eMedNY CORE Web Services Connectivity Guide v.2.2

Version Number: 2.2

April 9, 2025

TABLE OF CONTENTS

For all eMedNY Enrollment or Billing questions, please contact the eMedNY Call Center through the contact information at the end of this document

This page left blank intentionally.

1. Introduction

eMedNY CORE Web Services facilitates the exchange of the *ASC X12N/005010X279A1 Health Care Eligibility Benefit Inquiry and Response (270/271)*, and the *ASC X12N/005010X212 Health Care Claim Status Request and Response (276/277)*, for both interactive (real-time) transactions and batch files.

The eMedNY CORE Web Services can also be used to retrieve the *ASC X12N/005010X221A1 Health Care Claim Payment/Advice (835)*, the *ASC X12N/005010X218 Payroll Deducted and Other Group Premium Payment for Insurance Products (820*), and the *ASC X12N/005010X220A1 834 Benefit Enrollment and Maintenance (834)* files. See **Section 6** for a full list of retrievable files.

PLEASE NOTE: eMedNY Core Web Services does not support 837 claim file submission. See Section 4 of this manual for supported and non-supported transactions in eMedNY CORE Web Services.

This document provides information on how to connect to the eMedNY CORE Web Service application using support of Simple Object Access Protocol + Web Services Description Language envelope standards (SOAP+WSDL) and support of Hypertext Transfer Protocol/Multipurpose Internet Mail Extensions (HTTP/MIME) Multi-part envelope standards.

The eMedNY CORE Web Services application follows the federally mandated *CAQH CORE Connectivity Rule Version C2.2.0*. For a copy of these federally mandated Operating Rules, please refer to:

https://www.caqh.org/core/connectivity

eMedNY CORE Web Services is available 24 hours a day, 7 days a week barring an outage or maintenance.

All questions regarding enrollment, connectivity and billing will go through the eMedNY Call Center at 1-800-343-9000. When necessary, a ticket will be escalated to a call back unit.

Important Note: It is solely the responsibility of the submitter or user to obtain/develop/create and maintain their own CORE Web Services compliant application. eMedNY will in no way support the end user application. Additionally, sample transactions are provided for your convenience, but eMedNY is unable to assist with the coding of your user application. It is strongly recommended that the Trading Partner take appropriate action to have available technical support.

2. Requirements and Limitations

2.1 General Information

- eMedNY CORE Web Services is available 24 hours a day, 7 days a week barring an outage or maintenance
- The submitter will need a client that is compliant with the CAQH CORE Connectivity vC2.2.0 outlined on the CAQH-CORE website. eMedNY cannot assist the user in coding this client.
- Please refer to the linked guides in Section 7 when creating/acquiring your application, client or software for CORE Web Services
- XML or WSDL sample files provided by CORE: CAQH CORE Connectivity vC2.2.0
- eMedNY has provided some samples in the Appendices
- eMedNY CORE Web Services cannot be accessed or utilized through a direct Portal URL or log-in
- Any submitter that will be using CORE Web Services must first enroll in the eMedNY Web Portal
- Once enrolled in the eMedNY Web Portal the user can then enroll in eMedNY CORE Web Services, through the user's eMedNY Web Portal account, and obtain a CORE Web ID (aka "System Username"). The user will then send their "System Username" credentials in the envelope of the user's CAQH-CORE application/client to access and utilize the eMedNY CORE Web Services application.
- **The use of the TA1 flag ('1') in the ISA14 is highly encouraged**

2.2 Testing

- **CORE Web Services has no X12 compliance transaction testing mechanism.**
- If the file is tagged as test (**'T'**) in the ISA15 one of two outcomes will happen:
  - ISA14 is set to **'0'** – eMedNY will not validate this file in any way – no response will be returned
  - ISA14 is set to **'1'** – eMedNY will return a TA1 file with a rejection code of ***'020'*** *– Invalid Test Indicator Value*
- Sending this 'test' file can be done with the intent of getting the TA1 rejection in order to test and verify that your CORE application/client connectivity to eMedNY is successful.

Testing the X12 transaction for compliance may **only** be done through submission to eMedNY eXchange. Please refer to the eMedNY Transaction Information Standard Companion Guide CAQH - CORE Companion Guide for additional information on testing the X12 transaction.

2.3 Enrollment in the eMedNY Web Portal

In order to connect to eMedNY CORE Web Services, the user **must have an eMedNY Web Portal account.** Enrollment in the Web Portal is explained in this document:

https://www.emedny.org/selfhelp/ProviderPortal/Enrolling_in_Web_Portal.pdf

2.4 Enrolling in eMedNY CORE Web Services

Once the submitter is enrolled in the Web Portal, use the following guide to create a System Username:

https://www.emedny.org/selfhelp/ProviderPortal/CORE_Web_Services_Enrollment.pdf

The “System Username” and password for eMedNY Core Web Services Transactions are sent in the envelope (MIME or SOAP) of the transaction – this supplies the required authentication. See the sample files in sections 11 and 12 for the usage of the username/password in the payload envelope.

2.5 Additional Assistance and Resources

For more information on accessing or enrolling in the eMedNY Web Portal/CORE Web Services; please visit our website www.emedny.org and use the Self Help link at the right-hand top of the page or by clicking the direct URL links provided here and in section 7.2.

On the Self Help page view the section titled “**Web Portal”** there are several user guides to assist with enrollment, accessing, maintenance and administration of the eMedNY Web Portal account and assistance with enrolling in the eMedNY CORE Web Services, through the user’s eMedNY Web Portal account. The guides also provide assistance with the functions and uses for the eMedNY Web Portal and eMedNY CORE Web Services.

If a user/submitter has additional questions or needs assistance with accessing or enrolling in the eMedNY Web Portal/CORE Web Services, please contact the **eMedNY Call Center at 1-800-343-9000** and choose the prompt for Provider Enrollment.

3. Payload Envelope Specifications

The envelope of the transaction must conform to CORE Version 2.2.0 standards as outlined on the CAQH-CORE website. eMedNY has samples of the envelope for the supported transactions in the Appendix section of this document.

