Centers for Medicare & Medicaid Services (CMS)
7500 Security Boulevard
Baltimore, MD 21244-1850

HIPAA Eligibility Transaction System (HETS) Submitter SOAP/MIME Connectivity Instructions

Version: 3.2

Last Modified: August 21, 2017

Revision History

Table 1 - Document Revision History

Version	Date	Revision/Change Description	Pages Affected
1.0	08/15/2013	Initial release.	All
1.1	02/24/2014	Added clarification that only those characters referenced in the Basic and the Extended Character Sets noted in the Appendix of the ASCX12 270/271 version 005010X279A1 TR3 including the 005010X279E1 Errata are acceptable within a HETS 270 inquiry.	5, 10 and 23
2.0	06/19/15	Added clarification on requirement of SHA2-256 and TLS 1.2 requirement as well as clarified existing SOAP/MIME processing.	Multiple
2.1	9/2/2015	Added Entrust L1K/L1M certificate list and added additional values which can return in the PayloadType field for MIME responses.	4, 12
3.0	3/24/16	Modified document for R2016Q300 Redesign Release. Changes Include: Updated Title Page to remove OIS and 508 compliant check Section 1 – Updated HETS Help website, removed references to HPG and replaced with HDT, added URL for HDT User Manual Section 2 – Removed references to TLS 1.1 and December 31, 2015 deadline to utilize TLS 1.2 and a SHA2-256 certificate as this deadline has passed Section 2.1 – Removed references to December 31, 2015 deadline to utilize TLS 1.2 and a SHA2-256 certificate as this deadline has passed Section 2.1.2 – Removed outdated Certification Authorities for Entrust Section 2.2 – Removed reference to January 1, 2016 requirement as this deadline has passed and replaced SHA1 with SHA2 Section 3 – Updated SOAP URL Section 3.1, Table 2 and 3 description – Updated 271 Responses to X12 Responses Section 3.1, Table 2 and 3 – Updated TimeStamp description Section 3.2, Table 4 – Updated HTTP Header Content with new SOAP URL, and SOAP Header Timestamp Content with generic value, PayloadID Content with an example, TimeStamp Content with generic value Section 4 – Updated MIME URL Section 4.1, Table 6 and 7 description – Updated 271 Responses to X12 Responses Section 4.1, Table 6 and 7 – Updated TimeStamp description Section 4.2 – Added note that examples are for illustrative purposes only Section 4.2, Table 8 – Updated MIME Header Content with new MIME URL, PayloadID Content with an example, TimeStamp Content with generic value Section 4.2, Table 9 – Updated MIME Header and Body Content Section 5.2, Table 10 - Updated VersionMismatch	Multiple

Version	Date	Revision/Change Description	Pages Affected
		Description and removed InvalidPayload Section 5.2, Table 11 – Added UnAuthorized error and removed other errors Section 5.4 – Removed MIME Specific Processing Errors as they no longer apply Appendix B – Updated FAQ 12 and 16 Appendix C – Replaced HPG User Guide Reference to HDT User Guide Appendix D – Replaced HPG with HDT	
3.1	12/5/2016	Table 8 – Updated MIME Body Content Table 9 – Updated MIME Body Content	13, 14
3.2	8/21/2017	Section 1 – Updated URL for HETS Companion Guide Table 4 – Updated URL for HETS Companion Guide Table 5 – Updated URL for HETS Companion Guide Table 6 – Updated URL for HETS Companion Guide Table 8 – Updated URL for HETS Companion Guide Table 9 – Updated URL for HETS Companion Guide Table 15 – Updated Symantec URLs	1, 9, 11, 12, 13, 14, 24

Table of Contents

List of Figures

List of Tables

1 Introduction

This document provides information on how to connect to the HIPAA Eligibility Transaction System (HETS) 270/271 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 SOAP and MIME protocols are offered in addition to the Centers for Medicare & Medicaid Services (CMS) Extranet connection. HETS Trading Partners will have the option of using any of the available connection methods to submit and receive eligibility data. The HETS 270/271 application will continue to only accept real-time transactions.

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). The HETS 270/271 follows the federally mandated Phase I CORE 153: Eligibility and Benefits Connectivity Rule and the Phase II CORE 270: Connectivity Rule. For a copy of these federally mandated Operating Rules, please refer to http://www.caqh.org/ORMandate_Eligibility.php

Specifically HETS 270/271:

- Supports SOAP and MIME protocol and associated errors.
- Requires Trading Partners transmitting with SOAP or MIME to obtain a digital certificate and send the transaction to the HETS 270/271 application via a secure internet connection.

It is important to note that this document is intended for use by a technical professional who has experience implementing secure, web-based connectivity.

HETS 270/271 application authenticates the Trading Partner via a unique HETS 270/271 Submitter ID and ensures that the Trading Partner is associated with valid National Provider IDs (NPIs) in the HETS database. If the Trading Partner is not authorized, or is not associated with valid NPIs, then the appropriate X12 error response is returned. Please refer to the HETS Companion Guide found on the HETS Help website (https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/Downloads/HETS270271CompanionGuide5010.pdf) for the errors returned in the above situations.

