<td><strong>6</strong></td>
<td>A Health Plan responds (synchronously to request message 5) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Results Retrieval Response) (PayloadType=X12_278_Response_005010X217E1_2 or X12_005010_Response_NoBatchResultsFile)</td>
<td>Health Care Services Review – Request for Review &amp; Response</td>
</tr>
<tr>
<td><strong>7</strong></td>
<td>A Healthcare Provider submits the acknowledgement (PayloadType= X12_999_SubmissionRequest_005010X231A1, or X12_TA1_SubmissionRequest_00501X231A1) to a Health Plan. This acknowledgment submission is required by the CAQH CORE Infrastructure Rules.</td>
<td>Implementation Acknowledgement Submission</td>
</tr>
<tr>
<td>Message Sequence</td>
<td>Description</td>
<td>Reference to Payload Type Table</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>8</td>
<td>A Health Plan responds (synchronously to request message 7) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_Response_ConfirmReceiptReceived)</td>
<td>Implementation Acknowledgement Submission</td>
</tr>
</tbody>
</table>

CAQH Committee on Operating Rules for Information Exchange (CORE)
Connectivity Rule vC3.1.0
### 7.3.2.2. Batch Interaction for Mixed Payload Types

The UML sequence diagram below shows a Mixed Payload Type Batch Interaction between a HIPAA-covered Healthcare Provider and a HIPAA-covered Health Plan.

**Healthcare Provider**

1. **Batch Submission** (PayloadType=X12_005010_Request_BatchSubmissionMixed)

2. **HTTP Error** or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_005010_Response_BatchSubmissionMixed)

3. **Batch Acknowledgement Request** (PayloadType=X12_005010_Request_Acks)

4. **HTTP Error** or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Response) (PayloadType=X12_005010_Response_Acks or X12_005010_Response_NoBatchAckFile)

5. **Batch Results Retrieval Request** (PayloadType=X12_005010_Request_BatchResultsMixed)

6. **HTTP Error** or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Results Retrieval Response) (PayloadType=X12_005010_Response_BatchResultsMixed or X12_005010_Response_NoBatchResultsFile)

7. **Batch Results Acknowledgement Submission** (PayloadType=X12_999_SubmissionRequest_005010X231A1 or X12_TA1_SubmissionRequest_005010X231A1)

8. **HTTP Error** or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_Response_ConfirmReceiptReceived)
The following describes the typical Mixed Batch interaction as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Message Sequence</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>1</td>
<td>A Healthcare Provider submits a Batch of requests to a Health Plan, using PayloadType=X12_005010_Request_BatchSubmissionMixed</td>
<td>Batch Submission (mixed payload types)</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan responds (synchronously to request message 1) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_005010_Response_BatchSubmissionMixed)</td>
<td>Batch Submission (mixed payload types)</td>
</tr>
<tr>
<td>3</td>
<td>A Healthcare Provider sends a Request to a Health Plan with PayloadType=X12_005010_Request_Acks to solicit the acknowledgement from a Health Plan (ASC X12C v5010 999 or ASC X12C TA1) for the Batch file that was just submitted.</td>
<td>General Acknowledgements Pick Up</td>
</tr>
<tr>
<td>4</td>
<td>A Health Plan responds (synchronously to request message 3) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Response) (PayloadType=X12_005010_Response_Acks or X12_005010_Response_NoBatchAckFile)</td>
<td>General Acknowledgements Pickup</td>
</tr>
<tr>
<td>5</td>
<td>A Healthcare Provider sends a Request to a Health Plan to solicit the Results for the Batch file that was submitted in message sequence 1 using PayloadType=X12_005010_Request_BatchResultsMixed.</td>
<td>Batch Results Retrieval (mixed payload types)</td>
</tr>
<tr>
<td>6</td>
<td>A Health Plan responds (synchronously to request message 5) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Results Retrieval Response) (PayloadType=X12_005010_Response_BatchResultsMixed or X12_005010_Response_NoBatchResultsFile)</td>
<td>Batch Results Retrieval (mixed payload types)</td>
</tr>
<tr>
<td>7</td>
<td>A Healthcare Provider submits the acknowledgement (PayloadType= X12_999_SubmissionRequest_005010X231A1 or X12_TA1_SubmissionRequest_005010X231A1) to a Health Plan. This acknowledgment submission is required by the CAQH CORE Infrastructure Rules.</td>
<td>Implementation Acknowledgement Submission</td>
</tr>
</tbody>
</table>
7.3.3. Generic Batch Interactions

The term Generic is used to denote the fact that the Batch Interactions defined herein can be used as building blocks to build more complex interactions if such interactions are needed to support current or future business use cases. Within the Generic Batch Interactions, there are two types:

1) **Generic Push**: this message interaction is characterized by the following steps:
   - Client submits, or “pushes” a Batch Payload to a Server
   - Client then retrieves an acknowledgment (or error) from the Server for the Batch Payload that it had previously submitted to the Server.

