



**eMedNY CORE Web
Services
Connectivity Guide**

TABLE OF CONTENTS

1. Introduction.....	4
2. Requirements and Limitations	5
3. Envelope Specifications.....	6
4. Inbound Transaction Types	7
5. Responses to Inbound Files.....	8
5.1 Real-Time	8
5.2 Batch	8
6. Retrieving eMedNY-Generated Files.....	9
7. URL for CORE Web Services.....	10
8. Useful Links.....	11
8.1 CAQH-CORE Standards for Web Services.....	11
8.2 Web Portal Links	11
8.2.1 Assistance enrollment in the Web Portal and activation of a “System User”	11
8.2.2 The Web Portal	11
8.3 X12 Information Links	11
8.3.1 Information on the eMedNY X12 envelope (ISA and GS).....	11
8.3.2 eMedNY Companion Guide for 270/271 and 276/277	11
8.3.3 eMedNY Companion Guide for the 835/820	11
8.3.4 eMedNY Companion Guide for the 835S & 820S.....	11
8.3.5 Washington Publishing Company	12
8.4 Production URL for eMedNY CORE Web Services	12
8.5 Useful eMedNY Forms	12
8.5.1 Certification Statement/Instructions for Existing ETINs	12
8.5.2 Electronic Remittance Advice Request	12
9. Appendix I – SOAP Web Services Description Language (WSDL) Updated 6/10/2014.....	13
10. Appendix II – Web Services Schema Definition Updated 6/10/2014.....	18
11. Appendix III - SOAP XML Request and Response Samples	26
11.1 Real-Time Examples	26

11.1.1	SOAP XML Real-Time Request Example.....	26
11.1.2	SOAP XML Real-Time Response Example	26
11.2	Batch Examples	27
11.2.1	SOAP XML Batch Request Example.....	27
11.2.2	SOAP XML Batch Response Example	28
11.2.3	SOAP XML Batch Retrieval Request Example	29
11.2.4	SOAP XML Batch Retrieval Response Example	29
11.3	eMedNY-Generated Response File Examples.....	30
11.3.1	SOAP XML Request File List 835 Example.....	30
11.3.2	SOAP XML Response File List 835 Example	31
11.3.3	SOAP XML X12 835 Request 005010X221A1	32
11.3.4	SOAP XML X12 835 Response 005010X221A1.....	33
12.	Appendix IV – Web Services MIME Request and Response Samples.....	35
12.1	Real-Time Examples	35
12.1.1	MIME Real Time Request Sample.....	35
12.1.2	MIME Real Time Response Sample	36
12.2	Batch Examples	37
12.2.1	MIME Batch Request Sample	37
12.2.2	MIME Batch Response Sample	38
12.2.3	MIME Batch Retrieval Request Sample	39
12.2.4	MIME Batch Retrieval Response Sample.....	40
12.3	eMedNY-Generated Response File Examples.....	41
12.3.1	MIME Request File List 835 Example.....	41
12.3.2	MIME Response File List 835 Example.....	42
12.3.3	X12 835 Request 005010X221A1.....	43
12.3.4	X12 835 Response 005010X221A1	44

For eMedNY Billing Guideline questions, please contact the eMedNY Call Center through the contact information at the end of this document.

1. Introduction

The eMedNY CORE Web Service is designed to facilitate the exchange of X12 270/271 Health Care Eligibility Benefit Inquiry and Response information, and 276/277 Health Care Claim Status Request and Response information, for interactive (real-time) transactions and batch files according to CORE standards. eMedNY CORE Web services can also be used to retrieve both 835 and 820 remittance advice files.

CORE standards require the use of Hypertext Transfer Protocol Secure (HTTPS) over the public Internet. The X12 payload is required to be enclosed within a message envelope that meets the CORE standards for the following two protocols:

- HTTP MIME Multipart
- SOAP + WSDL

The purpose of this document is to supply the needed data to connect to eMedNY's CORE Web Services that adhere to these standards. The general CORE Standards can be found at the following website:

<http://www.cagh.org/pdf/CLEAN5010/270-v5010.pdf>

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

Important Note: It is solely the responsibility of the submitter or user to develop or create their CORE Web Services compliant application. eMedNY will in no way support the end user application, therefore it is strongly recommended that the Trading Partner take appropriate action to have available technical support.