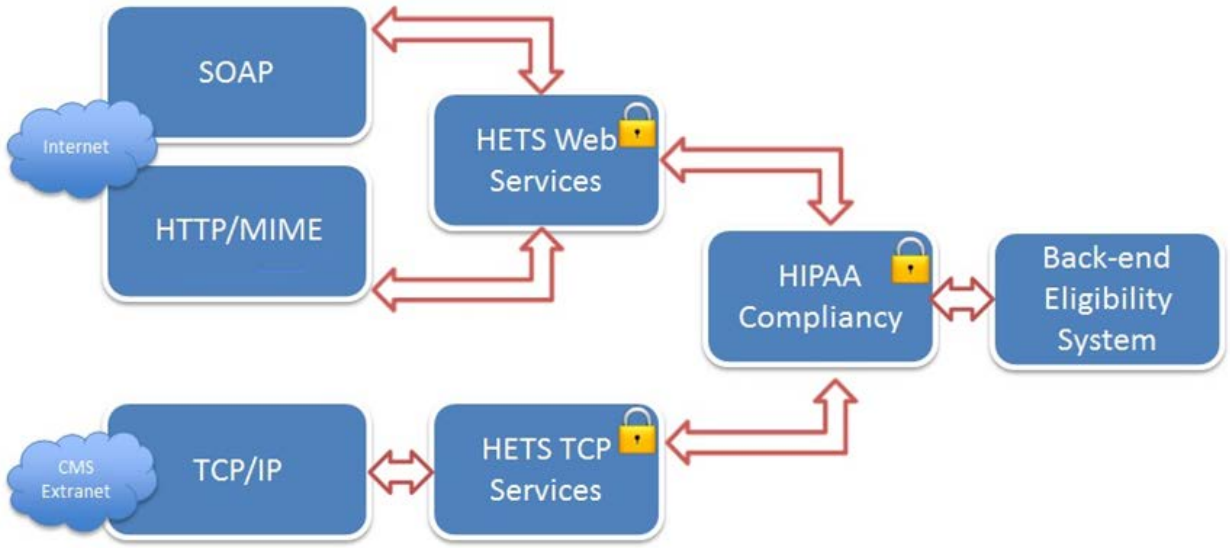
Before submitting a 270 request to the HETS 270/271 application the submitter must ensure that all valid SID/NPI relationships have been added under their new Web Submitter ID in the HETS Desktop (HDT). A Web Submitter ID indicates that the submitter has been set up to submit 270 requests to the HETS 270/271 application using SOAP or MIME. Existing clearinghouse submitters that have access to batch NPI Management can perform actions (query, add, and/or

terminate) for multiple NPIs at one time through batch file functionality. They can use the same HDT user account, mailbox number and file naming convention that they use for their non-Web Submitter IDs. Submitters should ensure that when submitting the HDT Batch file, they include their Web submitter ID rather than their non-Web submitter ID. All other submitters should set up their individual SID/NPI relationships via HDT. The user manual for the HDT application can be found at the following link:

https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/downloads/HDTUserGuide.pdf

Figure 1 illustrates the high-level process for communicating with the HETS 270/271 application. The lock icons represent system checkpoints that must be passed before eligibility information is returned on the 271 response.

Figure 1 – HETS 270/271 Communication Process



2 Authentication and Authorization Handling

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) 1.2 open standard for client certificate-based authentication. TLS 1.2 is required for compliance per the federally-mandated NIST Special Publication 800-52r1.

The Trading Partner's IP address will be verified by CMS prior to allowing the 270 inquiry through to the HETS 270/271 application. Note that the Trading Partner's IP address must be an address from the organization's Production (not Testing) environment. Also note that the supplied Trading Partner IP address must be a public address.

2.1 X.509 Digital Certificates

The Trading Partner's digital certificate will need to be provided to CMS in advance by contacting the MCARE Help Desk during the on-boarding process.

MCARE Contact Information:
Monday to Friday 7am to 7pm ET
1-866-324-7315
MCARE@cms.hhs.gov

HETS Help Website Contact Page:
http://cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/ContactUs.html

MCARE will verify the digital certificate and initiate the process to properly configure Trading Partner access to the HETS system. The same digital certificate is also required for digitally signing the SOAP message timestamp and payload fields as specified in *Section 3.3 Web Services Security Policy*. The SOAP response will also be digitally signed by CMS for authenticity of the message.

The information provided in the following steps should allow the Trading Partners to locate proper digital certificates for HETS connectivity. Trading Partners will need to generate a Certificate Signing Request (CSR) for obtaining the digital certificate for their organization. The CSR generation process is platform-specific. Please review the CSR generation process for your Certificate Authority (CA) carefully, as shown in the links found in the following three subsections, and contact the CAs directly in order to obtain the digital certificate. CMS requires that all Trading Partners using SOAP or MIME use a SHA2-256 digital certificate.

The Trading Partners will need to procure a digital certificate from one of the following CAs detailed in sections 2.1.1 to 2.1.3 in order to allow their infrastructure to connect to the HETS servers. Information on certificate procurement and platform-specific CSR generation processes can be found on each CAs webpage. The links to their home pages has been provided in sections 2.1.1 to 2.1.3.

