



Batch Claim Status Inquiry Setup

Document Revision History

Version	Date	Author	Description
1	2002	Tracey Miller	Initial Authoring
2	June 17, 2004	Laurance Stuntz	Updated Template, minor edits

Table of Contents

Purpose	3
Batch Record	3
Layout	3
Guidelines	4
Sample File	4
Extract Process Setup Guidelines	4
Database	6
Data Model	6
Database Tables	7
<i>Transactions</i>	7
<i>Codes</i>	8
<i>ClaimStatusRequests</i>	8
<i>ClaimStatusProviderRequests (New)</i>	8
<i>ClaimStatusHeaderResponses (New)</i>	9
<i>ClaimStatusLineResponses (New)</i>	10
Batch Applications	11
ProviderSend276	11
StoreReceive277	11
<i>Claim Status Response Codes</i>	11
Appendix A – Batch Script	12
Purpose	12
Sample VB Script (file extension of .vbs)	12

Purpose

The batch extract is a batch-mode solution for submitting claim status inquiry transactions to any payers connected to NEHEN. The extract file is created from the hospital or practice management system based on a defined set of criteria. An extract file is created for each payer, which provides the flexibility to schedule each batch at various times. The files are transferred using FTP to a known directory on the NEHEN production server.

The Send276 application is used to process the batch files. The application opens the file and sends a claim status inquiry request to the appropriate payer for each record in the file. The response records are loaded into NEHENLite, which the users can utilize for reviewing the information.

Batch Record

Layout

FIELD NAME	REQUIRED	VALID VALUES OR FIELD FORMAT
User	Y	To be Determined
Payer	Y	NEHEN002 (Tufts) NEHEN003 (HPHC) NEHEN013 (NHP) NEHEN042 (Network Health)
Provider's Last Name or Institution Name	Y	
Provider's First Name	N	
Provider's Identification Number	Y	
Subscriber's Last Name	N	
Subscriber's First Name	N	
Subscriber's Policy Number	N	
Patient's Last Name	Y	
Patient's First Name	N	
Patient's Policy Number	Y	
Patient's Date of Birth	Y	CCYYMMDD
Patient's Gender	Y	F, M or U
Account Number	N	
Payer Claim Identification Number	N	
Starting Date of Service	Y	CCYYMMDD
Ending Date of Service	N	CCYYMMDD
Claim Amount	N	

Guidelines

1. Create a separate file for each payer. This is not required, but makes it easier to submit batches to different payers at different times, and to resubmit batches if necessary.
2. A comma is used to separate fields in the record. Each record is one line in the file.
3. Put double quotes before and after each field in the record.
4. The user field may contain a user's ID, service code, pre-defined batch ID or any value available in your information system. These values are stored in the UserDetails table in the NEHEN database. For example if the extract process populates this field with user IDs, all user IDs would need to be stored in the NEHEN database.
5. If the patient is not the subscriber, the subscriber elements are required.
6. If the ending date of service is blank, the application will use the start date as the end date.
7. The record must contain a placeholder for each required and non-required field. When a field does not contain a value, leave it empty. Do not put spaces between the double quotes. For example:

"bdupre", "NEHEN003", "Hospital", "909999", "", "", "", "Dupre", etc...

Sample File

*"Batch", "NEHEN003", "Hospital", "", "999999", "", "", "", "Dupre", "Bonz", "HP99998
9200", "19630202", "F", "", "", "20010501", "",*

*"Batch", "NEHEN002", "Hospital", "", "888888", "", "", "", "Stuntz", "Laurance", "9998
8877700", "19700101", "M", "", "", "20010501", "20010502",*

Extract Process Setup Guidelines

1. Develop the batch process to extract information from your hospital system and create a file using the above record format. Create a separate file for each payer.
2. Create an automated method to run the batch process nightly against the hospital system and use FTP to transfer the file to the NEHEN server where the eGateway is installed and running.
3. Configure the NEHEN server to accept the nightly batch file by activating the FTP service in Internet Information Server and setting security to allow the batch extract process to FTP the extract file.