2) **Generic Pull**: this message interaction is characterized by the following steps:
   - Client retrieves, or “pulls” a Batch Payload from a Server
   - Client then submits an acknowledgment (or error) to the Server for the Batch Payload that the Client has previously retrieved from the Server.

Both of these message interactions can be used either for Specific Transaction Batch Payload Types (with a single type of transaction set), or for Mixed Batch Payload types (using multiple transaction sets within the same Batch Payload). For simplicity, the examples shown below are limited to Specific Transaction Batch Payload Types.

Two example transactions are shown in the following sub-sections:

   a) Benefit Enrollment and Maintenance (ASC X12N v5010 834)
   b) Payroll Deducted and Other Group Premium Payment for Insurance Products (ASC X12N v5010 820)

Both of these transactions can use either the **Generic Push or Generic Pull** interactions. Depending on the interaction being used, the business actors that use these interactions will need to assume the roles of Client or Server.

7.3.3.1. **Generic Push**

This message interaction is characterized by the following steps:

- Client submits, or “pushes” a Batch Payload to a Server
- Client then retrieves an acknowledgment (or error) from the Server for the Batch Payload that it had previously submitted to the Server.

The UML sequence diagrams below show examples of the Generic Push Interactions.
Example: Benefit Enrollment and Maintenance (ASC X12N v5010 834)

The following describes the **Benefit Enrollment and Maintenance** transaction using the **Generic Push** interaction as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Health Plan Sponsor (Client) submits to a Health Plan (Server) a Batch of Benefit Enrollment and Maintenance requests using PayloadType=X12_834_Request_005010X220A1.</td>
<td>Benefit Enrollment and Maintenance</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan (Server) responds synchronously in Real time either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_BatchReceiptConfirmation)</td>
<td>Batch Receipt Confirmation Response</td>
</tr>
<tr>
<td>3</td>
<td>A Health Plan Sponsor (Client) sends a Request to a Health Plan (Server) with PayloadType=X12_999_RetrievalRequest_005010X231A1 to solicit the acknowledgement (ASC X12C v5010 999 or ASC X12C TA1) for the Batch file that was just submitted.</td>
<td>Implementation Acknowledgement Retrieval</td>
</tr>
</tbody>
</table>
CAQH Committee on Operating Rules for Information Exchange (CORE)  
Connectivity Rule vC3.1.0

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</th>
</tr>
</thead>
</table>
| 4                | A Health Plan (Server) responds (synchronously to request message 3) to the request either with an: HTTP Error  
or HTTP Successful Response with the CORE Envelope Error  
or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Retrieval Response)  
(PayloadType= X12_999_Response_005010X231A1  
or X12_TA1_Response_00501X231A1  
or X12_005010_Response_NoBatchAckFile) | Benefit Enrollment and Maintenance |

**Example:** Payroll Deducted and Other Group Premium Payment for Insurance Products (ASC X12N v5010 820)

| Health Plan Sponsor (Client) | Health Plan Sponsor initiates the Submission (or Pushes)  
ASC X12N v5010 820 Batch to Health Plan | Health Plan (Server) |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Batch Submission (PayloadType=X12_820_Request_005010X218A1)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2. HTTP Error  
or HTTP Successful Response with the CORE Envelope Error  
or HTTP Successful Response with the CORE Envelope Response (PayloadType= X12_BatchReceiptConfirmation) |  
3. Batch Acknowledgement Retrieval Request (PayloadType=X12_999_RetrievalRequest_005010X231A1) |  
4. HTTP Error  
or HTTP Successful Response with the CORE Envelope Error  
or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Retrieval Response) (PayloadType= X12_999_Response_005010X231A1  
or X12_TA1_Response_00501X231A1  
or X12_005010_Response_NoBatchAckFile) |

The following describes the Payroll Deducted and Other Group Premium Payment for Insurance Products transaction using the Generic Push interaction, as shown in the above diagram.
<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The Health Plan Sponsor (Client) submits to a Health Plan (Server) a Batch of Payroll Deducted and Other Group Premium Payment for Insurance Products requests using PayloadType= X12_820_Request_005010X218A1.</td>
<td>Payroll Deducted and Other Group Premium Payment for Insurance Products</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan (Server) responds synchronously in Real time either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_BatchReceiptConfirmation)</td>
<td>Payroll Deducted and Other Group Premium Payment for Insurance Products</td>
</tr>
<tr>
<td>3</td>
<td>A Health Plan Sponsor (Client) sends a Request to a Health Plan (Server) using (PayloadType=X12_999_RetrievalRequest_005010X231A1) to solicit the acknowledgement (ASC X12C v5010 999 or ASC X12C TA1) for the Batch file that was just submitted.</td>
<td>Implementation Acknowledgement Retrieval</td>
</tr>
<tr>
<td>4</td>
<td>A Health Plan (Server) responds (synchronously to request message 3) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (Batch Acknowledgement Retrieval Response) (PayloadType= X12_999_Response_005010X231A1 or X12_TA1_Response_00501X231A1 or X12_005010_Response_NoBatchAckFile)</td>
<td>Implementation Acknowledgement Retrieval (Response)</td>
</tr>
</tbody>
</table>

**7.3.3.2. Generic Pull**

This message interaction is characterized by the following steps:
- Client retrieves, or “pulls” a Batch Payload from a Server
- Client then submits an acknowledgment (or error) to the Server for the Batch Payload that the Client has previously retrieved from the Server.