2.1.1 DigiCert

Information on digital certificates provided by DigiCert can be found using the following link:
http://www.digicert.com

Digital certificates issued by the following DigiCert Intermediate certificates are accepted:

- DigiCert SHA2 Assured ID CA
- DigiCert SHA2 Secure Server CA
- DigiCert SHA2 Extended Validation Server CA
- DigiCert SHA2 High Assurance Server CA
- DigiCert Assured ID CA G2

- DigiCert Global CA G2

2.1.2 Entrust

Information on digital certificates provided by Entrust can be found using the following link: http://www.entrust.net

Digital certificates issued by the following Entrust Intermediate certificates are accepted:

- Entrust Certification Authority – L1K
- Entrust Certification Authority – L1M

2.1.3 Symantec

Information on digital certificates provided by Symantec can be found using the following link: http://www.symantec.com

Digital certificates issued by the following Symantec Intermediate certificates are accepted:

- Symantec Class 3 EV SSL CA – G3
- Symantec Class 3 Secure Server CA – G4
- Symantec Class 3 Extended Validation SHA256 SSL CA

2.2 Overall HETS Web Services Security Policy

HETS Web Services Security Policy assertions will use both transport level and message level security bindings. The information provided for Transport Level Security applies to SOAP and MIME requests. The information provided for Message Level Security applies only to SOAP.

Transport Level Security (Transport Binding) – SOAP and MIME

- Create a SSL connection using a RSA 2048 bit certificate
- CMS requires TLSv1.2 and supports the following cipher suites:
  - TLS_RSA_WITH_3DES_EDE_CBC_SHA
  - TLS_RSA_WITH_AES_128_CBC_SHA
  - TLS_RSA_WITH_AES_256_CBC_SHA
  - TLS_RSA_WITH_AES_128_GCM_SHA256
  - TLS_RSA_WITH_AES_256_GCM_SHA384
  - TLS_RSA_WITH_AES_128_CBC_SHA256
  - TLS_RSA_WITH_AES_256_CBC_SHA256

Message Level Security (Asymmetric Binding) – SOAP ONLY

- Digitally sign the timestamp and payload using an RSA-SHA2 signature algorithm. The submitter's signature algorithm shall be RSA-SHA256.
- Include a Binary Security Token inside the Web Services Security Header
- Include a TimeStamp node in the Web Services Security Header

3 SOAP

The HETS 270/271 application will support transactions formatted according to SOAP Version 1.2, conforming to standards set forth by WSDL for Extensible Markup Language (XML) envelope formatting, submission and retrieval. The X12 payload data must be embedded using the Inline method (CDATA element), and the XML schema and WSDL definitions formatted according to the Phase II CORE 270: Connectivity Rule. The following links should be used as reference:

- 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
- CORE Connectivity Rule: http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf

HETS 270/271 Submitters connecting via SOAP will use the following link to connect and send their 270 requests:

https://soap.hets-270-271.cms.gov/eligibility/realtime/soap

3.1 SOAP Data Requirements

Submitters will need to specify appropriate SOAP headers. The SOAP specifications are precise and require that the headers and the body be constructed perfectly.

SOAP Header
The following link should be used as a reference when constructing the SOAP Header:
http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf

The SOAP Header must include the timestamp element which must be digitally signed. The Web Services Security Binary Security Token must be added to the SOAP Header which is used for verification of the signature.

SOAP Body

The following link should be used as a reference when constructing the SOAP Body: http://www.w3.org/TR/soap12-part1

Only those characters referenced in the Basic and the Extended Character Sets noted in the Appendix of the ASCX12 270/271 version 005010X279A1 TR3 including the 005010X279E1 Errata are acceptable within a HETS 270 inquiry.

Table 2 and *Table 3* describe the required HETS-specific body elements for 270 requests and X12 responses using SOAP.

Table 2 - Required Body Elements for 270 Requests Using SOAP

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/ for more information.
SenderID	This is a submitter defined alphanumeric field. The value must be 10 characters in length. Recommended value is your HETS 270/271 SOAP Submitter ID plus trailing zeros for a total of 10 characters.
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 3 - Required Body Elements for X12 Responses Using SOAP

Element Name	Description
PayloadType	X12_271_Response_005010X279A1, X12_TA1_Response_00501X231A1, X12_999_Response_005010X231A1
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/ for more information.
SenderID	CMS
ReceiverID	This field must be 10 characters in length. The same as the 270 Sender ID.
CORERuleVersion	2.2.0
Payload	X12 response

3.1.1 SOAP Digital Signature

The SOAP communication protocol requires Trading Partners embed their certificate within the eligibility request and digitally sign the SOAP Body Payload and SOAP Header Timestamp using their private key. CMS will embed their certificate in the 271 response enabling the Trading Partner to verify that it came from CMS. Trading Partners can obtain a copy of CMS' Certificate in advance by contacting the MCARE Help Desk.

Refer to the following link for details related to digital signatures as they relate to SOAP:
http://www.w3.org/TR/SOAP-dsig/

3.2 SOAP Examples

Examples of a real time SOAP request and response can be found in Sections 4.2.2.3 and 4.2.2.4 of the CORE Phase II Connectivity Rule at the following link:
http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf

Table 4 provides an example of a 270 request using SOAP. Carriage returns should not be used in the SOAP Body Payload field. They appear in the example information in the HETS Companion Guide for readability purposes only. Also, it is important that the Content-Type line of the HTTP Header and the namespace declaration in the Envelope begin tag contain values associated with SOAP 1.2 as shown below. Using values for SOAP 1.1 or different values may cause the SOAP message to be rejected by HETS. Note: The below example is for illustrative purposes only. All of the variable data will be unique per transaction and should not be copied verbatim and sent to HETS. Lastly, it is highly recommended that the encodingStyle attribute for the Envelope begin tag not be specified.

