



Nationwide Health Information Network (NHIN)

Medicaid Eligibility Verification

Web Service Interface Specification

V 1.0

1/13/2010



NHIN Medicaid Eligibility Verification Web Service Interface
Specification
v 1.0

Contributors

Name	Organization	Area
Seonho Kim	CHIC/MEDNET	Specification
John Fraser	CHIC/MEDNET	Specification
Laura Megas	FHA	Specification
Jackie Key	ONC/Deloitte	Specification
Bill Branch	CMS	Specification

Document Change History

Version	Date	Changed By	Items Changed Since Previous Version
0.5	12/23/2009	Seonho Kim	Contents – first draft
0.5	12/26/2009	John Fraser & Seonho Kim	Most sections – major updates from four sets of comments on the first draft.
0.9	1/4/2010	Seonho Kim & John Fraser	Changes based on 12/30/2009 call, and submitted comments. Add proscriptive SAML assertion design and some spec intended usage statements plus other changes.
0.9	1/12/2010	Seonho Kim	Changes based on 1/6/2010 call and submitted comments. WS-Security Username/Password Token authentication/authorization requirement is replaced by SAML Assertion based approach using NPI
1.0	1/13/2010	Seonho Kim, Jackie Key	Minor updates based on group feedback. Applied consistent formatting/language and enhanced clarity.

Document Approval

Version	Date	Approved By	Role



Table of Contents

1	PREFACE	4
1.1	INTRODUCTION	4
1.2	INTENDED AUDIENCE	4
1.3	BUSINESS NEEDS SUPPORTED BY THIS SPECIFICATION	4
1.4	REFERENCED DOCUMENTS AND STANDARDS	5
1.5	RELATIONSHIP TO OTHER NHIN SPECIFICATIONS	5
2	INTERFACE DESCRIPTION	6
2.1	DEFINITION	6
2.2	DESIGN PRINCIPLES AND ASSUMPTIONS	6
2.3	TRIGGERS	6
2.4	TRANSACTION STANDARD	7
2.4.1	ANSI ASC X12 270 and 271	7
2.4.2	CAQH CORE Phase II Rules	7
2.5	TECHNICAL PRE-CONDITIONS	8
2.6	TECHNICAL POST-CONDITIONS	8
3	INTERFACE DEFINITION	8
3.1	CAQH CORE ELIGIBILITY VERIFICATION TRANSACTION:	8
3.2	CORE METADATA	9
3.3	CONNECTIVITY RULE	10
3.4	SOAP + WSDL BASED MESSAGE ENVELOPE	10
3.5	REAL-TIME (REAL-TIME MODE, REAL-TIME PROCESSING MODE)	10
3.6	SAML ASSERTION BASED USER AUTHENTICATION AND AUTHORIZATION	10
3.6.1	National Provider Identifier (NPI) Attribute	11
3.6.2	NPI Provider Name Attribute	11
4	ERROR HANDLING	11
5	AUDITING	12
APPENDIX A:	SAMPLE MESSAGES	13
	SAMPLE CORE SOAP + WSDL REAL-TIME REQUEST	13
	SAMPLE CORE SOAP + WSDL REAL-TIME RESPONSE	13
APPENDIX B:	WSDL	14
APPENDIX C:	CORE PHASE II COMPLIANT XML SCHEMA SPECIFICATION	16



1 Preface

1.1 Introduction

The Nationwide Health Information Network (NHIN) Web Service Interface specifications define services which can be implemented by each node on the NHIN network in order exchange interoperable health information over the Internet. Health Information Organizations (HIOs) which act as nodes on the NHIN are termed NHIOs. These functional services provide discovery and information exchange capabilities and rest upon a foundational set of messaging, security, and privacy services.

This document presents the NHIN Medicaid Eligibility Verification Web Service Interface Specification. This service will allow health care providers and other authorized users to determine the enrollment status of an individual patient in any of the 54 different Medicaid systems operated by US states and territories using a real-time request/response service across the NHIN.

This specification is designed such that it can be extended in the future to provide eligibility status of individuals in Medicare and other public and private insurance plans¹.

1.2 Intended Audience

The primary audiences for NHIN Specifications are the individuals responsible for implementing software solutions that realize these interfaces at Health Information Organizations (HIOs) who are, or seek to be, nodes on the NHIN network. This specification document is intended to provide an understanding of the context in which the Web Service Interface is meant to be used, the behavior of the interface, the Web Services Description Language (WSDLs) used to define the service, and any Extensible Markup Language (XML) schemas used to define the content.