2. Requirements and Limitations

The user will need a client that is compliant with the CORE Standards outlined on the CAQH-CORE website. eMedNY cannot assist the user in coding this client, nor does eMedNY have sample code available.

In order to connect to eMedNY CORE Web Services, the user will need to obtain a “System User” Username and Password from the eMedNY Web Portal. Enrollment in the Web Portal is explained in this document:

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

Once you are enrolled in the Web Portal, use the following guide to create a System User:

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

The 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. The Web Services URL will not allow a transaction without a correct Username and password.

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

3. 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. Please see Appendix II for SOAP + WSDL samples, and Appendix III for HTTP/MIME samples. The following items are specific to eMedNY Web Services

- The *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 the links in Chapter 2 for more details.
- The eMedNY *ReceiverID* is “eMedNY.” Please note this instruction applies only to the envelope (SOAP or MIME) of the transaction. The receiver ID within the X12 transaction should still meet eMedNY specifications as noted in the [eMedNY Trading Partner Information Standard Companion Guide](#).
- The *Payload ID* conforms to CORE standards – for Batch transactions it is then used to retrieve the responses to the inbound file.

4. Inbound Transaction Types

Please refer to the [eMedNY Transaction Information Standard Companion Guide CAQH - CORE Companion Guide](#) for additional information that is specific to New York State Medicaid.

eMedNY CORE Web Services will accept the following inbound Transaction types:

- eMedNY Interactive 270 – Real-Time Eligibility Inquiry
- eMedNY Interactive 276 – Real-Time Claim Status Inquiry
- eMedNY Batch 270 – Batch Eligibility Inquiry
- eMedNY Batch 276 – Batch Claim Status Inquiry

The envelope needs to contain a properly formatted X12 transaction in either 270 or 276 format. The X12 coding of batch files sent through the File Transfer Service are exactly the same as for other transmission methods. Please refer to the ASC X12 Standard Implementation Guides from [Washington Publishing](#) to obtain the correct file and record formats for inbound and outbound eMedNY transactions. The [eMedNY Transaction Information Standard Companion Guide CAQH - CORE Companion Guide](#) supplies details specific to eMedNY 270 and 276 requests.

At this time, eMedNY CORE Web Services is limited to 1 Real-Time Transaction per second and two batch files per minute. Batch files are limited in size according to the transaction limits in the Implementation guide from Washington Publishing. 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.

5. Responses to Inbound Files

When a file is transmitted through eMedNY CORE Web Services, an immediate response will be generated. Other responses may be available, depending on the type of inbound file. Please see below for specifics.

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 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 will be sent through this method:

- 820 Premium Payment Advice (820 File)
- 820 Premium Payment Advice Supplement (820S File)
- 835 Remittance Advice (835 File)
- 835 Remittance Advice Supplement (835S File)
- 834 Benefit Enrollment and Maintenance (834 File)
- Pharmacy Carve-out Report (DWPX)

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 [Washington Publishing](#) to obtain the correct file and record formats for 835 and 820 files. The [eMedNY Transaction Information Standard Companion Guide X12](#) supplies details specific to eMedNY 835 and 820 files. The [Pended Claims Report: Specification](#) supplies information on the formatting of the 835S and 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 will need to 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 has a copy of this form that can be transmitted electronically.

To download an eMedNY-Generated response file, the user will send a "Request File List <payload tpe>" 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.URL for CORE Web Services

The URL for eMedNY CORE Web Services follows

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

This URL is for Production transactions only. At this time, there are no means of sending Test files through CORE Web Services. If a Test file is sent to the above URL it will receive a Front End or TA1 rejection for “Invalid Test Indicator.” Providers wishing to test their connectivity can use a Test Indicator with the intent of getting a rejection.

8. Useful Links

Most of the below links appear elsewhere in the document, this is a compilation of links the user may need for the system.