The following items are specific to eMedNY Web Services

- The *System Username* and *Password* are the respective Username and Password of the System User sending the transaction. A System User is set up via the Web Portal – see Section 2 for more details.
- The *SenderID* should conform to the current CORE Version 2.2.0 standards
- The *ReceiverID* for the payload envelope is "eMedNY."
- The *PayloadID* conforms to the current CORE Version 2.2.0 standards – for Batch transactions it is then used to retrieve the responses to the inbound file.
- See Appendix II for SOAP + WSDL samples, and Appendix III for HTTP/MIME samples for the placement of the Username, Password, SenderID, ReceiverID and PayloadID

PLEASE NOTE: these instructions apply only to the envelope (SOAP or MIME) of the payload. The *Receiver ID/SenderID* within the X12 transaction (ISA06/ISA08) must meet the eMedNY specifications as outlined in section 6 of the eMedNY Trading Partner Information Standard Companion Guide.

4. File Formats and Limitations for Batch and Real-Time

Please refer to the eMedNY Transaction Information Standard Companion Guide CAQH - CORE Companion Guide for the X12 components specific to New York State Medicaid. The payload in the CORE envelope needs to contain a properly formatted X12 transaction in either 270 or 276 format.

4.1 Supported Inbound Transactions

eMedNY CORE Web Services accepts the following inbound X12 transaction types:

- **eMedNY Interactive 270** – Real-Time Eligibility Inquiry (*X12N/005010X279A1 Health Care Eligibility Benefit Inquiry and Response (270/271)*)
- **eMedNY Interactive 276** – Real-Time Claim Status Inquiry *(X12N/005010X212 Health Care Claim Status Request and Response (276/277))*
- **eMedNY Batch 270** – Batch Eligibility Inquiry (*X12N/005010X279A1 Health Care Eligibility Benefit Inquiry and Response (270/271)*)
- **eMedNY Batch 276** – Batch Claim Status Inquiry *(X12N/005010X212 Health Care Claim Status Request and Response (276/277))*

4.2 Real-Time Submission

270 Eligibility Inquiry: eMedNY expects no more than 1 Subscriber Levels (Loop ID 2000C) per Transaction Set (ST/SE).
276 Claim Status Inquiry: eMedNY expects no more than 1 Subscriber Levels (Loop ID 2000D) per Transaction Set (ST/SE).

4.3 Batch File Submission

270 Eligibility Inquiry: eMedNY expects no more than 5000 Subscriber Levels (Loop ID 2000C) per Transaction Set (ST/SE).
276 Claim Status Inquiry: eMedNY expects no more than 5000 Subscriber Levels (Loop ID 2000D) per Transaction Set (ST/SE).

eMedNY places no further limits on batch files. However, please keep in mind that the larger the file is the longer it will take to process in full.

4.4 Non-Supported Inbound Transactions

eMedNY does not support the following inbound transactions.

- 837 Professional, Dental, Institutional claim files
- 278 Requests
- NCPDP D.0
- Provider Affiliation Files
- Any other file not explicitly named in section 4.1

5. Responses to Inbound Files

When a file is transmitted through eMedNY CORE Web Services, an immediate response will be generated (unless the ISA15 has a Test indicator. See Section 2 for testing limitations). eMedNY responses to inbound files are as follows:

- TA1 Interchange Acknowledgment – ONLY returned if both the TA1 flag is set AND if there is an Interchange error to report.
- 999 *(X12C/005010X231A1 Implementation Acknowledgment for Health Care Insurance (999))*
- 271 *(X12N/005010X279A1 Health Care Eligibility Benefit Inquiry and Response (270/271))*
- 277 *(X12N/005010X212 Implementation Guide for Health Care Claim Status Response (277))*

5.1 Real-Time

The immediate response to a Real-Time Transaction will be the last level of validation for the inbound file.

- If there is an issue with the envelope (e.g. improper formatting or an issue with the Username and password,) the response will contain an envelope fault (SOAP or MIME).
- If the X12 packet fails the front-end editing, the response will contain a TA1 or Front-end rejection response.
- If the X12 packet fails syntax validation, the response will contain the 999.
- If the X12 packet passes, the response will be the information requested – either a 271 for an inbound 270, or a 277 for an inbound 276.

5.2 Batch

The immediate response to a Batch Transaction will be either an envelope error or a notification that the transaction was received. All of the other responses must be requested through Web Services – if they are available they will be returned as the response to the request. eMedNY uses the *Payload ID* from the request to correspond to the *Payload ID* in the original inbound file – it is incumbent on the user to know the *Payload ID* of their inbound files.

If the User submits multiple Functional Groups (GS-GE) in the inbound inquiry, we will return the responses (999 and 271 or 277) as Message Transmission Optimization Mechanism (MTOM) attachments. Each Functional Group will be its own attachment.

6. Retrieving eMedNY-Generated Files

eMedNY CORE Web Services can be utilized in retrieving a provider's X12 Remittance Advice (835 for most providers, 820 for Managed Care plans) and certain Managed Care encounter data files. If the provider designates their files to be sent through eMedNY Core Web Services, the following types of files can be sent through this method:

- 820 = 820 Premium Payment Advice
- 820S = 820 Premium Payment Advice Supplemental
- 835 = 835 Remittance Advice
- 835S = 835 Remittance Advice Supplemental
- 834 = 834 Benefit Enrollment and Maintenance (834R)
- 834 = 834 Managed Care Roster (834R)
- RXCO = Managed Care Organizations Data File (DWPX)
- ADRR = Data Warehouse First Fill File (DWXF)
- FFRR = Data Warehouse Adherence Report (DWXA)

Remittance files (820/820S and 835/835S) will be enclosed within a *.tar* file. This *.tar* file can include multiple files, and will include all files of that type sent to the Web Services User Name for that cycle.

The X12 formatting of this file is exactly the same as those retrieved by other methods. Please refer to the ASC X12 Standard Implementation Guides from X12 (www.x12.org) to obtain the correct file and record formats for 835 and 820 files.

The **eMedNY Transaction Information Standard Companion Guide CAQH - CORE CG X12** supplies the business rules specific to eMedNY 835 and 820 files.

The **834 Companion Guide** supplies business rules specific to eMedNY 834 files.

The **Pended Claims Report: Specification** supplies information on the formatting of the 835S.

The **Managed Care Capitation Premium Pended and Denied Claims Report: Specification** supplies information on the formatting of the 820S.

In order to designate these files to be sent to the eMedNY CORE Web Services System User Name, the provider must fill out the following form to designate the routing:

https://www.emedny.org/info/ProviderEnrollment/ProviderMaintForms/700201_ELECRMTREQ_ERa_PDF_Remit_Form.pdf

The **Web Portal** also has a copy of this form that can be filled out and transmitted electronically.