1.3 Business Needs Supported by this Specification

When a patient schedules an appointment at a clinician's office, the clinician needs to be able to determine who will pay for the services if the patient is covered by insurance. This transaction is called Insurance Eligibility Verification or Eligibility Verification, which is classified as an "administrative service", as opposed to a clinical transaction.

The general work flow for this transaction is as follows:

- 1) A MITA Management Information Systems (MMIS) System receives a single 270 formatted request for eligibility verification from authorized providers, programs or business associates over NHIN, as defined in this specification
- 2) The MMIS System performs inquire on their Medicaid databases (not specified herein)
- 3) The MMIS system returns a 271 formatted response over the NHIN indicating whether the member is eligible for some health benefit plan coverage under Medicaid along with other information, including patient liability for deductible, co-pay and co-insurance amounts for a defined base set of general or specific benefits of service.

¹ This specification is constrained to address the needs for the MITA-Connect demonstration of this system at the 2010 Annual HIMSS Conference & Exhibition, March 1 – 4 in Atlanta. These constraints may be changed in future versions. This specification will also be updated, for example to adopt the version 5010 of ASC X12 standard, as newer standards are adopted in the community.



1.4 Referenced Documents and Standards²

The following documents and standards were referenced during the development of this specification. Deviations from or constraints upon these standards are identified below.

1) **Org/SDO name:** CAQH

Reference # / Spec Name: CORE 260 Phase II Eligibility & Benefits (270/271) Data Content Rule

Version #: v2.0.0 (July, 2008)

NHIN Deviations or Constraints:

Underlying Specs:

Link: <http://www.caqh.org/pdf/260.pdf>

2) **Org/SDO name:** CAQH

Reference # / Spec Name: CORE 270 Phase II Connectivity Rule

Version #: v2.0.1 (March 16, 2009)

NHIN Deviations or Constraints:

Underlying Specs:

Link: <http://www.caqh.org/pdf/270.pdf>

1.5 Relationship to Other NHIN Specifications

This specification is related to other NHIN specifications as described below:

- **Messaging Platform** – specifies a base set of messaging standards and web service protocols which must be implemented by each NHIN node and applies to all transactions. All NHIN inter-nodal messages are SOAP messages over HTTP using web services, must be encrypted and digitally signed.
- **Authorization Framework** – defines the exchange of metadata used to characterize each NHIN request. The purpose of that exchange is to provide the responder with the information needed to make an authorization decision for the requested function. Each initiating message must convey information regarding end user attributes and authentication using SAML 2.0 assertions.

Together, the Messaging Platform and the Authorization Framework define the foundational messaging, security and privacy mechanisms for the NHIN.

- **Web Services Registry** – enables nodes to discover each other through interactions with the NHIN UDDI registry, which lists NHIN nodes, the NHIN web services supported by each node, and how to reach those service end points. In this context, it might be needed to identify target nodes. This registry lists NHIN services which in this case will consist of a single entry for each of the 54 state and territorial Medicaid service endpoints that will respond to the 270/271 messages as defined herein.

² HITSP transaction specifications T40 *Patient Health Plan Eligibility Verification Transaction* and T85 *Administrative Transport to Health Plan Transaction* will be used to inform the implementation of this NHIN interface specification. The HITSP and NHIN teams will collaborate on updates to the relevant specifications at the conclusion of the initial demonstration of this interface to insure appropriate alignment.



2 Interface Description

2.1 Definition

This specification describes how Eligibility and Benefits (270/271) data can be exchanged using the NHIN. This exchange supports a patient-specific health plan eligibility verification request initiated from one HIO to a MITA MMIS requesting a verification indicating whether the member is eligible for some health benefit plan coverage under Medicaid along with other details. The interface will return a response containing a payload in the format of a 271 message. MITA MMIS systems may be acting as a single endpoint or as part of an HIO

This specification does not define Eligibility and Benefits (270/271) data sets, nor does this specification provide details on how HIOs should collect, store or format Eligibility and Benefits (270/271) data for exchange. For guidance on the structure and format of Eligibility and Benefits (270/271) data see the Transaction Standards defined below.