Table 4 - SOAP Request Message Structure

SOAP Structure Element	Content
HTTP Header	POST https://soap.hets-270-271.cms.gov/eligibility/realtime/soap HTTP/1.1 Accept-Encoding: gzip,deflate Content-Type: application/soap+xml;charset=”UTF-8”;action="RealTimeTransaction" Content-Length: 4808 Host: soap.hets-270-271.cms.gov 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 Web Services 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> yyyy-MM-dd'T'hh:mm:ss'Z'</wsu:Created> <wsu:Expires> yyyy-MM-dd'T'hh:mm:ss'Z'</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/2001/04/xmldsig-more#rsa-sha256"/> <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 Body PayloadType	<PayloadType>X12_270_Request_005010X279A1</PayloadType>
SOAP Body ProcessingMode	<ProcessingMode>RealTime</ProcessingMode>
SOAP Body PayloadID	<PayloadID> d5cf23d4-240d-1d9e-b7d5-ab0f8185296b</PayloadID>
SOAP Body TimeStamp	<TimeStamp> yyyy-MM-dd'T'hh:mm:ss'Z'</TimeStamp>

SOAP Structure Element	Content
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[*270 Request Detail: See Appendix A of the Companion Guide located on HETS Help for an example of the data that would appear here - http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/Downloads/HETS270271CompanionGuide5010.pdf*]]></Payload>
SOAP Body End	</ns1:COREEnvelopeRealTimeRequest> </soap:Body>
SOAP Envelope End	</soap:Envelope>

Table 5 provides an example of a 271 response using SOAP. Carriage returns should not be used in the SOAP Body Payload field. They appear in the example information in the HETS Companion Guide for readability purposes only.

Table 5 - SOAP Response Message Structure

SOAP Structure Element	Content
HTTP 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: application/soap+xml Date: Wed, 27 Jan 2016 15:45:25 GMT X-Client-IP: 111.11.1.1,111.11.1.1 X-Archived-Client-IP: 111.11.1.1
SOAP Envelope Begin	<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
SOAP Header Begin	<soap:Header>
SOAP Header Web Services 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>2016-01-27T15:45:25Z</wsu:Created> <wsu:Expires>2016-01-27T15:46:25Z</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-168"> <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/2000/09/xmldsig#sha1"/> <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/2000/09/xmldsig#sha1"/> <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: COREEnvelopeRealTimeResponse xmlns:ns1="http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd">
SOAP Body PayloadType	<PayloadType> X12_271_Response_005010X279A1</PayloadType>
SOAP Body ProcessingMode	<ProcessingMode>RealTime</ProcessingMode>
SOAP Body PayloadID	<PayloadID> d5cf23d4-240d-1d9e-b7d5-ab0f8185296b </PayloadID>
SOAP Body TimeStamp	<TimeStamp> yyyy-MM-dd'T'hh:mm:ss'Z'</TimeStamp>

SOAP Structure Element	Content
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[*271 Response Detail: See Appendix B of the Companion Guide located on HETS Help for an example of the data that would appear here - http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/Downloads/HETS270271CompanionGuide5010.pdf*]]></Payload>
SOAP Body End	</ns1: COREEnvelopeRealTimeResponse> <ErrorCode>Success</ErrorCode> <ErrorMessage/> </soap:Body>

4 MIME

HETS will support standard MIME messages. The MIME format used must be multipart/form-data.

Only those characters referenced in the Basic and the Extended Character Sets noted in the Appendix of the ASCX12 270/271 version 005010X279A1 TR3 including the 005010X279E1 Errata are acceptable within a HETS 270 inquiry.

CORE does not specify the naming conventions as a mandate. HETS will implement the MIME body parts with the same field names as the SOAP element nodes. The response will be returned as MIME multipart/form-data, with the Payload body part containing the X12 response.

HETS 270/271 Submitters connecting via MIME will use the following link to connect and send their 270 requests:

https://mime.hets-270-271.cms.gov/eligibility/realtime/mime

4.1 MIME Data Requirements

Submitters must specify appropriate MIME headers. The MIME specification is very precise, and requires that the headers and the body be constructed perfectly. The HETS implementation of MIME allows for the use of the Basic and Extended Character Sets as noted in the Appendix of the ASCX12 270/271 version 005010X279A1 TR3 including the 005010X279E1 Errata only. Please refer to the RFC 2388 – returning values from Forms: multipart/form-data to review header and body specifications. The RFC 2388 can be found at the following link: http://www.faqs.org/rfcs/rfc2388.html

MIME Header
MIME Messages will have standard HTTP header data elements, such as POST, HOST, Content-Length and Content-Type. The supported Content-Type is MIME multipart/form-data.

MIME Body
Required HETS-specific body elements for 270 requests and X12 responses using MIME are defined in *Table 6* and *Table 7*.

