HIPAA Eligibility Transaction System (HETS) 270/271

R2013Q100

Optional New Service Offering

Document Version 1-0 FINAL
Implementation Date: First Quarter 2013

R2013Q100 HETS 270/271 Optional New Service Offering

Impact on Trading Partners

IMPORTANT NOTE: This document is intended for use by a technical professional who has experience implementing secure, web-based connectivity.

The purpose of this summary document is to inform Trading Partners of an optional new service offering for the HIPAA Eligibility Transaction System (HETS) 270/271 application. The R2013Q100 HETS 270/271 release will address both federally mandated Operating Rule 153: Eligibility and Connectivity and Operating Rule 270: Connectivity. The Department of Health and Human Services (HHS) has named the Council for Affordable Quality Healthcare/ Committee on Operating Rules for Information Exchange (CAQH/CORE) the authoring entity of the Operating Rules mandated under the Patient Protection and Affordable Care Act (ACA).

For a copy of the mandated Operating Rules, please refer to http://www.caqh.org/ORMandate_Eligibility.php

Changes in this release include:

- Support of Simple Object Access Protocol (SOAP) + Web Services Description Language (WSDL) envelope standards.
- Support of Hypertext Transfer Protocol (HTTP)/Multipurpose Internet Mail Extensions (MIME) Multi-part envelope standards.
- Trading Partners transmitting with SOAP or MIME must obtain a digital certificate and send the transaction to the HETS 270/271 application via a secure internet connection.
- New error codes to support SOAP + WSDL and HTTP/MIME Multi-part communication protocols.

These new optional connection methods will be offered in addition to the CMSNet connection method currently in place. HETS Trading Partners will have the option of using any of the connection methods to submit and receive eligibility data. The HETS 270/271 application will continue to only accept real-time transactions.

Please refer to the following summary for additional information. Due to the extent of the upcoming changes, information included in this document is subject to change. Changes will be communicated as necessary.

1 SOAP + WSDL

Effective with the R2013Q100 release, the HETS 270/271 application will support transactions formatted according to Version 1.2 of SOAP conforming to standards set forth by WSDL for Extensible Markup Language (XML) envelope formatting, submission and retrieval. X12 payload data must be embedded using the Inline method (CDATA

element) and the XML schema and WSDL definitions according to Phase II CORE 270: Connectivity Rule version 2.2.0 March 2011 located at this link: http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf.

For additional information on constructing a 270 request using SOAP, please refer to the following links:

SOAP XML Schema: http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd

WSDL Information: http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.wsdl

SOAP Header Requirements: http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf
The SOAP Header should include the timestamp element and should be digitally signed.

SOAP Body Requirements: http://www.w3.org/TR/soap12-part1

Table 1 defines HETS-specific body elements for 270 requests using SOAP or MIME.

Table 1 – Required Body Elements for 270 Requests Using SOAP or MIME

Element Name	Description
PayloadType	X12_270_Request_005010X279A1
ProcessingMode	RealTime
PayloadID	Refer to section 4.4.2 of the Phase II CORE 270: Connectivity Rule for structural guidelines for CORE envelope metadata.
TimeStamp	Format is YYYY-MM-DDTHH:MMSSZ. Refer to http://www.w3.org/TR/xmlschema11-2/#dateTime for more information.
SenderID	This field should be 10 characters in length.
ReceiverID	CMS
CORERuleVersion	2.2.0
Payload	X12 request. This element must be digitally signed and the entire payload should be enclosed within a CDATA tag.

Table 2 defines HETS-specific body elements for 271 responses using SOAP or MIME.

Table 2 – Required Body Elements for 271 Responses Using SOAP or MIME

Element Name	Description
PayloadType	X12_270_Request_005010X279A1
ProcessingMode	RealTime
PayloadID	Refer to section 4.4.2 of the Phase II CORE 270: Connectivity Rule for structural guidelines for CORE envelope metadata.
TimeStamp	Format is YYYY-MM-DDTHH:MMSSZ. Refer to http://www.w3.org/TR/xmlschema11-2/#dateTime for more information.
SenderID	CMS
ReceiverID	This field should be 10 characters in length.
CORERuleVersion	2.2.0
Payload	X12 response

SOAP Digital Signature: http://www.w3.org/TR/SOAP-dsig/

SOAP Request and Response Examples:

Table 3 provides an example of a 270 request using SOAP.