To download an eMedNY-Generated response file, the user will send a *"Request File List <payload type>"* transaction to retrieve a list of available files. They then can download the specific available files with *"Request <payload type>"* transactions. See the Appendices for samples.

7. Useful Links

This section provides links that users may need to enroll, access, code and general self-help documentation. Please note that most of the links below are also included elsewhere in this document.

7.1 CAQH-CORE CAQH CORE Infrastructure Operating Rules

https://www.caqh.org/core/operating-rules

https://www.caqh.org/core/connectivity

7.2 eMedNY Web Portal

https://www.emedny.org/selfhelp/

See Section labeled **“Web Portal”** for all quick reference sheets.

7.2.1 The Web Portal Log-In URL

https://portal.emedny.org/provider

7.2.2 Assistance enrollment in the Web Portal and activation of a “System User”

https://www.emedny.org/selfhelp/ProviderPortal/Enrolling_in_Web_Portal.pdf

https://www.emedny.org/selfhelp/ProviderPortal/CORE_Web_Services_Enrollment.pdf

7.3 X12 Information

7.3.1 eMedNY X12 envelope (ISA and GS) Information

https://www.emedny.org/HIPAA/5010/transactions/eMedNY_Trading_Partner_Information_CG.pdf

7.3.2 eMedNY Companion Guides for X12 Transactions

For information on the 270/271/276/277/835/820:

https://www.emedny.org/HIPAA/5010/transactions/eMedNY_Transaction_Information_CAQH-CORE_CG_X12_version_5010.pdf

For information on the 834:

https://www.emedny.org/HIPAA/5010/transactions/MCE_834_Companion_Guide.pdf

7.3.3 eMedNY Companion Guide for the 835S & 820S

For information on the 835S:

https://www.emedny.org/HIPAA/5010/transactions/Pended_Claims_Report_Specifcation.pdf

For information on the 820S:

https://www.emedny.org/HIPAA/5010/transactions/MCO_Premium_Claims_Pend_and_Deny%20Report_Specifcation.pdf

7.3.4 X12 Implementation Guides

https://x12.org/

7.4 Production URL for eMedNY CORE Web Services

https://caqh.emedny.org/eMedNY/EDIGateway

7.5 Useful eMedNY Forms

7.5.1 Certification Statement/Instructions for Existing ETINs

The ETIN (Electronic Transmitter Identification Number) must be renewed yearly for all electronic transactions

https://www.emedny.org/info/ProviderEnrollment/ProviderMaintForms/490501_ETIN_CERT_Certification_Statement_Cert_Instructions_for_Existing_ETINs.pdf

7.5.2 Electronic Remittance Advice Request

This form determines the routing of all eMedNY-Generated remittance files for the stated ETIN

https://www.emedny.org/info/ProviderEnrollment/ProviderMaintForms/700201_ELECRMTREQ_ERa_PDF_Remit_Form.pdf

7.6 eMedNY Call Center Phone Tree

https://www.emedny.org/selfhelp/ProvSrvcContact/callcentertelephonetree.aspx

8. Change Log

Date	Version	Modification
1/8/2025	**2.1**	• **Update to Section 3 to add information about the SenderID in the Payload Envelope** • **Updated the Important Note in Section 1 for a clearer message** • **Reiterated coding assistance note from section 1 to Appendix Sections 9-13** • **Cleaned up the change log; grammar, punctuation, order** • **Cleaned up formatting**
12/09/2024	2.0	Major revisions in this update are as follows: • Introduction – modification of the paragraphs and updated links to CAQH/CORE • Section 4 – complete section refresh • Section 6 – added references to the 834 and updated the list with additional information • Section 7 – complete section refresh; corrected outdated links, added more content • Appendix I – updated the SOAP Web Services Description Language (WSDL), removed references to a comparison of old version vs. new version • Appendix II – updated the Web Services Schema Definition), removed references to a comparison of old version vs. new version • Appendix III – corrected link to CORE • Appendix V - removed 'before update' and 'after update' references in sections 13.1, 13.2, 13.3 • Section 14 – New - Call Center details • There are several different fonts in the guide, updated to all one font (Calibri) • Grammar, formatting and typos corrected • Corrected references for 'Washington Publishing' to 'X12'
2/14/2023	1.6	Altered a few entries in 6. "Pharmacy Carve-out Report (DWPX)" was replaced with "Managed Care Organizations Data File (DWPX,)" and the following entries were added: "Data Warehouse First Fill File (DWXF,)" "Data Warehouse Adherence Report File (DWXA,)" "Managed Care Roster (834R File)". In 11 the file types "MEDA" and "MEDS" were removed and the above were inserted.
5/15/2020	1.5	Made more changes to the samples to reflect the responses from the new system – removed a few samples as they can be extrapolated from the CAQH-CORE information.
1/17/2020	1.4	Updated all the samples to reflect an update in the responses. Added Appendix V to give "before" and "after" examples of the code changes due to a server migration. Added this Change Log.
4/12/2017	1.3	Added 834 files to file types. Changed some language on the end page.
9/3/2014	1.2	Minor revisions to the language of the document, changed company name, updated links.

6/23/2014	1.1	Initial Version

NOTE: Sample transactions are provided for your convenience; eMedNY is unable to assist with the coding of your user application.

9. Appendix I – SOAP Web Services Description Language (WSDL) Updated 12/04/2024

The Web Services Description Language (WSDL) for SOAP follows:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wsdl:definitions name="eMedNYCORERule"

targetNamespace="http://emedny.org/CORERule.wsdl"
   xmlns:tns="http://emedny.org/CORERule.wsdl"
   xmlns:sch="http://emedny.org/CORERule.xsd"
   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap12/"
   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
   xmlns:wsp="http://www.w3.org/ns/ws-policy"
   xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">

  <wsdl:documentation>
   eMedNY CORE Rule

   Version: 1.2

   Change History:
   09/16/13 -- Version 1.0 (CORE Rule Version 2.2.0)
   11/22/13 -- Version 1.1 Added Generic Transactions
   08/09/19 -- Version 1.2 Added changes for CXF web service
  </wsdl:documentation>