Table 6 - Required Body Elements for 270 Requests Using 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/ for more information.
SenderID	This is a submitter defined alphanumeric field. The value must be 10 characters in length. Recommended value is your HETS 270/271 MIME Submitter ID plus trailing zeros for a total of 10 characters.
ReceiverID	CMS
CORERuleVersion	2.2.0
Payload	X12 request. The X12 request must be submitted as part of the MIME request and not as an attachment. If an attachment is received, the transaction will be rejected. The request does not need to be enclosed within a CDATA tag. *See Appendix A of the Companion Guide located on HETS Help for an example of the 270 request that would appear here - http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/Downloads/HETS270271CompanionGuide5010.pdf*

Table 7 - Required Body Elements for X12 Responses Using MIME

Element Name	Description
PayloadType	X12_271_Response_005010X279A1, X12_TA1_Response_00501X231A1, X12_999_Response_005010X231A1
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/ for more information.
SenderID	CMS
ReceiverID	This field must be 10 characters in length. The same as the 270 Sender ID.
CORERuleVersion	2.2.0
Payload	X12 response

4.2 MIME Examples

Examples of a real time MIME request and response can be found in Sections 4.2.1.1 and 4.2.1.2 of the CORE Phase II Connectivity Rule at this link:
http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf

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

Refer to *Table 8* in this document for the HETS-specific body elements.

MIME Request and Response Examples:

Table 8 and *Table 9* provide examples of a 270 request and 271 response using HTTP MIME Multipart. The below examples are for illustrative purposes only. All of the variable data will be unique per transaction and should not be copied verbatim and sent to HETS.

Table 8 - MIME Request Message Structure

MIME Structure Element	Content
MIME Header	POST https://mime.hets-270-271.cms.gov/eligibility/realtime/mime HTTP/1.1 Connection: keep-alive Content-Length: 1392 Content-Type: multipart/form-data; boundary=COSZiva9NdnYzPXUEGy-tLBO8n4-czud Host: mime.hets-270-271.cms.gov User-Agent: Apache-HttpClient/4.2.1 (java 1.5)
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" d5cf23d4-240d-1d9e-b7d5-ab0f8185296b --COSZiva9NdnYzPXUEGy-tLBO8n4-czud Content-Disposition: form-data; name="TimeStamp" 2016-02-25T19:50:40.611Z --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 ****The 270 request will appear here beginning with the ISA segment and ending with the IEA segment as shown in the example from Appendix A of the HETS Companion Guide located on HETS Help at http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/Downloads/HETS270271CompanionGuide5010.pdf****

Table 9 - 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-Client-IP: 111.11.1.1,111.11.1.1 X-Global-Transaction-ID: 237915383 (User Defined) Content-Type: multipart/form-data; boundary="7aaeaf96-1e54-4567-a8d0-e93de77cd66a" POST: https://mime.hets-270-271.cms.gov/eligibility/realtime/mime
MIME Body	--7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name="PayloadType" X12_TA1_Response_005010X279A1 --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name="ProcessingMode" RealTime --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name="PayloadID" d5cf23d4-240d-1d9e-b7d5-ab0f8185296b --7aaeaf96-1e54-4567-a8d0-e93de77cd66a Content-Disposition: form-data; name="TimeStamp" 2016-02-25T19:50:40.611--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" ****The 271 response will appear here beginning with the ISA segment and ending with the IEA segment as shown in the example from Appendix B of the HETS Companion Guide located on HETS Help at http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/Downloads/HETS270271CompanionGuide5010.pdf**** --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

5 Common Error Processing for SOAP and MIME

The HETS 270/271 application will process SOAP and MIME transactions and return errors as described in this section.

5.1 HTTP Status and Error Codes

The processing and error codes for the HTTP layer are defined as part of the HTTP specifications as noted at the following link:
http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html

The intended use of these status and error codes in processing transactions is specified in Table 4.3.3.1 of the Phase II CORE 270: Connectivity Rule. This document is located at the following link:
http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf

5.2 CORE Envelope Processing Status and Error Codes

Table 10 describes envelope processing status and error codes specific to the HETS 270/271 application for SOAP and MIME transactions.

Table 10 - Envelope Process Status and Errors

Element Name	Description
<FieldName>Illegal	Illegal value provided for <FieldName>.
<FieldName>Required	The field <FieldName> is required but was not provided.
VersionMismatch	The CORERuleVersion sent is not acceptable to the Receiver.
Success	Envelope was processed successfully.

5.3 SOAP Specific Processing Errors

Table 11 describes examples of SOAP processing errors.

Table 11 - SOAP Specific Processing Errors

Element Name	Description
UnAuthorized	The signature could not be verified.

5.4 SOAP and MIME Transaction (X12) Error Processing

Refer to the HETS Companion Guide for additional information on the transaction processing errors that will be returned as a SOAP message or MIME Multipart/form-data containing the related response.

The HETS Companion Guide can be found in the downloads section of the CMS HETS Help web site at the following link:
http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/index.html

6 General On-boarding Checklist

If the Trading Partner is a new HETS submitter, they must first follow the traditional enrollment processes, which can be found at on the "How to Get Connected – HETS 270/271" page of the HETS Help website and includes the completion of the Trading Partner Agreement (TPA). It will take approximately two weeks to complete this process. The HETS Help website can be found at the following link:
http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/index.html

If the Trading Partner already has a HETS Submitter ID (SID), or if it has just completed the traditional enrollment process, the following steps can serve as a general guide to the on-boarding process for SOAP/MIME submissions. It will take approximately 2 weeks to complete this process.