Table 3 - SOAP Request Message Structure

SOAP Structure Element	Content
HTTP Header	POST /eligibility/realtime/soap HTTP/1.1 Accept-Encoding: gzip,deflate Content-Type: application/soap+xml;charset=UTF8;action="RealTimeTransaction" Content-Length: 4808 Host: 123.12.123.12:2121 Connection: Keep-Alive User-Agent: Apache-HttpClient/4.1.1 (java 1.5)
SOAP Envelope Begin	<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
SOAP Header Begin	<soap:Header>
SOAP Header WS-Security	<wsse:Security soap:mustUnderstand="true" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity- utility-1.0.xsd">
SOAP Header TIMESTAMP	<wsu:Timestamp wsu:Id="id-155"> <wsu:Created>2012-12-10T17:11:08.796Z</wsu:Created> <wsu:Expires>2012-12-10T17:11:28.796Z</wsu:Expires> </wsu:Timestamp>
SOAP Header Binary Security Token	<wsse:BinarySecurityToken EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token- profile-1.0#X509v3" wsu:Id="X509-0E4E74F95B0421C31C135515946875040">{{{BST HERE}}} </wsse:BinarySecurityToken>

SOAP Structure Element	Content
SOAP Header Signature	<ds:Signature Id="SIG-44" xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> <ds:SignedInfo> <ds:CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n- 20010315"/> <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/> <ds:Reference URI="#id-43"> <ds:Transforms> <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc- c14n#"> <InclusiveNamespaces PrefixList="ns1 soap" xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/> </ds:Transform> </ds:Transforms> <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/> <ds:DigestValue>cKtVDws5KS70zUTfNB90jcz/F5K/GwIiDF09aEV2fMA=</ds:DigestValue> </ds:Reference> <ds:Reference URI="#id-155"> <ds:Transforms> <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"> <InclusiveNamespaces PrefixList="ns1 soap" xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/> </ds:Transform> </ds:Transforms> <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/> <ds:DigestValue>tu65ngGe0dl2f2f3iwN/phOQBDXEPFVw2u6/1ZKmX/A=</ds:DigestValue> </ds:Reference> </ds:SignedInfo>
SOAP Header Signature Value	<ds:SignatureValue>{{{Encoded Signature Value }}} </ds:SignatureValue>
SOAP Header KeyInfo	<ds:KeyInfo Id="KI-0E4E74F95B0421C31C135515946875041"> <wsse:SecurityTokenReference wsu:Id="STR0E4E74F95B0421C31C135515946875042"> <wsse:Reference URI="#X509-0E4E74F95B0421C31C135515946875040" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token- profile-1.0#X509v3"/> </wsse:SecurityTokenReference> </ds:KeyInfo>
SOAP Header End	</ds:Signature> </wsse:Security> </soap:Header>
SOAP Body Begin	<soap:Body> <ns1:COREEnvelopeRealTimeRequest xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd">

SOAP Structure Element	Content
SOAP Body PayloadType	<PayloadType>X12_270_Request_005010X279A1</PayloadType>
SOAP Body ProcessingMode	<ProcessingMode>RealTime</ProcessingMode>
SOAP Body PayloadID	<PayloadID>10</PayloadID>
SOAP Body TimeStamp	<TimeStamp>2012-12-10T12:53:03.964Z</TimeStamp>
SOAP Body SenderID	<SenderID>ABCDEFGHIJ</SenderID>
SOAP Body ReceiverID	<ReceiverID>CMS</ReceiverID>
SOAP Body CORERuleVersion	<CORERuleVersion>2.2.0</CORERuleVersion>
SOAP Body Payload	<Payload wsu:Id="id-43"xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility1.0.xsd"><![CDATA[ISA*00* *00* *ZZ*…….IEA*1*000005014~]]></Payload>
SOAP Body End	</ns1:COREEnvelopeRealTimeRequest> </soap:Body>
SOAP Envelope End	</soap:Envelope>

The SOAP response format will look similar to the request format outlined in Table 3. Only the SOAP body will contain response-specific information. Table 4 provides an example of a 271 response using SOAP.

