

270/271 Enhanced Eligibility Initiative Batch Transaction Connectivity Specifications and Supporting Documents

The MN HIPAA Collaborative members, and other participating organizations, have agreed to endorse the recommendations for the “Enhanced Eligibility” search criteria, transaction data content, and the connectivity protocol OpenSSH for Batch processing of the 270/271 transaction. The purpose of this document is to define the various parameters around using OpenSSH on the public Internet to connect directly to participating payer systems for transmission and receipt of the ANSI ASC X12N 270/271 Eligibility Inquiry and Response transaction.

This document is intended for the use of provider information systems departments, practice management or hospital information system vendors, and clearinghouses that provide eligibility verification services.

Payers Supporting This Connection Method

Each payer’s production ready time frame to support this specific method of connection varies. Please contact directly the payer to which you wish to connect to determine their target implementation date.

- Blue Cross Blue Shield of Minnesota (an independent licensee of the Blue Cross and Blue Shield Association)
- HealthPartners
- Medica
- Minnesota Department of Human Services
- PreferredOne
- UCare Minnesota

Guiding Principles

The Minnesota HIPAA Collaborative’s intention is to:

- Support the use of connectivity standards that will incur little or no cost to trading partners.
- Support protocols that are readily available and supported by most hardware/software vendors or available on the Internet, and are widely accepted and used across the industry.
- Be supported by all payers and minimize the “customization” needed for direct connection to each payer.
- Offer the ability to connect to all participating payers directly in a similar or identical way.
- Provide a method for supporting the HIPAA transaction standard for system-to-system EDI.
- Define a method of connection that is acceptable to all participants, but is not exclusive. Participating payers may offer other options for direct connection to perform Batch 270/271 transactions. You should contact each payer directly to determine other available options.

General Description of the Connection Method

- A detailed definition of OpenSSH and how to use it can be found at OpenSSH.com.
- OpenSSH supports full encryption of transmissions over the Internet, including User IDs and Passwords. If you are using a commercially available version of SSH, there may be issues of interoperability. Please contact your trading partner to determine if the use of a version of SSH other than OpenSSH version 3.9 is compatible with their environment.
- This OpenSSH connection method is designed for use in a traditional EDI (machine-to-machine) process.
- This Batch 270/271 transaction process is designed for either single patient/member inquiries, or for more traditional, large batch inquiry submission. It is not designed to support Real Time 270/271. Capacity and response time may vary by payer, as their internal system environments will differ.
- It is recommended that batch files contain no more than 99 inquiries per Transaction Set (ST-to-SE loop) as is defined in the ASC X12N 004010X092A1 Implementation Guide. One or more transaction sets can be contained within a Functional Group and Interchange. However trading partners should avoid creating very large files as the translation and processing of such large files could impact system performance and response time. Please contact each trading partner directly to determine if there is a maximum file size limitation.

Security and Authentication Requirements

- All Collaborative members will use ID and Password as the default authentication requirements. Participating organizations have agreed to support ID & Password authentication as a minimum criteria.
- User ID and Password authentication may be scripted to be executed as a part of the OpenSSH login process, or the ID and Password may be entered interactively (see Example SCP Login Script).
- Issuance, maintenance and control of Password requirements may vary by trading partner. You should contact each trading partner to determine their ID and Password policies and requirements.
- You should contact each trading partner directly to obtain an appropriate ID and Password.
- Some trading partners may require additional security for file transmissions, such as the use of Gnu Privacy Guard (GPG), or PGP for additional encryption. Please contact each trading partner directly to determine if such additional encryption is required for this transaction.

Response Time, Time Out Parameters and Re-Transmission

- The ASC X12N 004010X092A1 Implementation Guide recommends a 24 hour or less response time for the Batch implementation of this transaction. All participating payers will attempt to generate a response to a Batch 270 Eligibility Inquiry transaction within the 24 hour time frame, and optimally far less than 24 hours. Participating payers understand the time sensitive nature and the business process needs of the provider community, and will attempt to respond to Batch 270's received in the least amount of time possible. Actual response time may vary from payer to payer.
- There may be circumstances under which a corresponding 271 Eligibility Response transaction may take longer than the recommended 24 hours. Therefore, we suggest that submitting providers or clearinghouses using this Batch process define an appropriate follow up procedure for inquiries not responded to within the recommended 24 hour period. Because the actual routing and processing of a

Batch 270/271 could exceed the 24 hour window, we recommend submitters wait a full 24 hours before resubmitting the specific patient inquiry.

- In the event that an OpenSSH batch reply message is not received within the maximum 24 hour response period, we recommend provider or clearinghouse submitters send a duplicate 270 transaction no sooner than 24 hours after the original attempt.
- If no response is received after the second attempt, we recommend providers or clearinghouses contact the payer by phone to determine if system availability problems exist.

Response Message Options and Error Notification

- The OpenSSH process is an asynchronous file submission and file retrieval process. The response sent back from the message receiver (e.g. the various payers) can fall into several different categories. The possible response messages/transactions could be one or more of the following:
 1. A negative 997 could be received if the submitted 270 transaction failed syntax compliance validation. A positive 997 or TA1 will be sent if the submitter has requested a TA1 (by assigning a value of 1 in the 14th element of the ISA), or the submitter has requested the receipt of a positive 997 functional acknowledgement.
 2. Payers may return a 271 to communicate semantic compliance failure to the submitter. Each payer will have their documented use of the 271 for semantic error notification on their respective web sites, or through their EDI support departments. The level of compliance validation for 270 transactions may vary by payer. Therefore, you should contact each payer to determine their compliance validation requirements.
 3. The optimum reply transaction is the 271 Eligibility Response. Participating payers have agreed to a common set of search criteria (see “270/271 Search Parameters” – [make a doc link on the Collab web site](#)) and eligibility and benefit information (see “MN HIPAA Collaborative Enhanced Eligibility Data Content Document” ([make a doc link on the Collab web site](#))). These documents can also be found on the MN HIPAA Collaborative web site. If processing errors occur, the 271 will contain AAA segments with standard code values describing the reason for failure. Please see the ASC X12N 004010X092A1 Implementation Guide for details.