Table 12 - General On-boarding Checklist

- ☐ When the Trading Partner contacts MCARE to request access to use SOAP/MIME, they must have already purchased an X.509 Digital Certificate and be prepared to provide the following information:
    - Organizational Legal Business Name
    - Organization Submitter ID (SID) if previously assigned
    - Organization originating IP address(es) that will be linked to the certificate
    - X.509 Digital Certificate Issuer Name
    - X.509 Digital Certificate Type
    - X.509 Digital Certificate Serial Number
- ☐ The Trading Partner should email the X.509 Digital Certificate to MCARE in (.PEM) format to MCARE at MCARE@cms.hhs.gov. The Trading Partner should not include the private key when sending the digital certificate.
- ☐ MCARE will review the digital certificate. If there are issues or errors, MCARE will notify the Trading Partner and assist in the resolution.
- ☐ Upon validation of the Digital Certificate, MCARE will work with the HETS team to provide access to the Trading Partner.
- ☐ Once access has been provided, MCARE will inform the Trading Partner and work with them to verify transactions can be sent successfully.

- After successfully implementing HETS via SOAP or MIME (i.e., sending a good 270 request and receiving a proper 271 response), the Trading Partner's Submitter ID status will be moved from 'Test' to 'Production. The Trading Partner may then send regular Medicare eligibility traffic to HETS.