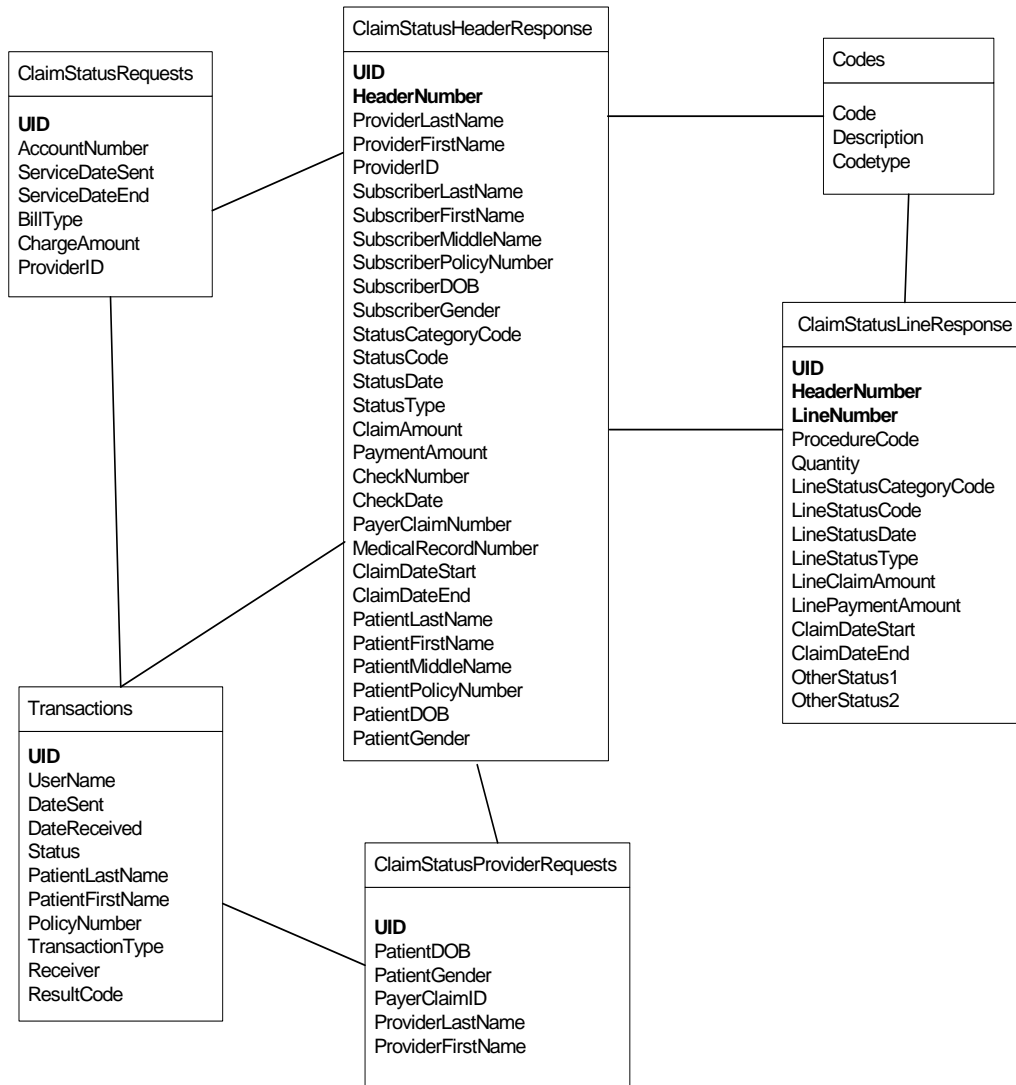
4. Setup an automated process on the NEHEN server to trigger the execution of the Send276 application after the batch extract arrives at the NEHEN server. Use a batch scheduler tool or the NT Task Scheduler. See Appendix A for a sample script.
5. The Send276 application resides in the [NEHEN core path]\NEHENLite\Bin directory. The application requires one parameter for the file name and location. Sample command to run Send276:

send276 d:\NehenExtract\extract.txt

This command will be run by the automated process on the NEHEN server and may also be run from the DOS command prompt.

Database

Data Model



Database Tables

Transactions

Description: This table contains each transaction that was sent to a trading partner using NEHENLite or during batch processing.

COLUMN NAME	DATA TYPE	DESCRIPTION
UID	Varchar	This value uniquely identifies a transaction.
UserName	Varchar	The user who submitted the transaction.
DateSent	DateTime	The date and time the 276 request transaction was sent to the payer.
DateReceived	DateTime	The date and time the 277 response transaction was received.
Status	Varchar	This code identifies the status of the transaction. Valid values: sent, received, reviewed
PatientLastName	Varchar	The last name of the patient. This field is populated with the information that was entered in the original request or returned from the payer.
PatientFirstName	Varchar	The first name of the patient. This field is populated with the information that was sent in the original request or returned from the payer.
PolicyNumber	Varchar	The policy number of the patient. This field is populated with the information that was sent in the original request or returned from the payer.
TransactionType	Integer	This code identifies the transaction type.
Receiver	Varchar	This code identifies the payer. Valid values: NEHEN002 – Tufts NEHEN003 – HPHC NEHEN013 – NHP NEHEN005 – Medicaid NEHEN006 – Medicare
ResultCode	Varchar	This code identifies the result of the claim status request. Valid values are stored in the ResultCodes table.