2.2 Design Principles and Assumptions

The following assumptions or design principles underlie this specification:

- There is no central or federated service that performs transactions across multiple HIOs.
- Each State or Territory Medicaid System shall have a single Medicaid Eligibility Verification service as defined by MITA, and registered in the NHIN Web Services Registry. The requesting NHIN Gateway shall be responsible for using the NHIN Web Services Registry to determine which of these Medicaid Services they will call for any specific 270 request message.
- Patient identity is established using an existing Medicaid patient identifier which is contained within encrypted payload in the request message. CMS Medicaid systems require the Initiating HIO to use this specific identifier for the patient of interest. The use of the NHIN Patient Discovery transaction is not required between an initiating HIO and a Medicaid HIO as these identifiers are already known to the initiating HIO through other verifiable means.
- Security Assertion Markup Language (SAML) Authorization assertion(s) will be included in the request message as specified by the Messaging Platform and Authorizations Framework specifications.
- Along with generic authorization statements, Attribute statement(s) specific to this specification will be included in the request. These specific attributes are National Provider Identifier (NPI) number and NPI Provider Name, which will be validated against National Plan & Provider Enumeration System (NPPES) by MITA.
- A single Medicaid Eligibility Verification transaction must be received, processed and the appropriate response provided back to the sender within response time requirements specified in CORE 270 Phase II Connectivity Rule section 4.3.5 and 4.3.6. These are assumed to occur at the client level.

2.3 Triggers

Any authorized provider (or program or business associate) will form a 270 message to verify Medicaid enrollment on a patient who is believed to be a Medicaid beneficiary. The authorized provider submits the 270 messages through an edge system (EHR system, Practice Management System, Web Portal, etc) to a HIO's NHIN Gateway using any method acceptable to the HIO. The HIO's Gateway then looks up in the NHIN Web Services Registry the appropriate Medicaid system to query. The HIO's NHIN Gateway then calls the appropriate service and submits a patient-specific Eligibility Verification Request and waits for the response. The response is then sent back to the authorized provider's edge system to complete the transaction.



2.4 Transaction Standard

This interface identifies the CAQH CORE 270 Phase II Connectivity Rule as the standard for Medicaid Eligibility Verification.

This specification adopts existing insurance eligibility verification message formats without changes, but defines transporting these messages using the current NHIN architecture.

These existing 270 and 271 message formats are defined in the following standards:

- Eligibility Verification requests and response messages shall adhere to the following existing standards. The American National Standards Institute (ANSI) Accredited Standards Committee (ASC) standards for the X12N 270/271 – Eligibility, Coverage or Benefit Inquiry and Response transaction set standard. This standard is defined in the X12N Implementation Guide 004010X92 plus Addenda 004010X92A1³, CAQH and with the additional constraints of the Council for Affordable Quality Healthcare (CAQH) CORE 270 Connectivity Rule, CAQH CORE 260 Data Content Rule. All references in this document to 270 and/or 271 shall mean these messages formatted to these standards.
- Eligibility Verification requests shall include sufficient attributes of the user performing the request to enable receiving system to authenticate and authorize the request.

2.4.1 ANSI ASC X12 270 and 271

ANSI ASC X12N 270/271 - Health Care Eligibility Benefit Inquiry and Response, Version 4010 is the HIPAA standard for eligibility inquiry and response. It is designed for eligibility inquiry submitters to determine 1) whether an information source organization (e.g., employer, payer, HMO) has a subscriber (or dependent) on file and to verify 2) the health care eligibility and benefit information about the subscriber (or dependent).

2.4.2 CAQH CORE Phase II Rules

This section describes CAQH CORE business rules defined by the CORE 260 Phase II Eligibility & Benefits (270/271) Data Content Rule and CORE 270 Phase II Connectivity Rule which this specification is based on (see http://www.caqh.org/CORE_phase2.php) with following constraints.

- This specification requires use of CORE Simple Object Access Protocol (SOAP) + WSDL Method (Envelope Standard B)
- This specification does not require use of CORE HTTP MIME Multipart (Envelope Standard A) This specification adopts the Real-Time Request/Request transaction Mode. Batch Processing Mode is not supported in this specification. For details, please refer CAQH CORE 270 Phase II Connectivity Rule, Version 2.0.1

2.4.2.1 CAQH CORE 260 Phase II Eligibility & Benefits (270/271) Data Content Rule

This CORE rule conforms with and builds upon the HIPAA-adopted X12N 270/271 version 004010X092 Health Care Eligibility Benefit Inquiry and Response Implementation Guide and its associated X12N 004010X092A1 Health Care Eligibility Benefit Inquiry and Response Addenda and specifies the minimum content that a CORE-certified entity must include in the 271 response (refer to <http://www.caqh.org/pdf/260.pdf>).

³ This specification will also be updated to adopt the 5010 X12N standards, as the community migrate from X12 4010 to 5010.



2.4.2.2 CORE 270 Phase II Connectivity Rule

This CORE rule provides a connectivity rule to facilitate connectivity standardization and interoperability across healthcare information exchange, including message envelope standards and metadata requirements (see <http://www.caqh.org/pdf/270.pdf>).