Appendix A: HETS Web Services Security Policy

The following text is an example of the XML Schema.

```
<?xml version="1.0" encoding="utf-8"?>
<wsp:Policy xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:sp="http://docs.oasis-open.org/ws-sx/ws- securitypolicy/200702"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
  <wsp:ExactlyOne>
    <wsp:All>
      <wsp:Policy wsu:Id="transport-ssl-client-cert">
        <sp:TransportBinding>
          <wsp:Policy>
            <sp:TransportToken>
              <wsp:Policy>
                <sp:HttpsToken>
                  <wsp:Policy>
                    <sp:RequireClientCertificate/>
                  </wsp:Policy>
                </sp:HttpsToken>
                <sp:HttpsToken RequestClientCertificate="true"/>
              </wsp:Policy>
            </sp:TransportToken>
            <sp:AlgorithmSuite>
              <wsp:Policy>
                <wsp:ExactlyOne>
                  <sp:Basic256Sha256/>
                </wsp:ExactlyOne>
              </wsp:Policy>
            </sp:AlgorithmSuite>
            <sp:IncludeTimestamp/>
          </wsp:Policy>
        </sp:TransportBinding>
      </wsp:Policy>
      <sp:AsymmetricBinding>
        <wsp:Policy>
          <sp:RecipientSignatureToken>
            <wsp:Policy>
              <sp:X509Token
                sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToInitiator">
                <wsp:Policy>
                  <sp:WssX509V3Token10/>
                </wsp:Policy>
```

```
          </sp:X509Token>
        </wsp:Policy>
      </sp:RecipientSignatureToken>
      <sp:InitiatorSignatureToken>
        <wsp:Policy>
          <sp:X509Token
            sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient">
            <wsp:Policy>
              <sp:WssX509V3Token10/>
            </wsp:Policy>
          </sp:X509Token>
        </wsp:Policy>
      </sp:InitiatorSignatureToken>
      <sp:AlgorithmSuite>
        <wsp:Policy>
          <sp:Basic256Sha256/>
        </wsp:Policy>
      </sp:AlgorithmSuite>
      <sp:Layout>
        <wsp:Policy>
          <sp:strict/>
        </wsp:Policy>
      </sp:Layout>
      <sp:IncludeTimestamp/>
    </wsp:Policy>
  </sp:AsymmetricBinding>
  <sp:EndorsingSupportingTokens
    xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702">
    <wsp:Policy>
      <sp:X509Token
        sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToRecipient">
        <wsp:Policy>
          <sp:WssX509V3Token10/>
        </wsp:Policy>
      </sp:X509Token>
    </wsp:Policy>
    <wsp:Policy>
      <sp:X509Token
        sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-
securitypolicy/200702/IncludeToken/AlwaysToInitiator">
        <wsp:Policy>
          <sp:WssX509V3Token10/>
        </wsp:Policy>
```

```
    </sp:X509Token>
   </wsp:Policy>
  </sp:EndorsingSupportingTokens>
  <wsp:Policy wsu:Id="request_parts">
   <sp:SignedElements>
    <sp:XPath>/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and local-
    name()='Envelope']/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and
    local-name()='Header']/*[namespace-uri()='http://docs.oasis-open.org/wss/2004/01/oasis-
    200401-wss-wssecurity-secext-1.0.xsd' and local-name()='Security']/*[namespace-
    uri()='http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd'
    and local-name()='Timestamp']</sp:XPath>
    <sp:XPath>/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and local-
    name()='Envelope']/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and
    local-name()='Body']/*[namespace-
    uri()='http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd' and local-
    name()='COREEnvelopeRealTimeRequest']/Payload</sp:XPath>
   </sp:SignedElements>
  </wsp:Policy>
  <wsp:Policy wsu:Id="response_parts">
   <wsp:ExactlyOne>
     <wsp:All>
      <sp:SignedElements>
       <sp:XPath>/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and
            local-name()='Envelope']/*[namespace-uri()='http://www.w3.org/2003/05/soap-
            envelope' and local-name()='Header']/*[namespace-uri()='http://docs.oasis-
            open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd' and local-
            name()='Security']/*[namespace-uri()='http://docs.oasis-
            open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd' and local-
            name()='Timestamp']</sp:XPath>
      <sp:XPath>/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and
            local-name()='Envelope'] /*[namespace-uri()='http://www.w3.org/2003/05/soap-
            envelope' and local-name()='Body']/*[namespace-
            uri()='http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd' and local-
            name()='COREEnvelopeRealTimeResponse']/Payload</sp:XPath>
     </sp:SignedElements>
    </wsp:All>
    <wsp:All>
     <sp:SignedElements>
      <sp:XPath>/*[namespace-uri()='http://www.w3.org/2003/05/soap-envelope' and
            local-name()='Envelope']/*[namespace-uri()='http://www.w3.org/2003/05/soap-
            envelope' and local-name()='Header']/*[namespace-uri()='http://docs.oasis-
            open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd' and local-
            name()='Security']/*[namespace-uri()='http://docs.oasis-
            open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd' and local-
            name()='Timestamp']</sp:XPath>
```

```
        </sp:SignedElements>
       </wsp:All>
      </wsp:ExactlyOne>
    </wsp:Policy>
    <wsaw:UsingAddressing xmlns:wsaw="http://www.w3.org/2005/08/addressing"/>
   </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Appendix B: Frequently Asked Questions

Table 13 - Frequently Asked Questions

Question Number	Question	Answer
1	Do I need my own digital certificate for exchanging 270/271 via SOAP and MIME methods?	Yes. The User ID and Password authentication method are not supported by HETS. Instead a Trading Partner must procure a digital certificate and configure their system to connect to HETS.
2	Are there specific Digital Certificates that can access HETS?	Sections 2.1.1 through 2.1.3 contain information regarding digital certificate issuance.
3	What specific connectivity configurations must I complete for a successful SOAP connection?	Trading Partners using SOAP are encouraged to ensure the following: • the SOAP communication protocol requires Trading Partners to send their certificate and digitally sign the payload and timestamp using their private key. This allows HETS to validate the contents of the received message and when it was sent. • the "wsu:ID" attribute is contained in both the timestamp and payload nodes. They should both match the "<Reference URI". • that their perimeter equipment IP range or subnet has been provided to MCARE for configuration within CMS firewall. • their application makes use of PKI and configures the Trading Partner keystore with the correct client certificate to sign the SOAP messages. • the trust store is correctly configured with the CMS certificate.
4	Do I need a VPN over internet for connection to HETS 270/271 Application?	A VPN connection to CMS is not required for connectivity to the HETS 270/271 Application.
5	What is the difference between SOAP and MIME transactions, specific to the HETS 270/271 Application?	From the Trading Partner's perspective, the HETS 270/271 Application has two different URLs for sending these transactions. The processing for both MIME and SOAP is the same.
6	How do I go about developing my SOAP or MIME client?	HETS does not require any specific tool for client side implementation. The Trading Partners are free to choose various COTS products or custom code to create the SOAP & MIME requests.
7	How do I wrap a 270 transaction for submission?	For SOAP transactions, the Trading Partners must ensure that the 270 transaction is contained in the payload tag and the "CDATA" tag is present. For MIME transactions, the Trading Partners must ensure that the 270 transaction is contained in the payload MIME boundary. MIME does not use CDATA tags and it should not be present.
8	Can I send more than one 270 in a single SOAP or MIME request?	No. Only one 270 should be submitted per SOAP or MIME request. The HETS 270/271 Application does not support batch.