8.1 CAQH-CORE Standards for Web Services

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

8.2 Web Portal Links

8.2.1 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

8.2.2 The Web Portal

<https://portal.emedny.org/provider>

8.3 X12 Information Links

8.3.1 Information on the eMedNY X12 envelope (ISA and GS)

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

8.3.2 eMedNY Companion Guide for 270/271 and 276/277

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

8.3.3 eMedNY Companion Guide for the 835/820

https://www.emedny.org/HIPAA/5010/transactions/eMedNY_Transaction_Information_CG_X12_version_5010.pdf

8.3.4 eMedNY Companion Guide for the 835S & 820S

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

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

8.3.5 Washington Publishing Company

<http://www.wpc-edi.com/>

8.4 Production URL for eMedNY CORE Web Services

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

8.5 Useful eMedNY Forms

8.5.1 Certification Statement/Instructions for Existing ETINs

The ETIN must be kept current for any kind of electronic transaction

https://www.emedny.org/info/ProviderEnrollment/ProviderMaintForms/401103_ETINDFLT_Default_ETIN_Selection_Form.pdf

8.5.2 Electronic Remittance Advice Request

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

https://www.emedny.org/info/ProviderEnrollment/ProviderMaintForms/401103_ETINDFLT_Default_ETIN_Selection_Form.pdf

9. Appendix I – SOAP Web Services Description Language (WSDL) Updated 6/10/2014

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: 1.1

  Change History:
  09/16/13 -- Peter Futerko - Original Version 1.0 (CORE Rule Version 2.2.0)
  11/22/13 -- Peter Futerko - Version 1.1
    Added Generic Transactions
</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>
```

```
</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:operation name="GenericBatchReceiptConfirmationTransaction">
    <wsdl:input message="tns:BatchResultsAckSubmissionMessage"/>
    <wsdl:output message="tns:BatchResultsAckSubmissionResponseMessage"/>
    <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:operation name="GenericBatchReceiptConfirmationTransaction">
  <soap:operation soapAction="GenericBatchReceiptConfirmationTransaction" 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>

<!-- ===== -->
<!-- SECURITY POLICY DEFINITION -->
<!-- ===== -->
<wsp:Policy>
  <sp:SupportingTokens>
    <wsp:Policy>
      <sp:UsernameToken sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient">
        <wsp:Policy>
          <sp:WssUsernameToken11/>
        </wsp:Policy>
      </sp:UsernameToken>
    </wsp:Policy>
  </sp:SupportingTokens>
</wsp:Policy>

</wsdl:definitions>

```

10. Appendix II – Web Services Schema Definition

Updated 6/10/2014

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: 1.1

      Change History:
      09/16/13 -- Peter Futerko - Original Version 1.0 (CORE Rule Version 2.2.0)
      01/16/14 -- Peter Futerko - Version 1.1
        Added payload types for retrieval of remittance advice, premium payment advice,
        835 supplemental, 820 supplemental, encounters, and pharmacy carve-out.
    </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:Uuid" 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:sequence>
  </xsd:complexType>
</xsd:element>

```

```

    <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:Uuid" 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:Uuid" 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="sch:ResultsRetrievalPayloadType" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="ProcessingMode" type="sch:BatchMode" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

```

<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:simpleType>

```

```
</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_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_835_Request_005010X221A1"></xsd:enumeration>
    <xsd:enumeration value="Request_820S"></xsd:enumeration>
    <xsd:enumeration value="Request_835S"></xsd:enumeration>
    <xsd:enumeration value="Request_MEDA"></xsd:enumeration>
    <xsd:enumeration value="Request_MEDS"></xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
```



```

<xsd:enumeration value="Request_DWPX"></xsd:enumeration>
<xsd:enumeration value="Request_File_List_820"></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_MEDA"></xsd:enumeration>
<xsd:enumeration value="Request_File_List_MEDS"></xsd:enumeration>
<xsd:enumeration value="Request_File_List_DWPX"></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>

```

11. Appendix III - SOAP XML Request and Response Samples

The following are various samples of SOAP XML Transactions.

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">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password Type="wsse:PasswordText">password</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>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</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* *00* *ZZ*NEHEN780 *ZZ*NEHEN003 ...IEA*1*000000031]]></cor:Payload>
    </cor:COREEnvelopeRealTimeRequest>
  </soap:Body>
</soap:Envelope>
```