  <!-- ================================================================= -->
  <!-- TYPE DEFINITIONS                                        -->
  <!-- List of types participating in this BPEL process .              -->
  <!-- The BPEL Designer will generate default request and response      -->
  <!-- types but you can define or import any XML Schema type and use    -->
  <!-- them as part of the message types.                            -->
  <!-- ================================================================= -->
  <wsdl:types>
   <xsd:schema>
    <xsd:import namespace="http://emedny.org/CORERule.xsd" schemaLocation="eMedNYCORERule.xsd"/>
   </xsd:schema>
   <xsd:schema targetNamespace="http://emedny.org/CORERule.wsdl">
    <xsd:element name="invokeFault">
     <xsd:complexType>
      <xsd:sequence>
       <xsd:element name="invokeFault" type="xsd:string"/>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
   </xsd:schema>
  </wsdl:types>

  <!-- ================================================================ -->
  <!-- MESSAGE TYPE DEFINITION                                          -->
  <!-- Definition of the message types used as part of the port type    -->
  <!-- type definitions                                                 -->
  <!-- ================================================================ -->
  <wsdl:message name="RealTimeRequestMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeRealTimeRequest"/>
  </wsdl:message>
  <wsdl:message name="RealTimeResponseMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeRealTimeResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchSubmission"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionResponseMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchSubmissionResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionAckRetrievalRequestMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchSubmissionAckRetrievalRequest"/>
  </wsdl:message>
  <wsdl:message name="BatchSubmissionAckRetrievalResponseMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchSubmissionAckRetrievalResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsRetrievalRequestMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchResultsRetrievalRequest"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsRetrievalResponseMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchResultsRetrievalResponse"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsAckSubmissionMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchResultsAckSubmission"/>
  </wsdl:message>
  <wsdl:message name="BatchResultsAckSubmissionResponseMessage">
   <wsdl:part name="body" element="sch:COREEnvelopeBatchResultsAckSubmissionResponse"/>
  </wsdl:message>
  <wsdl:message name="invokeFault">
   <wsdl:part name="parameters" element="tns:invokeFault"/>
  </wsdl:message>
```

```
<!-- ================================================================== -->
<!-- PORT TYPE DEFINITION                                     -->
<!-- A port type groups a set of operations into a logical service     -->
<!-- unit.                                                    -->
<!-- ================================================================== -->
<wsdl:portType name="CORETransactions">
 <wsdl:operation name="RealTimeTransaction">
  <wsdl:input message="tns:RealTimeRequestMessage"/>
  <wsdl:output message="tns:RealTimeResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
 <wsdl:operation name="BatchSubmitTransaction">
  <wsdl:input message="tns:BatchSubmissionMessage"/>
  <wsdl:output message="tns:BatchSubmissionResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
 <wsdl:operation name="BatchSubmitAckRetrievalTransaction">
  <wsdl:input message="tns:BatchSubmissionAckRetrievalRequestMessage"/>
  <wsdl:output message="tns:BatchSubmissionAckRetrievalResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
 <wsdl:operation name="BatchResultsRetrievalTransaction">
  <wsdl:input message="tns:BatchResultsRetrievalRequestMessage"/>
  <wsdl:output message="tns:BatchResultsRetrievalResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
 <wsdl:operation name="BatchResultsAckSubmitTransaction">
  <wsdl:input message="tns:BatchResultsAckSubmissionMessage"/>
  <wsdl:output message="tns:BatchResultsAckSubmissionResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
 <wsdl:operation name="GenericBatchSubmissionTransaction">
  <wsdl:input message="tns:BatchSubmissionMessage"/>
  <wsdl:output message="tns:BatchSubmissionResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
 <wsdl:operation name="GenericBatchRetrievalTransaction">
  <wsdl:input message="tns:BatchResultsRetrievalRequestMessage"/>
  <wsdl:output message="tns:BatchResultsRetrievalResponseMessage"/>
  <wsdl:fault name="fault" message="tns:invokeFault"/>
 </wsdl:operation>
</wsdl:portType>

<!-- ================================================================== -->
<!-- BINDING DEFINITION                                       -->
<!-- ================================================================== -->
<wsdl:binding name="CoreSoapBinding" type="tns:CORETransactions">
```

```
 <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
 <wsdl:operation name="RealTimeTransaction">
  <soap:operation soapAction="RealTimeTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
 <wsdl:operation name="BatchSubmitTransaction">
  <soap:operation soapAction="BatchSubmitTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
 <wsdl:operation name="BatchSubmitAckRetrievalTransaction">
  <soap:operation soapAction="BatchSubmitAckRetrievalTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
 <wsdl:operation name="BatchResultsRetrievalTransaction">
  <soap:operation soapAction="BatchResultsRetrievalTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
 <wsdl:operation name="BatchResultsAckSubmitTransaction">
  <soap:operation soapAction="BatchResultsAckSubmitTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
 <wsdl:operation name="GenericBatchSubmissionTransaction">
  <soap:operation soapAction="GenericBatchSubmissionTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
 <wsdl:operation name="GenericBatchRetrievalTransaction">
  <soap:operation soapAction="GenericBatchRetrievalTransaction" style="document"/>
  <wsdl:input><soap:body use="literal"/></wsdl:input>
  <wsdl:output><soap:body use="literal"/></wsdl:output>
  <wsdl:fault name="fault"><soap:fault name="fault" use="literal"/></wsdl:fault>
 </wsdl:operation>
</wsdl:binding>

<!-- ============================================================ -->
<!-- SERVICE DEFINITION                                     -->
<!-- ============================================================ -->
```