Codes

Description: This table contains different status codes and their descriptions. Codetype 507 contains the status description information.

COLUMN NAME	DATA TYPE	DESCRIPTION
Code	Varchar	Code.
Description	Varchar	Translation of the code.
Codetype	Varchar	Type of code. The claim status inquiry status has a code type of "507".

ClaimStatusRequests

Description: This table contains the claim status request information from the 276 transaction that was sent from either NEHENLite, the web-based application, or the batch process.

COLUMN NAME	DATA TYPE	DESCRIPTION
UID	Varchar	This value uniquely identifies a transaction.
AccountNumber	Varchar	Account number
ServiceDateSent	Datetime	Beginning date of service.
ServiceDateEnd	Datetime	Ending date of service.
BillType	Varchar	Bill Type
ChargeAmount	Varchar	Total amount of service.
ProviderID	Varchar	The provider's identification number.

ClaimStatusProviderRequests (New)

Description: This table contains the claim status request information from the 276 transaction that was sent from either NEHENLite or during batch processing. Upon receiving the additional fields required to be stored from the provider, they will be added to this table.

COLUMN NAME	DATA TYPE	DESCRIPTION
UID	Varchar	This value uniquely identifies a transaction.
PatientDOB	Datetime	The patient's date of birth.
PatientGender	Char	The patient's gender.
PayerClaimID	Varchar	Payer's claim identification number.
ProviderLastName	Varchar	Either the provider's last name OR the institution name.
ProviderFirstName	Varchar	Provider's first name. Null if the ProviderLastName field contains the institution name.

ClaimStatusHeaderResponses (New)

Description: This table contains the claim status header response information from the 277 response transaction that was sent from the trading partner. For every 277 transaction, there can be zero to many header response records.

COLUMN NAME	DATA TYPE	DESCRIPTION
UID	Varchar	This value uniquely identifies a transaction.
HeaderNumber	Varchar	This number starts at 1 and then increments by 1 for every header in a transaction. For every new transaction, the numbering starts over with 1. There can be zero to many headers per transaction.
ProviderLastName	Varchar	The last name of the provider, the business entity name or the full name of the provider.
ProviderFirstName	Varchar	The first name of the provider.
ProviderID	Varchar	The provider's identification number.
SubscriberLastName	Varchar	The last name of the patient.
SubscriberFirstName	Varchar	The first name of the patient.
SubscriberMiddleName	Varchar	The middle initial of the patient.
SubscriberPolicyNumber	Varchar	The subscriber's policy number.
SubscriberDOB	DateTime	The subscriber's date of birth.
SubscriberGender	Varchar	The subscriber's date of birth.
StatusCategoryCode	Varchar	This code identifies the status description for the claim. Valid values are stored in the Codes table (Codes.codetype = '508' or '999')
StatusCode	Varchar	This code identifies the overall status for the claim. Valid values are stored in the Codes table (Codes.codetype = '507')
StatusDate	Datetime	Date of when the status was last updated.
StatusType	Varchar	This code identifies the type of code stored in the StatusCategoryCode field. (Codes.codetype = '508' or '999')
ClaimAmount	Numeric	Total charge amount for all the claims for the date range.
PaymentAmount	Numeric	Total amount paid of the charged amount.
CheckNumber	Varchar	Payment check number.
CheckDate	Datetime	Payment check date.
PayerClaimNumber	Varchar	Payer's claim control number.
MedicalRecordNumber	Varchar	Medical record number.
ClaimDateStart	Datetime	Beginning date of service.
ClaimDateEnd	Datetime	Ending date of service.
PatientLastName	Varchar	The last name of the patient.
PatientFirstName	Varchar	The first name of the patient.
PatientMiddleName	Varchar	The middle initial of the patient.
PatientPolicyNumber	Varchar	The patient's policy number.
PatientDOB	Datetime	The patient's date of birth.
PatientGender	Varchar	The patient's gender. Valid values: M, F or U

ClaimStatusLineResponses (New)

Description: This table contains the claim status line response information from the 277 transaction that was sent from the trading partner. For every 277 header transaction, there can be one to many line records.