11.1.2 SOAP XML Real-Time Response Example

Content-Type: application/soap+xml; action="RealTimeTransaction"

Content-Length: 1269

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:cor="http://emedny.org/CORERule.xsd">
  <soap:Body>
    <cor:COREEnvelopeRealTimeResponse>
      <cor:PayloadType>X12_277_Response_005010X212</cor:PayloadType>
      <cor:ProcessingMode>RealTime</cor:ProcessingMode>
      <cor:PayloadID>f81d4fae-7dec-11d0-a765-00a0c91e6bf6</cor:PayloadID>
      <cor:TimeStamp>2013-10-09T20:32:22Z</cor:TimeStamp>
      <cor:SenderID>eMedNY</cor:SenderID>
      <cor:ReceiverID>HospitalA</cor:ReceiverID>
      <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
      <cor:Payload><![CDATA[ISA*00* *00* *ZZ*NEHEN780 *ZZ*NEHEN003 ...IEA*1*000000031]]></cor:Payload>
      <cor:ErrorCode>Success</cor:ErrorCode>
      <cor:ErrorMessage>None</cor:ErrorMessage>
    </cor: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

POST /eMedNY/EDIGateway HTTP/1.1

Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml";
action="BatchSubmitTransaction"; boundary="MIMEBoundaryAbCd"

--MIMEBoundaryAbCd

Content-Type: application/xop+xml; type="application/soap+xml"; action="BatchSubmitTransaction"

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">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password Type="wsse:PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <cor:COREEnvelopeBatchSubmission xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>X12_270_Request_005010X279A1</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
    </cor:COREEnvelopeBatchSubmission>
  </soap:Body>
</soap:Envelope>

```

```

<cor:PayloadID>f82d4fae-7dec-11d0-a765-00a0c91e6bf6</cor:PayloadID>
<cor:PayloadLength>1551254</cor:PayloadLength>
<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:Checksum>db3e770d531f230fb0e9c528d096f7eafe4d6d1a</cor:Checksum>
<cor:Payload><xop:Include href="cid:A2" xmlns:xop="http://www.w3.org/2004/08/xop/include"/></cor:Payload>
</cor:COREEnvelopeBatchSubmission>
</soap:Body>
</soap:Envelope>

```

```

--MIMEBoundaryAbCd
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <A2>

```

```

ISA*00*...~
--MIMEBoundaryAbCd--

```

11.2.2 SOAP XML Batch Response Example

```

Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml";
action="BatchSubmitTransaction"; boundary="E8DC6CC3-51A3-F033-3A53-53E7DBBC80BC"
Content-Length: 921

```

```

--E8DC6CC3-51A3-F033-3A53-53E7DBBC80BC
Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"
Content-Transfer-Encoding: binary
Content-ID: <A1>

```

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <cor:COREEnvelopeBatchSubmissionResponse xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>X12_BatchReceiptConfirmation</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
      <cor:PayloadID>f82d4fae-7dec-11d0-a765-00a0c91e6bf6</cor:PayloadID>
      <cor:TimeStamp>2013-10-09T20:40:46Z</cor:TimeStamp>
      <cor:SenderID>eMedNY</cor:SenderID>
      <cor:ReceiverID>HospitalA</cor:ReceiverID>
      <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
      <cor:ErrorCode>Success</cor:ErrorCode>
      <cor:ErrorMessage>None</cor:ErrorMessage>
    </cor:COREEnvelopeBatchSubmissionResponse>
  </soap:Body>
</soap:Envelope>