2.5 Technical Pre-conditions

The following technical pre-conditions exist for this interface specification:

- Responding Medicaid HIOs must publish in the NHN Web Services Registry data containing descriptive and technical information about their NHIN Medicaid Eligibility Verification service.
- The Medicaid HIO to which the query will be directed has been selected and applicable service end points have been identified using the NHN Web Services Registry.
- The Requesting HIOs have a pre-established trust relationship with the Medicaid HIO they are calling, that enables them to participate in an eligibility verification data exchange.
- The Requesting HIOs will format a standardized 270 request message as defined herein.
- The Initiating HIO must include SAML assertion containing user-level credentials sufficient to enable authentication and/or authorization by the receiving Medicaid system.

2.6 Technical Post-conditions

The following technical post-conditions will result after the execution of this interface specification:

- Errors encountered will be handled and included in the response as specified in CAQH CORE 270 Phase II Connectivity Rule.
- The response to this request is a payload containing an 271 message and some metadata describing the transaction such as Payload ID, Sender ID, Receiver ID, etc. (required CORE Metadata are described in section 3.3.2 in detail).

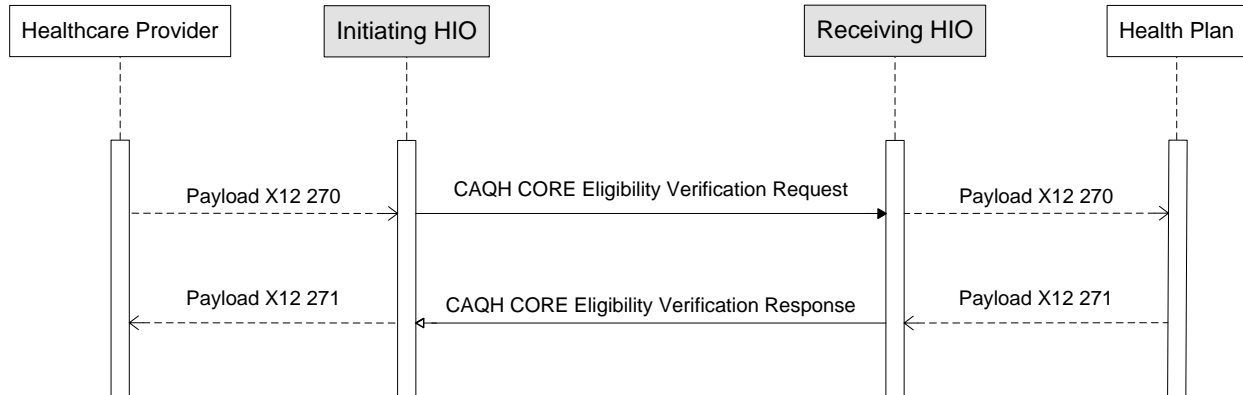
3 Interface Definition

3.1 CAQH CORE Eligibility Verification Transaction:

This transaction is described in CAQH CORE 270 Phase II Connectivity Rule Section 4. The figure below illustrates the actors and transactions involved in CORE SOAP + WSDL Real-time Request/Request transaction. Note that the Health Plan represents a Medicaid program.



NHIN Medicaid Eligibility Verification Web Service Interface Specification v 1.0



3.2 CORE Metadata

The required CORE Metadata is described in the table below. The schema definition for the required CORE Metadata can be found in Appendix C.

Element	Description	Field Name	Optionality	Data Type
Payload Type	The type of payload included within a request/response. This shall be "X12_270_004010X092A1" for a request and "X12_271_004010X092A1" for a response	PayloadType	R	Coded Set
Processing Mode	The mode of processing. This shall be "RealTime"	ProcessingMode	R	Coded Set
Payload ID	A payload ID assigned by the sender. This shall conform to ISO UUID standards with hexadecimal notation, generated using a combination of local timestamp as well as the hardware (MAC) address	PayloadID	R	String
Time Stamp	A single coordinated Universal Time (UTC) time stamp including time zone. This does not require a shared time server	TimeStamp	R	dateTime
Sender Identifier	A unique business entity (trading partner) identifier	SenderID	R	String
Receiver Identifier	A unique business entity (trading partner) identifier	ReceiverID	R	String
CORE Rule Version	The CORE Rule version that the envelope is using	CORERuleVersion	R	Coded Set
Error Code	The error code indicating the error when processing the envelope	ErrorCode	R	Coded Set
Error Message	Text error message	ErrorMessage	R	String



3.3 Connectivity Rule

This transaction is described in detail in CORE 270 Phase II Connectivity Rule Version 2.0.1 section 4.2.2.