```
  <wsdl:service name="CoreService">
    <wsdl:port name="CoreSoapPort" binding="tns:CoreSoapBinding">
      <soap:address location="https://caqh.emedny.org/eMedNY/EDIGateway"/>
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>
```

NOTE: Sample transactions are provided for your convience; eMedNY is unable to assist with the coding of your user application.

10. Appendix II – Web Services Schema Definition Updated 12/04/2024

The XML Web Services Schema Definitions follow:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified"
 targetNamespace="http://emedny.org/CORERule.xsd"
 xmlns:sch="http://emedny.org/CORERule.xsd">

 <xsd:annotation>
  <xsd:documentation>
  eMedNY CORE Rule

  Version: 1.3

  Change History:
  09/16/13 -- Original Version 1.0 (CORE Rule Version 2.2.0)
  01/16/14 -- Version 1.1 Added payload types for retrieval of remittance advice, premium payment advice,
          835 supplemental, 820 supplemental, encounters, and pharmacy carve-out.
  06/04/15 -- Version 1.2 Added payload types for X12 834 enrollments and effectuations.
  08/09/19 -- Version 1.3 Added changes for CXF web service.
  </xsd:documentation>
 </xsd:annotation>

 <!--#############################################################-->
 <!--#                                    #-->
 <!--#            Root Nodes                #-->
 <!--#                                    #-->
 <!--#############################################################-->

 <xsd:element name="COREEnvelopeRealTimeRequest">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="PayloadType" type="sch:SubmissionPayloadType" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ProcessingMode" type="sch:RealTimeMode" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="Payload" type="xsd:string" minOccurs="1" maxOccurs="1"/>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>

 <xsd:element name="COREEnvelopeRealTimeResponse">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ProcessingMode" type="sch:RealTimeMode" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="Payload" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ErrorCode" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ErrorMessage" type="xsd:string" minOccurs="1" maxOccurs="1"/>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>

 <xsd:element name="COREEnvelopeBatchSubmission">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="PayloadType" type="sch:SubmissionPayloadType" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PayloadLength" type="xsd:int" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="1" maxOccurs="1"/>
   </xsd:sequence>
  </xsd:complexType>
 </xsd:element>

 <xsd:element name="COREEnvelopeBatchSubmissionResponse">
  <xsd:complexType>
   <xsd:sequence>
    <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
    <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
```

```
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ErrorCode" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ErrorMessage" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeBatchSubmissionAckRetrievalRequest">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PayloadType" type="sch:AcknowledgementRetrievalPayloadType" minOccurs="1"
maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeBatchSubmissionAckRetrievalResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ErrorCode" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ErrorMessage" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeBatchResultsRetrievalRequest">
    <xsd:complexType>
      <xsd:sequence>
```

```
        <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeBatchResultsRetrievalResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ErrorCode" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ErrorMessage" type="xsd:string" minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeBatchResultsAckSubmission">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PayloadType" type="sch:AcknowledgementSubmissionPayloadType" minOccurs="1"
maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeBatchResultsAckSubmissionResponse">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadLength" type="xsd:int" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CheckSum" type="sch:Sha1Checksum" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="Payload" type="xsd:base64Binary" minOccurs="0" maxOccurs="1"/>
        <xsd:element name="ErrorCode" type="xsd:string"  minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ErrorMessage" type="xsd:string"  minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="COREEnvelopeError">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="PayloadType" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="PayloadID" type="sch:IsoUuid" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="TimeStamp" type="xsd:dateTime" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="SenderID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ReceiverID" type="sch:CoreString" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="CORERuleVersion" type="sch:CoreRuleVersion" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ErrorCode" type="xsd:string"  minOccurs="1" maxOccurs="1"/>
        <xsd:element name="ErrorMessage" type="xsd:string"  minOccurs="1" maxOccurs="1"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <!--############################################################-->
  <!--#                                          #-->
  <!--#              Simple Types                      #-->
  <!--#                                          #-->
  <!--############################################################-->

  <xsd:simpleType name="IsoUuid">
    <xsd:annotation>
      <xsd:documentation>
        ISO UUID standard (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, to ensure uniqueness.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"></xsd:pattern>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="Sha1Checksum">
    <xsd:annotation>
      <xsd:documentation>
        Algorithm is SHA-1, Encoding is Hex. Checksum must be computed only on the payload
        and not on the metadata.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[0-9a-fA-F]{40}"></xsd:pattern>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="CoreString">
    <xsd:annotation>
      <xsd:documentation>
        Maximum length 50 characters.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="CoreRuleVersion">
    <xsd:annotation>
      <xsd:documentation>
        CORE Rule Version = 2.2.0.
      </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="2.2.0"></xsd:pattern>
    </xsd:restriction>
  </xsd:simpleType>

  <xsd:simpleType name="RealTimeMode">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="RealTime"/>
    </xsd:restriction>
```

```
</xsd:simpleType>

<xsd:simpleType name="BatchMode">
 <xsd:restriction base="xsd:string">
  <xsd:pattern value="Batch"/>
 </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="SubmissionPayloadType">
 <xsd:annotation>
  <xsd:documentation>
   Submission payload types.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:token">
  <xsd:enumeration value="X12_270_Request_005010X279A1"></xsd:enumeration>
  <xsd:enumeration value="X12_276_Request_005010X212"></xsd:enumeration>
  <xsd:enumeration value="X12_005010_Request_Effectuation_Submit_834"></xsd:enumeration>
  <xsd:enumeration value="X12_005010_Request_BatchSubmissionMixed"></xsd:enumeration>
 </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="AcknowledgementRetrievalPayloadType">
 <xsd:annotation>
  <xsd:documentation>
   Acknowledgement retrieval payload types.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:token">
  <xsd:enumeration value="X12_999_RetrievalRequest_005010X231A1"></xsd:enumeration>
  <xsd:enumeration value="X12_TA1_RetrievalRequest_00501X231A1"></xsd:enumeration>
 </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="ResultsRetrievalPayloadType">
 <xsd:annotation>
  <xsd:documentation>
   Results retrieval payload types.
  </xsd:documentation>
 </xsd:annotation>
 <xsd:restriction base="xsd:token">
  <xsd:enumeration value="X12_005010_Request_Batch_Results_271"></xsd:enumeration>
  <xsd:enumeration value="X12_005010_Request_Batch_Results_277"></xsd:enumeration>
  <xsd:enumeration value="X12_005010_Request_BatchResultsMixed"></xsd:enumeration>
  <xsd:enumeration value="X12_999_RetrievalRequest_005010X231A1"></xsd:enumeration>
  <xsd:enumeration value="X12_820_Request_005010X218A1"></xsd:enumeration>
  <xsd:enumeration value="X12_005010_Request_Enrollment_Retrieve_834"></xsd:enumeration>
  <xsd:enumeration value="X12_835_Request_005010X221A1"></xsd:enumeration>
```

```
   <xsd:enumeration value="Request_820S"></xsd:enumeration>
   <xsd:enumeration value="Request_835S"></xsd:enumeration>
   <xsd:enumeration value="Request_DWPX"></xsd:enumeration>
   <xsd:enumeration value="Request_DWXA"></xsd:enumeration>
   <xsd:enumeration value="Request_DWXF"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_820"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_834"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_835"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_820S"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_835S"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_DWPX"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_DWXA"></xsd:enumeration>
   <xsd:enumeration value="Request_File_List_DWXF"></xsd:enumeration>
  </xsd:restriction>
 </xsd:simpleType>

 <xsd:simpleType name="AcknowledgementSubmissionPayloadType">
  <xsd:annotation>
   <xsd:documentation>
    Acknowledgement submission payload types.
   </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="xsd:token">
   <xsd:enumeration value="X12_999_SubmissionRequest_005010X231A1"></xsd:enumeration>
   <xsd:enumeration value="X12_TA1_SubmissionRequest_00501X231A1"></xsd:enumeration>
  </xsd:restriction>
 </xsd:simpleType>

</xsd:schema>
```

NOTE: Sample transactions are provided for your convience; eMedNY is unable to assist with the coding of your user application.