```

--E8DC6CC3-51A3-F033-3A53-53E7DBBC80BC--

11.2.3 SOAP XML Batch Retrieval Request Example

POST /eMedNY/EDIGateway HTTP/1.1

Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml";
action="BatchResultsRetrievalTransaction"; 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">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password Type="wsse:PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <cor:COREEnvelopeBatchResultsRetrievalRequest xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>X12_005010_Request_Batch_Results_271</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
      <cor:PayloadID>f82d4fae-7dec-11d0-a765-00a0c91e6bf6</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:COREEnvelopeBatchResultsRetrievalRequest>
  </soap:Body>
</soap:Envelope>
```

--MIMEBoundaryAbCd--

11.2.4 SOAP XML Batch Retrieval Response Example

Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml";
action="BatchResultsRetrievalTransaction"; boundary="40787A7E-65BA-1587-3F86-9B0F496CA666"

Content-Length: 2280

--40787A7E-65BA-1587-3F86-9B0F496CA666

Content-Type: application/xop+xml; charset=UTF-8; type="application/soap+xml"

Content-Transfer-Encoding: binary

Content-ID: <A1>

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <cor:COREEnvelopeBatchResultsRetrievalResponse xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>X12_271_Response_005010X279A1</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
      <cor:PayloadID>f82d4fae-7dec-11d0-a765-00a0c91e6bf6</cor:PayloadID>
      <cor:PayloadLength>993</cor:PayloadLength>
      <cor:TimeStamp>2013-10-09T20:46:31Z</cor:TimeStamp>
      <cor:SenderID>eMedNY</cor:SenderID>
      <cor:ReceiverID>HospitalA</cor:ReceiverID>
      <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
      <cor:Checksum>db3e770d531f230fb0e9c528d096f7eafe4d6d1a</cor:Checksum>
      <cor:Payload>
        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:A2"/>
      </cor:Payload>
      <cor:ErrorCode>Success</cor:ErrorCode>
      <cor:ErrorMessage>None</cor:ErrorMessage>
    </cor:COREEnvelopeBatchResultsRetrievalResponse>
  </soap:Body>
</soap:Envelope>

```

```

--40787A7E-65BA-1587-3F86-9B0F496CA666
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-ID: <A2>

```

```

ISA*00*...~
--40787A7E-65BA-1587-3F86-9B0F496CA666—

```

11.3 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.

11.3.1 SOAP XML Request File List 835 Example

```

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">
    <wsse:UsernameToken>
      <wsse:Username>username</wsse:Username>
      <wsse:Password Type="wsse:PasswordText">password</wsse:Password>
    </wsse:UsernameToken>
  </wsse:Security>
</soap:Header>
<soap:Body>
  <cor:COREEnvelopeBatchResultsRetrievalRequest xmlns:cor="http://emedny.org/CORERule.xsd">
    <cor:PayloadType>Request_File_List_835</cor:PayloadType>
    <cor:ProcessingMode>Batch</cor:ProcessingMode>
    <cor:PayloadID>17140000-0000-0000-0000-000000000003</cor:PayloadID>
    <cor:TimeStamp>2014-01-23T10:20:34Z</cor:TimeStamp>
    <cor:SenderID>sender</cor:SenderID>
    <cor:ReceiverID>eMedNY</cor:ReceiverID>
    <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
  </cor:COREEnvelopeBatchResultsRetrievalRequest>
</soap:Body>
</soap:Envelope>

```

--MIMEBoundaryAbCd--

11.3.2 SOAP XML Response File List 835 Example

HTTP/1.0 200 OK

Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml";
 action="GenericBatchRetrievalTransaction"; boundary="121194DD-BC2F-FDB2-E767-A38B39D5689B"
 Content-Length: 1401

--121194DD-BC2F-FDB2-E767-A38B39D5689B

content-id: <A1>

content-type: application/xop+xml; type="application/soap+xml"
 content-transfer-encoding: binary

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <cor:COREEnvelopeBatchResultsRetrievalResponse xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>Response_File_List_835</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
      <cor:PayloadID>17140000-0000-0000-0000-000000000003</cor:PayloadID>
      <cor:PayloadLength>131</cor:PayloadLength>
      <cor:TimeStamp>2014-01-23T10:13:52Z</cor:TimeStamp>
      <cor:SenderID>eMedNY</cor:SenderID>
      <cor:ReceiverID>sender</cor:ReceiverID>
      <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
      <cor:Checksum>eb7df9835304ae9add89b8b6b6e2899798b21f49</cor:Checksum>
    </cor:COREEnvelopeBatchResultsRetrievalResponse>
  </soap:Body>
</soap:Envelope>

```

```
<cor:Payload>
  <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:A2"/>
</cor:Payload>
<cor:ErrorCode>Success</cor:ErrorCode>
<cor:ErrorMessage>None</cor:ErrorMessage>
</cor:COREEnvelopeBatchResultsRetrievalResponse>
</soap:Body>
</soap:Envelope>
```

--121194DD-BC2F-FDB2-E767-A38B39D5689B

content-id: <A2>

content-type: application/octet-stream

content-transfer-encoding: binary