3.4 SOAP + WSDL based Message Envelope

The SOAP + WSDL based method requires SOAP version 1.2 as specified by the NHIN Messaging Platform specification. There is no MITA specific requirement for the SOAP header. The SOAP body contains the remaining metadata defined by CORE Phase II compliant XML Schema Specification (see Appendix C). The message envelope structure is defined in the CORE Phase II WSDL file. (see Appendix B).

Note: CORE 270 Phase II Connectivity Rule requires that WS-Security Username and Password token must be added to the SOAP Header in the request for user authentication/authorization. This requirement is replaced with SAML Assertion based user authentication/authorization described in section 3.6 in this specification

3.5 Real-time (Real-time Mode, Real-time Processing Mode)

Real time requests must include a single inquiry or submission (one eligibility inquiry to one information source for one patient). In this model the response from the message receiver is either an error response or the corresponding 271 message response. Real-time Mode is when an entity is required to immediately send a single transaction and receive a single, 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. In this specification, a Medicaid HIO (Health Plan) receives a Real-time request directly from an Initiating HIO (Health Provider) and responds synchronously (as part of the same connection). Both request and response shall contain ProcessingMode element with field value "RealTime" to specify the real-time transaction mode as part of transaction metadata. Following is an excerpt from a sample Real-time request.

```
<nsl:COREEnvelopeRealTimeRequest
  xmlns:nsl="http://www.caqh.org/SOAP/WSDL/CORERule2.0.1.xsd">
  . . . . .
  <ProcessingMode>RealTime</ProcessingMode>
  . . . . .
</nsl:COREEnvelopeRealTimeRequest>
```

3.6 SAML Assertion based User Authentication and Authorization

For submitter authentication/authorization, CORE 270 Phase II Connectivity Rule specifies two methods: 1) Username/Password based authentication and 2) X.509 Certificate based authentication over SSL. This specification will use SAML Assertion instead for that purpose⁴.

⁴ This specification is constrained to demonstrate the use of SAML Assertions containing NPI attributes for user authentication/authorization. For the HIMSS demo, WS-Security Username and Password token based authentication will NOT be used. The HIMSS demonstration is expected to inform the Medicaid and MITA stakeholder community on the adequacy of SAML for this purpose and/or identify the need to incorporate WS-Security Username and Password token in subsequent specifications to meet the CORE 270 Phase II Connectivity Rule requirements.



The SAML Assertion will include additional <Attribute> elements required by MITA MMIS systems. The additional <Attribute> elements required by this specification include 1) National Provider Identifier (NPI) Attribute and 2) NPI Provider Name Attribute. These are further defined in subsequent sections of this specification.

Other elements required by this specification include 1) User Organization ID Attribute, 2) User Role Attribute and 3) Purpose of Use Attribute, all of which are defined in NHIN Authorization Framework Specification. User Authentication will be performed based on an NPI Attribute and NPI Provider Name Attribute pair against the NPPES by the State Medicaid MMIS system. State Medicaid system will validate requests per NPI Attribute, User Role Attribute, and Purpose of Use Attribute. Details on submitter authentication/authorization by the State Medicaid systems are out of scope of this specification.

3.6.1 National Provider Identifier (NPI)⁵ Attribute

This <Attribute> element shall have the Name attribute set to “urn:oasis:names:tc:xspa:1.0:subject:npi” and a NPI number shall be placed as a string in plain text in the value of the <AttributeValue> element. An example of the syntax of this element is as follows:

```
<saml:Attribute Name="urn:oasis:names:tc:xspa:1.0:subject:npi">  
  <saml:AttributeValue>1467433888</saml:AttributeValue>  
</saml:Attribute>
```

3.6.2 NPI Provider Name Attribute

This <Attribute> element shall have the Name attribute set to “urn:nhin:names:saml:npiProviderName” and a provider name, which is associated with an identifier provided in NPI <Attribute>, shall be placed as a string in plain text in the value of the <AttributeValue> element. An example of the syntax of this element is as follows:

```
<saml:Attribute Name="urn:nhin:names:saml:npiProviderName">  
  <saml:AttributeValue>Family Medical Clinic</saml:AttributeValue>  
</saml:Attribute>
```

4 Error Handling

This section follows error handling specified in the section 4.3.3 of the CORE 270 Phase II Connection Rule. The error codes relevant to the Medicaid Eligibility Verification are listed in the following table:

⁵ NPI is a US Government issued unique provider identifier required for all Health Insurance Portability and Accountability Act (HIPAA) Privacy Disclosure Accounting transactions.