11. Appendix III - SOAP XML Request and Response Samples – updated 1/7/2020

General examples can be found at the following link:

https://www.caqh.org/hubfs/43908627/drupal/core/phase-ii/policy-rules/Connectivity-Rule-vC220.pdf?token=bNhpo5kH

The following are actual samples from the eMedNY Instance of CAQH/CORE:

11.1 Real-Time Examples

The following are examples of Real-Time transactions. The examples are 270/271, but a 276 can be sent by changing the payload type.

11.1.1 SOAP XML Real-Time Request Example

```
POST /eMedNY/EDIGateway HTTP/1.1
Content-Type: application/soap+xml; action="RealTimeTransaction"

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Header>
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" soap:mustUnderstand="true">
<wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-21621663">
<wsse:Username>******</wsse:Username>
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">******</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soap:Header>
<soap:Body>
<cor:COREEnvelopeRealTimeRequest xmlns:cor="http://emedny.org/CORERule.xsd">
<cor:PayloadType>X12_270_Request_005010X279A1</cor:PayloadType>
<cor:ProcessingMode>RealTime</cor:ProcessingMode>
<cor:PayloadID>60000000-0000-0000-0000-000000000005</cor:PayloadID>
<cor:TimeStamp>2007-08-30T10:20:34Z</cor:TimeStamp>
<cor:SenderID>HospitalA</cor:SenderID>
<cor:ReceiverID>eMedNY</cor:ReceiverID>
<cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
<cor:Payload><![CDATA[ISA*00…~]]></cor:Payload>
</cor:COREEnvelopeRealTimeRequest>
</soap:Body>
</soap:Envelope>
```

11.1.2 SOAP XML Real-Time Response Example

```
HTTP/1.1 200
Content-Type: application/soap+xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 12 May 2020 19:23:41 GMT

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
```

```
   <ns3:COREEnvelopeRealTimeResponse xmlns:ns1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:ns2="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:ns3="http://emedny.org/CORERule.xsd" xmlns:ns5="http://emedny.org/CORERule.wsdl">
     <ns3:PayloadType>X12_271_Response_005010X279A1</ns3:PayloadType>
     <ns3:ProcessingMode>RealTime</ns3:ProcessingMode>
     <ns3:PayloadID>60000000-0000-0000-0000-000000000005</ns3:PayloadID>
     <ns3:TimeStamp>2020-05-12T19:23:41Z</ns3:TimeStamp>
     <ns3:SenderID>eMedNY</ns3:SenderID>
     <ns3:ReceiverID>HospitalA</ns3:ReceiverID>
     <ns3:CORERuleVersion>2.2.0</ns3:CORERuleVersion>
     <ns3:Payload><![CDATA[ISA*00…~]]></ns3:Payload>
     <ns3:ErrorCode>Success</ns3:ErrorCode>
     <ns3:ErrorMessage>None</ns3:ErrorMessage>
   </ns3:COREEnvelopeRealTimeResponse>
  </soap:Body>
</soap:Envelope>
```

11.2 Batch Examples

The following are examples of Batch transactions. The examples are 270/271, but a 276 can be sent by changing the payload type.

11.2.1 SOAP XML Batch Request Example with MTOM

```
POST /eMedNY/EDIGateway HTTP/1.1
Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml"; action="GenericBatchRetrievalTransaction"; boundary="MIMEBoundaryAbCd"

--MIMEBoundaryAbCd
Content-Type: application/xop+xml; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <A1>

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Header>
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" soap:mustUnderstand="true">
<wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="UsernameToken-21621663">
<wsse:Username>******</wsse:Username>
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">******</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
</soap:Header>
<soap:Body>
<cor:COREEnvelopeBatchResultsRetrievalRequest xmlns:cor="http://emedny.org/CORERule.xsd">
```

```
<cor:PayloadType>Request_File_List_DWPX</cor:PayloadType>
<cor:ProcessingMode>Batch</cor:ProcessingMode>
<cor:PayloadID>AAA00000-0000-0000-0000-000000000000</cor:PayloadID>
<cor:TimeStamp>2020-03-30T10:20:34Z</cor:TimeStamp>
<cor:SenderID>HospitalA</cor:SenderID>
<cor:ReceiverID>eMedNY</cor:ReceiverID>
<cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
</cor:COREEnvelopeBatchResultsRetrievalRequest>
</soap:Body>
</soap:Envelope>
--MIMEBoundaryAbCd--
```

11.2.2 SOAP XML Batch Response Example with MTOM

```
HTTP/1.1 200
Date: Tue, 31 Mar 2020 13:06:04 GMT
Server: Apache/2.4.6 (Red Hat Enterprise Linux) OpenSSL/1.0.2k-fips mod_auth_gssapi/1.5.1 mod_nss/1.0.14 NSS/3.28.4 PHP/5.4.16 mod_wsgi/3.4 Python/2.7.5
Content-Type: multipart/related; type="application/xop+xml"; boundary="uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1"; start="<root.message@cxf.apache.org>"; start-info="application/soap+xml"
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked

--uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
   <ns3:COREEnvelopeBatchResultsRetrievalResponse xmlns:ns1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:ns2="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:ns3="http://emedny.org/CORERule.xsd" xmlns:ns5="http://emedny.org/CORERule.wsdl">
      <ns3:PayloadType>Response_File_List_DWPX</ns3:PayloadType>
      <ns3:ProcessingMode>Batch</ns3:ProcessingMode>
      <ns3:PayloadID>AAA00000-0000-0000-0000-000000000000</ns3:PayloadID>
      <ns3:PayloadLength>72</ns3:PayloadLength>
      <ns3:TimeStamp>2020-03-31T13:06:04Z</ns3:TimeStamp>
      <ns3:SenderID>eMedNY</ns3:SenderID>
      <ns3:ReceiverID>HospitalA</ns3:ReceiverID>
      <ns3:CORERuleVersion>2.2.0</ns3:CORERuleVersion>
      <ns3:CheckSum>62bb53f35ff9649155bff5f4bf9baf753517b70f</ns3:CheckSum>
      <ns3:Payload>
```

```
        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:abc54148-29fd-43d9-96fd-8daed0429f52-45@emedny.org"/>
      </ns3:Payload>
      <ns3:ErrorCode>Success</ns3:ErrorCode>
      <ns3:ErrorMessage>None</ns3:ErrorMessage>
    </ns3:COREEnvelopeBatchResultsRetrievalResponse>
  </soap:Body>
</soap:Envelope>

--uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <abc54148-29fd-43d9-96fd-8daed0429f52-45@emedny.org>

<list><payloadid>20190802-0900-4600-0000-000000000001</payloadid></list>
--uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1--
```