The UML sequence diagrams below show examples of the *Generic Pull* Interactions.
**Example: Benefit Enrollment and Maintenance (ASC X12N v5010 834)**

The following describes the Benefit Enrollment and Maintenance transaction using the Generic Pull interaction, as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Health Plan (Client) sends a Health Plan Sponsor (Server) a retrieval request for a Batch of Benefit Enrollment and Maintenance requests using PayloadType=X12_834_Request_005010X220A1.</td>
<td>Benefit Enrollment and Maintenance</td>
</tr>
<tr>
<td>2</td>
<td>Health Plan Sponsor (Server) responds synchronously in Real time either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_834_Response_005010X220A1)</td>
<td>Benefit Enrollment and Maintenance</td>
</tr>
</tbody>
</table>
CAQH Committee on Operating Rules for Information Exchange (CORE)  
Connectivity Rule vC3.1.0

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table Transaction Name Column in the Processing Modes and Payload Types Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>A Health Plan (Client) submits to a Health Plan Sponsor (Server) the acknowledgement (PayloadType X12_999_SubmissionRequest_005010X231A1 or X12_TA1_SubmissionRequest_005010X231A1) to the Health Plan. This acknowledgment submission is required by CAQH CORE Infrastructure Rules.</td>
<td>Implementation Acknowledgement Submission</td>
</tr>
<tr>
<td>4</td>
<td>Health Plan Sponsor (Server) responds (synchronously to request message 3) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_Response_ConfirmReceiptReceived)</td>
<td>Implementation Acknowledgement Submission</td>
</tr>
</tbody>
</table>

**Example: Payroll Deducted and Other Group Premium Payment for Insurance Products (ASC X12N v5010 820)**

Health Plan (Client) Health Plan Retrieves (or Pulls) ASC X12N v5010 820 Batch from Health Plan Sponsor

1. Batch Retrieval (PayloadType=X12_820_Request_005010X218A1)

2. HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_820_Response_005010X218A1)

3. Batch Acknowledgement Submission (PayloadType=X12_999_SubmissionRequest_005010X231A1, or X12_TA1_SubmissionRequest_005010X231A1)

4. HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=X12_Response_ConfirmReceiptReceived)
The following describes the *Payroll Deducted and Other Group Premium Payment for Insurance Products* transaction using the *Generic Pull* interaction, as shown in the above diagram.

<table>
<thead>
<tr>
<th>Message Sequence</th>
<th>Description</th>
<th>Reference to Payload Type Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A Health Plan (Client) sends a Health Plan Sponsor (Server) a retrieval request for a Batch of <em>Payroll Deducted and Other Group Premium Payment for Insurance Products</em> using PayloadType=\textit{X12_820_Request_005010_X218A1}.</td>
<td>Payroll Deducted and Other Group Premium Payment for Insurance Products (Retrieval Response)</td>
</tr>
<tr>
<td>2</td>
<td>A Health Plan Sponsor (Server) responds synchronously in Real Time either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=\textit{X12_820_Response_005010_X218A1}).</td>
<td>Payroll Deducted and Other Group Premium Payment for Insurance Products (Retrieval Response)</td>
</tr>
<tr>
<td>3</td>
<td>A Health Plan (Client) submits to a Health Plan Sponsor (Server) the acknowledgement (PayloadType=\textit{X12_999_SubmissionRequest_005010_X231A1} or \textit{X12_TA1_SubmissionRequest_005010_X231A1}) to a Health Plan. This acknowledgment submission is required by CAQH CORE Infrastructure Rules.</td>
<td>Payroll Deducted and Other Group Premium Payment for Insurance Products (Batch Results Acknowledgment Submission)</td>
</tr>
<tr>
<td>4</td>
<td>A Health Plan Sponsor (Server) (responds synchronously to request message 3) to the request either with an: HTTP Error or HTTP Successful Response with the CORE Envelope Error or HTTP Successful Response with the CORE Envelope Response (PayloadType=\textit{X12_Response_ConfirmReceiptReceived}).</td>
<td>Payroll Deducted and Other Group Premium Payment for Insurance Products (Batch Results Acknowledgment Response)</td>
</tr>
</tbody>
</table>