```
<list><payloadid>20140114-1327-3700-0000-000000000001</payloadid><payloadid>20140113-1643-2200-0000-000000000001</payloadid></list>
```

--121194DD-BC2F-FDB2-E767-A38B39D5689B—

11.3.3 SOAP XML X12 835 Request 005010X221A1

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">
      <wsse:UsernameToken>
        <wsse:Username>username</wsse:Username>
        <wsse:Password Type="wsse:PasswordText">password</wsse:Password>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>
  <soap:Body>
    <cor:COREEnvelopeBatchResultsRetrievalRequest xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>X12_835_Request_005010X221A1</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
      <cor:PayloadID>20140114-1327-3700-0000-000000000001</cor:PayloadID>
      <cor:TimeStamp>2014-01-23T10:20:34Z</cor:TimeStamp>
      <cor:SenderID>sender</cor:SenderID>
      <cor:ReceiverID>eMedNY</cor:ReceiverID>
      <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
```



```
</cor:COREEnvelopeBatchResultsRetrievalRequest>
</soap:Body>
</soap:Envelope>
```

--MIMEBoundaryAbCd--

11.3.4 SOAP XML X12 835 Response 005010X221A1

HTTP/1.0 200 OK
Content-Type: multipart/related; type="application/xop+xml"; start="<A1>"; start-info="application/soap+xml";
action="GenericBatchRetrievalTransaction"; boundary="BE215BE7-1129-862F-1877-35FB0AAD4446"
Content-Length: 11519

--BE215BE7-1129-862F-1877-35FB0AAD4446
content-id: <A1>
content-type: application/xop+xml; type="application/soap+xml";
content-transfer-encoding: binary

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <cor:COREEnvelopeBatchResultsRetrievalResponse xmlns:cor="http://emedny.org/CORERule.xsd">
      <cor:PayloadType>X12_835_Response_005010X221A1</cor:PayloadType>
      <cor:ProcessingMode>Batch</cor:ProcessingMode>
      <cor:PayloadID>20140114-1327-3700-0000-000000000001</cor:PayloadID>
      <cor:PayloadLength>10240</cor:PayloadLength>
      <cor:TimeStamp>2014-01-23T10:16:14Z</cor:TimeStamp>
      <cor:SenderID>eMedNY</cor:SenderID>
      <cor:ReceiverID>sender</cor:ReceiverID>
      <cor:CORERuleVersion>2.2.0</cor:CORERuleVersion>
      <cor:Checksum>426d6759be519fdd60621666a4597955a5906e4a</cor:Checksum>
      <cor:Payload>
        <xop:Include xmlns:xop="http://www.w3.org/2004/08/xop/include" href="cid:A2"/>
      </cor:Payload>
      <cor:ErrorCode>Success</cor:ErrorCode>
      <cor:ErrorMessage>None</cor:ErrorMessage>
    </cor:COREEnvelopeBatchResultsRetrievalResponse>
  </soap:Body>
</soap:Envelope>
```

--BE215BE7-1129-862F-1877-35FB0AAD4446
content-id: <A2>
content-type: application/octet-stream
content-transfer-encoding: binary

SIVOSwAA
AAAAAAAAAAAAAAAAAAAAEpVTksAAA
AA

AAAAAAAAASIVOSwAA
AAAAAAAAAAAAAAAAABKVU5LAA
AA
AA
OMIT BYTES
AA
AAAAAAAAAAAAAAAAAAAAA=
--BE215BE7-1129-862F-1877-35FB0AAD4446—

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

12.1 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.

12.1.1 MIME Real Time Request Sample

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

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

```
X12_270_Request_005010X279A1
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="ProcessingMode"
```

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

```
f81d4fae-7dec-11d0-a765-00a0c91e6bf6
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="TimeStamp"
```

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

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

```
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

Content-Disposition: form-data; name="Payload"