NHIN Medicaid Eligibility Verification Web Service Interface
Specification
v 1.0

Error Codes	Description
<FieldName>Illegal	Illegal value provided for <FieldName>.
<FieldName>Required	The field <FieldName> is required but was not provided.
<FieldName>NotUnderstood	The field <FieldName> is not understood at the receiver. In the case of SOAP, this error is returned as a NotUnderstood SOAP fault.
VersionMismatch	The version of the envelope sent is not acceptable to the receiver. If the SOAP version is not valid at the receiver, a SOAP fault is returned with this fault code.
Unauthorized	The username/password or Client certificate could not be verified.
Sender	The envelope sent by the sender 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.
Receiver	The message could not be processed for reasons attributable to the Receiver (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.

5 Auditing

This transaction shall be audited as specified in the section 4.3.4 of the CORE 270 Phase II Connection Rule.



Appendix A: Sample Messages

Sample CORE SOAP + WSDL Real-time Request

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeRealTimeRequest
      xmlns:ns1="http://www.cagh.org/SOAP/WSDL/CORERule2.0.1.xsd">
      <PayloadType>X12_270_004010X092A1</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>2.0.1</CORERuleVersion>
      <Payload><![CDATA[ISA*00* *00* *ZZ*NEHEN780 *ZZ*NEHEN003 ...IEA*1*000000031]]></Payload>
    </ns1:COREEnvelopeRealTimeRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample CORE SOAP + WSDL Real-time Response

```
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Body>
    <ns1:COREEnvelopeRealTimeResponse
      xmlns:ns1="http://www.cagh.org/SOAP/WSDL/CORERule2.0.1.xsd">
      <PayloadType>X12_271_004010X092A1</PayloadType>
      <ProcessingMode>RealTime</ProcessingMode>
      <PayloadID>a81d44ae-7dec-11d0-a765-00a0c91e6ba0</PayloadID>
      <TimeStamp>2007-08-30T10:20:34Z</TimeStamp>
      <SenderID>PayerB</SenderID>
      <ReceiverID>HospitalA</ReceiverID>
      <CORERuleVersion>2.0.1</CORERuleVersion>
      <Payload><![CDATA[ISA*00* *00* *ZZ*NEHEN780 *ZZ*NEHEN003 ...IEA*1*000000031]]></Payload>
      <ErrorCode>Success</ErrorCode>
      <ErrorMessage></ErrorMessage>
    </ns1:COREEnvelopeRealTimeResponse>
  </soapenv:Body>
</soapenv:Envelope>
```



NHIN Medicaid Eligibility Verification Web Service Interface Specification v 1.0

Appendix B: WSDL

CAQH CORE Phase II Connectivity Rule provides a WSDL definition⁶. Since Batch Processing Mode is not supported in this specification, some of this WSDL (i.e. Batch Submission) shall not be used.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:CORE="http://www.caqh.org/SOAP/WSDL/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" \
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:CORE-XSD="http://www.caqh.org/SOAP/WSDL/CORERule2.0.1.xsd"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  name="CORE"
  targetNamespace="http://www.caqh.org/SOAP/WSDL/">
  <wsdl:types>
    <xsd:schema xmlns="http://schemas.xmlsoap.org/wsdl/"
      elementFormDefault="qualified"
      targetNamespace="http://www.caqh.org/SOAP/WSDL/">
      <xsd:import namespace="http://www.caqh.org/SOAP/WSDL/CORERule2.0.1.xsd"
        schemaLocation="http://www.caqh.org/SOAP/WSDL/CoreRule2.0.1.xsd"/>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="RealTimeRequestMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeRealTimeRequest"/>
  </wsdl:message>
  <wsdl:message name="RealTimeResponseMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeRealTimeResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchSubmission"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionResponseMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchSubmissionResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionAckRetrievalRequestMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchSubmissionAckRetrievalRequest"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionAckRetrievalResponseMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchSubmissionAckRetrievalResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsRetrievalRequestMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchResultsRetrievalRequest"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsRetrievalResponseMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchResultsRetrievalResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsAckSubmissionMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchResultsAckSubmission"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsAckSubmissionResponseMessage">
    <wsdl:part name="body" element="CORE-XSD:COREEnvelopeBatchResultsAckSubmissionResponse"/>
  </wsdl:message>
  <wsdl:portType name="CORETransactions">
    <wsdl:operation name="RealTimeTransaction">
      <wsdl:input message="CORE:RealTimeRequestMessage"/>
      <wsdl:output message="CORE:RealTimeResponseMessage"/>
    </wsdl:operation>
    <wsdl:operation name="BatchSubmitTransaction">
      <wsdl:input message="CORE:BatchSubmissionMessage"/>
      <wsdl:output message="CORE:BatchSubmissionResponseMessage"/>
    </wsdl:operation>
  </wsdl:portType>
</wsdl:definitions>
```

⁶ The CORE Phase II Connectivity WSDL can be downloaded at <http://www.caqh.org/SOAP/WSDL/CORERule2.0.1.wsdl>.



NHIN Medicaid Eligibility Verification Web Service Interface Specification v 1.0