NOTE: Sample transactions are provided for your convience; eMedNY is unable to assist with the coding of your user application.

12. Appendix IV – Web Services MIME Request and Response Samples

General examples can be found at the following link:

http://www.caqh.org/sites/default/files/core/phase-ii/policy-rules/270-v5010.pdf

The following are actual samples from the eMedNY Instance of the system:

12.1 eMedNY-Generated Response File Examples

These are examples of requesting a file list of and requesting an eMedNY-Generated response file. The examples use an 835 file, but this can be varied by changing the payload type.

12.1.1 MIME Request File List 835 Example

```
POST /eMedNY/EDIGateway HTTP/1.1
Content-Type: multipart/form-data; boundary="MIMEBoundaryAbCd"

--MIMEBoundaryAbCd
Content-Disposition: form-data; name="PayloadType"

Request_File_List_835
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="ProcessingMode"

Batch
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="PayloadID"

20190802-0901-4600-0000-000000000001
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="TimeStamp"

2007-08-30T10:20:34Z
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="UserName"

******
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="Password"

******
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="SenderID"
```

```
HospitalA
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="ReceiverID"

eMedNY
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="CORERuleVersion"

2.2.0
--MIMEBoundaryAbCd--
```

Response

```
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D--
```

12.1.2 MIME Request File List 835 Example

```
HTTP/1.1 200
Server: nginx
Date: Tue, 12 May 2020 18:42:23 GMT
Content-Type: multipart/form-data; boundary=lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D;charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive

--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="PayloadType"
Content-Type: text/plain;charset=UTF-8
Content-Length: 22

Response_File_List_835
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="ProcessingMode"
Content-Type: text/plain;charset=UTF-8
Content-Length: 5

Batch
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="PayloadID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 36

20190802-0901-4600-0000-000000000001
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="PayloadLength"
Content-Type: text/plain;charset=UTF-8
Content-Length: 2
```

```
13
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="TimeStamp"
Content-Type: text/plain;charset=UTF-8
Content-Length: 20

2020-05-12T18:42:19Z
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="SenderID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 6

eMedNY
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="ReceiverID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 9

HospitalA
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="CORERuleVersion"
Content-Type: text/plain;charset=UTF-8
Content-Length: 5

2.2.0
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="CheckSum"
Content-Type: text/plain;charset=UTF-8
Content-Length: 40

2ae0811dcbdcf48b1e31a5cbbbf60e2c29328a26
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="Payload"
Content-Type: text/plain;charset=UTF-8
Content-Length: 13

<list></list>
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="ErrorCode"
Content-Type: text/plain;charset=UTF-8
Content-Length: 7

Success
--lwbkInuxo8setmgQs28mTjVG9_zwL2ors5CnSD4D
Content-Disposition: form-data; name="ErrorMessage"
Content-Type: text/plain;charset=UTF-8
Content-Length: 4
```

None

12.1.3 MIME Request 835 file

```
POST /eMedNY/EDIGateway HTTP/1.1
Content-Type: multipart/form-data; boundary="MIMEBoundaryAbCd"

--MIMEBoundaryAbCd
Content-Disposition: form-data; name="PayloadType"

X12_835_Request_005010X221A1
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="ProcessingMode"

Batch
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="PayloadID"

20190802-0901-4600-0000-000000000001
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="TimeStamp"

2007-08-30T10:20:34Z
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="UserName"

******
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="Password"

******
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="SenderID"

HospitalA
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="ReceiverID"

eMedNY
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="CORERuleVersion"

2.2.0
--MIMEBoundaryAbCd--
```

12.1.4 X12 835 Response 005010X221A1

```
HTTP/1.1 200
Content-Type: multipart/form-data; boundary=WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 12 May 2020 19:15:54 GMT

--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="PayloadType"
Content-Type: text/plain;charset=UTF-8
Content-Length: 29

X12_835_Response_005010X221A1
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="ProcessingMode"
Content-Type: text/plain;charset=UTF-8
Content-Length: 5

Batch
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="PayloadID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 36

20190802-0901-4600-0000-000000000001
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="PayloadLength"
Content-Type: text/plain;charset=UTF-8
Content-Length: 5

10240
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="TimeStamp"
Content-Type: text/plain;charset=UTF-8
Content-Length: 20

2020-05-12T19:15:54Z
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="SenderID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 6

eMedNY
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="ReceiverID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 9
```

```
HospitalA
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="CORERuleVersion"
Content-Type: text/plain;charset=UTF-8
Content-Length: 5

2.2.0
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="CheckSum"
Content-Type: text/plain;charset=UTF-8
Content-Length: 40

4fb722c683f49b035241f7cc731004e2f646ef5e
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="Payload"; filename="R160629150126.2025.835-.tar"
Content-Type: application/x-tar
Content-Length: 10240

R160629150116.2025.835.0000.x12...
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="ErrorCode"
Content-Type: text/plain;charset=UTF-8
Content-Length: 7

Success
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd
Content-Disposition: form-data; name="ErrorMessage"
Content-Type: text/plain;charset=UTF-8
Content-Length: 4

None
--WCEXufLrZ8MpOyxDWa7bkyco6U3jidwO-jhLd--
```

NOTE: Sample transactions are provided for your convience; eMedNY is unable to assist with the coding of your user application.