ISA*00* *00* *ZZ*NEHEN780 *ZZ*NEHEN003 ...IEA*1*000000031

--MIMEBoundaryAbCd--

12.1.2 MIME Real Time Response Sample

Content-Type: multipart/form-data; boundary="A847AD50-9F55-07FD-C0D3-2934F27F870B"

Content-Length: 1694

--A847AD50-9F55-07FD-C0D3-2934F27F870B

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

X12_277_Response_005010X212

--A847AD50-9F55-07FD-C0D3-2934F27F870B

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

RealTime

--A847AD50-9F55-07FD-C0D3-2934F27F870B

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

f81d4fae-7dec-11d0-a765-00a0c91e6bf6

--A847AD50-9F55-07FD-C0D3-2934F27F870B

Content-Disposition: form-data; name="TimeStamp"

2013-10-09T20:52:55Z

--A847AD50-9F55-07FD-C0D3-2934F27F870B

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

eMedNY

--A847AD50-9F55-07FD-C0D3-2934F27F870B

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

HospitalA

--A847AD50-9F55-07FD-C0D3-2934F27F870B

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

2.2.0

--A847AD50-9F55-07FD-C0D3-2934F27F870B
Content-Disposition: form-data; name="Payload"

ISA*00* *00* *ZZ*NEHEN780 *ZZ*NEHEN003 ...IEA*1*000000031
--A847AD50-9F55-07FD-C0D3-2934F27F870B
Content-Disposition: form-data; name="ErrorCode"

Success
--A847AD50-9F55-07FD-C0D3-2934F27F870B
Content-Disposition: form-data; name="ErrorMessage"

None
--A847AD50-9F55-07FD-C0D3-2934F27F870B—

12.2 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.

12.2.1 MIME Batch Request Sample

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

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

X12_270_Request_005010X279A1
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="ProcessingMode"

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

f83d4fae-7dec-11d0-a765-00a0c91e6bf6
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="PayloadLength"

5000
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="TimeStamp"

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

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

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
Content-Disposition: form-data; name="Checksum"

db3e770d531f230fb0e9c528d096f7eafe4d6d1a
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="Payload"

ISA*00*...~
--MIMEBoundaryAbCd--

12.2.2 MIME Batch Response Sample

Content-Type: multipart/form-data; boundary="6E9A221C-CD98-26F7-7045-12627B7178B4"
Content-Length: 1022

--6E9A221C-CD98-26F7-7045-12627B7178B4
Content-Disposition: form-data; name="PayloadType"

X12_BatchReceiptConfirmation
--6E9A221C-CD98-26F7-7045-12627B7178B4
Content-Disposition: form-data; name="ProcessingMode"

Batch
--6E9A221C-CD98-26F7-7045-12627B7178B4
Content-Disposition: form-data; name="PayloadID"

f83d4fae-7dec-11d0-a765-00a0c91e6bf6
--6E9A221C-CD98-26F7-7045-12627B7178B4

Content-Disposition: form-data; name="TimeStamp"

2013-10-09T21:01:58Z

--6E9A221C-CD98-26F7-7045-12627B7178B4

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

eMedNY

--6E9A221C-CD98-26F7-7045-12627B7178B4

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

HospitalA

--6E9A221C-CD98-26F7-7045-12627B7178B4

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

2.2.0

--6E9A221C-CD98-26F7-7045-12627B7178B4

Content-Disposition: form-data; name="ErrorCode"

Success

--6E9A221C-CD98-26F7-7045-12627B7178B4

Content-Disposition: form-data; name="ErrorMessage"

None

--6E9A221C-CD98-26F7-7045-12627B7178B4--

12.2.3 MIME Batch Retrieval Request Sample

POST /eMedNY/EDIGateway HTTP/1.1

Content-Type: multipart/form-data; boundary="MIMEBoundaryAbCd"

--MIMEBoundaryAbCd

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

X12_005010_Request_Batch_Results_271

--MIMEBoundaryAbCd

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

Batch

--MIMEBoundaryAbCd

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

f83d4fae-7dec-11d0-a765-00a0c91e6bf6

--MIMEBoundaryAbCd

Content-Disposition: form-data; name="TimeStamp"