What is Needed to Set Up This Method of Connection

- Access to the public Internet.
- If a trading partner elects to use a client software package for transmission/receipt of files using OpenSSH, an OpenSSH compatible client software package can be obtained at no cost at one of the following sites:
 - OpenSSH.com
 - OpenBSD.com
- The MN HIPAA Collaborative does not endorse or recommend any specific client software application. The OpenSSH site indicates several software products which are compatible with OpenSSH version 3.9 at the time this document is issued. Each trading partner is responsible for verifying that the client software they elect to use is compatible with the currently recommended version of OpenSSH. The following are client software applications available on the OpenSSH or OpenBSD web sites for download and use:
 - PuTTY, TTSSH, MSSH, WinSCP, or Cygwin
- We recommend you provide this document to your information systems department, your practice management system or hospital system vendor, or your clearinghouse as they may need to create or

modify an applications software program to support this connection method or incorporate the use of a client software application defined above.

- This Batch 270/271 is designed for single or multiple inquiries using the HIPAA standard transaction in true EDI mode (machine-to-machine). It is not intended to support web browser-driven inquiries or screen-scraping utilities.
- It is recommended that all parties using OpenSSH upgrade to any future releases within six (6) months of the formal release of a new version of the software, unless significant flaws or issues are uncovered with version 3.9, or future versions. In the event of the discovery of any such flaw or serious issue, upgrading to the newer replacement version should occur within 30 days or less. Participating payers will attempt to monitor the OpenSSH site for announcement of any such issues, and will contact their trading partners in the event an upgrade is required.

Examples of SSH Secure Copy Usage (SCP Login Script)

- Interactive usage of scp, the client provided with OpenSSH:

```
CommandPrompt> scp user1@172.23.102.33:/tmp/somefile .
User1@172.23.102.33's password:
somefile          100% 1571  310.3KB/s  00:00
CommandPrompt>
```

This example transfers the file named /tmp/somefile from the host with IP address 172.23.102.33 and copies it to the current directory. A hostname or domain URL could have been used instead of an IP address.

- **Scripted usage of scp using the expect command. (This allows scripted usage of a user ID and password.)**

When scripting use of the OpenSSH client, scp, on Unix, the password cannot be supplied by redirecting input from a file containing the password. A utility like the GNU expect command must be used.

The expect script is as follows:

```
#!/usr/bin/expect -f
# scp.put-file.exp
#
# Usage: scp.put-file.exp LocalFile LoginID HostName Password RemoteFile

set timeout 5

if {[llength $argv] != 5} then {
    puts "Usage: scp.put-file.exp LocalFile LoginID HostName Password RemoteFile"
    exit}

set LocalFile [lindex $argv 0] ; # Local file to put
set LoginID [lindex $argv 1] ; # Remote host login ID
set HostName [lindex $argv 2] ; # Remote hostname
set Passwd [lindex $argv 3] ; # Remote host login ID password
```

```

set RemoteFile [lindex $argv 4] ; # Remote File
set Prompt "($HostName>|:>)"
set GetPass "(passwd|Passwd)"
# expect -re $Prompt

puts "LocalFile: $LocalFile"
puts "login: $LoginID"
puts "HostName: $HostName"
puts "Passwd: Passwd"
puts "RemoteFile: $RemoteFile"
puts "Prompt: $Prompt"

# Connect to host

spawn scp $LocalFile $LoginID@$HostName:$RemoteFile

expect "password:"
send "$Passwd\n"
expect
wait
interact

```

An example execution of the script is:

```

prompt> ./scp.exp /etc/passwd someuser transfer1 password /home/someuser/ppppppj
LocalFile: /etc/passwd
login: someuser
HostName: transfer1
Passwd: password
RemoteFile: /home/someuser/ppppppj
Prompt: (transfer1>|:>)
spawn scp /etc/passwd someuser@transfer1:/home/someuser/ppppppj
someuser@transfer1's password:
passwd          100% |*****| 2914    00:00

```

This example copies a file from a local host to a remote host.

- **Interactive usage of pscp, the file transfer client provided as a companion to putty:**

```

C:\>pscp username1@172.23.102.33:/etc/somefile .

```

The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.

The server's rsa2 key fingerprint is:
ssh-rsa 1024 f3:fc:a3:f9:22:bd:77:a7:6a:ba:80:82:91:49:eb:bc

If you trust this host, enter "y" to add the key to PuTTY's cache and carry on connecting.

If you want to carry on connecting just once, without adding the key to the cache, enter "n".

If you do not trust this host, press Return to abandon the

connection.

Store key in cache? (y/n) y

username1@172.23.102.33's password:

somefile | 1 kB | 1.5 kB/s | ETA: 00:00:00 | 100%

This example includes the dialog that allows the client to accept the server side ssh key. A similar dialog also appears when first connecting to a host when using scp.