13. Appendix V - Responses

13.1 SOAP Responses

```
HTTP/1.1 200
Content-Type: application/soap+xml;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 12 May 2020 19:23:41 GMT

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns3:COREEnvelopeRealTimeResponse xmlns:ns1="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:ns2="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-
1.0.xsd" xmlns:ns3="http://emedny.org/CORERule.xsd" xmlns:ns5="http://emedny.org/CORERule.wsdl">
      <ns3:PayloadType>X12_271_Response_005010X279A1</ns3:PayloadType>
      <ns3:ProcessingMode>RealTime</ns3:ProcessingMode>
      <ns3:PayloadID>60000000-0000-0000-0000-000000000005</ns3:PayloadID>
      <ns3:TimeStamp>2020-05-12T19:23:41Z</ns3:TimeStamp>
      <ns3:SenderID>eMedNY</ns3:SenderID>
      <ns3:ReceiverID>HospitalA</ns3:ReceiverID>
      <ns3:CORERuleVersion>2.2.0</ns3:CORERuleVersion>
      <ns3:Payload><![CDATA[ISA*00…~]]></ns3:Payload>
      <ns3:ErrorCode>Success</ns3:ErrorCode>
      <ns3:ErrorMessage>None</ns3:ErrorMessage>
    </ns3:COREEnvelopeRealTimeResponse>
  </soap:Body>
</soap:Envelope>
```

13.2 SOAP MTOM Response

```
HTTP/1.1 200
Date: Tue, 31 Mar 2020 13:06:04 GMT
Server: Apache/2.4.6 (Red Hat Enterprise Linux) OpenSSL/1.0.2k-fips mod_auth_gssapi/1.5.1 mod_nss/1.0.14
NSS/3.28.4 PHP/5.4.16 mod_wsgi/3.4 Python/2.7.5
Content-Type: multipart/related; type="application/xop+xml"; boundary="uuid:cf568db6-d575-412b-a6fb-
acd4cb00efd1"; start="<root.message@cxf.apache.org>"; start-info="application/soap+xml"
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked

--uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <root.message@cxf.apache.org>

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
```

```
  <soap:Body>
   <ns3:COREEnvelopeBatchResultsRetrievalResponse xmlns:ns1="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-utility-1.0.xsd" xmlns:ns2="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:ns3="http://emedny.org/CORERule.xsd"
xmlns:ns5="http://emedny.org/CORERule.wsdl">
     <ns3:PayloadType>Response_File_List_DWPX</ns3:PayloadType>
     <ns3:ProcessingMode>Batch</ns3:ProcessingMode>
     <ns3:PayloadID>AAA00000-0000-0000-0000-000000000000</ns3:PayloadID>
     <ns3:PayloadLength>72</ns3:PayloadLength>
     <ns3:TimeStamp>2020-03-31T13:06:04Z</ns3:TimeStamp>
     <ns3:SenderID>eMedNY</ns3:SenderID>
     <ns3:ReceiverID>HospitalA</ns3:ReceiverID>
     <ns3:CORERuleVersion>2.2.0</ns3:CORERuleVersion>
     <ns3:CheckSum>62bb53f35ff9649155bff5f4bf9baf753517b70f</ns3:CheckSum>
     <ns3:Payload>
       <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:abc54148-29fd-43d9-96fd-
8daed0429f52-45@emedny.org"/>
     </ns3:Payload>
     <ns3:ErrorCode>Success</ns3:ErrorCode>
     <ns3:ErrorMessage>None</ns3:ErrorMessage>
   </ns3:COREEnvelopeBatchResultsRetrievalResponse>
  </soap:Body>
</soap:Envelope>

--uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <abc54148-29fd-43d9-96fd-8daed0429f52-45@emedny.org>

<list><payloadid>20190802-0900-4600-0000-000000000001</payloadid></list>
--uuid:cf568db6-d575-412b-a6fb-acd4cb00efd1--
```

13.3 MIME Responses

```
HTTP/1.1 200
Content-Type: multipart/form-data; boundary=-VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA;charset=UTF-8
Transfer-Encoding: chunked
Date: Tue, 07 Jan 2020 17:39:24 GMT

---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="PayloadType"
Content-Type: text/plain;charset=UTF-8
Content-Length: 29

X12_271_Response_005010X279A1
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="ProcessingMode"
Content-Type: text/plain;charset=UTF-8
```

```
Content-Length: 8
RealTime
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="PayloadID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 36

f81d4fae-7dec-11d0-a765-00a0c91e6bf6
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="TimeStamp"
Content-Type: text/plain;charset=UTF-8
Content-Length: 20

2020-01-07T17:39:24Z
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="SenderID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 6

eMedNY
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="ReceiverID"
Content-Type: text/plain;charset=UTF-8
Content-Length: 9

HospitalA
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="CORERuleVersion"
Content-Type: text/plain;charset=UTF-8
Content-Length: 5

2.2.0
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="Payload"; filename=”R160629150126.2025.835-.tar”
Content-Type: application/x-tar
Content-Length: 10240

ISA*00...~IEA*1*000000320~
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="ErrorCode"
Content-Type: text/plain;charset=UTF-8
Content-Length: 7

Success
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA
Content-Disposition: form-data; name="ErrorMessage"
Content-Type: text/plain;charset=UTF-8
Content-Length: 4
```

```
None
---VzUw7Ii8qgRhv_Ty28_HjOqVwvKfexA--
```

14. Call Center Phone Number

eMedNY Call Center: (800) 343-9000

Hours of Operation

F**or provider inquiries pertaining to non-pharmacy billing or claims, or provider enrollment:**
Monday through Friday: 7:30 a.m. - 6:00 p.m., Eastern Time (excluding holidays)

For provider inquiries pertaining to eligibility and pharmacy claims:
Monday through Friday: 7:00 a.m. - 10:00 p.m., Eastern Time (excluding holidays)
Weekends and Holidays: 8:30 a.m. - 5:30 p.m., Eastern Time

eMedNY Call Center Phone Tree

eMedNY is the name of the electronic New York State Medicaid system. The eMedNY system allows New York Medicaid providers to submit claims and receive payments for Medicaid-covered services provided to eligible clients.

eMedNY offers several innovative technical and architectural features, facilitating the adjudication and payment of claims and providing extensive support and convenience for its users.

The information contained within this document was created in concert by eMedNY and DOH. More information about eMedNY can be found at www.emedny.org.