2007-08-30T10:20:34Z

--MIMEBoundaryAbCd

Content-Disposition: form-data; name="UserName"

username

--MIMEBoundaryAbCd

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

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--

12.2.4 MIME Batch Retrieval Response Sample

Content-Type: multipart/form-data; boundary="0CBCBA69-994C-5F79-E39E-7C1D6521173A"

Content-Length: 2342

--0CBCBA69-994C-5F79-E39E-7C1D6521173A

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

X12_271_Response_005010X279A1

--0CBCBA69-994C-5F79-E39E-7C1D6521173A

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

Batch

--0CBCBA69-994C-5F79-E39E-7C1D6521173A

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

f83d4fae-7dec-11d0-a765-00a0c91e6bf6

--0CBCBA69-994C-5F79-E39E-7C1D6521173A

Content-Disposition: form-data; name="PayloadLength"

993

--0CBCBA69-994C-5F79-E39E-7C1D6521173A

Content-Disposition: form-data; name="TimeStamp"

2013-10-09T21:06:37Z

--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="SenderID"

eMedNY
--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="ReceiverID"

HospitalA
--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="CORERuleVersion"

2.2.0
--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="Checksum"

db3e770d531f230fb0e9c528d096f7eafe4d6d1a
--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="Payload"

ISA*00*...~
--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="ErrorCode"

Success
--0CBCBA69-994C-5F79-E39E-7C1D6521173A
Content-Disposition: form-data; name="ErrorMessage"

None
--0CBCBA69-994C-5F79-E39E-7C1D6521173A—

12.3 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.

12.3.1 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"

17140000-0000-0000-0000-000000000001
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="TimeStamp"

2014-01-23T10:20:34Z
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="UserName"

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

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

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

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

2.2.0
--MIMEBoundaryAbCd--

12.3.2 MIME Response File List 835 Example

HTTP/1.0 200 OK
Content-Type: multipart/form-data; boundary="FDB0151A-A60C-CFDC-3934-22C9D2A85856"
Content-Length: 1470

--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="PayloadType"

Response_File_List_835
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="ProcessingMode"

Batch
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="PayloadID"

17140000-0000-0000-0000-000000000001
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="PayloadLength"

131
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="TimeStamp"

2014-01-23T10:08:06Z
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="SenderID"

eMedNY
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="ReceiverID"

sender
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="CORERuleVersion"

2.2.0
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="Checksum"

eb7df9835304ae9add89b8b6b6e2899798b21f49
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="Payload"

<list><payloadid>20140114-1327-3700-0000-000000000001</payloadid><payloadid>20140113-1643-2200-0000-000000000001</payloadid></list>
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="ErrorCode"

Success
--FDB0151A-A60C-CFDC-3934-22C9D2A85856
Content-Disposition: form-data; name="ErrorMessage"

None
--FDB0151A-A60C-CFDC-3934-22C9D2A85856--

12.3.3 X12 835 Request 005010X221A1

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"

20140114-1327-3700-0000-000000000001
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="TimeStamp"

2014-01-23T10:20:34Z
--MIMEBoundaryAbCd
Content-Disposition: form-data; name="UserName"

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

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

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

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

2.2.0
--MIMEBoundaryAbCd—

12.3.4 X12 835 Response 005010X221A1

HTTP/1.0 200 OK
Content-Type: multipart/form-data; boundary="467700E7-B493-9ECD-6FD9-D31E6DF20C3D"
Content-Length: 11588

--467700E7-B493-9ECD-6FD9-D31E6DF20C3D
Content-Disposition: form-data; name="PayloadType"

X12_835_Response_005010X221A1
--467700E7-B493-9ECD-6FD9-D31E6DF20C3D

--467700E7-B493-9ECD-6FD9-D31E6DF20C3D
Content-Disposition: form-data; name="ErrorMessage"

None

--467700E7-B493-9ECD-6FD9-D31E6DF20C3D--

For eMedNY Billing Information:

Institutional: 1-800-343-9000

Clinics, hospitals, rate-based; ordered/referred ambulatory

Practitioner: 1-800-343-9000

MDs, RNs, dentists

Professional: 1-800-343-9000

others (DME, non-MDs - phys. Therapy, etc.)

Business hours are Monday through Friday from 7:00 a.m. to 6:00 p.m..



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.