Table 4 - SOAP Response Message Structure

SOAP Structure Element	Content
SOAP Body Begin	<soap:Body> <ns1: COREEnvelopeRealTimeResponse xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd">
SOAP Body PayloadType	<PayloadType>X12_999_Response_005010X231A1</PayloadType>
SOAP Body ProcessingMode	<ProcessingMode>RealTime</ProcessingMode>
SOAP Body PayloadID	<PayloadID>10</PayloadID>
SOAP Body TimeStamp	<TimeStamp>2012-12-10T12:53:05.964Z</TimeStamp>
SOAP Body SenderID	<SenderID>CMS</SenderID>
SOAP Body ReceiverID	<ReceiverID>ABCDEFGHIJ</ReceiverID>
SOAP Body CORERuleVersion	<CORERuleVersion>2.2.0</CORERuleVersion>
SOAP Body Payload	<Payload wsu:Id="id-168 " xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"">"><![CDATA[ISA*00* *00* *ZZ*…….

SOAP Structure Element	Content
	IEA*1*000005014~]]></Payload>
SOAP Body End	</ns1: COREEnvelopeRealTimeResponse> <ErrorCode>Success</ErrorCode> <ErrorMessage/> </soap:Body>

For additional examples, refer to http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf (sections 4.2.2.3 and 4.2.2.4).

2 HTTP/MIME Multipart

Effective with the R2013Q100 release, the HETS 270/271 application will support standard HTTP MIME messages. The required MIME format is multipart/form-data. Responses to request transactions sent via this protocol will be returned in a MIME multipart form, which contains the payload as an X12 document.

For additional information on constructing a 270 request using MIME, please refer to the following links:

MIME Data Requirements for Header and Body:
http://www.faqs.org/rfcs/rfc2388.html

Since CORE does not specify naming conventions, HETS will implement MIME with the same field names as SOAP. Refer to Table 1 in this document for the HETS-specific body elements.

MIME Request and Response Examples:

Table 5 provides an example of a 270 request using HTTP MIME Multipart.

Table 5 - MIME Request Message Structure

MIME Structure Element	Content
MIME Header	POST https://hets.cms.cmstest/eligibility/realtime/mime HTTP/1.1 Connection: keep-alive Content-Length: 1392 Content-Type: multipart/form-data; boundary=COSZiva9NdnYzPXUEGy-tLBO8n4-czud Host: hets.cms.cmstest User-Agent: Apache-HttpClient/4.2.1 (java 1.5)

MIME Structure Element	Content
MIME Body	--COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="PayloadType" X12_270_Request_005010X279A1 --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="ProcessingMode" RealTime --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="PayloadID" hets1232123234 --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="TimeStamp" 2012-11-31T23:00:5Z --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="SenderID" HETS00001 --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="ReceiverID" CMS --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="CORERuleVersion" 2.2.0 --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="Payload"; filename="MIMETest.txt" Content-Type: text/plain <actual file content, not shown here> --COSZiva9NdnYzPXUEGy-tLBO8n4-czud--

Table 6 provides an example of a 271 response using HTTP MIME Multipart.

Table 6 - MIME Response Message Structure

MIME Structure Element	Content
MIME Header	HTTP/1.1 200 OK X-Backside-Transport: OK OK,OK OK Connection: Keep-Alive Transfer-Encoding: chunked X-Powered-By: Servlet/2.5 Content-Type: multipart/form-data; boundary="7aaeaf96-1e54-4567-a8d0-e93de77cd66a" Date: Wed, 02 Jan 2013 20:28:28 GMT X-Client-IP: 10.17.2.7,172.23.11.253 X-Archived-Client-IP: 10.17.2.7 POST: http://172.16.104.99:2122/eligibility/realtime/mime

MIME Structure Element	Content
MIME Body	--7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=PayloadType X12_TA1_Response_00501X231A1 --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=ProcessingMode RealTime --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=PayloadID hets1232123234 --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=TimeStamp 2013-01-02T15:28:28.906Z --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=SenderID CMS --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=ReceiverID HETS000001 --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=CORERuleVersion 2.2.0 --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=Payload ISA*00* *00* *ZZ*CMS *ZZ*H000000001 *130102*1528*^*00501*000012379*0*T*|~TA1*000005014*050516*0734*R* 017~I EA*0*000012379~ --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=ErrorCode Success --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name=ErrorMessage --7aaeaf96-1e54-4567-a8d0-e93de77cd66a

For additional examples, refer to http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf (sections 4.2.1.1 and 4.2.1.2).