```
<wsdl:operation name="BatchSubmitAckRetrievalTransaction">
  <wsdl:input message="CORE:BatchSubmissionAckRetrievalRequestMessage" />
  <wsdl:output message="CORE:BatchSubmissionAckRetrievalResponseMessage" />
</wsdl:operation>
<wsdl:operation name="BatchResultsRetrievalTransaction">
  <wsdl:input message="CORE:BatchResultsRetrievalRequestMessage" />
  <wsdl:output message="CORE:BatchResultsRetrievalResponseMessage" />
</wsdl:operation>
<wsdl:operation name="BatchResultsAckSubmitTransaction">
  <wsdl:input message="CORE:BatchResultsAckSubmissionMessage" />
  <wsdl:output message="CORE:BatchResultsAckSubmissionResponseMessage" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="CoreSoapBinding" type="CORE:CORETransactions">
  <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="RealTimeTransaction">
    <soap12:operation soapAction="RealTimeTransaction" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BatchSubmitTransaction">
    <soap12:operation soapAction="BatchSubmitTransaction" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BatchSubmitAckRetrievalTransaction">
    <soap12:operation soapAction="BatchSubmitAckRetrievalTransaction" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BatchResultsRetrievalTransaction">
    <soap12:operation soapAction="BatchResultsRetrievalTransaction" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="BatchResultsAckSubmitTransaction">
    <soap12:operation soapAction="BatchResultsAckSubmitTransaction" style="document" />
    <wsdl:input>
      <soap12:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap12:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="Core">
  <wsdl:port name="CoreSoapPort" binding="CORE:CoreSoapBinding">
    <soap12:address location="http://URL_OF_WEB_SERVICE" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```



Appendix C: CORE Phase II Compliant XML Schema Specification⁷

Since Batch Processing Mode is not supported in this specification, some of this schema (i.e. Batch Submission) shall not be used.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.cagh.org/SOAP/WSDL/CORERule2.0.1.xsd"
  targetNamespace="http://www.cagh.org/SOAP/WSDL/CORERule2.0.1.xsd">
  <xs:element name="COREEnvelopeRealTimeRequest">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string"/>
        <xs:element name="ProcessingMode" type="RealTimeMode"/>
        <xs:element name="PayloadID" type="xs:string"/>
        <xs:element name="TimeStamp" type="xs:string"/>
        <xs:element name="SenderID" type="xs:string"/>
        <xs:element name="ReceiverID" type="xs:string"/>
        <xs:element name="CORERuleVersion" type="xs:string"/>
        <xs:element name="Payload" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeRealTimeResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string"/>
        <xs:element name="ProcessingMode" type="RealTimeMode"/>
        <xs:element name="PayloadID" type="xs:string"/>
        <xs:element name="TimeStamp" type="xs:string"/>
        <xs:element name="SenderID" type="xs:string"/>
        <xs:element name="ReceiverID" type="xs:string"/>
        <xs:element name="CORERuleVersion" type="xs:string"/>
        <xs:element name="Payload" type="xs:string" minOccurs="0"/>
        <xs:element name="ErrorCode" type="xs:string" minOccurs="0"/>
        <xs:element name="ErrorMessage" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeBatchSubmission">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string"/>
        <xs:element name="ProcessingMode" type="BatchMode"/>
        <xs:element name="PayloadID" type="xs:string"/>
        <xs:element name="PayloadLength" type="xs:int" />
        <xs:element name="TimeStamp" type="xs:string"/>
        <xs:element name="SenderID" type="xs:string"/>
        <xs:element name="ReceiverID" type="xs:string"/>
        <xs:element name="CORERuleVersion" type="xs:string"/>
        <xs:element name="Checksum" type="xs:string" />
        <xs:element name="Payload" type="xs:base64Binary"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="COREEnvelopeBatchSubmissionResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PayloadType" type="xs:string"/>
        <xs:element name="ProcessingMode" type="BatchMode"/>
        <xs:element name="PayloadID" type="xs:string"/>
        <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
        <xs:element name="TimeStamp" type="xs:string"/>
        <xs:element name="SenderID" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

⁷ The schema specification file is available at <http://www.cagh.org/SOAP/WSDL/CORERule2.0.1.xsd>.