COLUMN NAME	DATA TYPE	DESCRIPTION
UID	Varchar	This value uniquely identifies a transaction.
HeaderNumber	Varchar	This number corresponds to the ClaimStatusHeaderResponse.HeaderNumber.
LineNumber	Varchar	Line number. There can be one to many lines per ClaimStatusHeaderResponses.
ProcedureCode	Varchar	Procedure code for the claim.
Quantity	Varchar	Quantity
LineStatusCategoryCode	Varchar	This code identifies the status description for the claim. Valid values are stored in the Codes table (Codes.codetype = '508' or '999')
LineStatusCode	Varchar	This code identifies the line status for the claim. Valid values are stored in the Codes table (Codes.codetype = '507')
LineStatusDate	Datetime	Date of when the status was last updated.
LineStatusType	Varchar	This code identifies the type of code stored in the StatusCategoryCode field. (Codes.codetype = '508' or '999')
LineClaimAmount	Numeric	Charge amount of this line.
LinePaymentAmount	Numeric	Amount that has been paid for this line.
ClaimDateStart	Datetime	Beginning date of service.
ClaimDateEnd	Datetime	Ending date of service.
OtherStatus1	Varchar	Additional claim status information.
OtherStatus2	Varchar	Additional claim status information.

Batch Applications

ProviderSend276

The Send276 application stores any provider custom fields from the claim status inquiry batch extract record. A method will be created to insert a record into the ClaimStatusProviderRequests table. This method will be called after a record is inserted into the Transaction table.

StoreReceive277

The StoreReceive277 application stores the claim status inquiry response information from the 277 transactions into the appropriate tables.

Claim Status Response Codes

The following section contains all the valid claim status codes (code type = “507”) that could be returned on a 277 transaction. Only transactions with a specific code will be stored in the new 277 response tables.

Status	Description	Stored in Response Table?
A0	Acknowledgement/Forwarded	Yes
A1	Acknowledgement/Receipt	Yes
A2	Acknowledgement/Acceptance into adjudication system	Yes
A3	Acknowledgement/Returned as unprocessable claim	No
A4	Acknowledgement/Not Found	No
E0	Error in Submitted Request Data	No
E1	Response not possible	No
F0	Finalized	Yes
F1	Finalized/Payment	Yes
F2	Finalized/Denial	Yes
F3F	Finalized/Forwarded	Yes
F3N	Finalized/Not Forwarded	Yes
F4	Finalized/Adjudication Complete	Yes
F5	Finalized/Cannot Process	Yes
P0	Pending: Adjudication/Details	Yes
P1	Pending/In Process	Yes
P2	Pending/In Review	Yes
P3	Pending/Requested Information	Yes
P4	Pending/Patient Requested Information	Yes
R0	Requests for additional Information/General Requests	No
R1	Requests for additional Information/Entity Requests	No
R3	Requests for additional Information/Claim/Line	No
R4	Requests for additional Information/Documentation	No
R5	Request for additional information/more specific detail	No

Appendix A – Batch Script

Purpose

This script searches the ftproot directory for a file called NehenExtract.txt. If the file exists, the process moves and renames the file then the Send276 application is executed. If the file does not exist, the process sleeps before checking for the file again. This script will continue to execute until the file is found. You may use the job scheduler or NT task scheduler to set the maximum run time for the script.

Sample VB Script (file extension of .vbs)

```
set fso = wscript.createobject("scripting.filesystemobject")
set ws = wscript.createobject("wscript.shell")
randomize(Timer)

Dim runflag
Dim newfilename
Dim filectr
Dim fname
Dim mypos

runflag = true
filectr = 0

Set f = fso.GetFolder("d:\Inetpub\ftproot\")
Set t = ls.OpenTextFile("d:\Batch Claim Status\Logs\extractscript.txt",8, true,
TristateFalse)
Set fs = f.Files

runflag = true

Do While runflag = true
  For each fl in fs
    fname = fl.name
    mypos = InStr(1,fname,"nehen", 1)
    if mypos > 0 then
      newfilename = "d:\Batch Claim Status\Temp\" &
"nehenextract" &
Cstr(Rnd) & ".txt"
      fl.Move (newfilename)
      ws.Run("Send276.exe " & newfilename)

      filectr = filectr + 1
      Wscript.Sleep(50000)
    end if
  Next
Next
```

```
If filectr = 0 then  
    Wscript.Sleep(100000)  
else  
    runflag = false  
end if  
Loop  
set fso = nothing  
set ws = nothing
```