3 Digital Certificates

To connect to the HETS 270/271 application via SOAP or MIME, Trading Partners will need to authenticate with an X.509 Digital Certificate using the Transport Layer Security (TLS) 1.0 open standard for client certificate-based authentication. TLS 1.0 is required for compliance with the federally-mandated with Submitter Authentication Standard D in the Conformance Requirements.

Before a certificate can be procured from a Certificate Authority (CA), Trading Partners will need to generate a platform-specific Certificate Signing Request (CSR). Trading

Partners are requested to review the CA-specific CSR process carefully and contact the CAs directly to obtain the certificate. The HETS 270/271 application requires a certificate enabled with a minimum 128-bit SSL encryption.

Trading Partners must use one of the following CAs to procure a Digital Certificate:

- **DigiCert:** DigiCert provides "SSL Plus Certificates" which can be procured from http://www.digicert.com/welcome/ssl-plus.htm.

  Before procuring a certificate, Trading Partners are advised to review the information on certificate procurement and platform-specific CSR generation at this link: http://www.digicert.com/csr-creation.htm.

- **Entrust:** Entrust provides "Advantage SSL Certificates" which can be procured from http://www.entrust.net/ssl-certificates/advantage.htm.

  Before procuring a certificate, Trading Partners are advised to review the information on certificate procurement and platform-specific CSR generation at this link: http://www.entrust.net/ssl-technical/csr_faq.cfm.

- **Symantec (VeriSign):** Symantec issues VeriSign "Secure Site" SSL certificates which can be procured from http://www.symantec.com/verisign/ssl-certificates/secure-site.

  Before procuring a certificate, Trading Partners are advised to review the information on certificate procurement and platform-specific CSR generation at this link: https://knowledge.verisign.com/support/ssl-certificates-support/index?page=content&actp=CROSSLINK&id=AR235.

Note: The certificates listed for each CA are the minimum level required to connect to the HETS 270/271 application. Trading Partners may choose to procure a higher level of certificate.

Before accessing the HETS 270/271 application via SOAP or MIME, new and existing Trading Partners must provide the Digital Certificate to CMS by contacting the Medicare Customer Assistance Regarding Eligibility (MCARE) Help Desk. MCARE will verify the certificate and initiate the process to configure Trading Partner access to the HETS 270/271 application. If the Trading Partner's Digital Certificate has not been approved and properly configured, connection to the HETS 270/271 application may be rejected.

4 Status and Error Codes

The HETS 270/271 application will process SOAP and MIME transactions and return errors as described in the next sections.

4.1 HTTP Status and Error Codes

The processing and error codes for the HTTP Layer are defined as part of the HTTP specifications which can be found at http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html. These status and error codes are defined in Table 4.3.3.1 in the document located at http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf.

4.2 Envelope Processing Status and Error Codes

Table 7 describes envelope processing status and error codes specific to the HETS 270/271 application for SOAP and MIME transactions.

Table 7 - Envelope Processing Status and Errors

Error Code	Error Message
<FieldName>Illegal	Illegal value provided for <FieldName>.
<FieldName>Required	The field <FieldName> is required but was not provided.
VersionMismatch	The version of the envelope sent is not acceptable to the receiver.
Invalid Payload	Payload is invalid or does not start with ISA.
Success	Envelope was processed successfully.

4.3 SOAP-Specific Processing Errors

Table 8 describes examples of SOAP processing errors.

Table 8 – SOAP-Specific Processing Errors

Error Code	Error Message
nonconforming.content	No signature in message!
nonconforming.content	No signature in the WS-Security message for the configured SOAP actor/role
nonconforming.content	Unsupported or unrecognized Signature signer format in the message
nonconforming.content	*Certificate not found*
nonconforming.content	Illegal value provided for ProcessingMode
nonconforming.content	Found <Fieldnamevalue> (in default namespace), but next item should be <Fieldname>
env:Client	There was an error in the incoming SOAP request
env:Client	Processing Mode cannot be empty. Value expected is RealTime

4.4 MIME-Specific Processing Errors

Table 9 describes examples of MIME processing errors.

Table 9 - MIME-Specific Processing Errors

Error Code	Error Message
nonconforming.content	ProcessingMode value <FieldValue> is not a valid instance of type RealTimeMode
env:Client	ProcessingMode of type RealTimeMode may not be empty

4.5 Transaction Processing Errors

Transaction processing errors, described in sections 8.1 through 8.4 of the HETS 270/271 Companion Guide, will be returned as a SOAP or MIME message containing the related response.