9	Can I send my transactions as SOAP or MIME attachments?	No. The 270 transactions should be sent as part of payload tag in SOAP requests. For MIME requests they should be sent in-line, as part of the payload element.

Question Number	Question	Answer
10	Do I need to use a User ID / Password when establishing a connection to HETS to submit SOAP or MIME transactions?	No. The HETS 270/271 Application connection authentication requirements are based only on digital certificates.
11	Does the SID used in the SOAP message body need to match the X12 SID?	Trading Partners should ensure that the submitter IDs match. However, the HETS 270/271 Application uses only the SID embedded in X12 270 transaction for authorization.
12	How can we ensure the digital certificate doesn't get activated until MCARE validates and authorizes the submitter?	The certificate will be active the day it was issued to the Trading Partner. However, MCARE will ensure that access to the firewall is allowed only after the certificate verification step is complete.
13	What happens when an organization is revoked by their CA?	The Certificate Revocation Lists for each CA will be loaded into the production environment infrastructure and those Trading Partners that attempt submission with a revoked digital certificate will be denied access through the CMS firewall.
14	How will the Trading Partner get the WS_Policy also known as the Web Services Security Policy?	The Trading Partners should receive a copy of the WS-Policy document during on-boarding process. See Section 6.
15	The submitter is receiving "Error getting response; javax.net.ssl.SSLHandshakeException: Received fatal alert: handshake_failure". What does that mean?	The 2-way SSL handshake process did not complete successfully. This is most likely due to either the submitter not having configured a 2-way SSL on their end or an invalid or revoked digital certificate is being used.
16	What types of attachments can be included in a MIME transaction?	No attachments can be included in the MIME transaction. The 270 request must be encoded in-line to the MIME message.If the MIME transaction is submitted with the X12 request in an attachment, the submitter must ensure that the attachment has a (.txt) file extension. If an attachment with a file extension other than (.txt) is received, the transaction will be rejected.

Appendix C: References

Table 14 - References

Document	Hyperlink
CORE Connectivity	http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf
CORE Mandated Operating Rules	http://www.caqh.org/ORMandate_Eligibility.php
CORE Phase II Connectivity Rule	http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf
Digicert Certificate Procurement	http://www.digicert.com/welcome/ssl-plus.htm
Digicert CSR Generation	http://www.digicert.com/csr-creation.htm
Entrust Certificate Procurement	http://www.entrust.net/ssl-certificates/advantage.htm
Entrust CSR Generation	http://www.entrust.net/ssl-technical/csr_faq.cfm
HDT User Guide	https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/HETSHelp/downloads/HDTUserGuide.pdf
HTTP Specifications	http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html
MIME Header & Body Specifications	http://www.faqs.org/rfcs/rfc2388.html
SOAP Body	http://www.w3.org/TR/soap12-part1
SOAP Header	http://www.caqh.org/pdf/CLEAN5010/270-v5010.pdf
SOAP XML Schema	http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.xsd
Symantec Certificate Procurement	https://www.websecurity.symantec.com/
Symantec CSR Generation	https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&id=INFO235&actp=LIST&viewlocale=en_US
Timestamp Element Format	http://www.w3.org/TR/xmlschema11-2/
WSDL Information	http://www.caqh.org/SOAP/WSDL/CORERule2.2.0.wsdl
XML Schemas	http://www.w3.org/TR/SOAP-dsig

Appendix D: Glossary of Terms

Table 15 - Glossary of Terms

Term	Acronym	Definition
Affordable Care Act	ACA	The Affordable Care Act puts families and small businesses in control of their own health care by allowing them to choose the insurance coverage that works best for them in an open, competitive insurance market.
Certificate Authority	CA	Issues digital certificates after verifying the identity of the applicant.
Council for Affordable Quality Healthcare – Committee on Operating Rules for Information Exchange	CAQH CORE	A multi-stakeholder initiative developing operating rules that streamline electronic healthcare administrative data exchange and support interoperability between payers and providers.
Centers for Medicare & Medicaid Services	CMS	CMS is a Federal agency within the United States Department of Health and Human Services that administers the Medicare program and works in partnership with state governments to administer Medicaid, the State Children's Health Insurance Program, and health insurance portability standards.
Certificate Signing Request	CSR	A message which is sent from an applicant to a Certificate Authority in order to apply for a digital certificate.
HIPAA Eligibility Transaction System	HETS	HETS allows the release of eligibility data to Medicare Providers, Suppliers, or their authorized billing agents for the purpose of preparing an accurate Medicare claim, determining Beneficiary liability, or determining eligibility for specific services. There are two ways to inquire for eligibility. See HETS 270/271 and HETS User Interface.
Department of Health and Human Services	HHS	HHS is a Cabinet department of the United States government with the goal of protecting the health of all Americans and providing essential human services.
Health Insurance Portability and Accountability Act	HIPAA	Title I of the Health Insurance Portability and Accountability Act protects health insurance coverage for workers and their families when they change or lose their jobs. Title II of the Health Insurance Portability and Accountability Act, known as the Administrative Simplification provisions, requires the establishment of national standards for electronic health care transactions and national identifiers for providers, health insurance plans, and employers.
Hypertext Transfer Protocol	HTTP	An application level protocol for distributed, collaborative, hypermedia information systems.
HIPAA Eligibility Transaction System (HETS) Desktop	HDT	The HDT is an Internet-facing application that assists clearinghouses in validating Medicare Legacy Provider and National Provider Identifier numbers.
Medicare Customer Assistance Regarding Eligibility Help Desk	MCARE Help Desk	The MCARE Help Desk is a single point of contact for all submitters facing inquiries regarding the HETS eligibility systems.
Multipurpose Internet Mail Extensions	MIME	An internet standard that extends the format of email to support header and text in email that uses text attachments and multiple part message bodies.

Term	Acronym	Definition
National Provider Identifier	NPI	An NPI is a unique 10-digit identification number issued to health care providers in the United States by Centers for Medicare & Medicaid Services. Covered health care providers and all health plans and health care clearinghouses must use the NPI in the administrative and financial transactions adopted under Health Insurance Portability and Accountability Act.
Submitter ID Number	SID	The submitter ID number is the value that identifies your organization to the HETS 270/271 system.
Simple Object Access Protocol	SOAP	A protocol specification for exchanging structured information in the implementation of Web Service in computer networks that relies on XML Information Set for its message format.
Transmission Control Protocol/Internet Protocol	TCP/IP	Defines the rules that computers must follow to communicate with each other over the internet.
Transport Layer Security	TLS	A protocol that ensures privacy between communicating applications and their user on the Internet. This protocol replaces Secure Socket Layer (SSL).
Web Services Description Language	WSDL	WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document or procedure oriented information.
Extensible Markup Language	XML	Defines a set of rules for encoding documents.