NHIN Medicaid Eligibility Verification Web Service Interface Specification v 1.0

```
<xs:element name="ReceiverID" type="xs:string"/>
<xs:element name="CORERuleVersion" type="xs:string"/>
<xs:element name="Checksum" type="xs:string" minOccurs="0"/>
<xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
<xs:element name="ErrorCode" type="xs:string" minOccurs="0"/>
<xs:element name="ErrorMessage" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="COREEnvelopeBatchSubmissionAckRetrievalRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PayloadType" type="xs:string"/>
      <xs:element name="ProcessingMode" type="BatchMode"/>
      <xs:element name="PayloadID" type="xs:string"/>
      <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
      <xs:element name="TimeStamp" type="xs:string"/>
      <xs:element name="SenderID" type="xs:string"/>
      <xs:element name="ReceiverID" type="xs:string"/>
      <xs:element name="CORERuleVersion" type="xs:string"/>
      <xs:element name="Checksum" type="xs:string" minOccurs="0"/>
      <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="COREEnvelopeBatchSubmissionAckRetrievalResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PayloadType" type="xs:string"/>
      <xs:element name="ProcessingMode" type="BatchMode"/>
      <xs:element name="PayloadID" type="xs:string"/>
      <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
      <xs:element name="TimeStamp" type="xs:string"/>
      <xs:element name="SenderID" type="xs:string"/>
      <xs:element name="ReceiverID" type="xs:string"/>
      <xs:element name="CORERuleVersion" type="xs:string"/>
      <xs:element name="Checksum" type="xs:string" minOccurs="0"/>
      <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
      <xs:element name="ErrorCode" type="xs:string" minOccurs="0"/>
      <xs:element name="ErrorMessage" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="COREEnvelopeBatchResultsRetrievalRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PayloadType" type="xs:string"/>
      <xs:element name="ProcessingMode" type="BatchMode"/>
      <xs:element name="PayloadID" type="xs:string"/>
      <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
      <xs:element name="TimeStamp" type="xs:string"/>
      <xs:element name="SenderID" type="xs:string"/>
      <xs:element name="ReceiverID" type="xs:string"/>
      <xs:element name="CORERuleVersion" type="xs:string"/>
      <xs:element name="Checksum" type="xs:string" minOccurs="0"/>
      <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="COREEnvelopeBatchResultsRetrievalResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PayloadType" type="xs:string"/>
      <xs:element name="ProcessingMode" type="BatchMode"/>
      <xs:element name="PayloadID" type="xs:string"/>
      <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
      <xs:element name="TimeStamp" type="xs:string"/>
      <xs:element name="SenderID" type="xs:string"/>
      <xs:element name="ReceiverID" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```



NHIN Medicaid Eligibility Verification Web Service Interface Specification v 1.0

```
<xs:element name="CORERuleVersion" type="xs:string"/>
<xs:element name="Checksum" type="xs:string" minOccurs="0"/>
<xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
<xs:element name="ErrorCode" type="xs:string" minOccurs="0"/>
<xs:element name="ErrorMessage" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="COREEnvelopeBatchResultsAckSubmission">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PayloadType" type="xs:string"/>
      <xs:element name="ProcessingMode" type="BatchMode"/>
      <xs:element name="PayloadID" type="xs:string"/>
      <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
      <xs:element name="TimeStamp" type="xs:string"/>
      <xs:element name="SenderID" type="xs:string"/>
      <xs:element name="ReceiverID" type="xs:string"/>
      <xs:element name="CORERuleVersion" type="xs:string"/>
      <xs:element name="Checksum" type="xs:string" minOccurs="0"/>
      <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="COREEnvelopeBatchResultsAckSubmissionResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PayloadType" type="xs:string"/>
      <xs:element name="ProcessingMode" type="BatchMode"/>
      <xs:element name="PayloadID" type="xs:string"/>
      <xs:element name="PayloadLength" type="xs:int" minOccurs="0"/>
      <xs:element name="TimeStamp" type="xs:string"/>
      <xs:element name="SenderID" type="xs:string"/>
      <xs:element name="ReceiverID" type="xs:string"/>
      <xs:element name="CORERuleVersion" type="xs:string"/>
      <xs:element name="Checksum" type="xs:string" minOccurs="0"/>
      <xs:element name="Payload" type="xs:base64Binary" minOccurs="0"/>
      <xs:element name="ErrorCode" type="xs:string" minOccurs="0"/>
      <xs:element name="ErrorMessage" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:simpleType name="RealTimeMode">
  <xs:restriction base="xs:string">
    <xs:pattern value="RealTime"/>
  </xs:restriction>
</xs:simpleType>
<xs:simpleType name="BatchMode">
  <xs:restriction base="xs:string">
    <xs:pattern value="Batch"/>
  </xs:restriction>
</xs:simpleType>
</xs:schema>
```